NBS SPECIAL PUBLICATION 503

COMPUTER SCIENCE AND STATISTICS: TENTH ANNUAL SYMPOSIUM ON THE INTERFACE

U.S. DEPARTMENT OF COMMERCE/National Bureau of Standards

NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards[1] was established by an act of Congress March 3, 1901. The Bureau's overall goal is to strengthen and advance the Nation's science and technology and facilitate their effective application for public benefit. To this end, the Bureau conducts research and provides: (1) a basis for the Nation's physical measurement system, (2) scientific and technological services for industry and government, (3) a technical basis for equity in trade, and (4) technical services to promote public safety. The Bureau consists of the Institute for Basic Standards, the Institute for Materials Research, the Institute for Applied Technology, the Institute for Computer Sciences and Technology, the Office for Information Programs, and the Office of Experimental Technology Incentives Program.

THE INSTITUTE FOR BASIC STANDARDS provides the central basis within the United States of a complete and consistent system of physical measurement; coordinates that system with measurement systems of other nations; and furnishes essential services leading to accurate and uniform physical measurements throughout the Nation's scientific community, industry, and commerce. The Institute consists of the Office of Measurement Services, and the following center and divisions:

Applied Mathematics — Electricity — Mechanics — Heat — Optical Physics — Center for Radiation Research — Laboratory Astrophysics[2] — Cryogenics[2] — Electromagnetics[2] — Time and Frequency[2].

THE INSTITUTE FOR MATERIALS RESEARCH conducts materials research leading to improved methods of measurement, standards, and data on the properties of well-characterized materials needed by industry, commerce, educational institutions, and Government; provides advisory and research services to other Government agencies; and develops, produces, and distributes standard reference materials. The Institute consists of the Office of Standard Reference Materials, the Office of Air and Water Measurement, and the following divisions:

Analytical Chemistry — Polymers — Metallurgy — Inorganic Materials — Reactor Radiation — Physical Chemistry.

THE INSTITUTE FOR APPLIED TECHNOLOGY provides technical services developing and promoting the use of available technology; cooperates with public and private organizations in developing technological standards, codes, and test methods; and provides technical advice services, and information to Government agencies and the public. The Institute consists of the following divisions and centers:

Standards Application and Analysis — Electronic Technology — Center for Consumer Product Technology: Product Systems Analysis; Product Engineering — Center for Building Technology: Structures, Materials, and Safety; Building Environment; Technical Evaluation and Application — Center for Fire Research: Fire Science; Fire Safety Engineering.

THE INSTITUTE FOR COMPUTER SCIENCES AND TECHNOLOGY conducts research and provides technical services designed to aid Government agencies in improving cost effectiveness in the conduct of their programs through the selection, acquisition, and effective utilization of automatic data processing equipment; and serves as the principal focus wthin the executive branch for the development of Federal standards for automatic data processing equipment, techniques, and computer languages. The Institute consist of the following divisions:

Computer Services — Systems and Software — Computer Systems Engineering — Information Technology.

THE OFFICE OF EXPERIMENTAL TECHNOLOGY INCENTIVES PROGRAM seeks to affect public policy and process to facilitate technological change in the private sector by examining and experimenting with Government policies and practices in order to identify and remove Government-related barriers and to correct inherent market imperfections that impede the innovation process.

THE OFFICE FOR INFORMATION PROGRAMS promotes optimum dissemination and accessibility of scientific information generated within NBS; promotes the development of the National Standard Reference Data System and a system of information analysis centers dealing with the broader aspects of the National Measurement System; provides appropriate services to ensure that the NBS staff has optimum accessibility to the scientific information of the world. The Office consists of the following organizational units:

Office of Standard Reference Data — Office of Information Activities — Office of Technical Publications — Library — Office of International Standards — Office of International Relations.

[1] Headquarters and Laboratories at Gaithersburg, Maryland, unless otherwise noted; mailing address Washington, D.C. 20234.
[2] Located at Boulder, Colorado 80302.

Computer Science and Statistics: Tenth Annual Symposium on the Interface

Proceedings of the 10th Annual Symposium
held at the National Bureau of Standards
Gaithersburg, Maryland
April 14-15, 1977

Edited by:

David Hogben, Institute for Basic Standards
and
Dennis W. Fife, Institute for Computer Sciences and Technology

National Bureau of Standards
Washington, D.C. 20234

U.S. DEPARTMENT OF COMMERCE, Juanita M. Kreps, Secretary

Dr. Sidney Harman, Under Secretary

Jordan J. Baruch, Assistant Secretary for Science and Technology

NATIONAL BUREAU OF STANDARDS, Ernest Ambler, Director

Issued March 1978

National Bureau of Standards Special Publication 503
Nat. Bur. Stand. (U.S.), Spec. Pub. 503, 467 pages (March 1978)
CODEN: XNBSAV

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 1978

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20434
Order by SD Stock No. 003-003-01908-9 Price $6.25
(Add 25 percent additional for other than U.S. mailing).

AVAILABILITY OF PROCEEDINGS OF PREVIOUS INTERFACE SYMPOSIA

The Proceedings of the Ninth Interface can be obtained from

Prindle, Weber & Schmidt, Incorporated

20 Newbury Street

Boston, MA 02116

The Proceedings of the Eighth Interface can be obtained from

Health Science Computing Facility

AV-111, Center for Health Sciences

University of California

Los Angeles, CA 90024

The Proceedings of the Seventh Interface can be obtained from

Statistical Numerical Analysis and Data Processing Section

117 Snedecor Hall

Iowa State University

Ames, IA 50010

The Proceedings of the Fourth, Fifth, and Sixth Interface can be obtained from

Western Periodicals Company

13000 Raymer Street

North Hollywood, CA 91605

PREFACE

Computer Science and Statistics: The Tenth Annual Symposium on the Interface was a continuation of a series of Interface Symposia which has developed rapidly both in quantity and quality. The objective of the Interface is to provide a forum for statisticians, computer scientists and numerical analysts to discuss important problems in the rapidly growing field of statistical computing. The workshop structure of the Interface, bringing together a variety of people from different disciplines, is an effective mechanism for consolidating and disseminating technical advances and for identifying important problems whose solution would benefit both statistics and computer science.

Attendance at the Tenth Interface was high with over 440 participants. As the list of the participants at the end of these proceeedings show, participants came from all over the United States and several foreign countries.

The highlight of the Tenth Interface was the superb Keynote Address, The Mathematization of Computer Science, presented by Anthony Ralston, Chairman, Department of Computer Science, SUNY Buffalo.

Following the format of the Ninth Interface, the Tenth Interface consisted of six Workshops and three Poster Sessions. There were three concurrent Workshops on the first day and three on the second day. The Workshops had 42 invited speakers and the Poster Sessions attracted 47 contributed papers.

The Evaluation of Statistical Software Workshop was divided into two sessions on Statistical Program Packages for Small Computers, Chaired by Ivor Francis, and Computing Approaches to the Analysis of Variance from Unbalanced Data, Chaired by Richard M. Heiberger. In the first session invited speakers reviewed statistical programs for small computers, their languages and portability and prospects for statistical systems for the new generation of minicomputers. In the second session authors of ANOVA programs discussed issues leading to the choice of appropriate hypotheses for a given set of data and the default decisions taken by their programs. The Nonlinear Models Workshop, Chaired by John M. Chambers and John E. Dennis, presented important new material for the fitting and analysis of nonlinear models. The Graphics Workshop, Chaired by Jane F. Gentleman, was concerned with the choice of graphics hardware and human engineering in graphics software. The Large Data Files Workshop, Chaired by Gordon Sande, Jr., emphasized computing for "messy data" obtained under incompletely controlled situations. The Numerical Analysis in Statistics Workshop, Chaired by Richard A. Tapia, was concerned with the exchange of ideas and experiences with the goal of determining directions for future education and research. The Maintenance and Distribution of Statistical Software, Chaired by Mervin E. Muller, featured invited discussions by developers and users of major software of techniques for more effective maintenance and distribution of software.

ACKNOWLEDGEMENTS

It is a pleasure to acknowledge the essential financial and professional support of the National Science Foundation, the Office of Naval Research, and the U. S. Army Research Office.

The Tenth Interface was held with the support and cooperation of the Statistical Computing Section, American Statistical Association; the Washington Statistical Society; the Association for Computing Machinery; SIGNUM, ACM; and the Washington, DC Chapter, ACM.

The success of the Tenth Interface was due primarily to the efforts of the eight Workshop Chairpersons: Ivor Francis, Richard M. Heiberger, John M. Chambers, John E. Dennis, Jane F. Gentleman, Gordon Sande, Jr., Richard A. Tapia, and Mervin E. Muller.

Past Interface Chairmen gave valuable guidance. In particular, we appreciate the guidance of David C. Hoaglin, Roy E. Welsch, James A. Frane, and William J. Kennedy.

Sarah R. Torrence was in charge of local arrangements and provided invaluable assistance. Doris M. Burrell supervised registration. Roy H. Wampler gave assistance for transportation arrangements. Marjorie E. Young gave accounting assistance. Barbara A. Uglik helped in many ways.

David Hogben
Dennis W. Fife

Chairpersons

Financial support from

National Science Foundation (MCS77-04441)

Office of Naval Research (NR 042-000)

U. S. Army Research Office (ARO 14862-M)

and the cooperation of

SIGNUM, ACM

Washington Statistical Society

Washington, D.C. Chapter of ACM

Association for Computing Machinery

Statistical Computing Section of the American Statistical Association

CONTENTS

EVALUATION OF STATISTICAL SOFTWARE WORKSHOP

Ivor Francis and Richard M. Heiberger, Chairpersons

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

AN INTERACTIVE STATISTICAL PROCESSOR FOR THE UNIX TIME-SHARING SYSTEM

Peter Bloomfield
Department of Statistics, Princeton University, Princeton, N. J. 08540

ABSTRACT

An interactive statistical processor has been developed for the Unix time-sharing system. A unified command syntax has been imposed by using a command-interpreting "shell" program, which communicates with user at his or her terminal and initiates execution of separate programs to carry out the required operations. Uniformity of these operational programs has been achieved by using a single structure for files and providing a library of subroutines for analyzing the standard syntax for specification of options.

Since the shell knows nothing about the programs that it executes, except for default places to find them, new commands may be added even during the course of a session. Users may develop and use their own commands without making them publicly available, and if the command has the same name as a publicly available command, the user version is found first and executed, thus effectively redefining the command for that user.

Key words: Interactive data analysis; data analysis on minicomputers.

1. INTRODUCTION

Isp, the interactive statistical processor described in this document, was developed to provide a flexible way of using an extensive collection of data analysis programs on a minicomputer. The requirements were that the data analyst should be able to enter, modify and save data on disk files, to reexpress the data in various ways, to select subsets of the data in various ways, and to carry out various analyses of the results of these operations. The processor has been in daily use from the day it was first installed and has been used for a large number of analyses, often accounting for the largest single component of the daily use of the minicomputer on which it is run.

2. DESIGN CONSIDERATIONS

The limited main memory of a minicomputer requires that a data analysis package such as isp be based to a large extent on a mass storage device, typically disk. In the case of isp, the only component of the system that is

resident in main memory is a command interpreting "shell" program. The shell carries on all the interaction with the user, accepting free-format commands with a simple syntax, essentially the same as is used by McNeil (1977). When a command is parsed successfully, the shell initiates the loading from disk and execution of the appropriate program, and passes arguments to it that specify the (disk-resident) data set to be operated on, what results are requested, and what options have been exercised by the user. The results may appear as output on the user's terminal, or may be placed in new disk files, or both.

The functions of the various programs have been chosen to avoid duplication as far as possible. Thus, for instance, the regression program may produce a file of residuals, but no plots. There are several programs for producing a variety of graphical outputs. If a user wishes to give a single command that will cause a sequence of programs to be run (possibly a re-expression followed by a regression followed by a plot of the residuals against the fit), a macro may be constructed to do this.

The shell program contains no information about the possible commands. In fact, a command may refer to either a macro file or an executable program file, and these may be in the user's area of disk or the system's area. The name of the file is the same as the name of the command, and the four possible locations are searched in turn for a file of that name. Thus commands may be added or changed simply by installing a file in the appropriate area. Also, a user may implement commands for himself that would not be available to other users. Furthermore, since the user area is searched before the system area, a user may install his own version of a system command.

For simplicity all data are handled in a single format internally to isp, and to avoid repeated conversions that format is binary. There are utility commands, "read" and "print", for converting from character format to binary and _vice versa_. Binary format files are called _variables_. All files are created in a temporary area on disk that is cleaned up and removed when the session terminates. Files may be copied into permanent areas, of which there is one for character-format files and one for binary-format files.

3. PORTABILITY

Isp was developed on a Digital Equipment Corporation PDP-11/40 minicomputer, operating under the Unix Time-sharing System (Ritchie and Thompson, 1974). Features of Unix that are important in the design of isp are

- the ability of one program to initiate loading and execution of another program
- the ability to create, delete, extend, truncate and otherwise modify disk files during the execution of a program
- a convenient, nonrestrictive file system structure.

A similar processor could be developed for any combination of computer and operating system that offered these facilities. In the programs that carry out actual analyses, system dependent aspects such as file-handling have been restricted almost completely to a few library routines. They are mostly

written in Ratfor (Kernighan, 1974; see also Kernighan and Planger, 1976; Ratfor is abbreviated from rational Fortran), but may easily be preprocessed into Fortran. The shell program, however, is liberally interspersed with Unix system-calls. It is written partly in the language C (Ritchie, 1974) and partly in the compiler-compiler language Yacc (Johnson, 1974). Since these compilers are not widely available other than on Unix systems, the shell would need almost complete rewriting for other systems.

4. AN EXAMPLE

The following is a short example of a session with isp, in which a set of data is typed in at the terminal, displayed and analyzed. (Characters typed at the keyboard are underlined. Lines typed at the keyboard are terminated by a 'carriage return'. The symbol '^D' marks a control-D, which, when typed immediately after 'carriage return', indicates the end of input.)

The first command is to 'make' a file called 'junk', which contains the data.

```
*make junk
1 0
2 3
3 2
4 3
5 6
^D
```

The second command is to 'read' the data in 'junk' (i.e., convert to binary format), and place it in a variable called 'var'. This line may be read as 'read junk onto var'.

```
*read junk > var
```

The next command is to 'list' the names and types of files that have been created. Variables (that is, binary format files) are listed as 'arrays'.

```
*list
junk            text
var             array (10)
```

Since we want to use 'var' as an array of 5 lines each with 2 entries, the utility 'let' is invoked to reshape 'var'.

```
*let var = var (5,2)
```

A second use of 'list' verifies that 'var' now has the right shape.

```
*list
junk            text
var             array (5,2)
```

The 'print' utility is used as a second check that 'var' is correct.

```
*print var
1.0000          0.0000
2.0000          3.0000
3.0000          2.0000
4.0000          3.0000
5.0000          6.0000
```

Simple typewriter scatter-plots may be produced by 'scat'. Since 'var' has only two columns, no options need be specified. If 'var' had more columns, the command might be 'scat var {x=3;y=4}' (options are always enclosed in {}). The defaults are x=1 and y=2.

```
          *scat var
 6.00       |                  *
            |
            |
            |
            |
            |    *       *
            |
            |        *
            |
 0.000      *----------------------
       1.000                   5.000
```

The 'regress' command below fits a straight line (i.e., regresses column 2 on column 1 with a constant term). The symbol '>' is used as in the 'read' command above to indicate the disposition of output variables. Since 'regress' may produce more than one output, we specify 'res @ vresids' to indicate that the output known internally as 'res' is to be produced and placed in a variable called 'vresids'. Since 'res' is in fact the first output, 'regress var > vresids' would have the same effect (any other outputs will not be produced, since no variable name is given).

```
          *regress var > res @ vresids

 variable      coeff.          corr.          t-stat

    1         1.20000        0.875190        3.13340


 intercept     -0.800000
 multiple r     0.875190
 f-statistic     9.81819
```

The output 'res', here in variable 'vresids', consists of two columns, the first containing the fitted values and the second containing the residuals (this is also true if 'regress' is used for multiple regressions). Thus the following command produces a scatter-plot of residuals against fitted values.

```
*scat vresids

1.40      |      *
          |
          |
          |
          |                    *
          |
          |
          |
          |
          |----------------------
          |
          |
          *
          |
          |
          |              *
          |
          |                   *
-1.00     |
    0.4000                      5.200
```

The 'delete' utility is used to remove files from the temporary area. Its use is redundant here because all files in the temporary area are removed on 'exit'.

```
*delete junk var
*list

resids                  array (5,2)
*exit
```

5. CURRENT ISP COMMANDS

The following commands are currently implemented in isp. The data analysis methods are based on those described by Tukey (1977). Similar commands are described by McNeil (1977). The robust methods are developments of techniques described in Andrews et al. (1972) and Huber (1973). Several other commands exist in an experimental state.

System Commands

make	create a text file
edit	edit a text file
read	read a text file, converting to isp variable
save	save an isp variable or text file for future use
load	load a previously saved variable or file
delete	delete active variable(s)
unsave	unsave saved objects
list	list contents of active, data, text, or system areas
print	print variable or string
let	algebraic and manipulative capability
explain	how to explain something
rename	rename a file
copy	make a copy of a file
echo	echo command arguments

Data Analysis

boxplot	schematic plots
stemleaf	stem and leaf displays
code	coded displays
fivenum	fivenumber summaries
biweight	robust estimates of location (biweight M-estimate)
compare	comparison (schematic) plots
scat	scatterplots

Robust Fitting

robust	regression
oneway	anova
twoway	anova

Least Squares

stat	basic statistics of batches
corr	correlation matrix
regress	regressions
eigen	eigenvalues (real symmetric matrix)
princo	principal components
svd	singular value decomposition
cancorr	canonical correlations

Time Series

smooth	Tukey's smoothers
fft	Fast Fourier Transform
pgram	periodogram
xpgram	cross periodogram
cohphs	phase stuff

Utilities

gplot	Gsi, Tektronix graphics
trans	matrix transposes
sort	sorting (Shell sort)

6. ACKNOWLEDGMENTS

The earliest version of isp was developed by Ken Birman, whose work was supported by National Institutes of Mental Health Grants 1R03 MH 26561-01 and 26692-01. The latest version of the shell and the library routines for Ratfor programs were developed by David Donoho. Command programs have been contributed by many people. The Energy Research and Development Administration contract EY76-S02-2310, awarded to the Department of Statistics, Princeton University, has supported much of the software development.

7. REFERENCES

ANDREWS, D.F., BICKEL, P.J., HAMPEL, F.R., HUBER, P.J., ROGERS, W.H., and TUKEY, J. W. (1972). Robust Estimates of Location. Princeton University Press.

HUBER, P.J. (1973). Robust regression: asymptotics, conjectures and Monte-Carlo. Ann. Statist., 1, 799-821.

JOHNSON, STEPHEN C. (1974). Yacc - yet another compiler-compiler. Bell Telephone Laboratories, Technical memorandum.

KERNIGHAN, BRIAN W. (1974). A preprocessor for a Rational Fortran. Bell Telephone Laboratories, Technical memorandum.

KERNIGHAN, BRIAN W. and PLAUGER, P.J. (1976). Software Tools. Reading, Mass.: Addison-Wesley.

McNEIL, DONALD R. (1977). Interactive Data Analysis. New York: Wiley.

RITCHIE, DENNIS M. (1974). C reference manual. Bell Telephone Laboratories, Technical memorandum.

RITCHIE, DENNIS M. and THOMPSON, K. (1974). The Unix time-sharing system. Comm. ACM, 17, 365-375.

TUKEY, J. W. (1977). Exploratory Data Analysis. Reading, Mass.: Addison-Wesley.

8. BIOGRAPHY

Peter Bloomfield received his Ph.D. in Statistics from the University of London in 1970. He has taught in the Department of Mathematics at the Imperial College of Science and Technology, London, and the Department of Statistics at Princeton University, where he is currently Associate Professor and Director of Graduate Studies.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

MINIBMD: A MINICOMPUTER STATISTICAL SYSTEM

R. Buchness, L. Engelman
Health Sciences Computing Facility, UCLA, Los Angeles, Ca. 90024

ABSTRACT

The falling prices of minicomputers, their evolving capabilities, and their increasing presence in biomedical settings has motivated the Health Sciences Computing Facility to develop a minicomputer statistical package. Minicomputers are commonly used in biomedical research for data acquisition and screening. For statistical analysis, users of minicomputers must either use inadequate vendor packages or write their own software.

The MiniBMD package will be a reliably crafted, well supported statistical system operating on a wide variety of minicomputers. It will be arranged into a set of FORTRAN modules, (such as data input, screening, editing, description and various statistical routines) tied together by a supervisory program having simplified problem specification and tailored output routines. The modular structure will make it easier for the researcher with a specialized problem to modify or plagiarize the necessary routines. Documentation will be provided at both the program and module level.

Because the biomedical researcher requires quality software, meticulous numerical crafting and testing will be used in developing the MiniBMD series. A manual including test runs and annotated output will be provided to assist the investigator in proper program usage. Input and output will be finely tuned for both the batch and interactive investigator.

Keywords: Biomedical; FORTRAN; minicomputers; software; statistics

1. INTRODUCTION

At present, the minicomputer user desiring a general statistical package is faced with the prospect of a) using vendor packages, b) hiring an application programmer, or c) becoming a proficient programmer. The first option is often rejected because vendor packages 1) do not exist, 2) are inflexible, 3) have insufficient scope, or 4) are not well tested and documented. Hiring an application programmer is often impossible due to lack of funds. Few investigators can afford both an application and system programmer for their minicomputers. This leads to a search for the unlikely blend of both programming types. Transient students are often used as application programmers but this is generally inefficient: their goals seldom include documentation, quality coding and adequate testing. After rejecting the first two options the investigator is often forced to become a proficient programmer himself. Although some knowledge is mandatory for proper facility management, a full time study is usually an unnecessary dilution of the investigator's research.

At best, none of the above alternatives is likely to provide the researcher with a full range of up to date techniques. These considerations point to the need for a reliably designed, well-supported statistical system for minicomputers, that will allow the researcher to screen, edit, examine and analyze his data. The MiniBMD system is being designed and developed to meet these requirements.

Minicomputer software has lagged behind hardware because of the enormous development cost to the vendor. To avoid this cost, the hardware oriented manufacturer often acquires his software from the user. However, the prospects of developing the MiniBMD package have brightened due to considerable advancement in system software for minicomputers. This software takes the form of monitors, compilers, editors and linkage routines which make the maintenance of such a package more feasible.

In conjunction with advanced software support, the vendors now supply extensive auxiliary storage management routines. This allows the user to add, delete and maintain a considerable number of auxiliary data files. These files can be accessed through minicomputer FORTRAN. Most minicomputer vendors now offer ANSI 3.9 1966 standard FORTRAN (ANSI 1966) to their users. A statistical package written using this standard would suffer from a rather weak FORTRAN definition. Fortunately, a majority of minicomputer vendors have uniformly extended the standard to produce a more powerful FORTRAN. These extensions include mixed mode expressions, direct access, and file and error handling, and several other features not provided by the ANSI standard. In addition, most vendors provide character manipulation either directly or by utility subroutines.

2. SYSTEM DESCRIPTION

2.1 Comparison with BMD and BMDP

The MiniBMD statistical package is an entirely new system and will be quite different internally from the existing BMD and BMDP statistical packages described in Dixon (1973, 1975) and Frane (1976). This difference takes several forms. including: 1) less use of main memory, 2) more extensive use of auxiliary storage for intermediate storage of data, 3) alternate batch or interactive modes of operation and 4) terse as well as extensive output.

2.2 Use of main memory

The average minicomputer has an address space of 32K 16 bit words. Considering that a typical BMD program has a 30K 16 bit word storage area, the new series will have a completely different storage philosophy. Some minicomputers provide dynamic storage allocation on a subroutine basis and this reduces the core storage requirements on these machines. But a general system of dynamic memory management may be required for machines with severe core restrictions.

In order to reduce core requirements, the MiniBMD package will be composed of relatively independent modules. This modular construction will allow the use of OVERLAY management. The specification of OVERLAY structure varies widely but the option exists on most minicomputers. It is possible that the new series will use some of the ideas developed in the BMD series which assist in core storage management. However, these ideas will have to be further developed.

2.3 Use of auxiliary storage

A standard option for a minicomputer system is a disk storage device capable of storing well over one million real values. The MiniBMD data base will be transferred from disk by a resident executive. The executive will create and process annotated data files which contain the origin, history and description of the data in question. The form of the data base on auxiliary storage will be expanded because of the powerful transformation capabilities of the package. Using the annotated file, an investigator will be able to evaluate a transformation that produced a new variable.

2.4 Batch and interactive modes

In a batch job, program flow is completely specified before submittal. An interactive job has the advantage of allowing conditional program flow based on user input during

execution. This advantage is often very important in an experimental situation where the investigator is probing the nature of the data base. Early minicomputers were entirely interactively oriented. A minicomputer user was given 'his' machine and allowed to interact with the system. This unfortunately implied a considerable amount of typing which some investigators were reluctant to perform. A recent development in minicomputer systems is the ability to accept batch files and job control language.

Unlike the BMD series, which is batch oriented, the MiniBMD series will have both batch and interactive modes. In the interactive mode, the system may prompt the user for missing values and may request respecification of values outside a specified range. Once a sequence of steps is defined for a given problem, the investigator is free to define these steps in a batch control file which requires no interaction.

The control commands and their syntax will be identical in both the interactive and in batch mode of execution.

3. INPUT AND OUTPUT

The design of a statistical package places a heavy emphasis on input and output. The data base of a minicomputer user may be either fixed or free field information. The user of the MiniBMD will be able to specify either interactively or from a batch control file, the manner in which the data are read and manipulated. As in the BMD series, fixed field information will be handled using FORMAT specifications provided by the user. Less rigid input will be allowed with values separated by commas or by blanks, as desired. In both cases, the program will provide correct handling of extreme values and missing values. The proper reading and verification of data will be completely under user control via control language statements.

Program output will also be under user control. This output will take two forms, namely, 1) compact and 2) extensive. Compact output is designed primarily for inspection at the terminal in the interactive mode. This output will guide the user in specifying the next step in his analysis. The user will be able to compose a summary table or plot of the exact item of interest by using the powerful grouping and screening abilities of the command language. In both the batch and the interactive mode, the investigator will be able to obtain comprehensive hard copy in the form of tables and graphs, tailored to the output device in his system.

4. QUALITY STATISTICS

The Health Sciences Computing Facility has been heavily involved in developing statistical programs for sixteen years. During that time, we have learned that quality statistical programs cannot be obtained from a routine translation of statistical texts into a programming language. Such an attempt ignores many of the 'real' requirements of the user - for ease of setup, intermediate results, a useful display of information, etc. It also ignores the fact that textbook methods are often inadequate for the needs of the problem. Finally, developing the algorithm is only the beginning. Extensive testing, maintenance, and documentation are essential, if the programs are to have real utility. We have a large statistical staff to assist in both the statistical accuracy and selection of output forms in the MiniBMD series. It will also insure that up-to-date statistical techniques will be available for the BMD user.

To maintain integrity, statistical software for any computer system must be carefully designed and tested. In a minicomputer system particular notice must be taken of both statistical design and computer characteristics. Typical minicomputer precision allows six digits of accuracy during real valued computation. When required, double precision can be requested using standard FORTRAN. The proper use of such changes in precision is an integral part of the existing BMD package. In the MiniBMD series, the tradeoff between speed and accuracy will be carefully considered.

5. COMMAND LANGUAGE

The proposed series will contain a command interpreter which will parse, verify and execute command lines in both batch and interactive modes. The parse stage deciphers expressions, recognizes and creates variables and prepares commands for execution. The verify stage will provide diagnostics and defaults when required. The execute stage will invoke the proper module and assure the presence of proper variables. Basic commands will be identified with a keyword followed by parameter for example,

```
PLOT   X=HEIGHT  Y=WEIGHT;
```

will plot the current variables 'height' versus 'weight'.

An important part of the system is the transform processor which is used for data transformations and interpreting complicated case selection and editing criteria. For example, the user's case selection rule might read

```
SELECT    SEX.EQ.1.AND.(AGE GT.20.AND.AGE.LT.50)
```

or to compute a patient's age at treatment onset from his birth year and the treatment date (only two digits are used to record the year).

```
TRANSFORM AGE = START - BIRTH,
      IF (AGE.LT.0) THEN AGE = AGE + 100;
```

The same set of routines is also used by a CALCULATOR function that allows the user to evaluate algebraic expressions at any time during the session. This processor accepts user commands in notation similar to FORTRAN. A preliminary version of the transform processor has been tried out on PDP-11, PDP-12 and MODCOMP computers.

6. IMPLEMENTATION

At present, most minicomputer FORTRAN systems suffer from poor run time diagnostics and debugging aides. A typical installation also lacks proper peripherals for extensive program development. These facts make the creation of a statistical package difficult on a small installation. We intend to use UCLA minicomputers from at least four different vendors, augmented by the power of a large host, to develop the MiniBMD package. Using a variety of computers will assist in cross-checking of FORTRAN code and statistical accuracy, as well as helping to insure portability.

Several verification routines exist which can be used in assessing the portability of software. The system described by Ryder (1974) is routinely used in the development of the BMDP series. It is possible that a new program will be developed for the more complicated portability required for diverse minicomputer systems.

7. DISTRIBUTION AND MAINTENANCE

A statistical package is useless unless it is properly documented and maintained. At UCLA, we have distributed documentation and revisions for BMD installations throughout the world. Because of the diverse nature of minicomputers, we plan to use redistribution centers for specific hardware configurations. This method of support has been effective for the BMDP series.

The MiniBMD series will be distributed in phases, allowing user feedback to influence the redirection and modification of later stages of the design. We will form an advisory committee of experts in statistical systems to advise us on our developments.

8. CONCLUSIONS

Minicomputer systems have reached a level which allows design and distribution of a quality statistical package. Such a package must contain reliable code which has been

ieticulously tested and statistically confirmed, and should be designed to perform in both batch and interactive modes. A necessary prerequisite to success is proper documentation, support and improvements after program distribution. The MiniBMD package will have the necessary blend of statistical accuracy, portability, documentation and support to produce a viable tool in biomedical research.

9. ACKNOWLEDGMENT

Research sponsored by the Health Sciences Computing Facility, UCLA, supported by NIH Special Resources Research Grant RR-3.

10. REFERENCES

American National Standard FORTRAN (1966). American National Standard Institute, New York, N. Y.

DIXON, W.J., Ed. (1973). BMD-Biomedical Computer Programs. Berkeley, University of California Press.

DIXON, W.J., Ed. (1975). BMDP-Biomedical Computer Programs. Berkeley, University of California Press.

FRANE, J.W. (1976). The BMD and BMDP Series of Statistical Computer Programs. Communications of ACM, 19:10, 570-576.

RYDER, B.G. (1974). The PFORT Verifier, Software Practice and Experience, 4:4, 359-378.

BIOGRAPHIES

R. Buchness received a B.A. in mathematics and an M.S. in computer science from UCLA. he has been an application programmer for ten years, specializing in minicomputers, computer graphics and statistical analysis. He is currently working on a Ph.D. in computer science at UCLA.

L. Engelman supervises the design and development of the BMD programs. His responsibilities include design of the basic program structures, numerical analysis, implementation of suitable algorithms, research in statistical computing, coordination and supervision of the overall production effort, and documentation and distribution of the programs to other facilities. He organizes and edits the BMD Communications newsletter and consults with users inside and outside the facility. He has sixteen years experience (twelve at HSCF) with computers and programming. (Graduate work in mathematics; publications in statistical and computing journals.)

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interfac
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 197

EXPERIENCE AND RECOMMENDED PRINCIPLES FOR THE DEVELOPMENT OF SOFTWARE FOR PROCESSING STATISTICAL DATA IN THE THIRD WORLD

Henry Elkins, Victor Matthews, and Joanna Pomeranz
Population Council, New York, N.Y. 10017 - Columbia University, New York, N.Y. 10032

ABSTRACT

Problems such as diversity, inadequate maintenance, and limited capacity of hardware, lack of trained personnel, installation difficulties, and poor international communications have hampered the implementation of software for statistical processing in the third world. Past experience demonstrates that programs written in low level FORTRAN can overcome some of the problems. Though decreasing hardware costs will yield major benefits, better international communications remains a crucial need. A regularly updated catalog describing available software for statistical processing would help meet that need.

Key words: communications; catalog; third world; software; statistical data processing; software development.

1. INTRODUCTION

Despite notable advances, many third world countries still lag far behind industriali countries in data processing. The disparity is particularly apparent in statistical as opposed to commercial applications. More advanced and less costly hardware coupled with more sophisticated operating systems offered by the major vendors will alleviate some of t problems, but as the discussion below points out, the persistent and complex maladies that plague statistical data processing require more comprehensive remedies.

2. PROBLEMS CONFRONTING STATISTICAL DATA PROCESSING IN THE THIRD WORLD

2.1 Variegated hardware. The computer hardware serving the third world is extreme diverse and includes new and obsolete offerings from nearly every manufacturer in the worl Political obligations rather than price, performance, and maintenance facilities have ofte determined the choice of manufacturer and increased the diversity. For example, war repar tions financed a Japanese computer in Manila, and the Polish government donated a compute to Dacca University in Bangladesh. The heterogeneity of hardware has limited the usefulne of software tailored to specific computers and has precluded the development of easily portable packages.

2.2 Inadequate hardware maintenance. Fragmented and limited third world markets imply minimal service facilities. Because replacement parts and qualified technicians are not readily available, a relatively minor problem may take days or even weeks to resolve. In 1973 the National Statistical Office in Thailand estimated a loss of 30 percent of effective operating time because of inoperative tape drives. Yet adequate training of servic personnel must be related to products of a particular manufacturer and requires an investmen often judged to be excessive in comparison to the size of the market.

2.3 Limited hardware capacity in relation to software requirements. Another major stacle to software implementation is the limited capacity of most third world computers in lation to the requirements of major software packages. In some cases the architectural sign of the computer itself restricts the capacity. In other cases the limited capacity rives from the tendency to install small computers in each government ministry so that ery minister may control his own fiefdom. More generally, the problem stems from a ronic lack of hard currency and the higher cost of computers for remote or semi-remote cations. The Population Council's experience indicates that computer costs in the third rld range from 50 to 100 percent higher than comparable costs in the United States, even thout taking into consideration the added expenses often necessary for power regulation or nerators. Unfortunately, even many package programs reduced to fit smaller machines have orage requirements far in excess of the capacity commonly available in the third world.

2.4 Inexperienced and poorly trained personnel. The lack of training facilities and portunities for experience are compounded in the case of those in governmental and univer- ty centers by economic disincentives. Differential pay scales between commercial and ientific endeavors are not uncommon in the West, but assume extreme dimensions in the ird world. In Thailand, where most statistical data processing is done in government ganizations, a survey of salaries in private and government organizations documented that ta processing personnel in private organizations earned two and one-half to four times as ch as their counterparts in government positions. These differentials held for all lev- s of personnel, from keypunch operators through data processing managers. It was not rprising that the same survey found unusual high turnover in government positions.

Some senior data processing management may be appointed because of their political nnections, not because of their experience or knowledge of data processing. Poor morale d inefficient utilization of the computer facility are the inevitable results. Bizarre nagement procedures further restrain efficiency. At one IBM 370 site in Africa the nager kept the manuals under lock and key, and did not permit disk storage for any user. though the users finally generated enough pressure to order SPSS, thereafter, only the nager or the assistant manager prepared all SPSS jobs and returned the output to users ter removing all related job control language from the output.

2.5 Installation difficulties. The installation of many packages requires a ystems programmer" who is intimately familiar with both the computer hardware and oper- ing system and who can devote a significant amount of time not only to the installation d testing but also to the maintenance of that package. Well written users manuals and t uncommonly oral instruction for users are generally required before the package can be ll utilized.

Many packages such as OSIRIS and DATA-TEXT are written, at least in part in Assembler nguage and/or machine-specific FORTRAN and thus are machine dependent. In recent years, me authors have attempted to overcome this problem by creating different versions of the ckage for specific target machines. The wide variety of computers in the third world ans that such an approach will have limited applicability.

2.6 Lack of communication. Those individuals who are interested and involved in atistical data processing in the third world are a small and isolated group. Unfortu- tely, much of the information concerning program development or adaptation and available chnical support is disseminated through informal communications networks to which few ird world data processers belong. Only occasionally is such information found in profes- onal publications such as Computer Survey, Proceedings of the Association for Computer chinery, Journal of the American Statistical Association, and SIGSOC. The recent tablishment of a statistical computing section in the American Statistician is a welcome t by no means adequate corrective. The newly created International Association for atistical Computing (a section of the International Statistical Institute) may perform a eful role. The proceedings of conferences are also difficult to obtain, even for owledgeable persons in the developed world. For the statistical data processer in the ird world journal subscriptions are expensive and attending conferences is out of the estion. Organizations such as the Population Council, the United Nations Statistical fice, and Bureau of the Census have provided and will doubtless continue to provide chnical assistance, but such assistance must first be requested and even when provided

often falls woefully short of fulfilling the need. The third world desperately requires new source of information -- information on what software will do, the machines for which it is suited, and where and how a prospective user may obtain it.

3. EXPERIENCE IN THE RECENT PAST

A number of different organizations and individuals have confronted the problems discussed above. Our own experience has been limited largely to the social sciences, and population in particular. We discuss below the work of organizations and individuals we know best and do not mean to imply that others have not done similar work.

In the mid-1960's Nathan Keyfitz at the University of Chicago developed a series of programs for demographic analysis. These programs written in simple FORTRAN and suitable for small computers, were later printed and widely used throughout both the developed and developing world. In 1968 the Community and Family Study Center of the University of Chicago through a field staff member located in Bogota, Colombia initiated development of what was to become the MINI-TAB series, a group of inter-related FORTRAN programs for small computers. The MINI-TAB series includes programs for data editing, frequencies (marginals), cross tabulation, multiple regression, and life table analysis. The Population Council became involved in the development and implementation of statistical software through its mandate to advance knowledge in population through research, training, and technical assistance. The Council's own IBM 1130 and FORTRAN E compiler and 16K word memory of 16 bit words (later supplanted by a PDP 11/45) provided the constrained environment for developing programs suitable for small computers in the field. The Population Council adopted the Keyfitz and MINI-TAB programs, enhanced them, and developed additional programs. The criteria for software development were portability, small core and storage requirements, modular programming, and extensive documentation and user aids designed for non-computer related personnel.

For portability the use of a low level FORTRAN overcame in large measure the problem of the variety of machines in the third world. Features such as object time formats and logical "IF" statements were avoided. FORTRAN is the language most universally known to statisticians and thus provided a basis for understanding and confidence in the alogarithm used, as well as opportunity for program modification at the local level.

In order to fit most of the small core machines in the third world, all Council programs were designed to run in 16K (16bit words) or 32K bytes. For users with greater available core storage, instructions provide for expansion of the DIMENSION statement to handle more variables and/or produce more tables. An important programming technique in common with the MINI-TAB programs is single rather than multiple dimension arrays to utilize core storage more efficiently.

A significant decision was to use modular programming, both in the macro and micro sense. At the macro level the Council decided not to offer an integrated package or system for file management and statistical analysis such as OSIRIS, P-STAT, and SPSS but instead to offer a set of independent programs. Although the integrated approach might be accomplished through heavy overlaying, additional disk storage would be required, and both portability and ease of installation and use would suffer.

At the micro level the Council utilized the concept of "structured programming." Each program includes three major sections: 1) program definition, including specification of options and checking of the set-up instructions; 2) data input, including necessary recoding and dealing with non-standard codes; and 3) output, including calculation of requested statistics. The modular approach facilitates program modification at the local level and enhances flexibility. For example, an error in the set-up instructions terminates the run with a readable description of the problem, a procedure that saves both user and computer time. Modular programming also facilitates the inclusion of options. An example is the option of directly analyzing data after having corrected the most serious data inconsistencies but without correcting data codes which might fall outside a specified range, e.g. an alpha code in a numeric field. Finally, following the example of P-STAT, modular

gramming has permitted the development of machine-dependent sub-routines (written in TRAN) which enhance program performance. Copies of these modules are integrated into ected programs when the user specifies the type of computer to be used.

A major consideration was the development of user documentation and aids for those who e not themselves highly competent in data processing, and who were presumed not to have ess to technicians who were. The user's manual provides step-by-step instructions, with mples, and contains sample set-up records, a test data file and the resulting computer put. The manual and programs incorporate heavy repetition of mnemonic symbols such as R for the number of variables, NCASE for the number of cases, and MIN and MAX for imum and maximum data values. Throughout the Council programs set-up records are hly standardized. This standardization aids the user, in that once the user has experi-e with one program, he may utilize similar set-up records with another program. The vision with each program of a sample of set-up cards, test deck, and copy of output has ilitated both local installation and user instruction, since each user can examine the -up instructions and run the sample data on his own computer to insure that the program working correctly.

4. AND OF THE FUTURE

It seems clear that decreasing hardware costs and the consequent easing of maintenance blems are likely to result in a proliferation of computer installations throughout the rd world. Remote installations will depend upon spare components stocked in duplicate or plicate and thereby become less dependent upon highly trained service personnel. Sec-dly, there seems little reason to expect the number of manufacturers and the variety of mputers to decrease. In fact, because of the easing of maintenance it seems likely that e companies will challenge the monopolies or near monopolies now enjoyed by some of the jor companies in developing nations.

A burgeoning number of users and increased demand for software seems likely to follow e increase in computer installations. The continuing diversity of computers expected in e third world would indicate that software tailored exclusively for a single type of mputer would fail to meet users' needs. Yet our current software is only partially equate and is becoming dated as the mini-computer movement begins to take effect, and the cro-processor demands its place. Some encouraging steps have already occurred: SPSS has en implemented on the PDP 11 series and the Data General Eclipse, and there are plans for MINI-BMD package. In addition, there are rumors that both Digital Equipment Corporation d Data General will soon offer 32 bit minicomputers. Such a development would expedite e adaptation of existing software packages.

In our view the greatest need of the third world is for better technical support, in-uding training and development of local expertise and improved communications.

One of the most important contributions for the third world would be the regularly up-ted publication of a catalog describing available software for statistical analysis. The talog should describe software capability, indicate the type of hardware for which it uld be suitable, the user documentation available, and where and how the user might tain the software. Ideally the editors of the catalog would be experienced in both data ocessing and statistics so that they could test and evaluate the software in accordance th evaluation standards now being developed.

The industrialized world has the opportunity of making a major contribution to the rd world by compressing the experiences of the last decade into one or two years. If equate communications are established, it is likely that benefits will flow in both rections.

BANGKOK METROPOLITAN WATER WORKS AUTHORITY: SPECIAL REPORT ON ELECTRONIC DATA PROCESSING SALARY SURVEY (April 1972), cited in Thavisakdi Thangsuphanich, "Problems in (DPC) Management in Government," Computer Journal (Thailand), I:1, 28 (November, 1972).

BUHLER, ROALD (1976). "Some Portability Issues Affecting the P-Stat System," Proceedir of the Ninth Interface Symposium on Somputer Service and Statistics, 93-95.

FRANCIS, IVOR and J. SEDRANSK (1976). "Software Requirements for the Analysis of Sutve in Proceedings, International Biometric Conference. Boston. 2, 228-252.

KEYFITZ, NATHAN and WILHELM FLIEGER (1971). Population: Facts and Methods of Demograph San Francisco: W.H. Freeman & Company.

MATTHEWS and POMERANZ (1975), "Upgrading to a Smaller Machine," Datamation (May), 73-75

WHITE, JAMES W. and G. DAVID RIPLEY (1977). "How Portable are Minicomputer Fortran Programs," Datamation (July), 105-107.

YATES, FRANK (1971). "The Use of Computers for Statistical Analysis: A Review of Aims Achievements." Proceedings of the 38th Session of the International Statistical Institute, Bulletin of the International Statistical Institute, 39-53.

BIOGRAPHIES

Henry Elkins received a Ph.D. in sociology from the University of Chicago in 1970. is currently involved in international family planning research at the Center for Popul and Family Health at Columbia University. He also serves as Consultant for the Populati Council, where he was Staff Associate 1972-76. He has had overseas assignments in Colom Bangladesh, Mexico, El Salvador and Thailand.

Victor Matthews was Head of the Computer Centre and Staff Associate at the Populati Council, 1971-77. He is currently Director of Computer Systems at Real-Time Systems. H received a Ph.D. in sociology from Washington State University in 1971.

Joanna Pomeranz was Manager of the Computer Centre at Population Council 1971-77 af having worked at the New York Medical College and IBM World Trade. She is currently Man of Computer Systems, Real-Time Systems. She holds a B.A. from the University of Michiga London School of Economics.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
roceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
ld at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

XTALLY - A MULTI-DIMENSIONAL CROSS TABULATION PACKAGE IN RPG-2

Michael R. Lackner
United Nations Statistical Office
New York, New York 10017

ABSTRACT

XTALLY produces fully-titled cross-tabulations of up to 7 dimensions and 100,000 cells, each summing 1 or 2 variables, complete with all sub-totals, percentages of over-all total, and automatic inflation/deflation of values proportionate to 1 or 2 pre-specified overall totals. The system requires only 24K byte primary storage, and 2 megabyte disk storage. XTALLY does not depend on either compilation or sorting, and only 3 statement formats are used with only two major procedures so users can learn XTALLY in only a few hours.

Data record formats and category-sets are recorded in a disk-stored dictionary of variable names and locations, category-set names, and category limits and titles.

A particular cross tabulation is specified with a single statement naming category-sets in hierarchical order for rows and for columns, identifying the 1 or 2 accumulation variables, and associated inflation/deflation totals if desired. Tabulation proceeds at from 15,000 to 150,000 records/hour, depending on the computer configuration and the dimensions of the table. Timing is a linear function of data file length.

XTALLY has been operational on the IBM System 3 since 1974 and the IBM System 32 since 1975, and has been used for survey or census processing in half a dozen countries. RPG-2 will enable its implementation on the IBM 360/370 (DOS), Honeywell-Bull 6000, ICL 2903, Hewlett-Packard 3000-II, Univac 9400, Burroughs 1700, NCR Century and Criterion Series, and other small to medium range computers. The portability of RPG-2, and the array and file access operations it offers, have led to its selection as the programming language for developing an edit package and a data-base package for census data processing on small computers.

1. INTRODUCTION

XTALLY was developed to provide an easy-to-use statistical cross tabulation capability computer users whose applications are mainly or entirely programmed in RPG-2 or whose rdware/software configurations cannot implement cross-tabulation packages requiring BOL, FORTRAN, BASIC or other compilers and more than 32K byte primary stores. The first rsion of XTALLY, completed in early 1974, runs on a 16K byte IBM System 3 or System 32 th disk. Later and much faster versions for IBM 3, 32 or 370, Honeywell-Bull 6000, L 2903, Hewlett-Packard 3000-II, UNIVAC 9400, Burroughs 1700 or NCR Century and Criterion ries require only 24 to 32K byte or equivalent primary storage plus 2 megabytes of disk orage.

XTALLY has been used for tabulating census, survey or administrative data in a numb of technical cooperation projects in developing countries supported by the UN Statistic Office. The system does not require on-site compilation and all programmes are interpre tive so that XTALLY has been installed by mail in most cases because of severely limited funds available for travel and demonstration. User instructions are stored on the XTALL disk, diskette or tape, enabling on-site generation of a brief but complete users-manual whenever one is needed.

Since use of XTALLY does not involve either compilation or sorting, and only 3 stat ment formats are used with only two major procedures, users can learn XTALLY in only a few hours.

The procedure for using XTALLY has 2 steps:

1. Define the source data record content and format, one card/record per item; and define category-sets -- value groupings -- to be used for various cross-tabulations, one card/record per category for each data item.

2. Specify particular cross-tabulations by naming, in one control card

   a. Names, in hierarchical order, of from 1 to 3 column category sets.

   b. Names, in hierarchical order, of from 1 to 4 row category sets.

   c. Names of one or two quantitative data items whose values are to be summed i the 2- to 7- dimensional cross-tabulation.

   d. If desired, arbitrary overall totals for either of the two quantitative ite can be specified to cause proportionate expansion of all subsidiary totals. This feature is intended to be used if the tabulated data comprise a sample

The first step in the XTALLY procedure -- data and category-set definition -- need be repeated while any number of cross-tabulations are produced. XTALLY stores the defin tions on the XTALLY disk and thus ensures consistency between different cross-tabulation that use one or more common category sets or accumulation items. Whenever it is desirab to modify or replace the data or category-set definitions, it is only necessary to re-ru the single procedure that stores them on the disk and prints them out for use in specify tabulations.

The major features of XTALLY are the following:

1. XTALLY cross-tabulations may include up to 99,999 individual cells. Each cell contains the summed values of either one or two specified accumulation variables If no accumulation variable is specified, only the count will appear in each cel If only one accumulation variable is specified, the count may also be included i wanted.

2. When the number of columns in the tabulation exceeds 15, XTALLY automatically divides the overall table into 'strips' of 15 columns each, repeating the row titles for each 'strip' to enable proper alignment or independent use. Rows con tinue vertically until all row cross-categories are complete.

3. Quantities are converted to percentages -- nearest 1/100 of 1% -- and automatically printed in identical format following print-out of the quantitative table.

4. 'Estimated' values are automatically produced if arbitrary overall totals for accumulation items are supplied in the control card.

5. Sub-totals and percentages of overall total are automatically included at all hierarchical row and column levels.

6. Each column in the printed cross-tabulation is titled according to all three hierarchical column categories, and each row is titled according to all four row category sets to ensure proper identification of values.

7. Extra copies of XTALLY cross-tabulations are produced at printer speed, without re-calculation of totals.

8. A copy of User Operating Instructions can be printed from the XTALLY disk any time one is wanted.

2. PLANNING AND PREPARING FOR TABULATIONS WITH XTALLY

The XTALLY cross-tabulation system is designed to be used directly by the statistician or analyst. Using XTALLY does not require any computer programming; the three forms used to define data records, establish category limits and category-sets, and to specify tables are intended to be learned and used first-hand by the statistician or analyst.

The tabulations wanted from a file of data are often specified by proforma or narrative descriptions. Such specifications are easily used as a basis for completing XTALLY forms for category-set definition or table specification, but they are not necessary. The following steps for planning and preparing tabulations can incorporate reference to such additional specifications as table proforma, but the tabulation planning chart can be produced and used independently.

2.1 Step 1 - Data Definition. This first step is the simple and obviously necessary one of identifying the items in the data record that will be used in some way to produce tables. Items that are not in any way used need not be defined: for example, family name is not used in census data tabulation and its position in the record or the values it may assume need not be established in preparing for tabulations.

Items that must be named and whose start positions and end-positions in the record must be identified are those that are used in either of the following ways:

1. accumulation variables: quantitative items (such as number of children ever born, amount of money earned or spent, number of days worked, etc.) that might be summed within categories

2. categorizing variables: quantitative or qualitative variables whose individual values (or sets of values) identify categories within which sums or counts might be accumulated.

A single item, such as age, expenditures, earnings, or number of children ever born, might be used both as an accumulation variable and as a categorizing variable. One data item may be part of another data item; for example, the first digit of a two-digit variable such as age-in-years may be one data item and the full two-digits may be another data item.

Any item that might be used as an accumulation variable or as a categorizing vari must be given a unique name. The name must consist of three alphabetic characters. T format for assigning data item names and stating start and end positions in the record given in the operating instructions.

2.2 Step 2 - Category Set Definition. A single category is established by a cla fying value, or set of values, of one of the categorizing variables. For example, a category might consist of the values 00, 01, 02, 03 and 04 for a two-digit data item named AGE; another category might consist of the value AABB for a four-character data item named COD.

A category-set is a set of separate and distinct categories that together account all possible values of one of the categorizing variables such that any particular valu the variable is in one and only one category of the set. For example, 20 five-year ag categories 00-04, 05-09. 10-14 . . . 95-99 may be a category set for the variable AGE. Another category set for the variable AGE might be 100 single-value categories 00, 01, 02, . . . , 99. A third category set for the same variable AGE might be 00-13, 14, 15 17, . . . , 39, 40-45, 46-99.

A single category set classified data along one dimension. Two category sets cla sify along two dimensions, so that a set of 20 age categories, for example, and a set four marital-status categories together yield 80 cross-classifications, or cells. Usi XTALLY, up to seven different category sets can be used to cross classify for any one table, yielding up to 99,999 cells; each cell may contain summed values for one or two accumulation variables, for one accumulation variable, or for the record count alone.

It is usually desirable to include sub-totals in tables of cross classifications. example, a table showing number of persons by sex, age and marital status usually inclu not only sums of never-married males and for each age group, but also the total of neve married males of all ages. For this reason, every XTALLY category set is automatically extended to include a "total" category. All sub-totals and grand totals are automatica produced: the sums accumulated in each of the individual categories are summed togethe to produce the "total" for the category set. The format for establishing category-sets given in the operating instructions.

2.3 Step 3 - Preparing a Tabulation Chart. For planning and preparing to use XTA it is useful to express the desired tables in chart form as follows:

1. Label the columns of the chart with the identification numbers or codes of the individual tables so that each column is identified with one table.

2. Label the rows of the chart with the names of the individual category sets followed by the names of the individual accumulation variables. Next to the name of each category set put the number (count) of individual categories in t set plus one (total).

3. Specify the format and composition of each table:

   a. locate the intersections of the table column with the rows labeled with category sets included in the table and the accumulation variables summed in the table;

b. for each category set used in the table, indicate its use as a row-heading or a column-heading category set by R or a C, and indicate its hierarchical position as a row- or column-heading by a 1, 2, 3 or 4 following the R or C;

c. for each accumulation variable summed in the table, indicate its relative print position in the table cells by a 1 (top) or a 2 (bottom).

4. Calculate the total number of columns in the table, which is the product of the numbers (recorded in 2 above) next to the names of the column category sets (C1, C2, or C3).

5. Calculate the total number of rows in the table, which is the product of the numbers next to the names of the row-category sets (R1, R2, R3, or R4).

6. Calculate the total number of cells in the table, which is the product of total columns x total rows. (This must not exceed 99,999, and a Halt will occur if it does.)

mple:

ume AGE01 is a category set of 100 single-years of age, and AGE05 is a category set of five-year age groups; SEX01 is a category set of the two sex categories; MAR01 is a egory set of five marital-status codes; EDU01 is a category set of four educational ainment codes; OCC01 is a category set of 80 occupational code groupings; IND01 is a egory set of 60 industrial code groupings; and STA01 is a category set of four activity tus groupings. Assume DAW is an accumulation variable measuring number of days worked t week, and CEB is an accumulation variable expressing number of children ever born. A rt for a set of tables using these category sets and accumulation variables might appear follows:

egory Sets:	Table 1	Table 2	Table 3	Table 4	Table 5
01 (101)	R2				
05 (21)				C1	C1
01 (3)	C1	C2		C2	
01 (6)	C2				
01 (5)	R1		R2		
01 (81)			R1	R1	R1
01 (61)		R1	C1		
01 (5)		C1			R2
. Variables:					
		1	2		2
		2			
sons (record count)	1		1	1	1
al Columns	18	15	61	63	21
al Rows	505	61	405	81	405
al Cells	9,090	915	24,705	5,103	8,505

The XTALLY control statements used to produce the tables would be:

	COLUMNS (cc 1-17)	ROWS (cc 33-55)	ACC. VARIABLES (cc 26-32)
Table 1	SEX01,.....,MAR01	EDU01,.....,AGE01,	
Table 2	STA01,.....,SEX01	.....,OCC01,.....,	DAW,CEB
Table 3	.....,.....,IND01	OCC01,.....,EDU01,	..,DAW
Table 4	AGE05,.....,SEX01	.....,.....,OCC01	
Table 5	.....,.....,SEX01	OCC01,.....,STA01	...,DAW

3. OPERATION

Data record formats and category-sets are recorded in a disk-stored dictionary of variable names and locations, category-set names, and category limits and titles. The automatic coupling of titles to category definitions eliminates an important source of error while also reducing the work required of a user to a minimum.

Tabulation proceeds at from 15,000 to 150,000 records/hour, depending on the compu configuration and the dimensions of the table. Timing is a linear function of data fil length.

The summary array is produced on the fixed disk and formatted and printed by a separate program. This not only facilitates reproduction of table printout but also provides the basis for extending the system to allow more flexibility in output format to provide additional functions of the one or two summed variables.

4. USES OF RPG-2

The portability of RPG-2, and the array and file access operations it offers, have led to its selection as the programming language for developing an edit package and a data-base package for census data processing on small computers. Important facilities provided by RPG-2 include the following:

a. logical and arithmetic operations on variables or arrays

b. easy direct access to disk-stored arrays and array segments

c. simple but powerful input and output format statements.

The most complex XTALLY programme, as an example, is coded with 507 RPG-2 statements. S of the statements are File Declarations, 8 are Array Declarations, 58 are Input Format a 33 are Output Format Statements. Of the 400-odd calculation statements, the great majo are concerned with calculation of array indices and their factors.

The original simple report-generator concept of RPG is still evident in RPG-2, but newer capabilities to deal with arrays in primary and secondary storage, coupled h the very large and fast disk stores available with new small computers have made -2 a very practical programming language for statistical computing. This has made possible to provide statistical software for small, so-called business-oriented puters, and XTALLY is an example of such user-oriented software.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interfa
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 19

CONSTRAINTS IN THE DESIGN AND IMPLEMENTATION OF INTERACTIVE STATISTICAL SYSTEMS FOR MINICOMPUTERS

Robert F. Ling
Department of Mathematical Sciences, Clemson University, Clemson, S.C. 29631

ABSTRACT

In this paper, attention is focussed on issues and problems relating to the design and implementation of interactive statistical systems (as opposed to batch systems or small batch or interactive programs) for minicomputers. In particular, constraints imposed by certain characteristics of existing minicomputers (such as size of main storage and data format) as well as related operating systems software and programming languages are discussed. Efforts to relax or eliminate these constraints may be considered as prospects for statistical systems for future generations of minicomputers.

Key words: Interactive statistical systems; minicomputer; statistical software design.

1. INTRODUCTION

During the past decade, the minicomputer industry has experienced an explosive peri of growth, in terms of technological advances and market volume. According to recent Da pro Research Corporation Reports, estimates of worldwide minicomputer market volumes are

1972 [1] $300 - $450 million

1975 [2] $800 million - $1.4 billion

1977 [2] $1.8 billion.

These figures are rather striking by themselves even if we do not take into account the rapid decrease in the cost of central processors. Kenney [10] wrote, "In 1966, for ample, the processor cost approximately $30,000, but six years later, 1972, its price wa only 20 percent of that cost, about $6,500." Monrad-Krohn [12] (1977) estimated, "The c tral processing element of a computer has decreased to the cost of about $20."-- of cours he was referring to the lower spectrum of present generation of micro computers.

During this period of explosive growth, technological advances in the hardware comp nents have far exceeded the development of software. The following quotes are fairly typ cal of current opinions about minicomputer software:

> "The present state of software development is far from being acceptable ... Develop ment of the software takes longer than anticipated and almost always the costs are more than expected. At times the finished product does not perform as expected, and there have been times when it didn't perform at all." [10, p. 76]

"Software, which had long received only cursory attention from the predominantly hardware-oriented minicomputer makers, is rapidly becoming the principal distinguishing factor between competitive product lines." [2, p. 70c-010-20d]

Given the state of general software development of minicomputers, it should be no surprise that existing statistical software for minicomputers is fragmented, localized, and of primitive. Some manufacturers (such as Hewlett-Packard) serve as the distributor of -contributed software, including statistical programs and systems. In such cases, the of quality control standards for contributed programs resulted in many library programs are low in quality, by any reasonable standards of evaluation. Portable statistical ems for minicomputers, interactive or not, are almost nonexistent. MiniBMD [5] is per- the first serious attempt at the creation of a portable, high quality, general purpose istical system specifically designed for minicomputers.

For the aforementioned reasons, instead of doing a survey of existing, non-portable, istical software, I shall consider some characteristics of portable statistical software minicomputers in the immediate future by focussing on constraints imposed by such com- rs on the design and implementation of interactive statistical systems. In my opinion, ractive systems are of paramount importance in the effective use of statistics on mini- uters, and the effective design of such systems must pay close attention to the con- ints.

2. WHAT IS A MINICOMPUTER?

One agreement within the minicomputer industry is that there is disagreement as to what titutes a minicomputer. For the purpose of the present discussion, I shall use the do-definition "minicomputers are machines whose mainframes sell for less than $50,000 some other arbitrary figure)" in the the spirit minicomputers are defined in [2]. A ty- l system configuration costs two to four times the cost of the mainframe. There are no r cutoff values that separate minis from micros and midis (see e.g. [12, 15]). For ex- e, Interdata 8/32 is classified as a mini in [2] and a midi in [15]. Given the trend of easing computer power and decreasing cost, the next generation of minis will likely be arable to some of today's maxis in capacity and performance.

The most important distinguishing characteristic of a mini is its word length. A "ty- l" mini currently on the market has a 16-bit word length, although minis with word ths of as many as 32 bits or as few as 8 bits are not rare. For a minicomputer which is ble of supporting a moderately versatile interactive statistical system, we may consider following to be some of its "typical" characteristics:

Software support: a time sharing operating system. BASIC and/or FORTRAN compilers.

Data Format: 16-bit word length (and up).

Main storage: magnetic core having a maximum storage capacity of 32768 words (and up).

I/O control: DMA channel and multilevels of external interrupt.

Peripheral: disk pack or cartridge drives, tape drives and other standard I/O devices.

3. CHOICE OF COMPUTER AND INTERACTIVE SYSTEM DESIGN -- WHICH COMES FIRST OR SHOULD IT MATTER?

From the system designer's point of view, two general optimization approaches are possible:

(A) Consider an ideal design of an interactive system and then choose a computer with characteristics are most suitable for the implementation of that design.

(B) Given a computer and its associated software, design an interactive system which attempts to make optimal use of the available features and resources.

In practice, approach (A) is generally not available to the statistical system designer and judging from the characteristics of existing interactive statistical software for large and small computers, approach (B) appears to be the norm. As a result, most of them (e.g., IDA (BASIC Version) [11], isp [4], MIDAS [6, 7], SAS [13], SIPS [9], and SPEAKEASY [14]) achieve certain desirable features or local optimality at the expense of severely limited portability.

If we use the criteria for evaluating statistical software in [8, 16] as guidelines in designing an interactive system, then neither approach (A) nor approach (B) would be appropriate. Instead, the system designer should first consider the constraints imposed by the requirement of portability to choose the software language used to code the interactive system (e.g., at the present time, neither APL nor PL/I would be an appropriate choice because most minicomputers do not have an interpreter or compiler for these languages, although purely from a programming language point of view, they are in many respects better than their counterparts BASIC and FORTRAN which are widely supported.)

Our experience with existing interactive systems should have taught us a lesson about the importance of portability. Far too often, system designers (myself included) exhibit systems with many desirable features but unfortunately have to inform those who are interested in using the system that it cannot run under machine ABC or operating system XYZ without substantial conversion efforts. In order to consider a truly portable system, we are not only constrained to use BASIC or FORTRAN, but we must sacrifice certain features of a system if their implementation would require non-standard features of those languages. Similarly, other constraints imposed by minicomputers should be carefully considered before a system is designed or implemented.

4. CONSTRAINTS IMPOSED BY MINICOMPUTERS

The major categories of evaluation criteria and their dependence on the characteristics of a "typical" minicomputer can be summarized by figure 1. The diagram suggests that the partition size (which is generally a function of the primary core size) plays an important role in all aspects of a statistical system design.

Figure 2 gives a schematic representation of some typical implementations (using BASIC or FORTRAN as the source language) that further restricts the space available for active data and system parameters. In general, the use of FORTRAN places much greater constraints on the total size (and hence extensibility) of a system while the most favorable language for modularizing a large system (BASIC with CHAIN and COMMON) is likely to have severe portability problems. The constraints that effect each of several major evaluation items will be elaborated below:

4.1 User interface.

4.1.1. Date structure and size of active data. The most distinguishing feature be- a statistical system on a minicomputer and one on a maxicomputer is probably the total of the "active" arrays (variables addressable in the primary core). For a system run- on a maxicomputer with a 256K partition size, say, the space allocatable to active ar- generally exceeds the space on a mini allocatable to the entire system. Thus, in order ave the capability of analyzing a moderate to large dataset on a mini (where the raw must be accessed repeatedly, such as required in various residuals analyses) the system be structured to interface efficiently with data stored in secondary memory locations devices, whereas a maxi system may have sufficient space to place the entire dataset in . Moreover, a BASIC system without the COMMON feature will require explicit I/O to pass ables and system parameters among modules or subprograms, thereby exacting a heavy over- on the performance of the system.

Figure 1

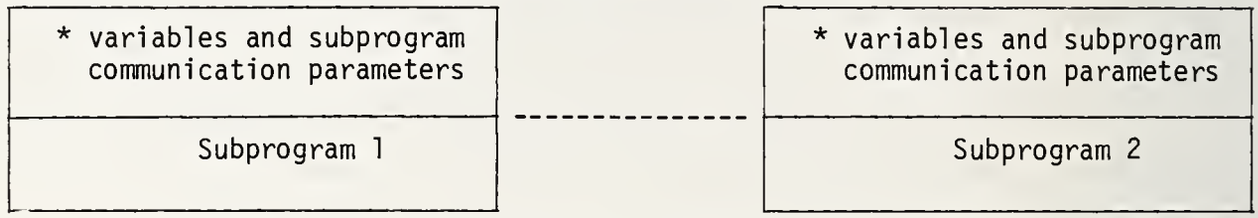
RELATION BETWEEN CONSTRAINTS
AND EVALUATION CRITERIA

Evaluation Criteria	Constraints: PARTITION SPACE UTILIZATION AND LIMITATION — SOURCE LANGUAGE OPERATING SYSTEM	WORD LENGTH	
INTERFACE			
Data Structure	X		X
Active Data			X
Command or Control Language			X
Level of Interaction	X		X
Internal Documentation			X
STICAL EFFECTIVENESS			
Versatility			X
Accuracy		X	X
EMENTATION			
Extensibility	X		X
Portability	X	X	X

Figure 2

EXAMPLES OF SOME TYPICAL IMPLEMENTATION
AND PARTITION SPACE UTILIZATION

Standard BASIC (without COMMON and CHAIN capabilities)

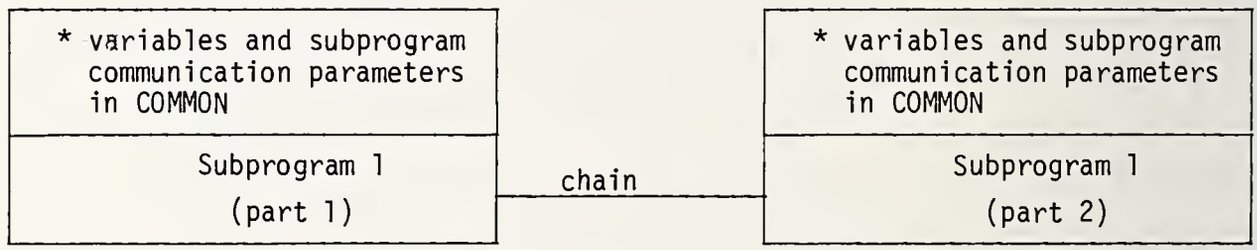


Explicit I/O is required to pass variables and parameters between subprograms

Size of source code for system virtually unlimited

Extensibility: good Portability: good

BASIC with COMMON and CHAIN (such as HP-2000 BASIC)



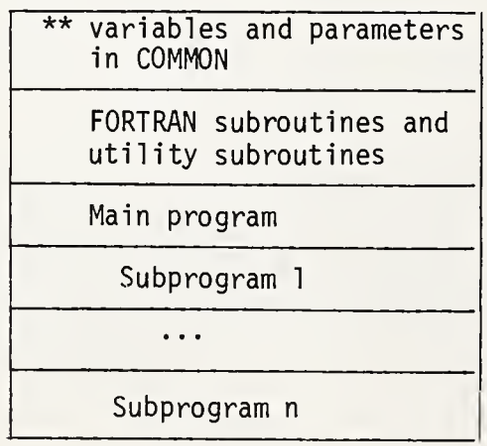
Even subprograms can be arbitrarily modularized through COMMON and CHAIN

Virtually unlimited size for source programs

Very small portion of partition needed for source

Extensibility: very good Portability: poor

FORTRAN Load Module (not overlayed)



high speed core for data, variables, and system parameters severely limited by siz of partition

versatility of system severely limited by the limited amount of space for subroutin

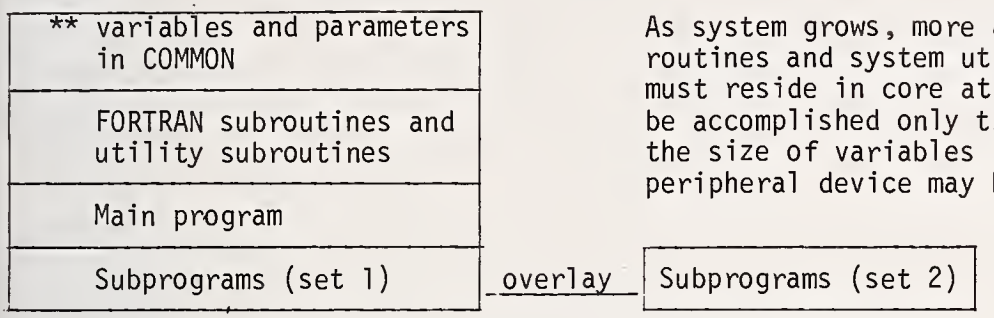
size of source code (function of load mod size) limited by size of partition

Extensibility: poor. Lack space. Also, must recompile main program and link

Portability: good if ANSI FORTRAN is use

Figure 2 (cont.)

AN Load Module (overlayed)



As system grows, more and more FORTRAN subroutines and system utility subroutines must reside in core at all times. This can be accomplished only through a reduction in the size of variables in COMMON. I/O on peripheral device may be necessitated

ce relative to partition size remains
ghly constant as system grows
imum usable space relative to partition
e diminishes as system grows

Extensibility: fair to poor

Portability: almost as good as non-overlayed

4.1.2 Command language structure. All interactive systems must have a command lan- structure. The syntax of the structure may range from simply a dictionary of COMMAND to one admitting flexible combinations of language phrases and arithmetic expressions. atter will require a parsing algorithm to interpret the command or control phrases. artition size of a minicomputer will greatly curtail the space allocatable to the algo- and thus will limit its complexity and generality.

4.1.3 Level of interaction between User and System. The minicomputer itself has re- ely small effect on this aspect of the software design. The source language used and ode of communication between the main (driver) program and subroutines (modules) and modules will determine the efficiency of the interaction (provided the system is opti- designed and coded for man-machine interaction). For example, of the two types of illustrated in figure 2, the one with CHAIN and COMMON is much more amenable to a ole structure for user-system interaction than its counterpart, the standard BASIC.

4.1.4 Internal documentation. Ideally, the user of an interactive system ought to le to access all relevant information and documentation about the system on line, with- ne necessity of a User's Manual or various reference manuals. In practice, no existing m accomplishes this ideal, though some (such as SPEAKEASY, with several hundred pages t in the HELP file, hierarchically organized in a tree structure) come much closer to ternally-documented system than others. For minicomputers, even considerable less than that in the SPEAKEASY system would be constrained by the limited partition size. only the most frequently accessed documentation can be kept in core while the others e retrieved from secondary or peripheral storage devices.

4.2 Statistical effectiveness.

4.2.1 Statistical versatility. The statistical versatility of a system is con- ed primarily by the partition space utilization as illustrated in figure 2, so that nstraint is much more severe for a FORTRAN system than one in BASIC.

A comment is perhaps necessary here to clarify the assertion that a system written in TRAN has greater constraints on added statistical capabilities than one written in BASIC. a FORTRAN environment, statistical as well as I/O tasks that are common to many

procedures (modules) are accomplished by a CALL SUBROUTINE statement within the module w the subroutines being called resident in core at all times. Thus, as a system grows, th will be more and more of such "utility" subroutines. In a BASIC environment, the implic subroutine call feature does not exist, so that often the identical codes (or codes with different names) are explicitly coded within each and every subprogram or module of the tem, as a matter of necessity imposed by the language. In theory, if we simulate this f of inefficiency in FORTRAN (by discarding the effective use of subroutines) then the ove lay structure in FORTRAN is no different from the chaining structure in BASIC insofar th programmer is concerned. However, it appears reasonable to assume that when one is work within a portable FORTRAN environment (having sacrificed many non-standard but more powe ful features) one is entitled to, and should, make effective use of the SUBROUTINE featu in FORTRAN while paying a price in the extensibility of a large system.

4.2.2 Numerical accuracy. The primary constraint is the word length of a minicom ter which limits the achievable numerical accuracy of the minicomputers. Typically, min computers do not have the option to perform computations in double-precision arithmetic while many statistical computational algorithms require double-precision to ensure a hig degree of accuracy. A secondary constraint may be considered to be the CPU speed of ari metic operations because algorithms capable of achieving a high degree of numerical accu at the expense of "number crunching" may have to be discarded in favor of less accurate, but much speedier algorithms.

4.3. Implementation.

4.3.1 Extensibility. The implementation of a system should make allowances for t types of modification or extension:

(A) Added system capabilities (new commands or procedures).
(B) Accommodations of user-supplied procedures or routines.

The feasibility and ease of implementing these depend heavily on the software langu used to code the system and to some extent on the operating system on which the package run. Typically such extensions are much more easily accomplished in BASIC (or any inter tive language) than in FORTRAN (which requires compilation, linking, and the creation of new load module for the entire system before execution of the new procedure can take pla At the present state of affairs, I would assess the extensibility of a FORTRAN system to moderately clumsy to fair for the system implementor, and difficult to impossible for th user. On the other hand, extending a system written in BASIC is generally simple and straightforward.

4.3.2 Portability. Among all of the evaluation criteria of a statistical system, portability is probably the most challenging one to satisfy as well as one which is much more restrictive than it may seem. The major constraint lies in the fact that even for commonly used languages such as BASIC and FORTRAN, different manufacturers of minicomput support different features of the languages). Consequently, to achieve portability, oft certain desirable features have to be sacrificed (e.g., efficient coding, efficient I/O, and optimal interrupt handling and error recovery) in order that the system can be run w out modification on different computers.

5. LOOKING AHEAD TOWARDS THE NEXT GENERATION

In this paper, I presented my impression of the constraints imposed by the present g eration of minicomputers on the design and implementation of interactive statistical sys tems. Given the present rate of technological advances and decrease in the cost of the hardware, it appears likely that the next generation of minicomputers will approach or s pass most of the present generation maxicomputers in capacity and performance. As a res many of the existing constraints will be partially or totally removed simply as a natura

equence of progress. However, constraints in the portability of software will likely in in the near future; and may be better or worse in the intermediate future, depending he demands of the "buyers" and the manufacturers' assessments of the needs of the existand potential market. In either case, the scientific computing community in general the statistical computing community in particular (both being small minorities in the uting market of consumers) will be unlikely to have any major impact on the manufacturhardware and software designs. Thus, even if it becomes technologically feasible to inate all of the constraints discussed in the paper for minicomputers, some of them will in because of the diversity of demands of different groups of users.

6. ACKNOWLEDGMENT

Preparation of this paper was supported in part by ONR Contract N00014-75-C-0451.

7. REFERENCES

All about minicomputers (June, 1972). Feature Report. Datapro Research Corporation, Moorestown, N.J.

All about minicomputers (September 1975). Feature Report, Addendum 1 (February 1976). Datapro Research Corporation, Dekran, N.J.

AVERY, K. R. and AVERY, C. A. (1975). Design and development of an interactive statistical system (SIPS). Proc. Computer Sci. and Statist.: 8th Ann. Symposium on the Interface. J. W. Frane, Ed., Health Sciences Computing Facility, UCLA, 49-55.

BLOOMFIELD, P. (1977). An interactive statistical processor for the Unix timesharing system. Proc. Computer Sci. and Statist.: 10th Ann. Symposium on the Interface. National Bureau of Standards, Gaithersburg, Maryland.

BUCHNESS, R. and ENGLEMAN, L. (1977). MiniBMD: A minicomputer statistical system. Proc. Computer Sci. and Statist.: 10th Ann. Symposium on the Interface. National Bureau of Standards, Gaithersburg, Maryland.

FOX, D. J. (1975). Some considerations in designing an interactive data analysis system. Proc. Computer Sci. and Statist.: 8th Ann. Symposium on the Interface. J. W. Frane, Ed., Health Sciences Computing Facility, UCLA, 61-5.

FOX, D. J. and GUIRE, K. E. (1974). Documentation for MIDAS, revised 2nd edition, Statistical Research Laboratory, the University of Michigan.

FRANCIS, I., HEIBERGER, R. M. and VELLEMAN, P. (1975). Criteria in the evaluation of statistical program packages. American Statistician, 29, 52-5.

GUTHRIE, D., AVERY, C., and AVERY, K. (1974). Statistical Interactive Programming System (SIPS), User's Reference Manual. Oregon State University Bookstore, Corvallis, Oregon.

KENNY, D. P. (1973). Minicomputers. New York: Amacon.

LING, R. F., and ROBERTS, H. V. (1975). IDA and user interface. Proc. Computer Sci. and Statist.: 8th Ann. Symposium on the Interface. J. W. Frane, Ed., Health Sciences Computing Facility, UCLA, 91-4.

[12] MONRAD-KROHN, L. (February 1977). The micro vs the minicomputer. Mini-Micro Syste 28-33.

[13] SERVICE, J. (1975). Adapting a batch-oriented statistical analysis system to inter active use: the case of SAS. Proc. Computer Sci. and Statist.: 8th Annual S posium on the Interface. J. W. Frane, Ed., Health Sciences Computing Facility UCLA.

[14] COHEN, S. and PIEPER, S. C. (1976). SPEAKEASY-3 Reference Manual Level Lambda IBM OS/VS Version. Argonne National Laboratories, Argonne, Illinois.

[15] THESIS, D. J. (February 1977). The minicomputer. Datamation, 73-82.

[16] VELLEMAN, P. and WELSCH, R. E. (1975). Some evaluation criteria for interactive st tistical program packages. Proc. Statist. Computing Section, American Statist cal Association, 10-2.

BIOGRAPHY

Robert F. Ling received a Ph.D. in statistics from Yale University in 1971. He was a Assistant Professor of Statistics at the Graduate School of Business, University of Chica 1970-75 before joining Clemson University as an Associate Professor in the Department of Mathematical Sciences. He designed and implemented IDA (Interactive Data Analysis, HP BAS Version) in the summer of 1972, and is currently a co-developer (with several people of th University of Chicago) of a portable FORTRAN Version of IDA. Ling is an Associate Editor the Applications Section and the Book Review Section of JASA.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
roceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
ld at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

INVITED CONTRIBUTION TO THE DISCUSSION
STATISTICAL PROGRAM PACKAGES FOR SMALL COMPUTERS

J. H. Maindonald
Victoria University of Wellington,
New Zealand

Designers of existing statistical systems have in most cases aimed too directly at viding capabilities at the level required by the ordinary user. Later attempts to ify the initial version of the command language in a way that will give needed flexi- ity then lead to highly complicated forms of statement. Such modifications will still satisfy the user who wants access to part only of the total computation so that he can it for his own purposes.

Rather one should begin by asking: "What are the optimal building blocks (primitive abilities) from which to piece together capabilities of the type that are finally re- red?"

many types of linear statistical computations suitable building blocks are:

an algorithm which, given X or X'X, finds the upper triangle matrix T (zeros below the diagonal) such that T'T = X'X;

) an algorithm for solving an upper triangle system of equations $T\underset{\sim}{c} = \underset{\sim}{d}$, and one for solving a lower triangle system $T'\underset{\sim}{g} = \underset{\sim}{h}$;

i) an algorithm for finding eigenvalues and eigenvectors of a symmetric matrix.

rix inversion would also be included, but for use only when the inverse is required for own sake. Various further capabilities might be added; for example one would like to be e to obtain from T the upper triangle matrix $\overline{T}$ which corresponds to permuting the columns X. Only the eigenvalue algorithm is at all complicated, and the list has already ex- ded far enough to cater for any of the matrix computations described in the textbooks on ssical multivariate analysis.

Similar considerations apply in the provision of facilities for manipulating data, and input and output.

No doubt the suggested capabilities could readily be provided within APL. But this is restrict the use of the final product to the limited number of installations where APL is ilable.

These ideas fit well with what I believe to be an excellent practical approach to the lding of a statistical system.

Initially the basic capabilities are made available as subroutines.

) Access is then provided to the subroutines by means of a rudimentary form of command language, which may consist largely of numeric codes.

i) Words replace numeric codes, giving a form of language that mirrors closely the mathematical or statistical operations involved. The level will be that of the "primitive capabilities" discussed earlier.

) Finally a facility is provided for grouping together a number of primitive commands in a single macro or super-command. This is used to provide immediate access to the type of command which is standard in existing systems. Options are catered for by allowing editing of the statements in any macro. A good editing facility will be essential.

The pattern of development thus follows closely the anticipated pattern of use. S
ges (ii) and (iii) make available a form of command language which will be useful to the
developers themselves in experimenting with the features which are required at level (iv
Documentation and testing will follow a sequence which should ensure a well-tested and t
oughly documented final product. Some users may find their equipment will not rise to a
level (iv) implementation; they may still be able to use the system at level (ii) or lev
(iii). Where a "hands on" type of operation is possible use at level (ii), aided by goo
step by step accounts of how to proceed and of the way in which any output should be use
may be an effective substitute for more sophisticated command language capabilities.

The new breed of hand calculators, of which the Texas Instruments SR52 and the Hewl
Packard HP67/97 are the first, are ideally suited for level (ii) type of use in handling
standard types of linear least squares and linear multivariate computations. Matrix ope
tions of the type discussed earlier will, where up to six variables are involved, fit on
HP67/97. I have not attempted to handle eigenvalue calculations; but I think that these
are within the capabilities of this equipment.

Facilities available on these very small machines are improving so rapidly that they may
very soon dominate the "small computer" scene. Time spent in coding and using algorithm
such small machines is in any case not wasted; it is an excellent training for anyone wh
hopes to code the same algorithms in Fortran.

Reference:

Maindonald, J.H. (1977). Statistical Computer Packages. CSIRO Division of Mathematics and Statistics Newsletter, No. 28, pp. 1-2.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

COMPUTING APPROACHES TO THE ANALYSIS OF VARIANCE FOR UNBALANCED DATA

Richard M. Heiberger and Larry L. Laster
University of Pennsylvania

ABSTRACT

Questions are raised on the appropriate analysis for cross-classified data with unequal and disproportionate sample sizes. A set of answers is offered.

Key words: Analysis of variance; unbalanced data.

1. PROLOGUE

The participants in this workshop were invited to respond to the following statement: In unbalanced data the sums of squares for effects (both main effects and interactions of all orders) are not orthogonal. No standard order for computation and presentation of the lines of the ANOVA table will be appropriate for all situations. For example in the model

$$Y_{ijkl} = \mu + \alpha_i + \beta_j + \gamma_k + (\alpha\beta)_{ij} + (\alpha\gamma)_{ik} + (\alpha\beta\gamma)_{ijk} + \varepsilon_{ijkl}$$

cogent arguments have been made for using each of the following sums of squares for the main effect of factor A:

$$R(\alpha|\mu)$$

$$R(\alpha|\mu,\beta)$$

$$R(\alpha|\mu,\beta,\gamma)$$

$$R(\alpha|\mu,\beta,\gamma,(\beta\gamma))$$

$$R^*(\alpha|\mu,\beta,\gamma,(\alpha\beta),(\alpha\gamma),(\beta\gamma))$$

$$R^*(\alpha|\mu,\beta,\gamma,(\alpha\beta),(\alpha\gamma),(\beta\gamma),(\alpha\beta\gamma))$$

where $R(.|.)$ indicates no restrictions have been imposed on the parameters of the model and $R^*(.|.)$ that overspecification of the parameters has been reduced by imposing restrictions on the parameters. Each of these sums of squares tests a different hypothesis about the parameters. The specification of some factors as fixed and others as random, or some as blocking factors and others as treatment factors, or nesting and crossing relationships among the factors helps reduce the number of potentially informative hypotheses but does not necessarily reduce the number to a unique one.

Any general computer program which claims to help in the analysis of unbalanced data

must at least implicitly take a stand on the statistical issues and select one or more of the potential sums of squares for computation. Some impose the choice of a single set of hypotheses which can be investigated by calculating only the sums of squares appropriate it. Others allow the user to override the default decision with one of a limited number options. Still others permit complete freedom of choice by requiring the user to specify set of dummy variables appropriate to the set of hypotheses of current interest.

The three questions that the panelists are asked to address in the context of three-w and higher unbalanced analysis of variance problems are:

1. Is there a statistically valid default decision short of fitting all possible orders main effects followed by all meaningful orders of interactions? On what features of the sign structure does it depend? Is there a default strategy for simultaneously determinin several interesting sets of hypothesis and computing their sums of squares?

2. If interactions are found significant is an automatic procedure for splitting the des into more homogeneous subdesigns feasible?

3. What is the appropriate criterion for the testing of hypotheses? The hypotheses test by the F statistics are orthogonal even with unbalanced data if the sums of squares for e line of the ANOVA table is computed by adjusting for all lines above it and ignoring all lines below it. The set of contrasts associated with almost any other set of sums of squares is not orthogonal and therefore open to ambiguity of interpretation.

2. EPILOGUE

Following the formal presentations and discussion and the informal continuations we h answered for ourselves some of the questions we raised. A fuller exposition of our posit will appear (Heiberger and Laster, 1977). Our answer is not to be taken as the consensus the opinions expressed at the workshop.

We distinguish between hypotheses about population parameters and contrasts of sample estimates used to test the hypotheses. This enables us to resolve an unfortunate phrasing common in the literature and used above in the invitation. We note that the null hypothes must be chosen prior to selection of the sample and must not depend on the observed sample frequencies. We therefore find confusing statements of the form: The sum of squares base on a specific set of contrasts tests a null hypothesis which is a function of observed sam ple size. Once we recognized that power functions can, and indeed should, depend on sampl frequencies we were able to rephrase that statement to: The power function of the sum of squares based on a specified set of contrasts has its minimum at points other than ones sa tisfying the null hypothesis. We now can recognize that certain types of sums of squares are inappropriate for testing the null hypothesis because they do not distinguish well between situations which do and do not satisfy the null. It is still accurate to say that they test the originally stated null hypothesis.

We personally would use the following sequence for testing main effects and interactio in the three-way design. We would first test for the $(\alpha\beta\gamma)$ interaction by using $R((\alpha\beta\gamma)|\mu,\alpha,\beta,\gamma,(\alpha\beta),(\alpha\gamma),(\beta\gamma))$. If $(\alpha\beta\gamma)$ is determined not to be significant, we would pr ceed to test for the two-way interactions by $R((\alpha\beta)|\mu,\alpha,\beta,\gamma,(\alpha\gamma),(\beta\gamma))$. Should all three $(\alpha\beta)$, $(\alpha\gamma)$, and $(\alpha\beta\gamma)$ be determined not to be significant the A effects, if present, will uniquely defined. We would then test for main effects by $R(\alpha|\mu,\beta,\gamma,(\beta\gamma))$.

These recommendations observe the marginality constraint that the residual from projec tion onto the AB space must be orthogonal to the A subspace. We might also consider using $R(\alpha|\mu,\beta,\gamma)$ when in addition $(\beta\gamma)$ is not significant; $R(\alpha|\mu,\beta)$ when γ and $(\beta\gamma)$ are not sign ficant; or $R(\alpha|\mu)$ when β,γ and $(\beta\gamma)$ are not significant. These last three options increas

he power of the test for (still uniquely defined) A effects.

In the presence of interaction involving A (either $(\alpha\beta)$, $(\alpha\gamma)$, or $(\alpha\beta\gamma)$)we note that he A effect is not uniquely defined. For example,in the two-way case

$$y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \varepsilon_{ijk}$$

he main effect α_i^* in the presence of interaction

$$\alpha_i^* = \alpha_i + (\alpha\beta)_{i.} = \sum_j t_{ij}(\alpha_i + (\alpha\beta)_{ij}) \quad \text{with } \sum_j t_{ij} = 1$$

s a function of the t_{ij} weights. When $\gamma_{ij} = 0$ for all i and j (that is, noninteraction) he dependence of the A effects on the a priori definition of t_{ij} vanishes.

If one is willing to accept the meaningfulness of a definition of main effects in the resence of interaction, the main effect can be tested by collapsing the tables of $_i + (\alpha\beta)_{ij}$ effects and cell frequencies to the A margin using the definition of the t_{ij} eights to get

$$\hat{\alpha}_i^* = \hat{\alpha}_i + (\hat{\alpha\beta})_{i.} = \sum_j t_{ij}(\hat{\alpha}_i + (\hat{\alpha\beta})_{ij})$$

$$n_{i.}^* = \left[\sum_j \frac{t_{ij}^2}{n_{ij}}\right]^{-1}$$

here $n_{i.}^*$ are effective sample sizes of the A effects estimates. We would then compute he one-way ANOVA of the $\hat{\alpha}_i + (\hat{\alpha\beta})_{i.}$ by $R(\alpha^*|\mu)$. When the a priori $t_{ij} = 1$ this rocedure is equivalent to computing $R^*(\alpha|\mu,\beta,(\alpha\beta))$.

If the definition of $\alpha_i + (\alpha\beta)_{i.}$ is not acceptable the only alternative is to xamine individual cell means.

3. ACKNOWLEDGMENT

)r. Heiberger's work is supported by National Science Foundation grant MCS-75-13994-A01.
)r. Laster's work is supported by Robert Wood Johnson Foundation grant #1947.

4. REFERENCE

EIBERGER, RICHARD M. and LASTER, LARRY L. (1977). "Maximizing power in the higher-way ANOVA." Proceedings of the Statistical Computing Section, American Statistical Association (to appear).

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

BMD AND BMDP APPROACHES TO UNBALANCED DATA

James W. Frane
Health Sciences Computing Facility, UCLA, Los Angeles, Ca. 90024

ABSTRACT

Appropriate treatment of balanced and unbalanced data is a function of the circumstances and the research questions being asked. BMD and BMDP provide a wide variety of approaches, but also report (as standard results) tests of hypotheses that are independent of cell sizes, as recommended by several authors. The same (orthogonal) hypotheses are tested for unequal cell size problems as are tested for equal cell size problems. Repeated measures (BMDP2V) and mixed model (BMDP3V) problems with unequal cell sizes are given special treatment. (P3V will be distributed for the first time in the fall of 1977.)

Keywords: ANOVA; contrast; hypothesis; mixed model; repeated measures; unbalanced

1. INTRODUCTION

The panelists for this workshop have been asked to respond to a two-page statement by Richard Heiberger and Larry Laster on unbalanced data for three-way and higher-way analysi of variance. Presumably, the two-way problem is thoroughly understood. One might suppose that this is so, given recent articles such as those by Kutner (1974), Speed and Hocking (1976), Green, Heiberger and Laster (1976), and the book by Searle (1971). While I suspec that most of these (and other) authors reasonably understand each other, I don't believe that general users of analysis of variance understand what is going on. This belief is based on the inquiries I receive regarding BMD, BMDP and other software.

The importance of unbalanced designs cannot be over emphasized. While many designs begin balanced, they frequently end up unbalanced. Moreover, the inclusion of covariates makes sums of squares nonorthogonal even for equal cell size problems. On the other hand, if a design is nearly balanced, several computing schemes provide approximately the same results. When the design is severely imbalanced due to missing data, results from any computing scheme can be seriously biased if the occurrence of missing data is related to the (unobserved) values of the dependent variable. When covariates are used it is important to investigate whether the distribution of the covariates is related to the analysis of variance design (as highlighted in Lord's paradox, 1967). When the covariates are random variables (which is frequently the case in the behavioral and health sciences), we can screen them by using them as dependent variables in an analysis of variance.

The key to understanding the BMD-BMDP approach is to consider the parameters of a model. For the two-way model we have

$$EY_{ijk} = \mu + \alpha_i = \beta_j + \gamma_{ij}$$

with some authors imposing the "usual" constraints

$$\Sigma\alpha_i = 0, \Sigma\beta_j = 0, \Sigma_i \gamma_{ij} = 0, \Sigma_i \gamma_{ij} = 0.$$

Placing constraints on the parameters disturbs some statisticians. If constraints are not desirable, then the model

$$EY_{ijk} = \mu_{ij}$$

can be used as in Kutner (1974). I much prefer this notation. Not only does it eliminate the need for constraints, but it also illustrates the fact that any use of such constraints in computer programs is made because of the numerical algorithm chosen (not because of the hypotheses tested) and that another algorithm could be used that does not involve constraints. There is no need to overparameterize the model, either for purposes of computation or exposition. Models without interaction can also be stated without constraints. For example, in the two-by-two case, we can state

$$EY_{ijk} = \mu + (3-2i)\alpha + (3-2j)\beta .$$

In general, there are many parameterizations (e.g., orthogonal polynomials) available that do not involve overparameterization or constraints.

The main effect hypotheses tested in BMD and BMDP correspond to those of Yates (1934) and to those labeled A and B by Kutner, and H1 and H2 by Speed and Hocking (1976). (Interaction hypotheses are defined the same way by virtually everyone.) They are also the hypotheses most recommended by these authors. Why? These hypotheses do <u>not</u> depend on cell sizes:

$$A: \Sigma_j \mu_{ij} = \Sigma_j \mu_{kj} \quad \forall \ i,k$$

$$B: \Sigma_i \mu_{ij} = \Sigma_i \mu_{ik} \quad \forall \ j,k .$$

These are exactly the same hypotheses that are tested (the same models that are considered) by virtually everyone when the data are balanced. As default models (or hypotheses), they have the great advantage that they can be stated exactly before the experiment is performed. Hypotheses that are functions of the cell sizes are unknown until all the data are gathered, and can in fact be random variables: if the availability of data had been different, the hypotheses would have been different. Hypotheses that are functions of cell sizes are usually unacceptable for experimental data and are not used in BMD and BMDP programs. Searle (1971, p. 317) notes that "This dependence of hypotheses on the structure of available data throws doubt on the validity of such hypotheses."

On the other hand, the sums of squares and mean squares corresponding to the hypotheses tested in BMD and BMDP are not orthogonal. This disturbs a number of people who prefer to partition the "total" sum of squares into a sequence of orthogonal components.

The orthogonality of the hypotheses is another question. For equal cell sizes, most schemes test hypotheses for the two-by-two case defined by setting the following contrasts equal to zero:

$$\text{mean: } \mu_{11} + \mu_{12} + \mu_{21} + \mu_{22}$$

$$\text{A: } \mu_{11} + \mu_{12} - \mu_{21} - \mu_{22}$$

$$\text{B: } \mu_{11} - \mu_{12} + \mu_{21} - \mu_{22}$$

$$\text{Interaction: } \mu_{11} - \mu_{12} - \mu_{21} + \mu_{22}$$

These contrasts are orthogonal in the sense that the inner products of the coefficient for distinct pairs of these contrasts are zero. Assuming that the availability of data is unrelated to the hypotheses of interest, orthogonality of hypotheses seems to be best defined in terms of the inner products of the coefficients for the contrasts that define the hypotheses.

Consider the following table of expected cell means:

1	-1
-1	1

Using the usual orthogonal contrasts to define hypotheses, the interaction contrast is four and the main effect contrasts are zero. For equal cell sizes, the sum of squares for main effects and interaction are not orthogonal. However, the tests for main effects are correct in the sense that the size (true alpha value) is as advertised. We are <u>not</u> led to erroneously believe that there are main effects in spite of the fact that main effect sums of squares are not independent of the interaction sum of squares. On the other hand, if orthogonal sums of squares are used the main effect contrasts are <u>not</u> orthogonal to the interaction contrast and so there appear to be both main effects and interaction.

Searle (1971), Green, Heiberger and Laster (1976), and others have used the R ( )-notation. As Speed and Hocking (1976) have noted, the R ( )-notation does not indicate the hypotheses being tested and many people misinterpret tests based on computing schemes that result from its use. Unfortunately, the Green, Heiberger and Laster paper and the statement of the problem to this workshop use the R ( )-notation. Indeed, the R ( )-notation yields rather clean statements of the way things are computed but does not state what is being tested. This notation easily lends itself to an orthogonal decomposition of the "total" sum of squares. This may sound like an "orthogonal solution" (Green, Heiberger and Laster), but as we have shown above, there are two mutually exclusive aspects of orthogonality for unbalanced data. Since BMD and BMDP test orthogonal hypotheses, it is not correct to refer to the solutions provided by these packages as nonorthogonal. It seems best to avoid ambiguity by referring only to the sums of squares or the hypotheses, rather than to the ambiguous term "solution."

Why is the R ( )-notation used at all? It leads to testing hypotheses that are functions of cell sizes. Searle (1971, p. 317) says that this might lead to valid F-statistics "only if the N_{ij}'s (as they occur in the data) are in direct proportion to the occurrence of the elements of the model in the population." Some authors like to go into great detail regarding computing procedures with the hope that this makes the resulting analysis clear. I don't believe it is necessary to go into great detail on the computing algorithm in analysis of variance any more than a user of principal components in a statistical package needs to know how to compute eigenvalues. What the user needs is a clear

statement of the hypotheses tested and the ability to specify other hypotheses to suit special needs.

In the Heiberger and Laster statement posing the problem for this workshop, mention is also made of an R ( )-notation to be used when "overspecification of the parameters has been reduced by imposing restrictions on the parameters." This would permit specification of orthogonal hypotheses in the general framework of the R ( )-notation, but we again recommend against it since it focuses attention on the computing procedure rather than the hypotheses. Also, imposition of constraints is most unfortunate since it is unnecessary and makes the discussion clumsy. When talking to a client, I don't find it easy to discuss the problem in terms of the computing procedure. I begin with a statement of the problem in English and translate it to a model and set of hypotheses. The translation to a computing algorithm is done by the computer program.

2. SPECIFIC ANSWERS TO THE HEIBERGER AND LASTER QUESTIONS

The questions were posed in the framework of the R ( )-notation and with the suggestion of hypotheses dependent on cell sizes. My general recommendation for experimental data is to test the same hypotheses for both unbalanced and balanced data. The availability of data should not (usually) affect the questions being asked. Sequential sums of squares methods test hypotheses that depend on cell sizes and should not (usually) be used for experimental data.

When interactions are significant, the analysis of variance table for any computing scheme or set of hypotheses must be viewed carefully. Consideration must be made of exactly what the research questions are. Computer programs are helpful to the statistician, but (as always) the same program can be dangerous to the untrained user. There are several possibilities for testing main effects (none of which should ordinarily depend on cell sizes) even in simple two-by-two problems. Here are a few:

a. The hypothesis $\mu_{11} + \mu_{12} = \mu_{21} + \mu_{22}$ is interesting because the first subscript refers to a method and the second refers to a two equally important laboratories that must use an identical method in future work. This is the hypothesis tested in most computer programs for balanced data. In BMD10V and BMDP2V this hypothesis is also tested for unbalanced data.

b. Each factor represents absence or presence of a drug. Given interaction, the main effect of the first drug might best be tested via the hypothesis $\mu_{11} = \mu_{21}$; i.e., the main effect for the first drug is tested without using the second drug. This is important in showing efficacy or safety of a particular drug. When patients are already being treated for another ailment, we may also be interested in the hypothesis $\mu_{12} = \mu_{22}$ (efficacy or safety of the first drug while using the second drug).

c. As in (b), we may be interested in hypotheses of the form $\mu_{11} \leq \mu_{21}$ and $\mu_{12} \leq \mu_{22}$: Regardless of whether the second drug is used, is it better to use the first drug than not to?

d. The columns are not equally important, so we test the hypothesis $p\mu_{11} + q\mu_{12} = p\mu_{21} + q\mu_{22}$ where p and q are usually prespecified and not dependent on the availability of data. This is similar to (a) above except that the laboratories are not equally important.

There are other examples, but these should suffice to show that it is frequently necessary to get complete control of hypotheses, as in the general linear hypothesis program BMD10V (formerly called BMDX64).

3. THE LAST QUESTION

"What is the appropriate criterion for the testing of hypotheses? The hypotheses tested by the F statistics are orthogonal even with unbalanced data if the sums of squares for each line of the ANOVA table is computed by adjusting for all lines above it and ignoring all lines below it. The set of contrasts associated with almost any other set of sums of squares is not orthogonal and therefore open to ambiguity of interpretation."

Searle (1971, pp. 306-312) gives the hypotheses tested by such a sequential sums of squares procedure. For the two-by-two case, the first two hypotheses are

$$\text{mean:}\quad N_{11}\mu_{11} + N_{12}\mu_{12} + N_{21}\mu_{21} + N_{22}\mu_{22} = 0$$

$$\text{row:}\quad \frac{N_{11}\,\mu_{11} + N_{12}\,\mu_{12}}{N_{11}+N_{12}} - \frac{N_{21}\mu_{21} + N_{22}\mu_{22}}{N_{21} + N_{22}} = 0 \quad .$$

The inner product of the coefficients for these contrasts is

$$\frac{N_{11}^2 + N_{12}^2}{N_{11} + N_{12}} - \frac{N_{21}^2 + N_{22}^2}{N_{21} + N_{22}}$$

which is not in general equal to zero so the hypotheses are <u>not</u> orthogonal. For unequal cell sizes, we have a paradox: orthogonal sums of dquares do not yield orthogonal hypotheses and orthogonal hypotheses are not tested by orthogonal sums of squares.

The discussion for three-way and higher-way ANOVA can be obtained as a generalization of our discussion for the two-way ANOVA.

4. REPEATED MEASURES

Repeated measures designs are frequently used in the behavioral and health sciences. They are used whenever multiple measurements of the same dependent variable are made for each subject. Repeated measures designs are similar to split plot designs. Consider the following simple experiment: A group of patients is randomly divided into two groups. The first group receives placebo and then treatment and the second receives treatment and then placebo. Group effect is synonymous with order effect. Let the outcome be denoted by Y_{ijk} where i refers to order, j refers to treatment, and k refers to patient. Let $EY_{ijk} = \mu_{ij}$. Some hypotheses of interest are

$$\text{order:}\quad \mu_{11} + \mu_{12} = \mu_{21} + \mu_{22}$$

$$\text{treatment:}\quad \mu_{11} + \mu_{21} = \mu_{12} + \mu_{22}$$

$$\text{interaction:}\quad \mu_{11} + \mu_{22} = \mu_{12} + \mu_{21}$$

which are the same ones used in the fixed effects case. However, BMDP2V insists on complete data for each patient because we are working with paired comparisons. Since the model no longer contains fixed effects only, a different computing procedure is required. We can reformulate the problem (i.e., transform the data) as

$$Z_{ij} = Y_{i1k} + Y_{i2k}$$

$$W_{ik} = Y_{i1k} - Y_{i2k}$$

The Z's are used in a two-sample t-statistic to test the order effect. The W's are also used in a two-sample analysis whose main effect corresponds to the drug vs. order interaction and whose "grand mean" effect is the treatment effect. Z and W are, of course, the zero and first order orthogonal polynomial decomposition for the repeated measures (trial) factor. When there are three levels of the repeated measures factor, we use three orthogonal polynomials, etc.

When there are two repeated measures factors, the same basic principles are applied. Suppose, for example, that we have two drugs and that each has an associated placebo treatment. Each of the two repeated measures factors has treatment and control. Let us ignore the order effect and include another effect, sex of patient. Let the outcome be denoted by $Y_{ijk\ell}$ where i denotes sex, j is drug one, k is drug two, and ℓ is patient.

$$EY_{ijk\ell} = \mu_{ijk\cdot}$$

To test drug one, we use the linear combination

$$Y_{ill\ell} + Y_{il2\ell} - Y_{i2l\ell} - Y_{i22\ell} \cdot$$

The drug interaction test uses

$$Y_{ill\ell} - Y_{il2\ell} - Y_{i2l\ell} + Y_{i22\ell} \cdot$$

The sex effect uses

$$Y_{ill\ell} + Y_{il2\ell} + Y_{i2l\ell} + Y_{i22\ell} \cdot$$

5. GENERAL MIXED MODEL

Repeated measures problems are a special class of the general mixed model. For repeated measures problems in BMDP2V, imbalance is allowed in the between group factors, but data must be complete for each case (patient, subject, or whatever the experimental unit is). When such a balance is not possible or when there is more than one random factor with any kind of imbalance, the general mixed model is usually required. For this purpose, a new program, BMDP3V, is being prepared by Robert Jennrich and Paul Sampson to be released in the Fall of 1977. In BMDP3V you can choose either maximum likelihood or restricted maximum likelihood estimation. Being a very general program, it is not intended to replace other programs that handle more elementary problems directly.

6. EMPTY CELLS AND SEVERE IMBALANCE

Some problems have empty cells by design. When such designs are used, it is frequently assumed that one or more interaction terms are zero or negligible in order to get an error sum of squares defined with adequate degrees of freedom. Sometimes empty cells are accidental and interactions are not assumed to be zero. What can packaged programs do in this case? Consider the following elementary example: Two rows and three columns with the cell corresponding to μ_{11} empty. The hypothesis for no row effect is

$$\mu_{11} + \mu_{12} + \mu_{13} = \mu_{21} + \mu_{22} + \mu_{23}$$

and cannot be tested. The hypothesis for no column effects is

$$\mu_{11} + \mu_{21} = \mu_{12} + \mu_{22} = \mu_{13} + \mu_{23} \quad ,$$

which is partially testable. The test for column effects would ordinarily have two degrees of freedom, but here only one degree of freedom can be used since we can only test the hypothesis

$$\mu_{12} + \mu_{22} - \mu_{13} + \mu_{23} = 0 \ .$$

Thus, there may be sufficient evidence of a column effect when the above contrast is nonzero, although there could be a column effect involving μ_{11} that cannot be tested. Similarly, there is one degree of freedom for testing interaction. BMD10V provides "reduced degrees of freedom" tests, but care must be taken when interpreting the results since the occurrence of missing data may not be random. In particular, when a design is originally balanced or nearly balanced and ends up severely unbalanced, it is unlikely that the occurrence of missing data is random.

For large samples, the occurrence of missing data can be studied from a frequency table approach. If the original design is balanced, two-way frequency analysis should be done in BMDP1F. Having identified a significant interaction in the two-way frequency table, this interaction can be further studied in BMDP2F, which removes cells from the two-way layout one at a time in order to determine whether the imbalance is due primarily to a small number of cells. For higher-way designs, the multi-way frequency program BMDP3F is recommended.

If the original design is not balanced, BMDP3F can also be used as follows:

a. Dichotomize the dependent variable so that zero means that data were missing, and one means that data were observed.

b. The multi-way frequency table is created by using the analysis of variance factors and the dichotomized dependent variable.

The occurrence of missing data is often related to the value of the dependent variable. For example, if we are studying efficacy and the treatment for a particular patient is not effective, the patient may go to another clinic. Thus, the sample mean for a cell can be a severely biased estimate and so everyone's method of doing analysis of variance could be incorrect.

While it is not always possible to determine whether the occurrence of missing data is related to the (observed or unobserved) values of the dependent variable, some things can be done, especially when there are significant covariates. For each case, we can examine

the relationship of the dichotomized version of the dependent variable with the covariates. The dichotomized dependent variable defines two groups. With random covariates (as is often the case in the behavioral and health sciences) we can compute univariate and multivariate t tests (BMDP3D) and perform stepwise discriminant analysis (BMDP7M). The careful data analyst may go on to study the relationship of the occurrence of missing data to both the covariates and the analysis of variance design variables simultaneously.

7. CONCLUSION

For experimental data, tests based on cell sizes are rarely desirable, unless the design is nearly balanced. Tests should be stated in terms of expected cell means and not with the R ( )-notation. For unbalanced data, you cannot have orthogonal hypotheses and orthogonal sums of squares. Repeated measures designs can easily be miscomputed unless the computer program (such as BMDP2V) checks for completeness of data for each case and selects the appropriate error term. There are many appropriate tests for main effects in the presence of interaction, so general linear hypothesis programs such as BMD10V are needed. A major addition to BMDP will be BMDP3V, General Mixed Model.

8. ACKNOWLEDGMENT

This work is supported by NIH Grant RR-3.

9. REFERENCES

GREEN, P.A., R.M. HEIBERGER, and L.L. LASTER (1976). Ambiguous computing of the ANOVA for proportional data. In Proceedings of the Statistical Computing Section of A.S.A., 165-170.

KUTNER, M.D. (1974). Hypothesis testing in linear models (Eisenhart Model I). The American Statistician 28, 98-100.

LORD, F.M. (1967). A paradox in the interpretation of group comparisons. Psychological Bulletin 68, 304-305.

SEARLE, S.R. (1971). Linear Models. New York, Wiley.

SPEED, F.M. and R.R. HOCKING (1976). The use of the R ( )-notation with unbalanced data. The American Statistician 30, 30-33.

YATES, F. (1934). The analysis of multiple classifications with unequal numbers in the different classes. JASA 29, 51-66.

BIOGRAPHY

James W. Frane is Supervising Statistician in charge of BMDP programming at Health Sciences Computing Facility, UCLA. He holds a Ph.D. degree in mathematics from the University of Kansas. He is the author of the BMDP programs for factor analysis, partial correlation, canonical correlation, all possible subsets regression, and description and estimation of missing data.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

HYPOTHESIS TESTING IN MULTI-WAY ANOVA MODELS

J. H. Goodnight
SAS Institute, Raleigh, NC 27605

Key words: Anova; hypothesis testing; missing cells; unbalanced data.

1. INTRODUCTION

Over the years various terms have been associated with the data or analysis arising from experimental designs. Some of the terms, for example, are: balanced, unbalanced, orthogonal, nonorthogonal, missing cells, and messy data. All of these terms are used in an attempt to categorize the data or analysis resulting from experimental designs. However, none of the terms are indicative of whether or not the questions for which the experiment was carried out can be answered.

If we must categorize data, then there are only two categories:

a. Sufficient data - data which suffices for testing all of our envisioned hypotheses, and

b. Insufficient data - data which is insufficient for testing all of our envisioned hypotheses.

With the availability of today's computing power, whether or not a design is balanced is no longer as critical a concern as it was just a few years ago. The power of the computer has freed us to return to the underlying problem facing the statistician when analyzing experimental design data; and that problem is: whether or not the data is sufficient to answer the questions for which the experiment was carried out, and if it is insufficient, what reasonable information can be salvaged.

2. AN EXAMPLE

Consider, for example, a randomized block design with two blocks of four treatments each. Further assume that the four treatments are actually a factorial combination of A (at two levels) and B (at two levels). Assuming that all factors are fixed, the mathematical model for the experiment is:

$$Y_{ijk} = u_{ijk} + e_{ijk}$$

where $u_{ijk} = u + Block_i + A_j + B_k + AB_{jk}$

and e_{ijk} is distributed NORMALLY(0, $I\sigma^2$).

Pictorially we have (before randomization):

Block 1

A \ B	1	2
1	Y_{111}	Y_{112}
2	Y_{121}	Y_{122}

Block 2

A \ B	1	2
1	Y_{211}	Y_{212}
2	Y_{221}	Y_{222}

Without further information describing the intricacies of the factors involved, one would assert that the appropriate tests of hypotheses would be:

Effect	Hypothesis
Block	$u_{111} - u_{211} = 0$ (if blocks were to be tested)
	or $u_{112} - u_{212} = 0$
	or $u_{121} - u_{221} = 0$
	or $u_{122} - u_{222} = 0$
	or $BLOCK_1 - BLOCK_2 = 0$

Hypothesis		(Weights on uijk)	
1		-1	

A $\quad \frac{1}{2}(u_{111} + u_{112} - u_{121} - u_{122}) = 0$

or $\frac{1}{2}(u_{211} + u_{212} - u_{221} - u_{222}) = 0$

or $A_1 - A_2 + \frac{1}{2}(AB_{11} + AB_{12} - AB_{21} - AB_{22}) = 0$

Hypothesis		(Weights on uijk)	
$\frac{1}{2}$	$\frac{1}{2}$		
$-\frac{1}{2}$	$-\frac{1}{2}$		

B $\quad \frac{1}{2}(u_{111} + u_{121} - u_{112} - u_{122}) = 0$

or $\frac{1}{2}(u_{211} + u_{221} - u_{212} - u_{222}) = 0$

or $B_1 - B_2 + \frac{1}{2}(AB_{11} + AB_{21} - AB_{12} - AB_{22}) = 0$

Hypothesis		(Weights on uijk)	
$\frac{1}{2}$	$-\frac{1}{2}$		
$\frac{1}{2}$	$-\frac{1}{2}$		

A*B $u_{111} + u_{122} - u_{112} - u_{121} = 0$

or $u_{211} + u_{222} - u_{212} - u_{221} = 0$

or $AB_{11} + AB_{22} - AB_{12} - AB_{21} = 0$

Hypothesis	
1	-1
-1	1

(Weights on uijk)

For this particular set of data, all hypotheses are testable since $E(Y_{ijk}) = u_{ijk}$. However, remember that Y_{ijk} is not the BLUE of u_{ijk} since the model does not contain the Block*Treatment interaction.

Suppose that during the experiment the observation Y_{222} were lost due to circumstances unrelated to the factors themselves. So that we now have:

Y_{111}	Y_{112}
Y_{121}	Y_{122}

Y_{211}	Y_{212}
Y_{221}	

a. Is the design balanced or unbalanced? Unbalanced.

b. Is the ANOVA orthogonal or nonorthogonal?

These terms are ambiguous at best. For any analysis, there is always a set of orthogonal quadratic forms. Whether or not the quadratic forms used as test statistics for the envisioned test are orthogonal is difficult to ascertain by inspection of the data. The terms are vacuous and a poor substitute for the terms balanced and unbalanced.

c. Does it have a missing cell?

Partly yes and partly no, since u_{222} is actually a replicate of a linear combination of other u_{ijk}'s.

Having answered all of these categorical questions, what do you know about whether or not the data is sufficient to test the hypotheses of interest? Nothing.

In fact, the data is sufficient, all u_{ijk}'s are estimable including u_{222}. All combinations of the parameters of u, Block, A, B, and A*B that were estimable in the balanced design are still estimable.

There is nothing to prohibit you from testing the hypotheses originally envisioned. So test the hypotheses and be done with it!

Can the "appropriate" tests of hypotheses be generated by use of the R( ) notation? Not entirely. It can be shown that:

R Notation	Hypothesis (weights on u_{ijk})	
R(Block | A,B,A*B)	1	-1
R(A | Block)	$\frac{1}{2}$ $\frac{1}{2}$ -7/10 -3/10	
R(A | Block,B)	5/8 3/8 -5/8 -3/8	
R(B | Block,A)	5/8 -5/8 3/8 -3/8	
R(A*B | Block,A,B)	1 -1 -1 1	

We see that the R( ) notation can be used to generate the "appropriate" hypothesis for two of the four effects; they are Block and A*B. However the R() notation fails for effects A and B.

3. CHARACTERISTICS OF THE HYPOTHESES EMPLOYED IN THE RANDOMIZED BLOCK EXAMPLE

In terms of the parameters associated with u, Block, A, B, and A*B, the following table shows the hypotheses employed for the randomized block example (with or without Y_{222}).

	u	BLOCK1	BLOCK2	A_1	A_2	B_1	B_2	AB_{11}	AB_{12}	AB_{21}	AB_{22}
Block	0	1	-1	0	0	0	0	0	0	0	0
A	0	0	0	1	-1	0	0	$\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$
B	0	0	0	0	0	1	-1	$\frac{1}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{2}$
A*B	0	0	0	0	0	0	0	1	-1	-1	1

Note first that the estimable function for Block involves only Block parameters. In fact, this is the only estimable function (except for a scalar multiple) involving block effects and no other parameters. The same observations also hold for the estimable function associated with A*B. Since both A and B are "contained" in A*B (i.e., the columns of the design matrix associated with A and B are linear functions of the columns associated with A*B), any estimable function involving A or B parameters must of necessity involve A*B parameters. In fact, all of the estimable functions above involve only parameters associated with the effect involved and parameters associated with effects which contain the effect.

The estimable function for A does not involve parameters associated with u, Block, or B. However, it is not unique, since any multiple of the A*B estimable function may be added to it. This new estimable function would still involve only the parameters of A and A*B and could legitimately be called an hypothesis about the factor A. But those coefficients on the interaction parameters do make a difference.

4. SOME OBSERVATIONS

On the Role of the Statistician: Too often in papers of this nature, we show all of the types of hypotheses which could be tested and then say, "It's up to the experimenter to determine which of these hypotheses he is interested in." This seems to be similar to the situation in which a person describes his symptoms to a doctor and the doctor then lists the possible diagnoses and medications from which the patient is to choose. When is it appropriate in the fixed effects model for the statistician to recommend unequal weighting on the u_{ij}'s when defining a border mean? What are some examples? What should we look for?

On Statistical Methods Texts: How to compute sums of squares and the different methods of computing sums of squares for balanced situations and unbalanced situations should be de-emphasized. In its place, the nature of the hypotheses to be tested for different designs should be stressed. For a given type of design, when is one hypothesis preferred to another?

On Computer Programs: A computer program which required the statistician to describe in detail the model, the restrictions (if any) on the parameters, and all hypotheses to be tested would win, hands down, in terms of flexibility. The statistician could test any hypothesis he wanted to. Unfortunately, few statisticians are willing to enter several hundred lines of restriction and hypotheses testing information to obtain an analysis for a few dozen observations of data.

Clearly we all want flexibility from a computer program, but at the same time we also want ease of use. We would all like the computer to be off and running on our problem, computing the exact hypotheses we want, with us having specified the minimum of information for it to do the task properly. This is what computers are for, and this is what we expect of them.

In the design of computer programs, flexibility and ease of use are often conflicting attributes. Many "easy to use" programs have tried to achieve flexibility by allowing the user to select from several different types of hypotheses. Situations will always exist though for which none of the hypotheses "pre-programmed" will be appropriate. This is another manifestation of the conflicting attributes of flexibility and ease of use. For those of us in the interface of computer science and statistics, there is still work to be done.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

ANALYSES OF VARIANCE OF UNBALANCED DATA FROM 3-WAY AND HIGHER-ORDER CLASSIFICATIONS*

Shayle R. Searle
Biometrics Unit, Cornell University, Ithaca, NY 14853

ABSTRACT

Answers are given to three questions about 3-way and higher-order classifications that were asked of the panel members of the workshop session "Computing Approaches to the Analysis of Variance for Unbalanced Data".

Key words: Automatic interaction procedures; default decisions; hypothesis testing; interactions; orthogonal contrasts; unbalanced data.

Question 1: Is there a statistically valid default decision short of fitting all possible orders of main effects followed by all meaningful orders of interactions? On what features of the design structure does it depend? Is there a default strategy for simultaneously determining several interesting sets of hypotheses and computing their sums of squares?

Answer: It seems hard to believe that there could ever be a statistically valid default decision – particularly just one such decision, unique for all purposes. And the phrase begs the question as to what is meant by "statistically valid". Any hypothesis $H: \underset{\sim}{K}'\underset{\sim}{\beta} = \underset{\sim}{m}$, for which the rank of $\underset{\sim}{K}$ is $r(\underset{\sim}{K}')_{s\times p} = s$ with $\underset{\sim}{K}'\underset{\sim}{\beta}$ estimable and $s \leq r(\underset{\sim}{X})$ for $E(\underset{\sim}{y}) = \underset{\sim\sim}{X}\underset{\sim}{\beta}$, can be validly tested under normality using $F(H) = Q/s\hat{\sigma}^2$ for $Q = (\underset{\sim}{K}'\underset{\sim}{b}^\circ - \underset{\sim}{m})'(\underset{\sim}{K}'\underset{\sim\sim}{G}\underset{\sim}{K})^{-1}(\underset{\sim}{K}'\underset{\sim}{b}^\circ - \underset{\sim}{m})$ where $\underset{\sim}{b}^\circ = \underset{\sim\sim}{G}\underset{\sim}{X}'\underset{\sim}{y}$ and $\underset{\sim}{X}'\underset{\sim\sim}{X}\underset{\sim\sim}{G}\underset{\sim}{X}'\underset{\sim}{X} = \underset{\sim}{X}'\underset{\sim}{X}$; then, under H the distribution of F(H) is Snedecor's F on s and $N - r(\underset{\sim}{X})$ degrees of freedom. Any default decision that leads to an H of this sort can be validly tested. But since there are many such H's, with boundless ideas for being interested in some rather than others, it is difficult to see how any computer program can contain unique specifications for a choice that will be suitable for all possible kinds of data.

---

* Paper No. BU-334 in the Biometrics Unit, Cornell University.

The suggestion that a computer program could choose "interesting sets of hypotheses" trikes an odd chord. "Interesting" to whom? To the person whose data are being analyzed, resumably. But isn't the choice of interesting hypotheses part of the scientific method, ndeed that very part which so often involves human conjecture? This, then, is the baili- ick of the experimenter, the data gatherer, the survey analyst, of the person who wants to ake a step forward in his understanding of nature. It is not even the statistician's job, et alone that of an inhuman, non-thinking, automaton computer. Certainly a statistician an help, not as an automaton but as a clear thinking scientist discussing nature with the esearcher, helping him formulate, i.e. put into formal terms, the hypotheses or conjectures bout nature that he has in mind. One large aspect of the statistician's help is to confine he scientists' hypotheses to ones that are testable – i.e., to those involving estimable unctions.

Question 2: If interactions are found significant, is an automatic procedure for splitting he design into more homogeneous subdesigns feasible?

Answer: Any answer to this question must be preceded by considering a more fundamental uestion such as "what is the meaning of interactions in high-order classifications and how an they be tested, especially when unbalancedness of data includes many empty cells?". For example, can one give a useful, practical meaning to a 4-way interaction; and if 30% of the sub-most cells of the data set have no data, what is the meaning of interactions being "found significant"?

The complexities of trying to understand interactions in 3-, 4-, 5-way and higher-order classifications do, I believe, overpower any consequences of what should be done "if interactions are found significant" – especially for unbalanced data in which there are many empty cells. Even suggesting that a computer program could be planned to split the "design into more homogeneous sub-designs" therefore seems somewhat absurd. To heighten the absurdity, what would it do if 5^{th}-order and 3^{rd}-order interactions were significant but 4^{th}-order ones were not?

It seems clear to me that contemplating interactions in high-order classifications having unbalanced data including empty cells highlights the absolute necessity to abandon

overparameterized models and to fall back on cell means. This is, of course, what Hocking and Speed (1975, 1976) have been advocating for years and indeed is precisely what Fisher did when he started this whole analysis of variance business anyway [see Urquhart et al. (1973)]. The model is then $E(\underset{\sim}{y}) = \underset{\sim}{\mu}$ with each element of $\underset{\sim}{\mu}$ being a population cell mean, $\mu_{ijk\ell m}$, say, for a 5-way classification. Then $\hat{\mu}_{ijk\ell m} = \bar{y}_{ijk\ell m.}$ is the b.l.u.e. of $\mu_{ijk\ell m}$ with variance $\sigma^2/n_{ijk\ell m}$. A hypothesis about any number of linear combinations of the μ's is then testable, $\underset{\sim}{K}'\underset{\sim}{\mu} = \underset{\sim}{m}$ say, with its F-statistic being $Q/s\hat{\sigma}^2$ where, for $\underset{\sim}{\bar{y}}$ being the vector of cell means, $Q = (\underset{\sim}{\bar{y}}'\underset{\sim}{K} - \underset{\sim}{m}')(\underset{\sim}{K}'\underset{\sim}{G}\underset{\sim}{K})^{-1}(\underset{\sim}{K}'\underset{\sim}{\bar{y}} - \underset{\sim}{m})$ and $\underset{\sim}{G}$ is the diagonal matrix of reciprocals of cell numbers, $1/n_{ijk\ell m}$. Under these circumstances the model is simple to learn, simple to understand and simple to use; and the task of what hypotheses are to be tested is laid fairly and squarely where it should be: at the foot of the researcher. However, his task is now easy, compared to his task in overparameterized models. Any hypothesis about the value of any linear combinations of the population cell means (the $\mu_{ijk\ell m}$'s) can be tested. He has only to state his conjectures in this form, without any limitation at all on what sort of linear combination (because each and every one of them is an estimable function) can be the basis of a hypothesis. No statistician need persuade him to be confined to just certain (estimable) kinds of linear combinations; they are all permissible.

Question 3: What is the appropriate criterion for the testing of hypotheses? The hypotheses tested by the F-statistics are orthogonal even with unbalanced data if the sums of squares for each line of the ANOVA table is computed by adjusting for all lines above it and ignoring all lines below it. The set of contrasts associated with almost any other set of sums of squares is not orthogonal and therefore open to ambiguity of interpretation.

Answer: Ambiguity of interpretation is built into the analysis of unbalanced data. Furthermore, in many kinds of data, empty cells are a virtual certainty. In family surveys, for example, the 72-year-old father with 4 children under 5, on welfare, living in Georgetown in a 1-room house with 5 cars, 2 yachts and a Lear Jet simply does not exist. The Howard Hughes of this world seldom get caught in survey data.

So what do we do, insofar as hypotheses are concerned? Fall back on cell means is undoubtedly the only rational thing to do; and, thankfully, it is an easy route to take.

The concern implicit in this third question is that of orthogonal hypotheses and/or orthogonal sums of squares. Traditionally, hypotheses $H_1 : \underset{\sim}{k}_1'\underset{\sim}{\beta} = 0$ and $H_2 : \underset{\sim}{k}_2'\underset{\sim}{\beta} = 0$ (for $\underset{\sim}{k}_1'$ and $\underset{\sim}{k}_2'$ being row vectors) would be considered orthogonal when $\underset{\sim}{k}_1'\underset{\sim}{k}_2 = 0$. However, the numerator sums of squares for the corresponding F-statistics are independent (under normality) if and only if $\underset{\sim}{k}_1'\underset{\sim}{G}\underset{\sim}{k}_2 = 0$; in which case they sum to that used for testing H_1 and H_2 simultaneously [see Searle (1971), Sec. 5.5g]. Therefore $\underset{\sim}{k}_1'\underset{\sim}{G}\underset{\sim}{k}_2 = 0$ seems an appropriate generalization of the orthogonality concept for unbalanced data – and it reduces, of course, to $\underset{\sim}{k}_1'\underset{\sim}{k}_2 = 0$ when $\underset{\sim}{G}$ is a scalar matrix (or when appropriate principal submatrices of $\underset{\sim}{G}$ are). In this sense, orthogonal hypotheses have independent numerator sums of squares – but not independent F-statistics (both denominators contain $\hat{\sigma}^2$). As a result, the concept of hypotheses being orthogonal seems to deserve less importance than is implied by this question. The criteria for testing a hypothesis should be (i) that it is testable and (ii) that it is meaningful and of interest to the experimenter.

Final Comment: An overriding comment is to tell computer jocks not to write fully general programs. They are too difficult to explain and are so fraught with dangers for possible erroneous use that they frequently do get used erroneously – and often without the user knowing of the errors perpetrated. Complementary advice for statisticians would be to encourage data gatherers to set up their own hypotheses, and to assist them by relying entirely upon the cell means model. It is straightforward, requires no computers, is easy to understand and is in direct line with the way in which most experimenters think about their data.

REFERENCES

HOCKING, R. R. and SPEED, F. M. (1975). A full rank analysis of some linear model problems. J. Amer. Stat. Assoc. 70, 706-712.

SEARLE, S. R. (1971). Linear Models. Wiley, New York.

SPEED, F. M. and HOCKING, R. R. (1976). The use of the R( )-notation with unbalanced data. The American Statistician 30, 30-34.

URQUHART, N. S., WEEKS, D. L. and HENDERSON, C. R. (1973). Estimation associated with linear models: a revisitation. Communications in Statistics 1, 303-330.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

ANOVA FOR NON-ORTHOGONAL DATA

G. N. Wilkinson
Bell Laboratories
Murray Hill, New Jersey 07974

ABSTRACT

An ANOVA is primarily an information summary and screening device. One pass with a model-fitting algorithm provides both a *forward* ANOVA and a *backadjusted* ANOVA. The forward ANOVA depends on the order of fit of the model terms, but if main effects are ranked in importance on either prior information or the magnitude of unadjusted mean squares, and interactions are assigned the corresponding induced order, these two ANOVA's often suffice for interpreting the data. It is not necessary to consider all possible ANOVA's that could arise from arbitrary reordering of model terms. The question of hypothesis testing does not arise at the stage of presenting estimated values but only at the prior stage of determining an adequate model fit. The extension to multiple error strata, covariates and missing values is briefly considered.

Key words: Covariance analysis; expected mean squares; factorial models; multiple error strata; nonorthogonal ANOVA.

1. INTRODUCTION

Much of the current confusion about ANOVA, particularly in the nonorthogonal case, is attributable to some misunderstanding of the basic role of ANOVA, and to faulty or inappropriate mathematical formulations for it, such as the misleading classification of models as fixed, mixed or random, and the unnecessary introduction of marginal constraints in specifying a model. There is also far too much emphasis on hypothesis testing, as opposed to estimation.

This confusion has given rise to an extraordinary proliferation of publications and I believe that an intensive effort should be made to restore to ANOVA the essential simplicity that is so often obscured. It is to be hoped that the present workshop will make a positive contribution in this regard.

2. THE ROLE OF ANOVA

As its inventor, R. A. Fisher, clearly perceived, the primary statistical role of an ANOVA is that of an *information summary.* It also serves as a screening device, suggesting to its interpreter just how far the data warrants more detailed examination, and providing a gauge of the adequacy of proposed models for summarizing the data.

Thus the primary domain of application of ANOVA is Estimation. The F ratios, which in engineering parlance are (signal + noise)/noise ratios, provide formal significance tests for the adequacy of a model fit, and this is their usual role. Only occasionally are they needed to test a genuine scientific hypothesis, say of independent action of two factors A and B.

It is for this reason, I believe, that Fisher favored the use of conventional significance levels (5%, 1%, .1%; usually indicated by *, **, ***). It is only with some critical scientific test in mind

that one would wish to determine the exact significance probability of a particular F-ratio.

It is perhaps unnecessary to stress that for an ANOVA to be fully effective as an information summary, the partitioning of variance should be extended as far as possible, splitting factorial terms into linear and curvilinear components, etc.

3. EXPECTED MEAN SQUARES

Significance tests in ANOVA have not only been overemphasized but also greatly complicated by confusion regarding the appropriate formulae for expected mean squares, chiefly as a result of faulty mathematical treatment (in the scientific, not logical sense). Yates (1965) pointed out that apparent anomalies in the formulae for expected mean squares do not arise if marginal constraints (unnecessary in any case) are not imposed on the random terms in a model. In fact the apparent anomalies are simply a notational artifact, attributable to variance symbols having changed meanings in different contexts, as I subsequently realized.* The same effect was noted by Hocking (1973). Nelder (1977) has provided I believe, a definitive resolution of the confusion in this area, in a paper read to the Royal Statistical Society in London on November 9, 1976.

As Nelder has noted, there is a crucial distinction between two kinds of random term, (i) error terms which determine the primary stratification of the data vector into error strata, and likewise the ANOVA (see Fisher (1935)); and (ii) treatment terms which are nevertheless to be summarized in terms of estimated variance parameters, as when a set of treatments has been randomly selected from a larger population of treatments about which inferences are to be made. Some genetic studies fall in this category.

It is the error mean square in each error stratum that is the appropriate divisor of F-ratios for treatments estimated in that stratum. Otherwise, as Nelder (1977) shows, treatment mean squares in an error stratum have the same kind of comparability regardless of their fixed or random status. This comparability is rendered explicit when expected mean squares are specified in terms of canonical components of variance ϕ, the same canonical formulae applying regardless of the random or fixed status of the various terms. In the case of random terms the effect of sampling from either a finite or infinite population is absorbed in the definition of the canonical parameters.

4. FORWARD AND BACKADJUSTED ANOVA'S

The general least-squares method of fitting linear factorial models is a stepwise extension process, one new model term being fitted in each step. There are two phases in this process, *forward* and *backward,* the latter necessary if there are nonorthogonal effects:

Forward: The effects of a new model term are estimated

Backward: Previously estimated effects are backadjusted to their correct values in the extended model fit.

The forward steps collectively define a *forward* ANOVA, which depends on the order in which model terms are fitted. For a 2-way table of data classified by factors A and B, it takes the form

* In an unpublished paper with J. A. Nelder, 'The Mixed Model Muddle', now superseded by Nelder (1977).

Forward ANOVA

A ignoring B

B eliminating A (1)

$A \times B$ (automatically orthogonal to A and B)

Subclass variation

The sums of squares in this analysis are additive. In particular, the pooled sum of squares for the A and B terms is the sum of squares for fitting an additive model $(A+B)$ comprising only A and B effects, with no interaction terms, and is independent of the order of fit of the terms, A,B.

If the factors A and B are nonorthogonal, backadjustment produces a nonadditive analysis of variance,

Backadjusted ANOVA

A elim. B	(by backadjustment after fitting B)	
B elim. A	(same as in forward ANOVA)	(2)
$A \times B$	(same as in forward ANOVA) ,	

which is independent of the order of fit of A and B except when there is partial aliasing of A and B effects - the aliased effects are then represented in the first term fitted (A) but not in the second, and there is a corresponding effect on degrees of freedom.

We can now consider the first question put to this panel by Heiberger and Laster - what to do about the multiplicity of possible forward ANOVA's and partially backadjusted ANOVA's that could in principle be produced with generally nonorthogonal data.

From a practical point of view I think that only two ANOVA's need be considered in conjunction with a single pass of a model-fitting algorithm, the forward and the fully backadjusted ANOVA, with perhaps the further option of nominating a break-point in the model for backadjustment, for it is sometimes the case that the analyst is interested only in the fit of a reduced model but wishes nevertheless to exclude further high-order effects from the estimate of error variance, to avoid possible contamination.

The forward ANOVA depends of course on the order of fit of the model terms, and to be fully effective as an information summary, certain ordering principles ought to be invoked. The marginality principle requires that any term be preceded by any terms marginal to it - for obvious reasons. A second ordering principle is justified by the generally smooth nature of the underlying response models in practice, and that is to group all main effect terms together, followed by all first order interactions and so forth, as in a Taylor-McLaurin expansion. (The exception is with pseudo-factorial components, say of Varieties, in a pseudo-factorial analysis - these must always maintain their juxtaposition since they will be summed together in the resultant ANOVA.) A further justification for this ordering is the progressive loss of *statistical* information as one proceeds from main effects to first order interactions, etc. Finally, main effects should be ranked in order of importance, either according to prior knowledge or by the magnitude of the unadjusted mean squares for main effects. Once this ordering is specified, interactions should be assigned the corresponding induced ordering. For instance the ordering

Term:	A	B	C	D	
Binary code:	0001	0010	0100	1000	(3)

induces the ordering, indicated by the ascending magnitude of the binary code,

$A\times B$	$A\times C$	$B\times C$	$A\times D$	$B\times D$	$C\times D$	
0011	0101	0110	1001	1010	1100	(4)

of the first order interactions, and so forth.

With the option on ordering of main effects as described above, the forward and backadjusted ANOVA's often suffice for interpreting the data, though occasionally a second pass of the model fitting algorithm will be needed to fit, say, a more reduced model with main effects in a different order.

5. PRESENTATION OF ESTIMATED VALUES

This is an area where some controversy continues. I shall illustrate with the simple case of a 2-way table of data with unequal subclass replication:

Means			*Replications*		
$\bar{x}_{11}$	$\bar{x}_{12}$	$\bar{x}_{1.}$	n_{11}	n_{12}	$n_{1.}$
$\bar{x}_{21}$	$\bar{x}_{22}$	$\bar{x}_{2.}$	n_{21}	n_{22}	$n_{2.}$
$\bar{x}_{.1}$	$\bar{x}_{.2}$	$\bar{x}_{..}$	$n_{.1}$	$n_{.2}$	N

(5)

(Marginal means weighted according to subclass replication.)

The analyst will usually want to see a table of estimated values $\hat{\mu}_{ij}$ bordered with appropriate marginal means, together with appropriate standard errors. The marginal entries he requires will usually be unweighted means

$$\hat{\mu}_{i.} = (\hat{\mu}_{i1} + \hat{\mu}_{i2})/2, \quad i = 1,2\ ,$$
$$\hat{\mu}_{.j} = (\hat{\mu}_{1j} + \hat{\mu}_{2j})/2, \quad j = 1,2\ , \tag{6}$$

or else entries weighted according to say specified population weights, distinct from the subclass replication values, which are usually only an artifact of the experiment and not otherwise of intrinsic scientific interest. In a computer implementation the user should be allowed to specify his requirements in this regard.

In answer to Heiberger and Laster's third query, in this workshop, it is crucial now to stress that no question of *hypothesis-testing* arises at this stage - we are fully in the domain of estimation. The only hypothesis tests would have been concerned with how the expected values μ_{ij} are to be estimated. If interaction is allowed for, $\hat{\mu}_{ij}$ is simply $\bar{x}_{ij}$. Otherwise, if the constrained model

$$\mu_{ij} = \mu + \alpha_i + \beta_j \tag{7}$$

were judged to be appropriate, with the inherent no-interaction constraint

$$\gamma = \mu_{11} - \mu_{12} - \mu_{21} + \mu_{22} = 0, \tag{8}$$

then the $\hat{\mu}_{ij}$ would have been determined as combinations of estimated A and B effects only and a common term, according to the rearranged form of model (7),

$$\mu_{ij} = (\mu+\bar{\alpha}+\bar{\beta}) + (\alpha_i-\bar{\alpha}) + (\beta_j-\bar{\beta}) \tag{9}$$

in which the bracketed terms are the statistically estimable quantities. The terms $\bar{\alpha}, \bar{\beta}$ are the

marginally weighted means which are completed aliased with μ in estimation. Hypotheses which *might* have been tested, given say that A dominates B in effect, are as follows, with $\alpha = \alpha_1 - \alpha_2$, $\beta = \beta_1 - \beta_2$:

$$\begin{aligned} &(i) \quad \gamma = 0 \quad (\textit{no interaction}), \\ &(ii) \quad \beta = 0 \quad \textit{given} \quad \gamma \approx 0, \\ &(iii) \quad \alpha = 0 \quad \textit{given} \quad \gamma \approx 0 \text{ and } \beta \approx 0. \end{aligned} \tag{10}$$

For these tests the forward ANOVA F-ratios are uniquely appropriate. A test of $\alpha = 0$ given only that $\gamma \approx 0$ would come from the backadjusted ANOVA.

Note that none of these hypotheses depend on subclass replication values for their definition. Any representation of them that appears to make them so dependent is misleading and may be scientifically inappropriate. (Note that here I disagree with Hocking (1975). The relevant hypotheses are most clearly stated relative to (7).)

There is of course a difficulty of interpretation which experimenters commonly encounter when considering the unweighted marginal means $\hat{\mu}_{i.}$, $\hat{\mu}_{.j}$ in (6). They often don't understand why say the marginal difference $\hat{\mu}_{1.} - \hat{\mu}_{2.}$ does not agree with the previously presented least-squares estimate $\hat{\alpha}$ of α, relating to model (7). In fact $\hat{\alpha}$ is the best statistically combined estimate derived from the individual estimates $\hat{\alpha}_{.1} = \bar{x}_{11} - \bar{x}_{21}$, $\hat{\alpha}_{.2} = \bar{x}_{12} - \bar{x}_{22}$ which have the same expected value if interaction is absent. Thus if $1/w_j = 1/n_{1j} + 1/n_{2j}$, $j = 1, 2$,

$$\hat{\alpha} = (w_1\hat{\alpha}_{.1} + w_2\hat{\alpha}_{.2})/(w_1 + w_2), \tag{11}$$

and this, not $(\hat{\mu}_{1.} - \hat{\mu}_{2.})$, is the appropriate quantity for a test of significance of A eliminating B when interaction is negligible. Of course such a test usually becomes pointless if interaction is present.

To round off this discussion I should add that there is an uncommon class of scientific contexts in which a table of data can exhibit large interactive fluctuations but of a compensatory nature, so that marginal entries exhibit very much less fluctuation. An example would be a table of cash flows (+ or -) relating to various products and corporations. In such cases the usual factorial analysis is likely to be quite inappropriate, because of the correlation patterns engendered by random shocks to the system under study. Some form of dynamic modelling is indicated instead.

6. EXTENSION TO MULTIPLE ERROR STRATA

Many experiments have some physical structure relating the experimental units, such as a division of blocks into plots, or an allocation of several tests to each patient. The appropriate models then have more than one error term, and the least squares analysis becomes a two-stage process.

In the first stage the error part of the model only is fitted, ignoring 'treatment' terms. This determines a primary partition of the data vector into *error strata,* and similarly in the ANOVA. In the second stage an extension of the least-squares process is applied to estimate treatment (and possibly also covariate) effects in all error strata that provide information on them. There may also be further stages in which factorial terms are further subdivided according to specified submodels; in which treatment information from different strata is statistically combined; and in which random treatment terms may be passed through a variance component estimation process. A proper modularization of the computing algorithms for all these stages is essential in a general computer implementation of ANOVA.

The preceding remarks in the paper apply to multiple error strata with the following modifications:

(i) In fitting the error model, no backadjustments are made to error effects. Thus in a lattice square design we may have orthogonal strata designated 'Rows ignoring Columns' and 'Columns eliminating Rows' if it happens, say, that the diagonal elements of each square are missing. (Incidentally it may be noted that the order of fit of Rows and Columns will be

immaterial in the final analysis when treatment information in different strata is combined.)

(ii) Interpretation of the terms in the forward and backadjusted ANOVA's for each error stratum is a little more complicated. Probably the best way of viewing the situation is to note that we have essentially, a series of statistically independent fits of the treatment model, corresponding to the different error strata, and that each fit is to a greater or less extent degenerate, since there is zero information on some terms in some strata. It is perhaps for these reasons that no fully general 'combination of information' procedure has yet been published, (though I do have a general solution if treatment terms are mutually orthogonal).

(iii) If all treatment effects are mutually orthogonal there is generally one unique ANOVA for each error stratum, the forward and backward ANOVA's being the same. However, there is an exception; the fitting of a treatment interaction term, $A \times B$ say, in an error stratum will induce a backadjustment of a term marginal to it in that stratum, A say, if both are nonorthogonal to another error term such as 'Blocks'.

(iv) An ordering of main effects will need to be based on canonical components of variance rather than mean squares, because of differing stratum variances.

7. COVARIANCE ANALYSIS (AND MISSING VALUES)

We need only consider the case of a single error stratum since with multiple strata the same form of covariance analysis can be applied independently to each error stratum.

Let y denote the response variable and X a matrix whose column vectors are the covariates. The best computing procedure is first to fit the factorial model and any related submodels to both the y and X data. Call these *standard* analyses, meaning 'not adjusted for covariates.' The vector of regression coefficients b in the covariance relation can then be estimated from the residual SS-SP matrix of the (Y,X) MANOVA.

The next step is to backadjust all relevant effects, contrasts and residuals computed in the standard analysis of y, with adjustments of the form $\tilde{y} = y - Xb$. These are the correct estimates in the covariate-extended model.

We may also compute the standard forward and backward ANOVA's from the backadjusted data $\tilde{y}$, but only the residual term $SS_R(\tilde{y})$ will be correct for the covariate-extended model. A further, subtractive correction of the sum of squares $SS_T(\tilde{y})$ for each treatment term T is required, as follows:

Let $\tilde{p}_T$ be the vector of treatment products of $\tilde{y}$ with X for the term T, and A_{R+T} the pooled 'residual + treatment T' SS-SP matrix of the X MANOVA. Since $\tilde{p}_{R+T} = \tilde{p}_T$ because $\tilde{p}_R = 0$, $\delta b_T = A_{R+T}^{-1}\tilde{p}_T$ is the vector of adjustments to the covariance coefficients b that would result from combining the term T with the residual term. The subtractive correction for $SS_T(\tilde{y})$ is then the sum of products of δb_T with $\tilde{p}_T$.

Note that covariance-backadjustment of the standard *forward* ANOVA for the y data is equivalent to promoting 'covariates' to be the leading term in the model.

Missing values can be handled in a similar way to covariates, except that the special form of the indicator covariates for missing values simplifies the calculations required (Wilkinson, 1961). When there are also genuine covariates X the best modularization is to first adjust in parallel for missing values in both the y and X data, before proceeding to estimate and backadjust for covariate effects (Wilkinson, 1957). Treatment sums of squares undergo a double correction, first for missing values and then for covariates.

8. CONCLUSION

I hope I have said enough to suggest that ANOVA has an essential simplicity, even with nonorthogonal data, if not obscured by extraneous issues or inappropriate mathematical formalism. At the same time I do not wish to under-emphasize the magnitude and complexity of the task of

developing general and powerful computer programs for ANOVA. I think that the fundamental canonical theory of ANOVA outlined in James and Wilkinson (1971) is important in this regard. It has provided a recursive and adaptive algorithm of remarkable simplicity (Wilkinson, 1970) which is specially suited to analyzing generally balanced nonorthogonal designs. The GENSTAT implementation of it currently handles several hundred ANOVA's each week, some with many error strata and extraordinary confounding patterns. However, the best exploitation of canonical properties for computing analyses of experiments lacking general balance is very much an open question, though Hemmerle's (1973) iterative approach is promising.

Heiberger and Laster's second query really raises the question of automatic refinement of the analysis process in a computer program, for detecting and taking account of aberrants, nonadditivity, heterogeneity of variance between local subsets of the data, etc. I think we would agree that although automatic actions of this kind would not cover all contingencies without further interaction with the data analyst, their inclusion in a computer program with adequate user-controls is both feasible and desirable, and would help to protect naive users from inadequate interpretations of their data.

REFERENCES

Fisher, R. A. (1935). Contribution to a discussion of F. Yates' paper on complex experiments. *J. Roy. Statist. Soc. Suppl. 2,* 229-231.

Heiberger, R. M. and Laster, L. L. (1977). Computing Approaches to the analysis of variance for unbalanced data. *Proc. Tenth Interface Symp. on Comp. Sc. and Statist.* (to appear).

Hemmerle, W. J. (1974). Nonorthogonal analysis of variance using iterative improvement and balanced residuals. *J. Amer. Statist. Assn., 69,* 772-778.

Hocking, R. R. (1973). A discussion of the two-way mixed model. *Amer. Statist., 27,* 148-152.

Hocking, R. R. (1975). A full rank analysis of some linear model problems. *J. Amer. Statist. Assn., 70,* 706-712.

Nelder, T. A. (1977). A reformulation of linear models (with discussion). *J. Roy. Statist. Soc. A, 140* (to appear in Part 1).

Wilkinson, G. N. (1957). The analysis of covariance with incomplete data. *Biometrics 13,* 363-372.

Wilkinson, G. N. (1961). Comparison of missing value procedures. *Aust. J. Statist., 2,* 53-65.

Wilkinson, G. N. (1970). A general recursive procedure for analysis of variance. *Biometrika, 57,* 19-46.

James, A. T. and Wilkinson, G. N. (1971). Factorization of the residual operator and canonical decomposition of nonorthogonal factors in the analysis of variance. *Biometrika, 58,* 279-294.

Yates, F. (1965). A fresh look at the basic principles of the design and analysis of experiments. *Proc. Fifth Berkeley Symp. on Math. Statist. and Prob., 4,* 777-790.

BIOGRAPHY

Graham Wilkinson (M.Sc., University of Adelaide) is currently a visiting scientist (1975-77) at Bell Laboratories, Murray Hill, New Jersey, and was previously a research scientist in the CSIRO (Australia) Division of Mathematical Statistics (1950-70) and at the Rothamsted Experimental Station, England (1971-75). He is a member of the International Statistical Institute and a Guy Medallist (bronze) of the Royal Statistical Society. His research interests include analysis with missing data, canonical theory for analysis of variance, a statistical computing system for scientific data analysis (GENSTAT) and, most recently, a generalized theory of inferential probability.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

THE ANALYSIS OF LINEAR MODELS WITH UNBALANCED DATA

R. R. Hocking, O. P. Hackney, and F. M. Speed
Mississippi State University, Mississippi State, MS 39762

ABSTRACT

The purpose of this paper is to describe the hypotheses commonly tested in linear models with unbalanced data, including the case of zero cell frequencies. Historically, the sums of squares for the test statistics have been developed either on heuristic principles or because of computational convenience. Precise statements of the corresponding hypotheses are rarely found in the literature and, in those cases where the hypotheses are stated, they are usually described in terms of the parameters of the non-full rank model which may be difficult to interpret. In this paper, the hypotheses associated with the R( ) notation for general sets of conditions are described in terms of the means of the observed populations. The discussion is restricted to two-way models.

Key words: Linear models; tests of hypotheses; unbalanced data.

1. INTRODUCTION

The purpose of this paper is to discuss the analysis of the classical, fixed effects, linear model for designed experiments when the data is unbalanced, including the case of zero cell frequencies. Following Hocking and Speed (1975), we describe the analysis in terms of the cell means model given by

$$Y = W \mu + e \tag{1}$$

subject to

$$G \mu = 0. \tag{2}$$

Here, μ is the q-vector of cell means, W is the nxq matrix of zeros and ones indicating the number of times a particular cell or population has been observed, and G is a matrix which describes any known linear relations that may exist on μ.

In the following we state a number of theorems which describe the hypotheses being tested by some of the standard computational procedures. The fact that this is even necessary is contrary to normal statistical analysis which suggests that the first step is to formulate the hypothesis and the second is to develop the test statistic. Unfortunately, this has not been the case with the analyses of unbalanced, ANOVA data. In most cases, the sums of squares for testing are dictated by computational convenience rather than a precise

statement of the hypothesis of interest. For simplicity, the theorems are stated for the two-way classification model with interaction given by

$$y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_{ij} + e_{ijk} \qquad (3)$$

$$i=1\ldots a$$
$$j=1\ldots b$$
$$k=0,1,\ldots n_{ij}.$$

In terms of the cell means model, we simply write

$$y_{ijk} = \mu_{ij} + e_{ijk} \qquad (4)$$

with the obvious definition of μ_{ij}.

2. GENERAL HYPOTHESIS THEOREMS

The first three theorems describe the hypotheses tested as a result of imposing a specific set of non-estimable conditions on the model (3) and then testing for row effects $\alpha_i = 0$. Theorem 4 develops the interaction hypothesis under the conditions in Theorem 1 and 2.

Theorem 1. In the model (3) suppose the design is connected and we impose the conditions

$$\sum_{i=1}^{a} c_{ij}\,\gamma_{ij} = \sum_{j=1}^{b} c_{ij}\,\gamma_{ij} = 0 \qquad (5)$$

for i=1...a, j=1...b and, in addition, set γ_{ij} =0 if $n_{ij} = 0$ for a total of a + b - 1 + m linearly independent, non-estimable conditions. Here m is the number of empty cells. In addition, one condition on the α_i and the β_j are adjoined to obtain a full rank model. The row effect hypotheses, $H_\alpha : \alpha_i = 0$, is then given by

$$H_\alpha : \sum_{j=1}^{b} d_{ij}\,\mu_{ij} = \sum_{j=1}^{b}\sum_{i'=1}^{a} d_{ij}\,d_{i'j}\,\mu_{i'j} / d_{.j} \qquad (6)$$

$$\text{for } i=1\ldots a-1.$$

Here,

$$d_{ij} = \begin{cases} c_{ij} & \text{if } n_{ij} \neq 0 \\ 0 & \text{if } n_{ij} = 0. \end{cases}$$

Theorem 2. In theorem 1, consider the conditions

$$\sum_{i=1}^{a} v_i \gamma_{ij} = \sum_{j=1}^{b} w_j \gamma_{ij} = 0. \tag{7}$$

Then the row effect hypothesis is given by

$$H_\alpha : \sum_{j=1}^{b} w_j \delta_{ij} \mu_{ij} = \sum_{j=1}^{b} \sum_{i'=1}^{a} v_{i'} w_j \delta_{i'j} \delta_{ij} \mu_{i'j} \Big/ \sum_{i=1}^{a} v_i \delta_{ij} \tag{8}$$

$$\text{for } i = 1 \ldots a-1.$$

Here,

$$\delta_{ij} = \begin{cases} 1 \text{ if } n_{ij} \neq 0 \\ 0 \text{ if } n_{ij} = 0 \end{cases}$$

Theorem 3. Suppose the non-estimable conditions are

$$\sum_{i=1}^{a} c_{ij} (\alpha_i + \gamma_{ij}) = \sum_{j=1}^{b} c_{ij} (\beta_j + \gamma_{ij}) = 0 \tag{9}$$

$$\sum_{i=1}^{a} c_{i.} \alpha_i = \sum_{j=1}^{b} c_{.j} \beta_j = 0 \tag{10}$$

and if $n_{ij} = 0$ set $\alpha_i + \gamma_{ij} = \beta_j + \gamma_{ij} = 0$ for a total of $a + b + 1 + m$ linearly independent conditions. Then,

$$H_\alpha : \sum_{j=1}^{b} d_{ij} \mu_{ij} / d_{i.} = \sum_{j=1}^{b} d_{i'j} \mu_{i'j} / d_{i'.} \tag{11}$$

$$\text{for all } i, i'.$$

Here,

$$d_{ij} = \begin{cases} c_{ij} & n_{ij} \neq 0 \\ 0 & n_{ij} = 0. \end{cases}$$

Theorem 4. Given any set of $a + b - 1$ linearly independent, non-estimable conditions on the γ_{ij} such as (5) or (7), along with $\gamma_{ij} = 0$ if $n_{ij} = 0$, the interaction hypothesis is obtained as follows:

1. Let $H\mu = 0$ denote any set of $(a - 1)(b - 1)$ linearly independent interaction constraints. For example,

$$\mu_{11} - \mu_{i1} - \mu_{1j} + \mu_{ij} = 0 \qquad (12)$$

for i=2...a, j=2...b.

2. Assume that the components of μ are ordered so that the m empty cells occur first and reduce H to obtain the following equivalent set of constraints:

$$\begin{bmatrix} H_{11} & H_{12} \\ 0 & H_{22} \end{bmatrix} \mu = 0 \qquad (13)$$

where H_{11} has m columns. Then the interaction hypothesis is given by

$$H_\gamma : [0\ H_{22}]\ \gamma = 0. \qquad (14)$$

Corollary 1. If the design is connected, $H_{11} = I_m$. If not, $H_{11} = (I\ P)$ where I has dimension m - p with p being the minimum number of cells to be filled to connect the design.

Corollary 2. The two-way model without interaction can be written in the form of (1) and (2) with (1) given by (4) and (2) given by (14).

3. SUMMARY

The intent of this paper has been to demonstrate the hypotheses being tested when standard computing procedures are used. The analyst is urged to study these hypotheses to see if they meet his needs. Ideally, the computer program should be sufficiently flexible to allow the specification of any linear hypothesis rather than restricting the user to one or more of those specified by the above theorems.

4. REFERENCES

BURDICK, D. S., HERR, D. G., O'FALLON, W. M., and O'NEILL, B. V. (1974). Exact methods in the unbalanced, two-way analysis of variance - a geometric approach. Comm. Statist. 3, 581-94.

COOK, R. D. and WEISBERG, S. (1975). Missing values in unreplicated orthogonal designs. Technical Report 253r. Department of Applied Statistics, University of Minnesota.

ELSTON, R. C. and BUSH, N. (1964). The hypotheses that can be tested when there are interactions in an analysis of variance model. Biometrics. 681-699.

FINNEY, D. J. (1948). Main effects and interactions. JASA. 43, 566-571.

GOSSLEE, D. G. and LUCAS, H. L. (1965). Analysis of variance of disproportionate data when interactions are present. Biometrics, 21, 115-133.

GRAYBILL, F. A. (1961). An Introduction to Linear Statistical Models. McGraw-Hill.

HACKNEY, O. P. (1976). Hypotheses testing in general linear model. Unpublished Ph.D. disseration. Emory University.

HERR, D. G. and KENDALL, A. C. (1975). On near orthogonality of two-way designs. Bulleti of the I.M.S. 4, 119.

HERR, D. G. (1976). A geometric characterization of connectedness in a two-way design. Biometrika. 63, 93-100.

HOCKING, R. R. and SPEED, F. M. (1975). A full rank analysis of some linear model problem JASA. 70, 706-712.

HOCKING, R. R., HACKNEY, O. P., and SPEED, F. M. (1977). The analysis of linear models with unbalanced data. To appear in Contributions to Survey Sampling and Applied Statistics -- Papers in Honor of H. O. Hartley.

JOHN, P. W. M. (1971). Statistical Design and Analysis of Experiments. Macmillan Co.

RUBIN, D. B. (1972). A non-iterative algorithm for least squares estimation of missing values in any analysis of variance design. Applied Statistics. 21, 136-141.

SEARLE, S. R. (1967). Linear Models. John Wiley & Sons, Inc.

SEARLE, S. R. (1972). Using the R( ) -notation for reductions in sums of squares when fitting linear models. Presented at spring regional meeting of ENAR, Ames, Iowa.

SPEED, F. M. and HOCKING, R. R. (1976). The use of the R( ) -notation with unbalanced dat The American Statistician. 30, 30-33.

SPEED, F. M., HOCKING, R. R., and HACKNEY, O. P. (1977). Methods of analysis of unbalance data. To appear in JASA.

BIOGRAPHIES

Ronald R. Hocking received a Ph.D. in Statistics from Iowa State University in 1962. He is currently Professor of Statistics and Head of Statistical Services at Mississippi State University. Hocking, a fellow of ASA, was also 1976 Chairman SPES and 1977 Program Chairman SPES.

Olga P. Hackney received a Ph.D. in Statistics and Biometry from Emory University in 1977. She is currently Assistant Professor of Statistics at Mississippi State University.

F. Michael Speed received a Ph.D. in Statistics from Texas A&M University in 1969. Speed is currently Associate Professor of Statistics at Mississippi State University. He was previously employed by Texas A&I University and NASA in Houston.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
eld at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

DISCUSSION FROM WORKSHOP ON ANALYSIS OF VARIANCE FOR UNBALANCED DATA

edited by
Richard M. Heiberger
University of Pennsylvania

The discussion, except as noted, was recorded and then transcribed and edited for ıoothness. The speakers have not reviewed the comments attributed to them.

)hn A. Nelder, Rothamstead Experimental Station (written comments read at the meeting by ıe session chair): It appears from your alternatives as if you regard the stratification ' A, B, C as the main determinant of the form of SS required. (I am using 'strata' as in ' 1965 paper to denote different subspaces of contrasts which define the error strata.) I)ubt if this is true. A more relevant factor, I suggest, is whether the parameters corres-)nding to a term are nuisance parameters or not. In the anova context, I speak of the for-:r kind as being part of the minimal model, i.e. a model that must be fitted first, before ıy further fitting of non-nuisance (i.e. interesting) parameters can begin. In BIBs, .ocks form the minimal model, and (ignoring inter-block information for the moment), blocks .im.treatments would not be calculated, but treatments elim.blocks would.

A second point is that there is no point in calculating A elim.B if B is effectively ıll. In fact there are good reasons for not doing so, because the effect will be to in-:ease the sampling variances of the A estimates when A and B are non-orthogonal. Thus no :quence can be defined a priori, without knowing the size of the effects concerned. The)nclusion I draw from this is that there can be no standard form of output for a program, ıly the ability to define and fit a sequence of models and to present in tabular form the .S. for the models in that sequence. (The user will not be able to avoid thinking!)

A further point is that often the assumption of a single error term is unjustified.)w, the estimation of means and variances, and the assignment of associated measures of un-:rtainty (in whatever form) is an unsolved problem for unbalanced designs. However, this ; not a justification for pretending that there is only one error term, when in fact there ; not, because this can lead to gross underestimation of the standard errors.

ırray Aitkin, University of Lancaster (comments based on his contributed paper which ap-:ars elsewhere in these Proceedings)

:iberger: The analysis recommended by Francis (1973), with which Professor Aitkin disa-rees, adjusted main effects for all interactions using BMD10V (BMDX64). I would like to sk Jim Frane to comment.

rane: As Professor Aitkin has pointed out there is an apparent loss of power if you do not)ol. However, that a non-significant interaction was observed may reveal instead that here was a loss of power due to sparcity of data. I think we have a philosophical differ-nce here. Are we interested primarily in being sure that there is no A effect or B effect?) we want to be conservative or liberal? We don't always want to do the same thing. It is nportant to have computer programs that can specify a number of different models. For ex-nple, we can build a set of contrasts to test Professor Aitkin's hypothesis using BMD10V.

ilkinson: The A main effect can be thought of as a combination of two independent esti-ates $\bar{x}_{11}-\bar{x}_{21}$ and $\bar{x}_{12}-\bar{x}_{22}$ which have the same expected value if no interaction is present. he statistically best overall estimate of the A effect then comes from weighting the two ndependent estimates according to the proper weighting of Fisherian information theory, hat is, proportional to the cell frequencies.

Frane: I think we have illustrated that we can think of circumstances in which different kinds of hypotheses are interesting. I think if we had a particular client who came in an talked to each of us that we would find in fact that for the particular problem we would b closer than we appear to be here.

Heiberger: It is very difficult to provoke controversy sometimes.

Kevin Price: Agriculture Canada, Ottawa (written comments based on his impromptu floor comments): For some clients the 'β-model', with its emphasis on the main effects, is a more natural representation of their interests than the 'μ-model', emphasizing cell means. Suppose, for example, the experiment has measured the yield of wheat of different varietie (factor A) sown on different dates (factor B). If there is no A*B interaction, the result of the experiment may be used to make recommendations such as 'A_j is the best variety' or 'B_j is the best seeding date'. If there is an interaction, these recommendations must be qualified: 'A_i is the best variety if sown on date B_j'.

Such a client surely has no trouble interpreting the 'alphas and betas', which correspond, more directly than do the cell means, to the purpose of his experiment.

The phrase 'two way classification with interaction' is a strange combination of concepts. If there is an interaction, we have to say that the cells cannot be classified in two dimensions, but must be treated as a single dimension, as $\mu_{k,k=1\to n\times m}$ rather than as $\mu_{ij,i\to m,j=1\to n}$. 'Two way classification' is what our client hoped he had; 'with interactio is our way of telling him that his hopes were in vain.

Hocking: If you've ever watched Billy Graham give a sermon, you notice after it's over that people come down from the audience and announce that they've been converted. Well, I think I've just got a convert.

Searle: In terms of the relationship between the, so to speak, μ_{ij} models and the β models, or this discussion of interactions, or the discussion that Ron just had of the relationships between some cells and other cells, all that is fine, but everyone has been talking of models with all cells filled. Let's take a ten by five with 24 cells filled and the other 26 not. Now try to write the relationships or do anything that will make any sense to anybody except in the μ_{ij} model. This is the topic of the meeting. I do agree that in some contexts we do know everything about the two way classification. I don't think any of us knows everything but if we could pool all our knowledge we would know everything about it. The danger of this computing is, as Jim Goodnight says, we can do th computing for these other models. If we are stupid enough to go ahead and do it with a si factor experiment with, say, 200 or 300 cells and 45 of them filled...(laughter). You laugh. The one little piece of data there is in the Linear Models is about somebody with some social science data with 9 factors, I can't really remember the number, after a littl bit of editing there were approximately 5000 cells. In the linear model with all interactions there were about eight million of the damn things. I made a suggestion that when we get beyond the two way classification, and when we get beyond the all-cells-filled-case we really can't make any sense out of interactions. In that case, yes, I am a convert to the model.

Aitkin: In a health sciences survey in Sidney, there were 22 variables and 2700 observations. In that survey many 3- and 4-way interactions were significant and interpretable.

Larry L. Laster: University of Pennsylvania: I heard some talk about two-way models, a little talk about three-way models and some holes all over the place where nothing would make any sense. I am still a little unsure about the analysis of my data. I deal with clinical trial studies with 2, 3 or 4-way layouts which have reasonably well-defined factor structures. All my cells are filled, though not exactly balanced. And in such situations I have heard strong commentary from certain speakers but not from others. I would appre-

ciate it if Dr. Searle would comment on that situation where all cells are filled. How would you treat certain experimental situations? What hypotheses would you seek to test? I understand that there are many, many faults but most of the real research that I carry on, and that a lot of other people are concerned with, is not based on that terrible loss situation.

Searle: From the way you describe it, a clinical situation where you have data everywhere, I suppose for the moment that every cell has some data in it and the number of observations in every cell are fairly similar. But what do we mean by fairly similar? The usual standard is to compare $1/\sqrt{n_{ij}}$ and if the numbers are more or less the same then we say that the n_{ij} are more or less equal. My own feeling in this case is that I would do anything to strive to get a balanced analysis because that, after all, is easy to understand (and it's almost an aside to say that the arithmetic is easy to do). If my numbers of observations were 5, 6, 7, 9 or 5, 8, 5, 6 I might set aside some data points and do a balanced data analysis. Then I would put back the observations and set aside some others at random. I would do this three or four times, hoping that the conclusions I might draw from these several analyses would be the same. If they weren't I might have compounded my difficulties. I think the balanced data analyses are so easy that this is one way of striving. Another thing one could do is unweighted analysis of means, that is, to simply take the cell means and treat each as if it were a single observation. Then the analyses of variance are very easy and, if I remember rightly, the tests of hypotheses that would come out of the F statistics are reasonably useful and interpretable. I would go, in fact, to the unbalanced data analysis almost as the last thing. Now, of course, if you are really in the unfortunate situation where there are 4 cells and the number of observations are 200, 200, 200, and 6 then you are up a gum tree as my Australian friends would say. But I do feel that this is something which you can do. Does that help?

Heiberger: I would like to thank everyone for participating. I am sure that all of us would be more than happy to continue the discussion after the close of the session.

The contributed paper by Michael Kutner appearing in this Proceedings also addresses the question discussed in this session. The following comment was received from David G. Herr of the University of North Carolina-Greensboro after the meetings were over.

COMMENTS ON SESSION 2 OF WORKSHOP 1 OF THE TENTH ANNUAL SYMPOSIUM OF THE INTERFACE

David G. Herr
UNC - Greensboro

The preceding papers have been interesting in demonstrating that, however well understood the analysis of unbalanced, two-way designs may be, there is the need for a perspective or overview from which the various advocacies can be considered. The geometric or coordinate free approach to linear models provides such a perspective. For example, consider the debate concerning the cell mean model (CMM) versus the over-parametrized or grand mean model (GMM).

Suppose that there are a total of n observations in the design to be analyzed. Let Y be the n x 1 vector of random variables used to model these observations. Let U be a subspace of R^n. Then a linear model is defined by requiring $EY \in U$. A linear hypothesis requires $EY \in W$ for W a subspace of U. Under the usual assumptions on Y-EY, the hypothesis H: $EY \in W$ is rejected for large values of

$$\frac{|| P_W Y ||^2/\dim W}{||(I-P_U)\ Y\ ||^2/(n-\dim U)}$$

This statistic is distributed as an F(dim W, n-dim U). Here P_W, P_U are the perpendicular

projections on the subspaces W and U respectively. Viewed in this way the CMM and GMM are simply equivalent ways of specifying U. The CMM has the advantage of specifying Y as the range of a full rank transformation (matrix). The GMM has the advantage of explicitly exhibiting parameters that are useful and familiar to many. The CMM makes sense even with empty cells in the design. The GMM suggests regression like model comparisons which appea to many. Careful consideration will show that each model is just a crutch to help specify subspaces W, ie. specify hypotheses, of interest to the investigator. There seems little reason not to use each crutch where it is most useful, i.e. why be crippled with only one crutch?

This geometric view applied to unbalanced, two-way designs has been considered by Burdick, Herr, O'Fallon, and O'Neill (1974) in the case of all cells filled and Herr (1976 in the case of empty cells. I have named the three analyses championed by Hocking, Wilkinson and Aitkin the standard parametric (STP), each adjusted for the other (EAD) and the hierarchical, rows first then columns (HRC) respectively. The following is a summary their attributes for an a x b design.

Analysis	Parametric hypothesis tested	Model Comparisons Made	Orthoganali properties
STP			
rows	$\mu_{1.} = .. = \mu_{a.}$ $(\alpha_p = 0)$	$y = \mu + \beta + \gamma + e$ vs. $y = \mu + \alpha + \beta + \gamma + e$	In general there is no orthoganal ty between any t of rows, columns and interactions
columns	$\mu_{.1} = ... = \mu_{.b}$ $(\beta_q = 0)$	$y = \mu + \alpha + \gamma + e$ vs. $y = \mu + \alpha + \beta + \gamma + e$	
EAD			
rows	$\mu_{p*} = (\mu_{*p})_{p*}$ $p = 1(1)(a-1)$	$y = \mu + \beta + e$ vs. $y = \mu + \alpha + \beta + e$	Each of rows and columns are orth gonal to interac tions but not to each other
columns	$\mu_{*q} = (\mu_{p*})_{*p}$ $q = 1(1)(b-1)$	$y = \mu + \alpha + e$ vs. $y = \mu + \alpha + \beta + e$	
HRC			
rows	$\mu_{1*} = ... = \mu_{a*}$	$y = \mu + e$ vs. $y = \mu + \alpha + e$	Completely orthogonal analysis
columns	$\mu_{*q} = (\mu_{p*})_{*q}$ $q = 1(1)(b-1)$	$y = \mu + \alpha + e$ vs. $y = \mu + \alpha + \beta + e$	

Here μ_{p*} is the weighted average of cell means in the pth row with weights proportional to the cell sizes. Also $(\mu_{p*})_{*q}$ is the weighted average of the μ_{p*} with weights proportional to the cell sizes in the qth column.

What seems clear from the discussion here today is that none of these analyses lack su porters. Furthermore it also seems clear that the mathematics of each analysis is well un derstood by many. The problem of choosing one or another of these analyses is not mathema tical, but philosophical in nature. What is needed then is a clear, concise explanation of

ıe philosophy, not the mathematics, of each. Then, as Frane and Searle suggested, the in-
ıstigator, in consultation with the statistician, could decide the analysis appropriate
ır the problem at hand.

REFERENCES

ŦRDICK, D. S., HERR, D. G., O'FALLON, W. M. and O'NEILL, B. V. (1974). "Exact methods in the unbalanced, two-way analysis of variance - a geometric approach." Comm. Statist. 3, 581-94.

ŁRR, D. G. (1976). "A geometric characterization of connectedness in a two-way design." Biometrika 63, 93-100.

NONLINEAR MODELS WORKSHOP

John M. Chambers and John E. Dennis, Chairpersons

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

NONLINEAR STATISTICAL DATA ANALYSIS

Roy E. Welsch
Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

ABSTRACT

This paper discusses how recent progress in nonlinear optimization methods can help data analysts working with nonlinear models and nonlinear estimation procedures. Some advances in estimation for linear models such as robust methods and diagnostic sensitivity analysis have been partially generalized to nonlinear models, but many problems remain. These problem areas are discussed along with certain ways in which nonlinear optimization algorithms could be modified to help the statistician.

Key words: Convergence; covariance; diagnostics; influence; leverage; nonlinear; regression; robust; sensitivity analysis.

1. INTRODUCTION

With the advent of robust estimation procedures, even linear regression models require the use of some form of nonlinear optimization routine. This presents some new problems, both for the statistician and the numerical analyst who specializes in nonlinear optimization. Our purpose in this paper is to discuss a few of these problems in order to encourage the interaction of these two groups of researchers.

Too often, optimization algorithms are developed with the primary emphasis on finding a local minimum in the most efficient way with little regard for the nature of the problem being solved. On the other hand, statisticians often take optimization algorithms as given and make little or no attempt to influence their development so that they will more nearly satisfy the needs of a statistician. Far too little of what is already known about nonlinear methods has found its way into routine statistical data analysis.

2. ROBUST METHODS

2.1 Notation. Since the notation used to describe nonlinear statistical problems is by no means standard, we will need to introduce some notation. The regression model is

$$y_i = m_i(\beta)+\varepsilon_i \qquad i=1,\ldots,n \tag{1}$$

where β is a p by 1 vector and $m_i(\beta) = x_i\beta$ in the linear case. Let $r_i(\beta) = y_i - m_i(\beta)$ and

$$R(\beta) = \left(r_1(\beta),\ldots,r_n(\beta)\right)^T.$$

For robust regression we generally need to minimize

$$f(\beta,s) = \sum_{i=1}^{n} g(s)\rho\left(\frac{r_i(\beta)}{s}\right) + h(s) \tag{2}$$

where g(s) and h(s) are related to the scale parameter, s (often estimated), and $\rho(t)$ is a robust loss function such as:

$$\begin{cases} \dfrac{t^2}{2} & |t| \le c_1 \\ c_1|t| - c_1^2/2 & |t| > c_1 \end{cases} \tag{3}$$

or

$$\frac{c_2^2}{2}\left[1-\exp(-t^2/c_2^2)\right] . \tag{4}$$

Both of these functions are discussed in detail in Holland and Welsch (1977). For traditional least-squares estimation

$$\begin{aligned} \rho(t) &= t^2 \\ g(s) &= 1 \\ h(s) &\approx \log s. \end{aligned} \tag{5}$$

Four problems related to the structure of (2) will be discussed here: scale, iteration, convergence and covariance.

2.2 Scale. Letting $h(s) \approx \log s$ does not lead to a robust scale estimate except for some nonconvex loss functions. However, Huber (1975) chose $g(s)=s$ and $h(s)\approx s$ and showed that this led to a robust scale estimate for the Huber loss function, (3). In addition, the scale estimation can be based on a single objective function involving both β and s just as in the least-squares case. No one has yet shown how to do this for other robust loss functions.

We should note that Huber did not actually use (2) to simultaneously estimate scale, but instead let

$$s^2_{new} \approx \sum_{i=1}^{n} \left[\rho'\left(\frac{r_i(\hat{\beta}_{old})}{s_{old}}\right)s_{old}\right]^2$$

after $\hat{\beta}_{old}$ had been found based on s_{old}.

Often the scale problem is avoided by estimating scale just once at the starting values, or using the Huber approach until convergence and then another robust loss function such as (4) with the scale fixed at the final value obtained from the Huber iterations. This procedure is often satisfactory in practice but gives us little to go on theoretically.

Another approach is to set $g(s)=1$ and $h(s)=0$ and add an equation to the normal equations of the form

$$\sum_{i=1}^{n} \left[\rho'\left(\frac{r_i(\beta)}{s}\right)\right]^2 \approx \text{constant}. \qquad (6)$$

This means that a system of nonlinear equations must be solved and they are not derived from a single objective function such as (2).

Scale estimation cannot be forgotten about [see Holland and Welsch(1977) for details] and the fact that scale is special is something that statisticians must communicate to developers of nonlinear algorithms.

2.3 Iteration. It is convenient to form weights by defining

$$w_i(t) = \rho'(t)/t \qquad (7)$$

and letting <w> denote a diagonal matrix of the $\{w_i\}_1^n$. At least three iteration schemes have been proposed for the robust estimation of linear models with fixed scale:

$$\hat{\beta}_{new} = \hat{\beta}_{old} + (X^T<\rho''>X)^{-1}X^T<w>R(\hat{\beta}_{old}) \quad (8$$

$$\hat{\beta}_{new} = \hat{\beta}_{old} + (X^T<w>X)^{-1}X^T<w>R(\hat{\beta}_{old}) \quad (9$$

$$\hat{\beta}_{new} = \hat{\beta}_{old} + (X^TX)^{-1}X^T<w>R(\hat{\beta}_{old}). \quad (10$$

Both w_i and ρ_i'' are functions of $r_i(\hat{\beta}_{old})/s$. The first is Newton's method, the second has been suggested by Beaton and Tukey (1974), and the third is due to Huber (1975).

Detailed theoretical and computational studies comparing these methods have not been made. The weighted approach, (9), has received the most attention because of its relation to weigh least-squares, but (10) is computationally simpler. A statistician forced to use a Gauss-Newton nonlinear regression program automatically gets (8).

Often just one step from a starting value is used. The asymptotic properties of one-step estimators using (1) and (3) are the same as fully iterated estimates (Bickel, 1975), but unknown for (2) since Holland and Welsch (1977) show by Monte Carlo that in this case the one-step and fully iterated asymptotics are most likely different. We also do not know how one-step estimators affect the rates of asymptotic convergence.

2.4 Convergence. Most optimization algorithms include several convergence criteri which are often based on the gradient or relative change in the parameters. These have little real meaning for a statistician. The gradient is not scale free and therefore especi ally troublesome for robust methods.

John Dennis (1976) has proposed a scale-free test involving the maximum of the cosines of th angles between $R(\hat{\beta})$ and the p columns of the Jacobian (J). Allen (1976) simplifies this by using the absolute cosine between the residual vector R and the J column space projection, $J(J^TJ)^{-1}J^TR$. Both of these solve the scale problem, but lack direct statistical appeal.

Pratt (1977) proposes using

$$\delta^T A \delta \leq \varepsilon \quad (11$$

where $\delta = \hat{\beta}_{new} - \hat{\beta}_{old}$ and A is most often taken to be the Hessian but logically could be the inverse of another measure of covariance. This has statistical appeal because

$$\max_\lambda (\lambda^T\delta)^2/\lambda^TA^{-1}\lambda = \delta^TA\delta$$

where λ is any linear combination of the parameters.

A more direct approach is used by Huber (1975). Let C be the current estimate of the covariance and check to see if

$$|\delta_j| < \varepsilon s\sqrt{C_{jj}} \qquad j=1,\ldots,p. \tag{12}$$

Since $s \cdot C_{jj}^{\frac{1}{2}}$ is the estimated standard error, convergence is measured in terms of statistical variability or, conversely, precision. Why can't we have some of these options available in the nonlinear procedures we use?

2.5 Covariance. The last section showed how the estimated covariance matrix can play a role in convergence criteria. Certain theoretical considerations have usually led statisticians to be more or less happy with the inverse of the Hessian as the estimated covariance matrix. However, this does not completely solve the problem.

In the first place, the exact Hessian (i.e. actual second derivatives) is rarely known and the only thing available is the current approximation to it. Most nonlinear optimization algorithms are designed for speed of convergence and not estimation of the Hessian. When choices are possible, as in the quasi-Newton update methods, the statistician's desires to have a good covariance estimate are not taken into account. Only close interaction between statisticians and numerical analysts can help to develop algorithms that balance speed of convergence against the need for accurate covariance estimation.

On the other hand, statisticians are unsure about how to estimate covariance, especially in the robust case. Many suggestions have been made including

$$(X^TX)^{-1},\ (X^T\langle w\rangle X)^{-1},\ \text{and}\ (X^T\langle \rho''\rangle X)^{-1} \tag{13}$$

in the linear case, and

$$H^{-1},\ H^{-1}J^T\langle \rho'^2\rangle JH^{-1},\ \text{and}\ (J^TJ)^{-1} \tag{14}$$

in the nonlinear case. Another intriguing proposal has recently been put forth by Hill (1977a). Statisticians cannot really expect too much help from numerical analysts on covariance estimation until they narrow this list. There is, however, no reason to limit the output of a nonlinear statistical package to just one estimate of covariance. Perhaps user control is in order here.

3. SPECIALIZED ALGORITHMS

3.1 Special structure. Statisticians often use general purpose algorithms to solve

problems with special structure. When nothing else is available, there is no other course of action. However, there is a lot to gain by encouraging the development of specialized algorithms. Loglinear models (DuMouchel, 1976, and Bishop, Fienberg and Holland, 1975) and generalized linear models (Nelder and Wedderburn, 1972) provide but two examples.

For years, Levenberg-Marquardt type algorithms have exploited the special structure of nonlinear least-squares. Further advances have recently been made in this area by Dennis and Welsch (1976) who use quasi-Newton methods to approximate only the second-order portion of the Hessian since the other part, J^TJ, is known exactly.

In robust estimation we may want to exploit special structure in other ways. With robust loss functions there is a need to decide on a useful range of values for the parameter c_1 or c_2 in (3) and (4). In the linear case, one way to do this is to compute the predicted residual

$$p_i(c) = y_i - x_i\hat{\beta}_{(i)}(c) \tag{15}$$

where $\hat{\beta}_{(i)}(c)$ is the robust estimate of β obtained without using the i^{th} observation (or some approximation to this). A criterion for choosing c is to examine the region of the minimum with respect to c of

$$\sum_{i=1}^{n} p_i^2(c). \tag{16}$$

This requires clever techniques for the successive removal of observations and for the evaluation of (16) at different values of c.

4. REGRESSION DIAGNOSTICS

4.1 Linear case. For the linear model, considerable progress has been made in understanding how to search for influential observations [Welsch and Kuh (1977), Cook (1977) and Hill (1977b)]. Some useful measures are: the diagonal elements, h_i, of the projection matrix

$$H = X(X^TX)^{-1}X^T; \tag{17}$$

the externally studentized residuals

$$r_i^* = r_i/s_{(i)}(1-h_i)^{\frac{1}{2}} \tag{18}$$

where $s_{(i)}^2$ is the estimate of residual variance obtained without using the i^{th} observation; the change in coefficient estimates

$$\hat{\beta}-\hat{\beta}_{(i)} = (X^TX)^{-1}x_i^Tr_i/(1-h_i); \tag{19}$$

and the change in fit

$$x_i(\hat{\beta}-\hat{\beta}_{(i)}) = \frac{h_i r_i}{1-h_i} . \tag{20}$$

Clearly the influence of subsets of more than one observation is also of interest, but we will not pursue that topic here.

4.2 Nonlinear case. How can these ideas be extended to nonlinear computations? We could always use the existing methods in the neighborhood of the solution (local minimum) by linearizing the model appropriately. However, this will not answer questions about what would happen if we were to rerun the nonlinear optimization procedure without the i^{th} observation. Naturally, one would be reluctant to run n separate nonlinear regressions to get this diagnostic information. (However, this information would provide a useful estimate of covariance using jackknife techniques.)

If we do not want to perform n+1 regressions we can compute diagnostics like (17) to (20) at each iteration, then note the observations that had a high influence at that iteration and accumulate this information at the end. Perhaps separate runs (from the original starting values) with each of these observations (or a group) removed could then be made. This technique has worked well in practice, and our early fears that every point would turn up as a leverage point at some iteration have proved to be unfounded.

We do not have to use a local linear model in order to compute diagnostics at each iteration. Recent work on the nonlinear least squares problem by Dennis and Welsch (1976) uses J^TJ+S to approximate the Hessian where S_+ (for the next (+) step) satisfies the quasi-Newton equation

$$S_+(\hat{\beta}_+-\hat{\beta}) = (J_+-J)^TR(\hat{\beta}_+) = z . \tag{21}$$

One local approximation to $\hat{\beta}_{(i)}-\hat{\beta}$ is given by

$$-(J_{<i>}^TJ_{<i>}+S_{(i)})^{-1}J_{<i>}^TR_{<i>}(\hat{\beta}) \tag{22}$$

where <i> denotes a vector or matrix with the i^{th} row removed. Of course, we do not know $S_{(i)}$ and one way around this is to replace it by S. Since rank two update formulas are used to modify S it is possible to build an approximation to $S_{(i)}$ as well. We have built a new approximation to $S_{(i)}$ at each iteration because of the desire not to store n separate

matrices. This is a fertile area for research on the clever use of numerical linear algebra

Clearly some rules are needed to determine when the differences, $\hat{\beta}-\hat{\beta}_{(i)}$, are large. Usually this brings us back to a need for an estimate of the covariance, a problem we have already addressed.

5. BOUNDED INFLUENCE ESTIMATES

5.1 Diagnostic estimates. The notions of leverage and influence lead one to consider estimation procedures which bound the influence of individual (and perhaps subsets of) observations. In the linear case one natural way to proceed is to solve the system of p equations

$$\sum_{i=1}^{n} \rho'\left[A_i x_i (y_i - x_i \beta)\right] = 0 \tag{23}$$

for β. Here $\rho(\cdot)$ is again a robust loss function and A_i is often proportional to

$$(X^T X)^{-1} \quad \text{or} \tag{24}$$

$$(X^T X)^{-1}/(1-h_i). \tag{25}$$

For the motivation behind these two forms see Hinkley (1976) and Welsch (1977).

We note again how a linear problem has turned into a system of nonlinear equations. There is much special structure in this particular problem that we would be wise to exploit. Naturally, we could also ask how to do bounded influence estimation for nonlinear models.

Bounded influence estimators are designed to provide alternative estimates and not to supplant least-squares or other traditional estimators. The alternative estimates are then compared to regular estimates to gain a deeper insight into the nature of the data and model being studied.

6. AVAILABILITY OF PROGRAMS

6.1 Problems. The major problem with many statistical packages is that they offer no way to perform nonlinear optimization. If they do, they often do not provide a language in which to write the model [a Fortran subroutine is needed]. Finally, the user must supply derivatives. This is totally unnecessary because either numerical or symbolic differentiation could be used instead.

The TROLL system provides a choice of nonlinear routines, a modeling language, and derivatives. SAS provides the first two but still requires user supplied derivatives.

When available, many Levenberg-Marquardt routines are not reliable for large residual problems. The Dennis and Welsch (1976) proposals may help with this. As our earlier discussions have indicated most nonlinear routines do not provide adequate convergence options, covariance estimation, and diagnostic (leverage and influence) information.

While we have seen a lot of progress in recent years on nonlinear optimization algorithms, this progress has not really been felt by the bulk of the users of statistics and especially the users of statistical packages. I think it is time we all work together to make nonlinear statistical modeling and optimization a reality.

7. ACKNOWLEDGEMENT

This work has been supported, in part, by National Science Foundation Grant MCS76-00324.

8. REFERENCES

ALLEN, D. M. (1976). Private communication.

BEATON, A. E. and TUKEY, J. W. (1974). The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data. Technometrics, 16, 147-185.

BICKEL, P. J. (1975). One-step Huber estimates in the linear model. J. Amer. Statist. Assoc., 70, 428-434.

BISHOP, Y. M. M., FIENBERG, S. E. and HOLLAND, P. W. (1975). Discrete Multivariate Analysis: Theory and Practice. Massachusetts Institute of Technology Press.

COOK, R. D. (1977). Detection of influential observations in linear regression. Technometrics, 19, 1-14.

DENNIS, J. E. (1976). Nonlinear least squares and equations. The State of the Art of Numerical Analysis. D. Jacobs and R. Schrinen, Eds., Academic Press.

DENNIS, J. E. and WELSCH, R. E. (1976). Techniques for nonlinear least squares and robust regression. 1976 Proceedings of the Statistical Computing Section. American Statistical Association, Washington, D. C., 83-87.

DUMOUCHEL, W. H. (1976). On the analogy between linear and log-linear regression. Technical Report (67), Department of Statistics, University of Michigan.

HILL, R. W. (1977a). On estimating the covariance matrix of robust regression M-estimates. Submitted for publication.

HILL, R. W. (1977b). Robust regression when there are outliers in the carriers. Unpublished doctoral thesis, Department of Statistics, Harvard University.

HINKLEY, D. V. (1976). On jackknifing in unbalanced situations. Technical Report (22), Division of Biostatistics, Stanford University.

HOLLAND, P. W. and WELSCH, R. E. (1977). Robust regression using iteratively reweighted least-squares. Communications in Statistics, A6(9), 813-827.

HUBER, P. J. (1975). Robust methods of estimation of regression coefficients. Presented 2nd Int. Summer School on Problems of Model Choice and Regress. Anal. at Rheinhardsbr G.D.R., November 8-18.

NELDER, J. A. and WEDDERBURN, R. W. M. (1972). Generalized linear models. J. Roy. Statis Soc. Ser. A, 135, 370-384.

PRATT, J. W. (1977). Private communication.

WELSCH, R. E. (1977). Regression sensitivity analysis and bounded influence estimation. Working Paper, Sloan School of Management, Massachusetts Institute of Technology.

WELSCH, R. E. and KUH, E. (1977). Linear regression diagnostics. Working Paper (923-77) Sloan School of Management, Massachusetts Institute of Technology.

BIOGRAPHY

Roy E. Welsch received his Ph.D. in mathematics from Stanford in 1969 and is an Assoc ate Professor at the Sloan School of Management, Massachusetts Institute of Technology. F the past four years, he has also been a Senior Research Associate at the National Bureau o Economic Research Computer Research Center which is responsible for the development of the TROLL system. Welsch, an associate editor of JASA, was 1975 Publications Liaison Officer and is 1978 Secretary-Treasurer of the Statistical Computing Section, ASA.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

MLP, A MAXIMUM LIKELIHOOD PROGRAM

G.J.S. Ross
Statistics Department, Rothamsted Experimental Station.
Harpenden, Herts, AL5 2JQ, England

ABSTRACT

MLP is designed to make it easy for the non-specialist to fit appropriate non-linear models to data. The program includes a wide range of standard models for fitting curves, distributions and assays, with appropriate statistics and graphical output. There is a user's language for fitting other models specified as functions of parameters. The methods used depend on both the model and the data. Optimization is used, but care is taken to ensure that the objective function is well-conditioned, approximately quadratic and bounded.

1. INTRODUCTION

The other speakers in this session have discussed the present state of the art with regard to the use of general optimization routines for constrained or unconstrained non-linear functions of several variables. The emphasis has been on the choice of algorithm for different problems, and the difficulties likely to be encountered.

I wish to present an alternative approach, the presentation of non-linear model-fitting problems in terms of objective functions which are readily optimized. Since this approach requires some detailed study and understanding of the relationship between each model and the data to which it is fitted, it is best provided as software through the medium of programs which treat each model as a special case and to provide the best fitting formulation that can be found. It is only by this approach that the non-specialist (in statistical computing, that is) may fit his models in a routine manner without meeting the problems associated with optimization.

The problem of deciding which models to include in an integrated modelling-package is partly decided by the users themselves. The models most commonly required must clearly be included, and in addition such specialisations or generalisations as are necessary to enable users to decide which models are adequate. In this way a library of models, mostly of few parameters but of wide application, is being built up. Some models are very easily fitted while others are very data-dependent in their ease of solution.

The Maximum Likelihood Program, MLP, is designed to provide solutions to a wide range of models, linear and non-linear, that arise in the biological sciences and elsewhere, with a standard user's language concealing the many different techniques used in obtaining solutions, which will now briefly be described.

2. CONDITIONING A NON-LINEAR PROBLEM TO FACILITATE OPTIMIZATION

In a previous paper (Ross (1970)) I discussed how the same optimization problem could be formulated in different ways sometimes easy and sometimes difficult to optimize.

The first important technique is parameter transformation, reparameterizing so that the log-likelihood function (or residual sum-of-squares) approximates a well

conditioned quadratic, leading to rapid convergence from arbitrary starting values. The original parameters are then recovered by an inverse transformation. The working parameters are called stable if their final values differ little from initial values, and these are found in practice as the expected values of descriptive statistics of the data. For example, for any satisfactory fit to a non-linear curve the fitted curve must pass close to the data points, and therefore each fitted value is a potential stable parameter, with finite range. The practical difficulty is to find a set of stable parameters which are approximately uncorrelated, and from which the defining parameters (and hence the set of fitted values and the likelihood) may be easily calculated. In each case it is necessary to make some preliminary analysis of the data from which suitable working constants (such as means or scaling factors) may be derived. When the algebraic problems are intractable a simpler approximation to the inverse transformation may provide a set of parameters which are nearly as efficient for the purpose of fitting.

The second technique, of particular use in curve fitting, is that of separability of linear parameters. Given trial values of non-linear parameters then any linear parameters may be estimated directly by linear regression, so that optimization is in the reduced space of the non-linear parameters only, as was first pointed out by Richards (1961). These reduced functions may be less easily optimized because they inevitably have local maxima or flattish regions, and so stable non-linear parameters must be sought.

The third technique is that of sequential optimization, use of a sequence of models, perhaps of increasing complexity, either to arrive at suitable initial values for the final optimization or to find a suitable transformation for it. Thus if the initial transformation fails to find a solution rapidly this may be because the working constants obtained from the data were not appropriate, and could be improved from the transformatio obtained after a few iterations of optimization.

In practice it has been found that use of these principles makes it easy to choose initial values and step lengths for optimization routines not requiring derivatives (for the transformations render analytical differentiation almost impracticable). For stable parameters lying in a defined a priori range, the step length may be a simple fraction of that range, provided the function is not too asymmetric about its minimum.

3. UNIQUENESS AND EXISTENCE OF SOLUTIONS

It is well-known that non-linear objective functions may have more than one solution, or no solution at all. Therefore the remedy of transforming to stable parameters would seem, as a general principle, to founder on such cases. In fact it provides the means of understanding how they arise.

If, for example, a curve with p parameters is fitted to p points exactly, the system of non-linear simultaneous equations may have unique solution, or no solution, or several solutions, according to the positions of the points. Thus if the curve is always monotone increasing but the points are not, then there can be no solution. But if the solution relies on a quadratic equation, two solutions may be possible. If there are more than p data points the result may still depend on the position of p critical predicted points.

Another situation is where the fit is so poor that contrasting sets of solutions are obtained in which one subset of points are fitted well while the others are poorly fitted.

When each problem is specially programmed it is easier to ensure that non-existence of solutions is rapidly detected, and that ambiguity is avoided where possible, sometimes by requiring the user to specify one of two alternative forms.

4. AN EXAMPLE OF THE TREATMENT OF A MODEL IN MLP

One detailed example of the treatment of a model will have to suffice: the fitting of the logistic growth curve,

$$y = \theta_3/(1 + \exp(-\theta_1 - \theta_2 x))$$

by least squares.

A preliminary data analysis checks the number of data points and fits a straight line of y on x. This establishes the mean and standard deviation of x, which indicates the range and scaling of the x values and also the sign of θ_2, governing whether y ascends or descends with x.

If the data resemble a logistic in shape, and are reasonably uniformly distributed on the range of x, then the predicted values at three equally spaced points ($\bar{x}$-s, $\bar{x}$ and $\bar{x}$+s) form the basis for a set of stable parameters which is algebraically tractable.

But since the parameter θ_3 is linear, and may be fitted by regression through the origin if θ_1 and θ_2 are known, only two working parameters are required. These may be the ratios of the first to the second predicted value and the second and the third, thus

$$\phi_1 = f(\bar{x}-s)/f(\bar{x})$$

$$\phi_2 = f(\bar{x})/f(\bar{x}+s)$$

and $$\theta_3 = \sum y\, z(\phi_1,\phi_2) / \sum (z(\phi_1,\phi_2))^2,$$

where z is the equation of the curve in terms of ϕ_1 and ϕ_2, apart from the scale, θ_3. Now $0 < \phi_1 < \phi_2 < 1$ for if $\phi_1 > \phi_2$ then the curve increases more rapidly than an exponential and cannot be of the right form. Therefore either a solution is found, or a bound is violated, or too many iterations are required, and in the latter case a second chance is allowed with new x values at which the critical predicted values are used. Ths second chance usually finds solutions that were missed because of the uneven distribution of data points.

5. MODELS FITTED BY MLP, AND EXAMPLES OF USERS LANGUAGE

The models in MLP are first classified by type of problem, for example, into curve fitting, quantal response models (biological assay), discrete distributions, continuous distributions, genetic frequency models, regression or general user models.

A comprehensive set of data manipulation facilities allow data to be read, generated or transformed prior to model fitting. Then each set of data may be fitted by several models of the same type, or several sets of data may be fitted by the same model and the results compared or amalgamated. The program is therefore a data analyst's tool in which individual details of modelling are subordinated to the wider task of interpreting data and obtaining reliable estimates of quantities that are of interest.

The curve fitting section, for example, allows up to 20 sets of data to be read at a time, analysing each set singly and then in combination by a 'parallel' curve analysis analogous to that of parallel lines. The curves include single and compound exponential curves, compartment models, growth curves such as the logistic, inverse polynomials (ratios of polynomials) and also simple linear polynomials. In most cases the curve may be constrained to pass through the origin or to possess a fixed asymptote, and all the models are related as a partially ordered network so that adequacy of fit may be assessed. Output includes a graph of the curve, estimates of slope and standard errors of prediction, the maximum of the curve if it has one, the positions of asymptotes and any desired extrapolated or interpolated value.

The distribution fitting section provides solutions to the problem of Normal mixtures, the lognormal with unknown origin and other models of practical importance; also a range of discrete distributions such as the Negative binomial and Neyman Type A.

Biological assay is provided in the form of comparison of probit regression lines, and other related models. The output reflects the traditional presentation (Finney, (1971)) in terms of median lethal dose and other percentage points. There are further models of interest to biologists such as dilution series estimation and genetic models.

To fit a standard model it is only necessary to supply the data in the form required for that model, and to specify the model by name. For example, to fit a logistic curve to a set of observations one need only write

```
DATA   1     2     2.9     4     4.7     6     10
       .72   2.5   4.1     7.8   8.3     10.2  10.6;
CMODEL=LOGISTIC  FIT CURVE
```

whereas to fit a negative binomial distribution to frequency data one could write

```
DATA  24  31  21  22  11  14  6  3  2  3  /  0  1  1  ;
DMODEL=NEGBIN  FIT DIST
```

The sign (/) indicates grouping of frequencies.

Additional options may be specified, and there is a wide range of data manipulation facilities for transforming or editing data. Each model has its own procedures for obtaining suitable transformations for efficient optimization, but for routine work the details do not have to concern the user, who is often a biologist or non-specialist. The program attempts to recognise data that will not fit, either at the preliminary stage or after failure of the optimization routine to converge in reasonable time. Failure diagnostic messages may suggest alternative models, where appropriate.

6. GENERAL NON-LINEAR MODELS

Models not provided in the standard sections may be fitted by the user who has to choose his own parameterisation. The model is specified as a set of instructions

(in high level language) to compute the 'fixed part' of the model as function of parameters, and an option to select the error distribution (or 'random part'). If the linear terms are separable only the non-linear parameters need be specified. Other functions of parameters may be specified and these are evaluated when the model has been fitted, so that even if a transformation has been used the parameters of interest may also be obtained.

Although the general model fitting procedure is a means of giving direct access to the optimization routine it is hoped that users will be encouraged to seek the most effective formulation of the problem, and there are several diagnostic aids which simplify this procedure, such as contour diagrams of the function being optimized and listing of the partial derivatives of the fitted values with respect to the parameters. As a measure of the accuracy of the quadratic approximation a plot is provided of the discrepancy between the actual log-likelihood or sum-of-squares and the predicted values from the information matrix at the solution.

7. PROGRAM AVAILABILITY

The program is written in ANSI Fortran IV and is currently implemented on the following machine ranges: ICL System 4-70, IBM 3.70, ICL 1906B, Burroughs B6700 and CDC (Cyber) 7600. It may be distributed under licence agreement by application to The Programs Secretary, Statistics Department, Rothamsted Experimental Station, Harpenden, Herts AL5 2JQ, England. A comprehensive user's guide is available.

8. ACKNOWLEDGMENTS

The author would like to thank the Symposium organisers for the invitation to attend.

REFERENCES

Finney, D.J. (1971). Probit Analysis (3rd Edition), Cambridge U.P.

MLP User's Guide (1978). Rothamsted Experimental Station.

Richards, F.S.G. (1961). A method of maximum likelihood estimation. J.R.Statist.Soc., A, 135, 370-384.

Ross, G.J.S. (1970). The efficient use of function minimisation in non-linear maximum likelihood estimation. J.R.Statist.Soc., C, 19, 205-221.

GRAPHICS WORKSHOP

Jane F. Gentleman, Chairperson

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

COMPUTER GRAPHICS AVAILABLE TO STATISTICIANS

Barbara F. Ryan
The Pennsylvania State University, University Park, Pa. 16802

ABSTRACT

This paper surveys the graphics capabilities available to statisticians in most of the widely available general purpose statistics packages and in a few graphics-oriented statistics packages. The emphasis is on pictures for data analysis rather than on pictures for data presentation.

1. INTRODUCTION

The statistics packages chosen for this study (Figure 1) include the most widely available general purpose statistics packages, and a few more graphics-oriented packages. Figure 1 also indicates information available to the author, the plotting devices used by the package, and a few other facts about the packages.

Each package developer was sent a small data set consisting of 50 observations on 7 variables, 4 test problems and 4 additional questions concerning histograms, 6 test problems and 6 additional questions on scatterplots, and 10 questions on other graphical displays.

The results of these test problems and the answers to the questions provided by the package developers form the basis for this paper. In some cases, results were obtained from versions of packages not yet released. Additional information was obtained from reference manuals and tests carried out by the author on those packages available at Penn State University.

2. HISTOGRAMS

Histograms are (or should be) widely used by statisticians for a variety of reasons: these include looking for outliers, examining the shape of a distribution, comparing several groups. In spite of this, 5 of the 14 packages did not have any histogram capability, and at least one of the remaining produced histograms which are useless for any of the purposes mentioned above. On the other hand, BMDP has a wide variety of histogram displays.

The first test problem asked for a default histogram of an integer-valued variable. The variable had two extreme values, one at 10 and one at -10; all other values were between -2 and +2. The Minitab histogram is shown in Figure 2. This histogram was designed primarily for screening purposes, so it is fairly compact and fast printing.

The worst result for this test problem was the histogram produced by SPSS, shown in Figure 3. The two extreme values were completely obscured. SPSS produces one bar for each distinct value in the data set, then places the bars an equal distance apart. This obscures information about the shape of the distribution and the presence of outliers. This histogram was apparently designed to display frequencies of nominal data with relatively few categories, although this is not completely clear from the manual.

Another problem with histograms is shown by Genstat, which requires using Genstat commands to group the data before doing the histogram. There is no default grouping. If the grouping is misspecified, outliers are put in the last group. Thus the user can detect outliers only if he expects them.

TROLL has the capability of producing histograms on Calcomp plotters, Tektronix graphic terminals, and typewriter terminals in such a way that all look fairly similar, yet each makes use of special capabilities of the device being used.

Another important use of histograms is to compare groups. All packages which have a histogram capability (except SPSS) allow the user to specify the scale of a histogram. Thus he can get a histogram for each group, all on the same scale, suitable for comparison. Groups can be compared more easily using BMDP7D's side-by-side histograms (Figure 4). Another technique is to have one histogram, with different characters to denote the different groups. An example from P-STAT is shown in Figure 5; BMDP5D also uses this technique.

3. OTHER UNIVARIATE PICTURES

All packages surveyed can plot X_i versus i. Probability plots can be done by most packages. Omnitab II and Statsystem can do probability plots for many distributions. BMDP, Genstat, Minitab, SAS, TROLL, and TSAM all can do normal probability plots. BMDP can also do detrended normal probability plots.

Stem and leaf displays and box plots can be produced by Omnitab II and the TROLL Experimental Programs. BMDP2D produces very useful displays for data screening. A portion of the output is shown in Figure 6. Omnitab II has a command STATPLOTS which produces 4 plots on one page of computer output: X_i versus i, X_i versus X_{i-1}, a histogram, and a normal probability plot.

4. SCATTERPLOTS

Every package surveyed does some form of scatterplot. These plots vary less between packages than do histograms, but there are important differences. A few major features are shown in Figure 7. The ability to control size is useful both to fit output on narrow terminals (or make use of wide printers) and to make plots suitable for inclusion in reports. The control of scale is, of course, useful to make the plots "pretty", but more importantly, it allows doing plots of different groups or different variables on the same scale.

All the packages have some form of control over the "window" -- the range of data to be included in the plot. All packages give some indication of count (e.g., 1,2,...,9,+) if more than one observation falls on the same printing position. (Packages which plot on devices such as a Calcomp plotter or a Tektronix scope had difficulty with this problem. One developer noted that the test problem, which had 14 replications of one observation, caused his pen plotter to tear a hole in the paper.)

A test problem to determine how each package handles missing data on the plots was unfortunately not included on the questionnaire. Based on reading the manuals, BMDP appears to be the only package which attempts to show the y values for the missing x (and vice versa), by putting a symbol in the axis. About half of the packages have the ability to distinguish several groups by using different plotting symbols, and about half can plot more than one pair of variables on the same plot. All the packages make residuals from some analyses such as regression available for plotting (but in some packages this can be difficult -- for example, in SPSS, appropriate job control language must be used to "punch" the residuals on a disk file, then they must be read back in for plotting).

The scaling of the axes in the default plot (see Figure 8) fell into 3 main groups. Three packages, Omnitab II, P-STAT, and SPSS, did no rounding of the scales. The minimum x value is put on the left of the plot, the maximum on the right, and the rest proportionally in between. Four packages, BMDP, Genstat, Minitab, and Speakeasy round the scales, and put labels on every fifth or every tenth space. This procedure makes it easy to see how much is represented by each printer space. BMDP is particularly successful in finding a scale which is reasonably rounded yet lets the data fill most of the space available. Note that Genstat allows data to be plotted in the axis, where it is effectively hidden. Three packages, Data-Text, TROLL, and SAS, vary the number of spaces between the labelled tick marks. This allows quite elegant looking plots, at the expense of having the amount represented by one space difficult to read. In a class by itself, unfortunately, is OSIRIS, which leaves the decimal points out on the plot scales.

Some packages have interesting features available in their scatterplot routines. One test problem involved plotting y versus x and the regression line for predicting y for x, on the same axis. Genstat plotted the data with an asterisk (*), and plotted the line using an apostrophe (') if the line was in the top half of the printer position and a comma (,) if it was in the bottom, which effectively doubles vertical resolution. In addition, when the data and the line occupied the same printer position, the line was suppressed.

Omnitab II has a FOURPLOTS command, which puts 4 small plots all on one page of computer output.

Contour plots and 3-dimensional plots are produced by BMDP, Genstat, Minitab, Omnitab II, SAS, and Speakeasy, although in some cases, it is necessary to use package commands to compute the plotting symbol.

Graphics devices can be used by several packages. Omnitab II and Soupac can use Calcomp plotters. TROLL, TSAM, Statsystem, and Speakeasy can use a variety of devices including Calcomp plotters and Tektronix terminals. Speakeasy commands to produce a plot of volume versus diameter (of black cherry trees), and put the regression line on the plot, are shown in Figure 9.

5. OTHER GRAPHICAL DISPLAYS

Other graphical displays are available on some packages. Cyphergraph (a companion program to TSAM) produces bar-graphs and other displays with elaborate labelling, shading, and even color, suitable for including directly in business reports, see Chamberlain (1975). TROLL has capabilities for displaying multivariate data as stars or faces, see Chernoff (1973) and Friedman (1972). TROLL also has a capability for rotating and masking data, patterned after the PRIM-9 system (1973). BMDP1M prints a correlation matrix compactly by printing only the first digit of the (absolute value of) the correlation. It also prints a "shaded" correlation matrix, where the choice of symbol and overstrikes are used to show high correlations as dark areas.

6. REFERENCES

CHAMBERLAIN, Richard. (1975). Computer graphics and time series analysis. Proceedings of Computer Science and Statistics: 8th Ann. Symposium on the Interface. pp. 20-26.

CHERNOFF, Herman. (1973). Use of faces to represent points in n-dimensional space graphically. JASA, 68, 361-368.

FRIEDMAN, H. P., et al. (1972). A graphic way of describing changing multivariate patterns. Proceedings of Computer Science and Statistics: 6th Ann. Symposium on the Interface. pp. 56-59.

PRIM-9. (1973). Film produced by Stanford Linear Accelerator Center, Stanford, California Bin 88 Productions.

	Mode	Devices	Question	Penn State has Package	Histogram	Main Purpose	Distributor
Data-Text	B	L	1/2		no	Soc. Sci.	D. Armor, RAND Corp
OSIRIS	B	L	yes		no	Soc. Sci.	Univ. of Michigan
P-STAT	B-I	L	yes		yes	Soc. Sci.	R. Buhler, Princeton University
SPSS	B	L	yes	yes	1/2	Soc. Sci.	SPSS, Inc., Chicag Illinois
BMDP	B	L	yes	yes	yes	Stat	HSCF, UCLA
Genstat	B	L	yes		yes	Stat	Cornell University Ithaca, N. Y.
Minitab	I-B	L	yes	yes	yes	Stat	T. Ryan, Penn Stat University
Omnitab II	B-I	LP	yes	yes	yes	Stat	National Bureau of Standards, D.C.
SAS	B	L	yes	yes	no	Stat	SAS Inst., Raleigh N.C.
Soupac	B	LP	No		No	Stat	Univ. of Illinois
STATSYSTEM	I	LPS	1/2		yes	Stat	G.E., Rockville, M
Speakeasy	I	LPS	yes		yes	Math,Stat	Argonne National Lab., Illinois
TROLL	I	LPS	1/2		yes	Econ	NBER, Cambridge, Massachusetts
TSAM	I	LPS	no		no	Econ	Cyphernetics, Ann Arbor, Michigan

Notes: Mode: B=primarily batch; I=primarily interactive
Devices: L=line printer or typewriter; P=plotter; S=scope
Question: yes=questionnaire sent to developer and answers received

Figure 1. Packages Surveyed

```
MIDDLE OF    NUMBER OF
INTERVAL     OBSERVATIONS
  -10.         1 *
   -9.         0
   -8.         0
   -7.         0
   -6.         0
   -5.         0
   -4.         0
   -3.         0
   -2.         1 *
   -1.        13 *************
    0.        23 ***********************
    1.         9 *********
    2.         2 **
    3.         0
    4.         0
    5.         0
    6.         0
    7.         0
    8.         0
    9.         0
   10.         1 *
```

Figure 2. Minitab Histogram

```
CODE
      I
-10.  ** (      1)
      I
      I
      I
 -2.  ** (      1)
      I
      I
      I
 -1.  ************* (     13)
      I
      I
      I
  0.  ************************ (     23)
      I
      I
      I
  1.  ********** (      9)
      I
      I
      I
  2.  *** (      2)
      I
      I
      I
 10.  ** (      1)
      I
      I
      I.........I.........I.........I..
      0         10         20         30
      FREQUENCY
```

Figure 3. SPSS Histogram

```
          * 1.0000      * 2.0000
          ............+............+
 MIDPOINTS
    6.400)
    6.000)
    5.600) *              **
    5.200) *              ***
    4.800) *              ****
    4.400) *              *******
    4.000) **             M***
    3.600) ****           ******
    3.200) M*****         ***
    2.800) ***
    2.400) *****
    2.000) **
    1.600)                *
    1.200) *
    0.800)
    0.400) *
   -0.000)
   -0.400)
```

Figure 4. BMDP7D Side-by-Side Histogram

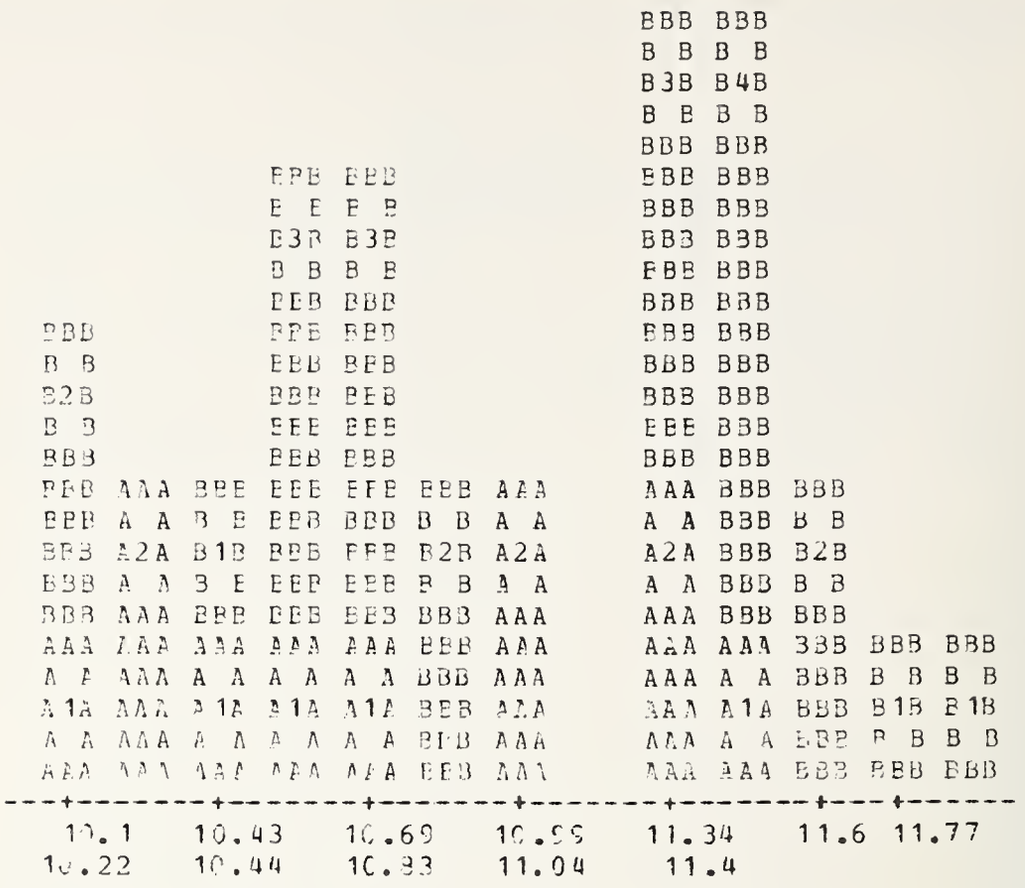


Figure 5. P-STAT Histogram of Two Groups

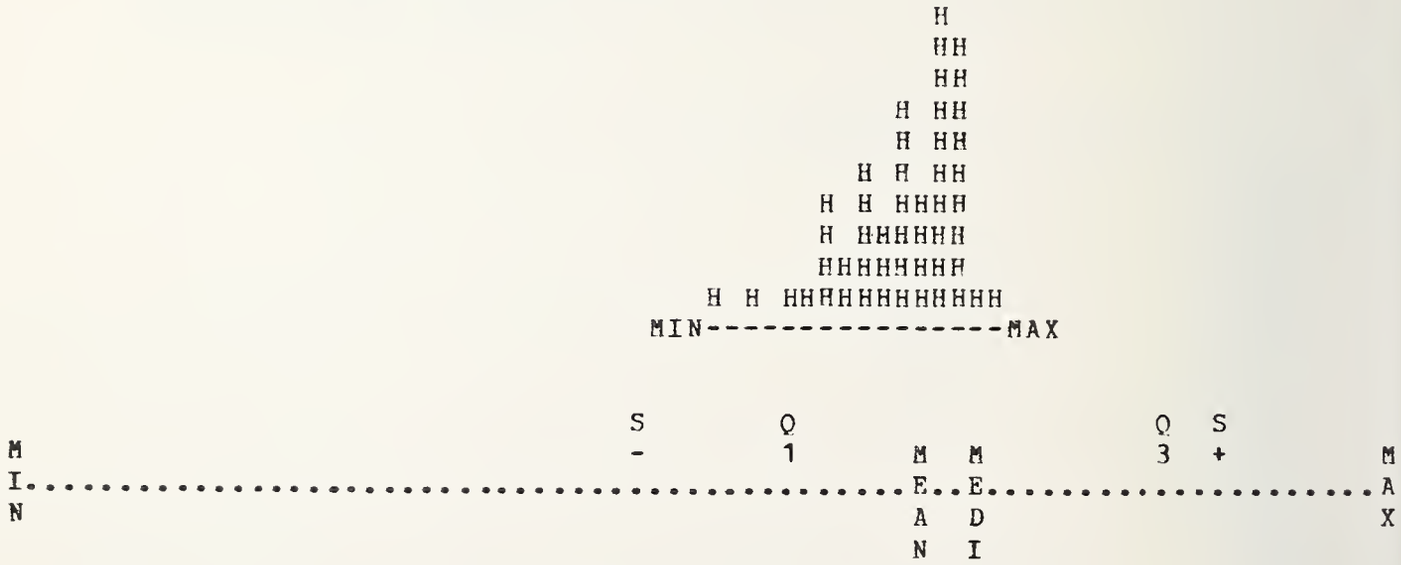


Figure 6. BMDP2D Univariate Data Display

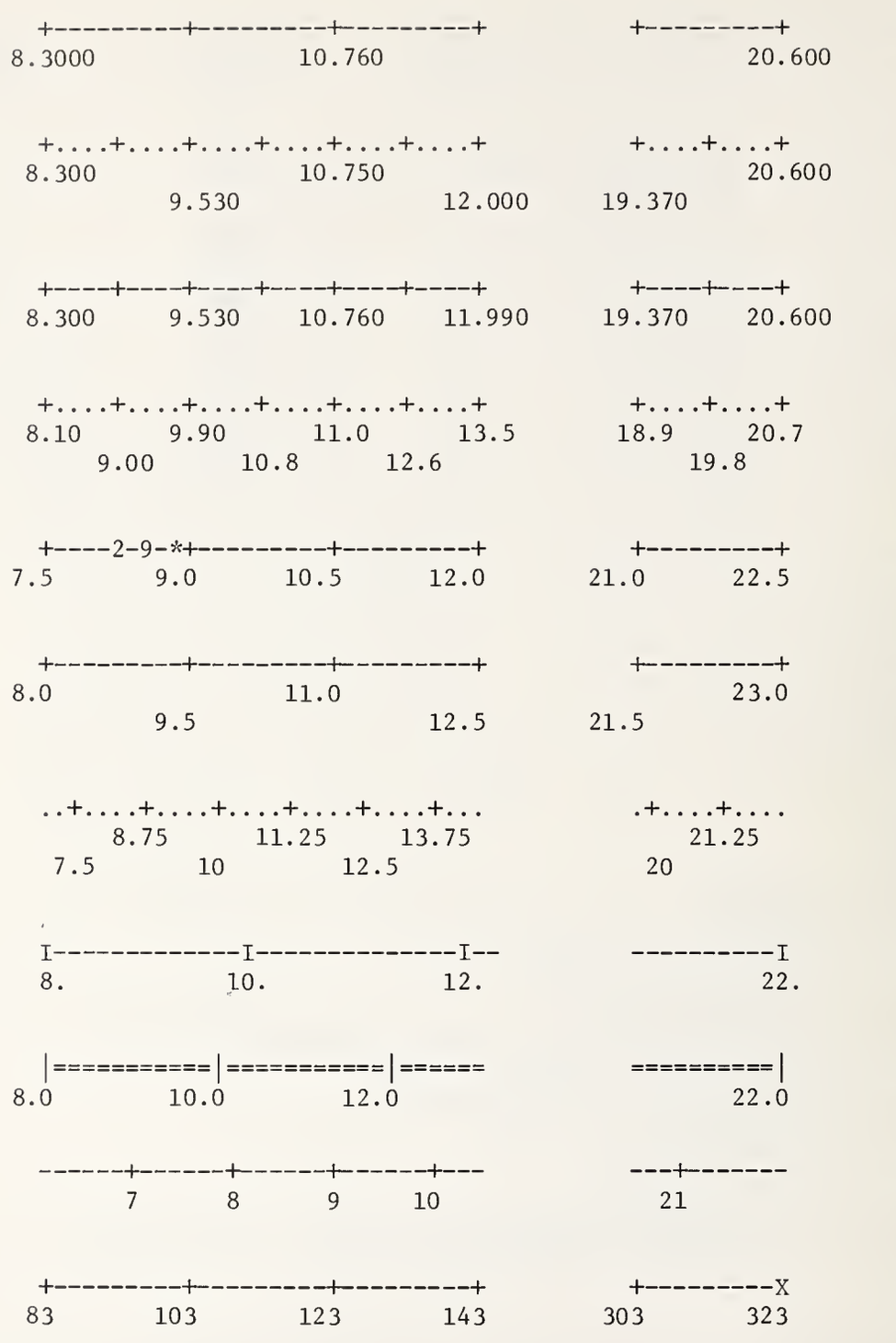
	Control Size	Control Scale	Several Groups	Multiple Plot
Data-Text	yes	yes	yes	no
OSIRIS	no	no	no	no
P-STAT	horz.	no	no	no
SPSS	no	yes	no	no
BMDP	yes	yes	yes	no
Genstat	yes	yes	yes	yes
Minitab	yes	yes	yes	yes
Omnitab II	yes	yes	yes	yes
SAS	yes	yes	yes	yes
Soupac	no	yes	no	no
STATSYSTEM	no	yes	no	yes
Speakeasy	yes	yes	yes	yes
TROLL	yes	yes	1/2	1/2
TSAM	yes	yes	?	yes

Figure 7. Plot Features

```
 1  PROGRAM
 2  LINECODE=-5
 3  COEF=MULTIREG(DIAMETER,VOLUME)
 4  SETXLABEL("DIAMETER")
 5  SETYLABEL("VOLUME")
 6  SETTITLE("VOLUME VS. DIAMETER WITH REGRESSION LINE")
 7  GRAPH(VOLUME:DIAMETER)
 8  PREDVOL=COEF(1)+COEF(2)*DIAMETER
 9  LINECODE=1
10  ADDGRAPH(PREDVOL:DIAMETER)
11  HARDCOPY
```

Note: A linecode of -5 means plot with a *, and 1 means plot with a continuous solid line.

Figure 9. Speakeasy Input for a Plotting Device



Test problem: Get default plot of Y versus X, where X is continuous from 8.30 to 20.60. Scaling on the X (horizontal) axis is shown.

Figure 8. Default Plotting

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

PORTABLE GRAPHICS

James E. George
Los Alamos Scientific Laboratory
P.O. Box 1663 - M.S. 272
Los Alamos, NM 87545

ABSTRACT

The portability of graphic software is discussed for the past, present and future. Early methods of portability are examined and contrasted with modern methods; the effects of the current graphic standards effort are surveyed. Representative portable graphic systems are discussed and example applications are utilized for illustration.

Key words: Graphics, portability, device independence, graphic software.

1. INTRODUCTION

Current graphical software practices are extremely diverse with little common ground. ıny installations have implemented their own packages which may or may not be device ıdependent and may or may not be portable. This may be adequate for a user who never ɔves, an environment which is static and a user who does not share his work with others or ry to import external software with graphics. As pointed out by Walsh(1972), this ɔndition has a high cost in manpower, money and time; he recommended that a portable, evice independent package be designed to solve this problem.

Device independence is a measure of the ease with which a new device can be utilized by program. For graphical applications, device independence is crucial since various forms f output are usually desired (e.g. film, high precision plotting, terminal). Additionally, evice independence allows an organization the freedom of competitive procurement of raphical devices.

Portability is a measure of the ease with which a program can be transferred from one ıvironment to another (Poole and Waite, 1973). Portability is important within an ganization as the computer environment changes (i.e. new computers are acquired). It can lso reduce retraining and allow economic mobility of programmers.

2. PORTABILITY TECHNIQUES

Various techniques to achieve portability have been discussed in the literature (Naur ıd Randell, 1969; Buxton and Randell, 1970; Brown, 1977a & b; Griswold, 1977; Griffiths, 977; Brown, 1970; Waite, 1970; Waite, 1973; Los Alamos, 1976; Aird, Battiste and Gregory, 977). Some of the approaches are:

Make use of a widely available high level language (e.g. Fortran, Cobol, Basic);

Make use of a verifier for a subset of a language which is considered safe (e.g. PFORT ; Ryder, 1974);

Utilize an abstract machine model or intermediate language.

ırther, portable software must be thoroughly tested by portable tests to assure that the ıchine independent constants have been properly initialized and that the assumed system ıcilities are provided correctly.

For graphic software, these must be extended to provide an identical user interface for ıe graphic facilities. This could be accomplished by extending the languages, transporting ıe graphic facilities or by adopting a set of graphic facilities as "standard". To date, ɔne of these have been very successful.

In the long term, the definition and acceptance of a graphic standard and its ıplementation for various languages is probably the best solution and has been initiated by CM/SIGGRAPH (Standards, 1977). These standards need be feared only if they are ill ɔnsidered or imposed (Ross, 1976).

3. DEVICE INDEPENDENT TECHNIQUES

Device independence is supported by several graphical software packages (e.g. Gino, 1976; Disspla, 1970a & b; NCAR, 1977; Caruthers, van den Bos and van Dam, 1977; GCS, 1974; Wright 1975a & b, 1977). All of these packages utilize an abstract machine model or intermediate file (i.e. intermediate language) to provide device independence which can be either low or high level.

The advantages of a low level file are that it is easy to support new devices, it is easy to learn how to implement these device drivers and memory space is minimal for multiple devices. The disadvantages are that devices with advanced features may be inefficiently utilized and transmission bandwidth may be wastefully used.

The advantages of a high level file is that advanced devices can be efficiently utilized and transmission bandwidth more effectively used. The disadvantages are that new devices are more difficult to support, it is more difficult to learn to implement the drivers and more memory may be used for multiple devices. A non-deterministic implementation of these drivers can remove all of their disadvantages (Gino, 1976).

The design of the "right" level for an environment and its application is non-trival, but the picture processing pipeline approach of the proposed standard (Standards 1977) may ease this problem.

4. EXAMPLE GRAPHIC SOFTWARE

Currently, graphic software is available from either hardware vendors or software vendors (an excellent in depth review of several packages is given in Standards(1977); additional graphical tools are discussed in Phillips(1976)). The software available from hardware vendors is typically portable but will only support that vendor's hardware; it should not be generally available to users, but can be effectively used in implementing device drivers.

The packages available from software vendors are usually widely available with radically different portability costs and are generally device independent. Before selecting a particular package, its portability to all of an environment's machines (present and future) should be examined; with today's technology, 8 and 16 bit computers are widely available and many of the popular packages are unavailable on these small computers.

5. Desirable Features

There are many desirable features in addition to portability and device independence and these are closely related to a particular environment's needs. Some minimal features are:

Portability

Device independence

2-D graphic primitives and text

Window

Viewport

Data graphing capabilities

Selectable character quality and fonts

Input facilities

Variable line texture

Automatic scale generation

Figures 1 thru 4 are samples illustrating many of these minimal features.

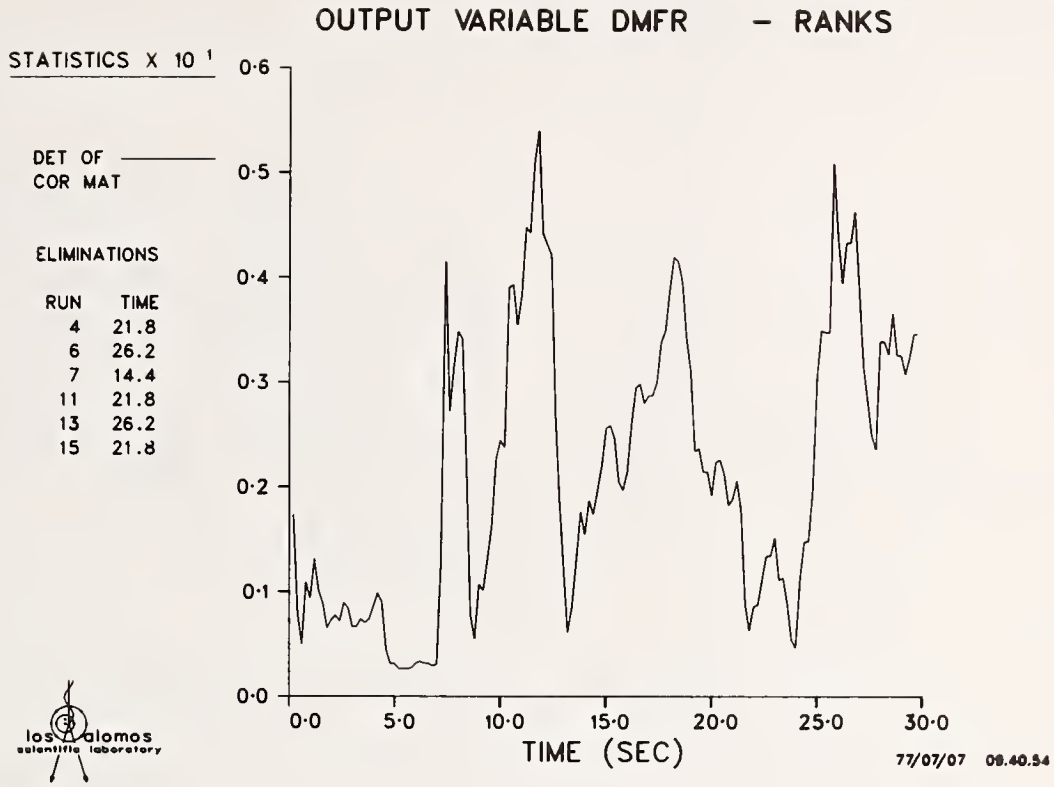


Figure 1

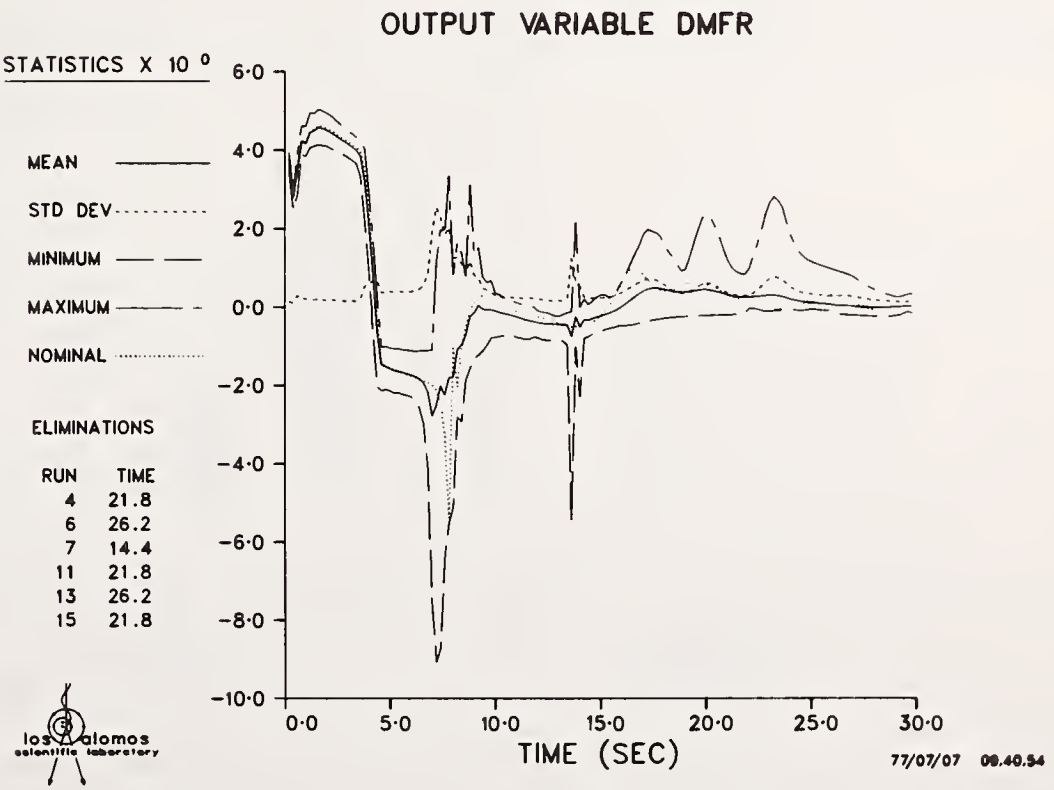


Figure 2

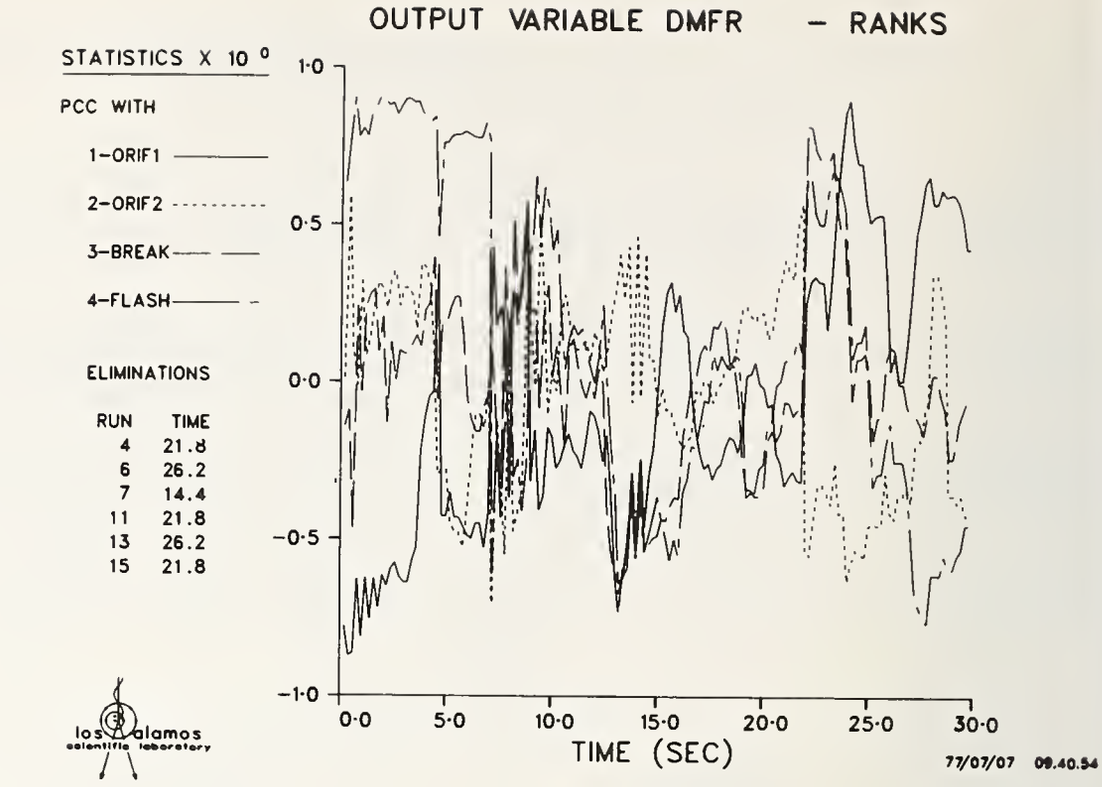


Figure 3

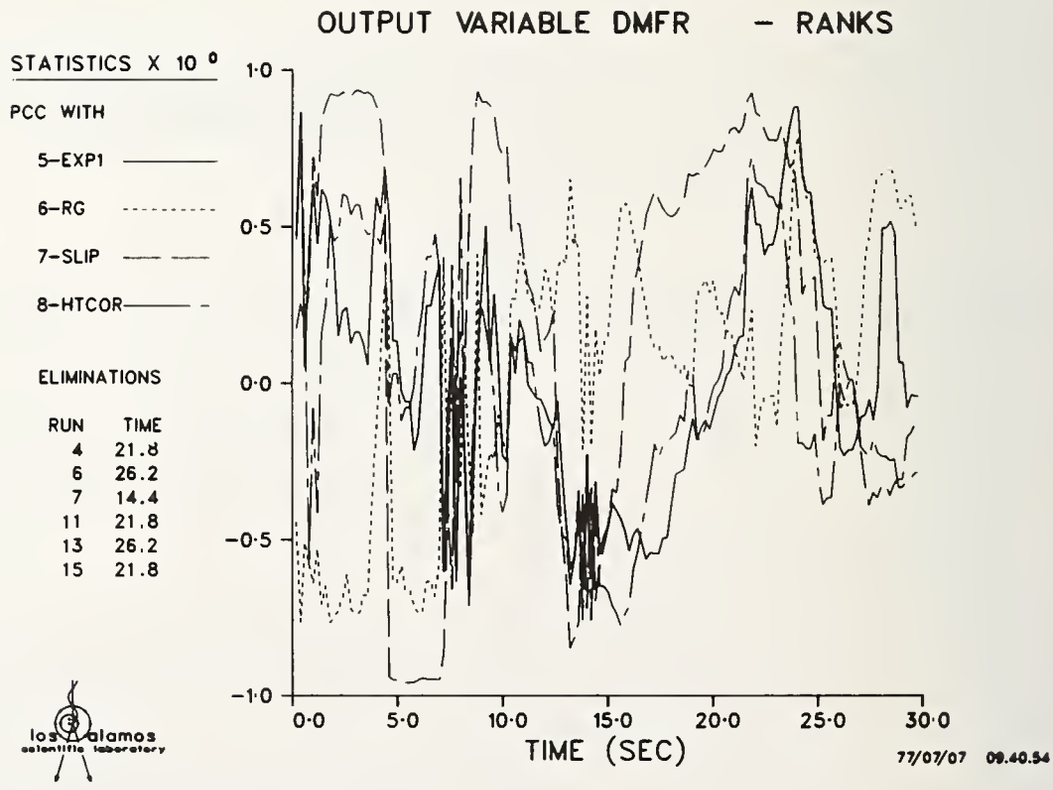


Figure 4

(All figures provided by M. McKay, LASL group Q-12)

Psychological theory suggests that five items may be distinguished (coded) with line types (Foley and Wallace, 1974; Martin, 1973), but the previous figures provide a convenient counter example. Color is a more effective coding method and is becoming widely used. It is easy to distinguish the above data with color; examples may be obtained by writing the author. Further additional features might be:

Color

3-D graphic primitives and text

Viewing transformations

Interaction

Picture segmentation

Surfaces

Curve fitting

These features may be used for many advanced applications including computer generated movies. The following films may be borrowed at no cost from:

Report Library
Los Alamos Scientific Library
P. O. Box 1663 - MS 364
Los Alamos, NM 87545

Y-306 Thermal Analysis in Mold Design

Y-285 Matrices and Their Singular Values

Y-281 Computer Movies: Aid to Energy Research

Y-277 Interactive Graphics at LASL

6. CONCLUSIONS

With modern software technology, portable device independent software should be provided in all environments; there is no excuse for any less. Further, the package should be carefully chosen to insure the minimal features and as many additional features consistent with each particular environment.

7. REFERENCES

AIRD, T. J., BATTISTE, E. L. AND GREGORY, W. C. (1977). Portability of Mathematical Software Coded in Fortran, ACM Transactions on Mathematical Software, 3, 2.

BROWN, P. J. (1977a). Software Portability , Cambridge University Press, Editor.

BROWN, P. J. (1977b). Basic Implementation Concepts, in Software Portability , Cambridge University Press.

BROWN, W. S. (1970). Software Portability, in Software Engineering Techniques , Nato Science Committee, Buxton, J. N. and Randell, B., Ed.

BUXTON, J. N. and RANDELL, B. (1970). Software Engineering Techniques, Nato Science Committee, Editors.

CARUTHERS, L. C., VAN DEN BOS, J. and VAN DAM, A. (1977). GPGS - A Device-Independent General Purpose Graphic System for Stand-alone and Satellite Graphics, Computer Graphics, 11, 2.

DISSPLA (1970a). Disspla Beginners/Intermediate Manual, Integrated Software Systems Corp., San Diego, CA.

DISSPLA (1970b). Disspla Advanced Manual, Integrated Software Systems Corp., San Diego, CA.

FOLEY, J. D. and WALLACE, V. L. (1974). The Art of Natural Graphic Man-Machine Conversation, Proceedings of the IEEE, 62, 4.

GINO (1976). GINO-F User Manual, Computer Aided Design Centre, Cambridge, England.

GRIFFITHS, M. (1977). Verifiers and Filters, in Software Portability , Brown, P. J. Ed. Cambridge University Press.

GRISWOLD, RALPH E. (1977). Engineering for Portability, in Software Portability , Brown, P J. Ed., Cambridge University Press.

LOS ALAMOS (1976). Working Papers from the Los Alamos Workshop on Software Portability.

MARTIN, JAMES (1973). Design of Man-Computer Dialogues , Prentice-Hall.

NAUR, PETER and RANDELL, BRIAN (1969). Software Engineering, Nato Science Committee Editors.

NCAR (1977). NCAR Graphics Software, National Center for Atmospheric Research, Boulder, CO

PHILLIPS, R. L. (1976). Software Tools for Computer Graphics, Computer Science an Scientific Computing , Academic Press.

POOLE, P. C. and WAITE, W. M. (1973). Portability and Adaptability, Advanced Course o Software Engineering, F. L. Bauer, Ed., Springer-Verlag.

ROSS, DOUGLAS T. (1976). Homilies for Humble Standards, Communications of the ACM, 19, 11.

RYDER, B. G. (1974). THE PFORT Verifier, Software Practice and Experience, 4.

STANDARDS (1977). Status Report of the Graphic Standards Planning Committee o ACM/SIGGRAPH, Computer Graphics, 11, 3.

WAITE, W. M. (1970). The Mobile Programming System: STAGE2, Communications of the ACM, 13 7.

WAITE, WILLIAM M. (1973). Implementing Software for Non-Numeric Applications , Prentice Hall.

WALSH, JOHN P. (1972). A Plea for Standards and Graphics vs. Freedom, Computer Graphics, 6 3.

WRIGHT, THOMAS (1975a). A Schizophrenic System Plot Package, Computer Graphics, 9, 1.

WRIGHT, THOMAS (1975b). Practical Computer Graphics for Scientific Users : Philosophy an Implementation, Computers & Graphics, 1, 2/3.

WRIGHT, THOMAS (1977). Machine-independent Metacode Translation, Computer Graphics, 11, 2.

8. BIOGRAPHY

James E. George received a BSEE from the University of Oklahoma in 1959, a MSEE fro New York University in 1961 and a Ph.D. in computer science from Stanford University i 1971. From 1959 until 1966, he was a Member of the Technical Staff at Bell Telephone Laboratories, and; from 1971 until 1974, he was an Assistant Professor of Computer Science at Colorado State University. In 1974, he he became a Staff Member at Los Alamos Scientific Laboratory where he is the Alternate Group Leader of the Computer Graphics Group in the Computer Sciences and Service Division. He has been active in the Association for Computing Machinery and is the current Chairman of ACM's Special Interest Group on Computer Graphics.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

TERMINAL AND COMPUTER INDEPENDENCE FOR INTERACTIVE GRAPHICS APPLICATIONS SOFTWARE

H. G. Bown, C. D. O'Brien, G. Thorgeirson and W. Sawchuk
Communications Research Centre, Ottawa, Canada

ABSTRACT

This paper describes an approach to provide terminal and computer independence for interactive graphics application software. The major goals of the software system are to achieve a high degree of environment independence through software portability and the concept of a virtual display terminal, and to simplify the writing of interactive graphics programs.

An overview of the programming language, IGPL is presented together with a description of the virtual terminal software. The commands (Graphical Task Interactions, GTI's) that are communicated between host and terminal are described where their intent is to separate the application-dependent and system-dependent functions.

Key words: Application; computer; displays; graphics; independence; interactive; language; portability; software; standards; terminal.

1. INTRODUCTION

This paper discusses methods to achieve a high degree of environment independence in the design of a general-purpose, interactive computer graphics, software system. Environment independence is achieved when the hardware and software implementation details are made invisible to the application programmer so that any new advances in the evolving computer graphics technology can be accommodated without adversely affecting the application programs.

Recently, there has been considerable activity related to the development of computer graphic standards (Status Report of the Graphic Standards Planning Committee of ACM/SIGGRAPH, Fall 1977). The emphasis in the above document is the definition of a core graphic system that will present a common set of function routines to the application programmer. Another equally important aspect of environment independence that will be presented in this paper relates to the independence of the programming system from the display terminal hardware being utilized.

A high level graphic programming language, IGPL (Interactive Graphical Programming Language) developed at the Communications Research Centre (see O'Brien and Bown, 1975a) and now marketed by Norpak Ltd (see Norton, 1976) is presented. This language simplifies the writing of interactive programs and offers a high degree of portability and device input/output independence.

2. VIRTUAL DISPLAY TERMINAL CONCEPT

A major requirement for environment independence can be achieved by separating the software system from the display terminal hardware. Figures 1 and 2 illustrate the dividing line between user application software and systems software. A separation between these two functions can be realized by defining a virtual display terminal with specific capabilities. All communications with this virtual display terminal are made in such a manner as to be independent of any particular realization of the virtual terminal. For example, an application program is unaware of the technique being employed in the virtual display terminal when it requests that a line, character or symbol be generated. In addition, the application program is unaware of whether a random access refresh, a raster refresh or a storage display is being utilized as the display medium. Different virtual

display terminals may perform the functions of vector and character generation in a totally different manner; one may use hardware techniques, whereas another may utilize software programs to perform the same function.

A set of instructions is provided to enable the virtual terminal to be referenced in this hardware-independent manner. These commands, GTI's (Graphical Task Instructions) presented in Table 1, are subdivided into the different categories as shown below:

1. System Initialization and Definition
2. Display Generation
3. Graphical Modifiers
4. Display File Modifiers
5. Interactive Device Control
6. Terminal Generated (Return).

These instructions are defined to be independent of any particular coding scheme and can be communicated over serial or parallel lines between the processors of a dual-processor computer graphics system. In addition, the GTI instructions form an extensible set allowing for future expansion to accommodate new hardware and software innovations.

Figure 1 presents a conceptualization of a graphics system where the intent is to separate the application-dependent and system-dependent functions. The definition of a virtual display terminal permits the separation of these functions into independent processes. These independent processes may be implemented in a single-processor system but with the rapid increase in the capabilities of micro-computers the separation of functions suggests a dual-processor system design as presented in Figure 2. One processor (usually a micro-computer) is dedicated to the task of providing the virtual terminal capability and the other processor, a mini-computer or large frame system, is responsible for the application program execution. The dual-processor system has the added advantage of providing faster performance because the display housekeeping and I/O device handling are now performed by the second processor. Also, this concept will promote the development of microprocessor virtual display terminals that can be treated in the same way we now consider ASCII alphanumeric teletype-like devices.

3. IGPL (Interactive Graphic Programming Language)

The IGPL language provides the following facilities to permit ease of interactive graphical programming:

a) graphical input response facilities
b) graphical drawing facilities
c) structured programming constructs
d) data manipulation facilities
e) easy access to external software.

These facilities are provided in such a manner as to be independent of hardware input/output devices by using the virtual device concept (see O'Brien and Bown, 1975b).

IGPL has been designed to provide a language in which a relatively untrained person can write interactive application programs. The language is designed for "application programmers", that is, persons who have some knowledge of programming but have most of their expertise in the field of their application. The language provides a basic set of graphical drawing commands and augments these with a powerful set of display modifiers.

The IGPL language is block structured and provides a very powerful display procedure capability. The example program shown in Figure 3 illustrates the block program structure and language syntax of IGPL. The translator for IGPL is written using the macro-processor, STAGE2 (see Waite, 1973), thus permitting portability of application programs. The intermediate code produced by the translator is standard ANSI Fortran, thus further enhancing portability and utilization of existing software packages.

4. CONCLUSION

The interactive graphics software system described in this paper has been in use at the Communications Research Centre and at a number of other locations for the past two years. A large number of different display devices are being supported including both point-to-point random displays and colour raster display equipment.

This approach of achieving environment independence suggests that the translator for application programs emit a standard GTI code, which provides a means of addressing an interactive graphic terminal, similar in concept to the ASCII code being used to address most alphanumeric terminals. Terminal and computer independence for interactive graphics applications software can be achieved by:

1) the use of the virtual terminal concept utilizing the GTI instructions presented in Table 1,
2) application program portability made possible throught a portable translator writing system or a standardized base language.

Application programs could then be moved from machine to machine and could take advantage of future advances in terminal displays technology without the need for expensive and time-consuming reprogramming.

5. REFERENCES

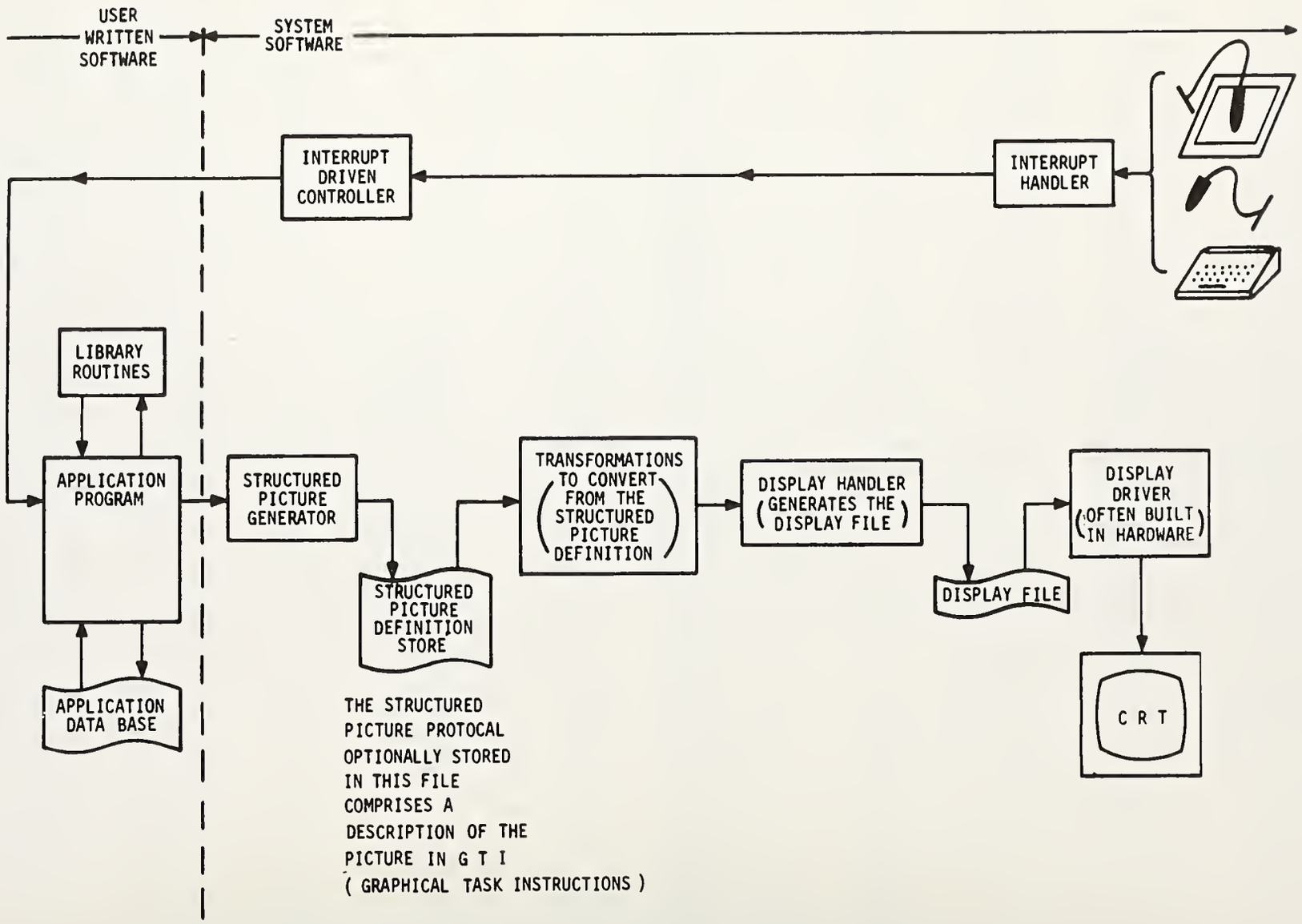
NORTON, J. IGPL Users Programming Manual, Norpak Ltd, Pakenham, Ontario, Canada, KOA 2XO.

O'BRIEN, C. D., BOWN, H. G. (1975a). Image, a language for the interactive manipulation of a graphic environment, 2nd Annual Conference on Computer Graphics and Interactive Techniques, ACM/SIGGRAPH 75, Bowling Green, Ohio, U.S.A

O'BRIEN, C. D., BOWN, H. G. (1975b). A device independent input structure for a high level graphics language, Proceedings of the 4th Man-Computer Communication Conference, National Research Council of Canada, Ottawa, Canada, May 1975.

Status report of the graphics standards planning committee of ACM/SIGGRAPH, Computer Graphics, Vol. 11, No. 3, Fall 1977.

WAITE, W. M., Implementing Software for Non-Numeric Applications, Prentice-Hall, Inc., Englewood Cliffs, N. J., U. S. A., 1973.



CONCEPTUALIZATION OF A GRAPHICS SYSTEM

ARROWS INDICATE INFORMATION FLOW

Figure 1 Comceptualization of a Graphics System

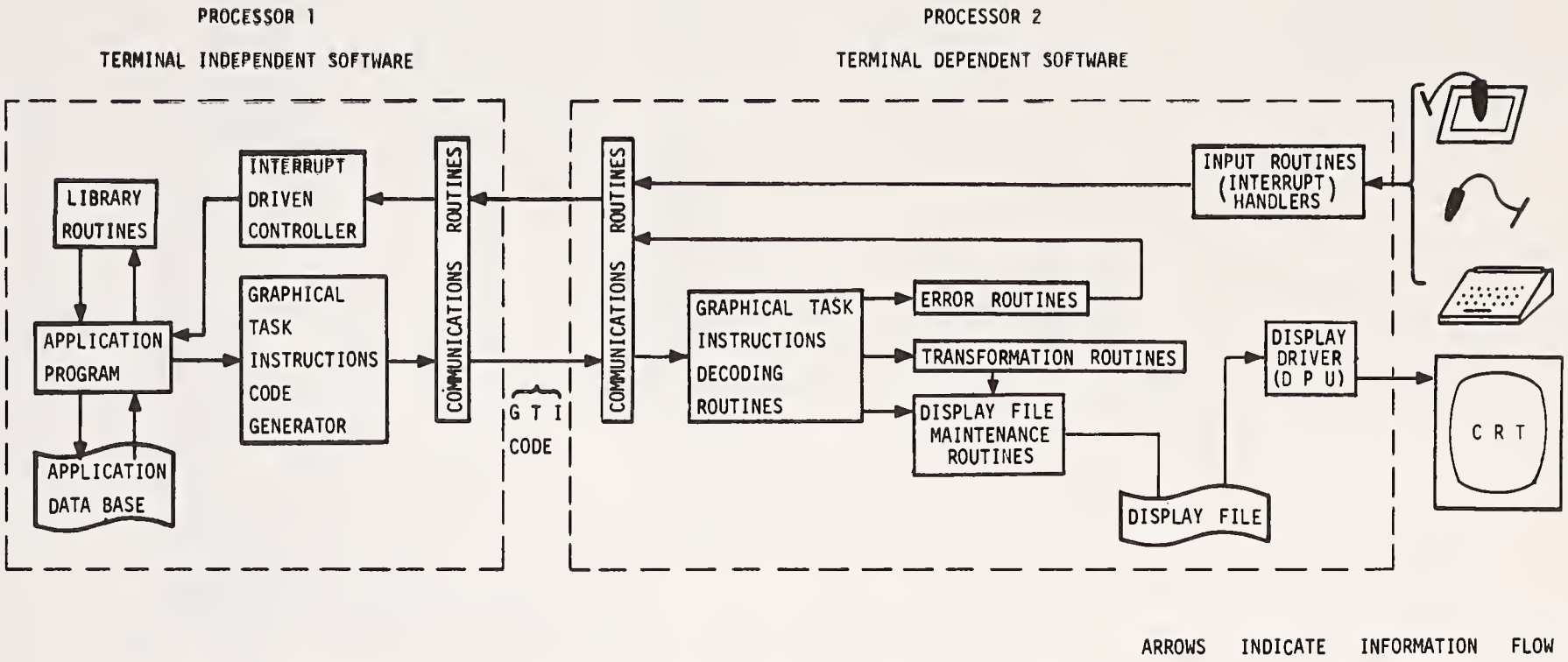
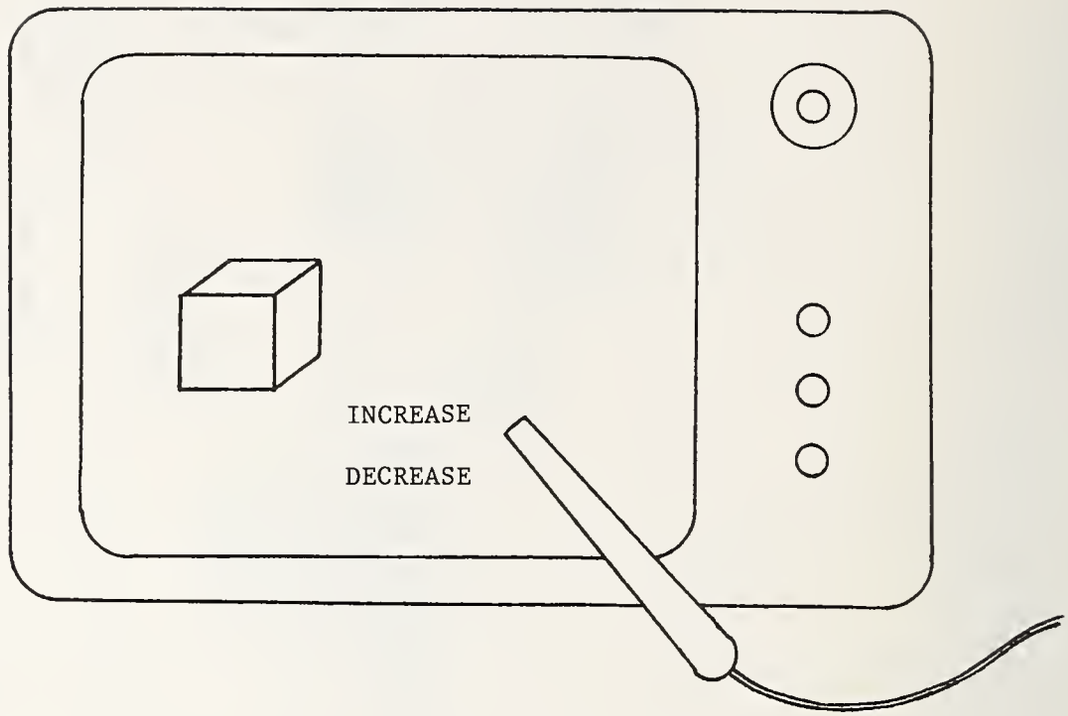


Figure 2 A Dual-Processor Graphics System




```
*
*   EXAMPLE IGPL PROGRAM
*
*       THIS PROGRAM DRAWS A CUBE ON THE SCREEN AND ALLOWS
*       THE USER TO INCREASE OR DECREASE THE SIZE OF THE CUBE
*
REAL SIZE
ENTRY
  LET SIZE = 1.                      ** INITIALIZE CUBE SIZE
OBJECT
  TEXT 'INCREASE';AT(700,450)        ** LIGHT BUTTON "INCREASE"
ACTION
  LET SIZE=SIZE*1.2                  ** INCREASE THE CUBE SIZE
  DISPLAY CUBE                       ** REDISPLAY THE CUBE
OBJECT
  TEXT 'DECREASE';AT(700,250)        ** LIGHT BUTTON "DECREASE"
ACTION
  LET SIZE=SIZE/1.2                  ** DECREASE THE CUBE SIZE
  DISPLAY CUBE                       ** REDISPLAY THE CUBE
OBJECT CUBE                          ** DEFINE THE CUBE
  LINES 0,80/-80,0/40,20/80,0/-40,-20;SCALE(SIZE);AT(375,450)
  LINES -80,0/0,80;SCALE(SIZE);AT(375,450)
  LINES 40,20/0,80;SCALE(SIZE);AT(375,450)
END
```

Figure 3 Example IGPL Program

Table 1 GRAPHICAL TASK INSTRUCTIONS (GTIs)

GTI SPECIFICATION	GTI STRUCTURE	REMARKS
System Initialization and Definition		
INITIALIZE	opcode	
I Display Generation		
SET BEAM POSITION	opcode x y	set beam to point x,y
POINT	opcode	intensified point
CHARACTER STRING	opcode N char 1 ⋮ char N	sequence of N characters
LINE TO	opcode x y	line to a point x,y
LINES	opcode N Δx_1 Δy_1 ⋮ Δx_N Δy_N	concatenated lines with x and y axis displacements
LINES THROUGH	opcode N x_1 y_1 ⋮ x_N y_N	concatenated lines through a sequence of points
AREA	opcode Δx Δy	rectangular area of sides $\Delta x, \Delta y$
ARC	opcode x_s y_s x_e y_e x_c y_c direction	circular arc with starting (x_s,y_s), ending (x_e,y_e), and centre (x_c,y_c), and direction
SYMBOL	opcode Symbol No.	symbol of number N
PLOT	opcode	hardcopy output

Table 1 GRAPHICAL TASK INSTRUCTIONS (GTIs) (continued)

	GTI SPECIFICATION	GTI STRUCTURE	REMARKS
III	Graphical Modifiers		
	PAGE	opcode x_{min} x_{max} y_{min} y_{max}	user co-ordinates system of rectangle x_{min}, y_{min} and x_{max}, y_{max}
	SCALE	opcode factor	scaling function
	TRANSLATE	opcode Δx Δy	translating function
	ROTATE	opcode Θ	rotating function
	REFLECT	opcode Θ	reflection function
	WINDOW	opcode x_{min} x_{max} y_{min} y_{max}	window function (rectangular)
	WITHIN	opcode x_{min} x_{max} y_{min} y_{max}	within (viewport) function (rectangular)
	INTENSITY	opcode N	colour of shade of colour N ($0 \leq N \leq 1.0$)
	COLOUR	opcode N	colour of value N
	LINE TEXTURE	opcode N	line of type N
	FLASH ON	opcode	enable flashing
	FLASH OFF	opcode	disable flashing
	CHARACTER SET SPECIFICATION	opcode N	character set no. N
	SYMBOL SET SPECIFICATION	opcode N	symbol set no. N
	END MODIFIER	opcode	modifier removal function

Table 1 GRAPHICAL TASK INSTRUCTIONS (GTIs) (continued)

GTI SPECIFICATION	GTI STRUCTURE	REMARKS
Display File Modifier		
TAG	opcode	display file segmentation
BEGIN INSERT	opcode	appending function
END INSERT	opcode	termination of insertion
ERASE	opcode begin tag No. end tag No.	delete tagged objects
MODIFY	opcode begin tag No. end tag No. modify type modify value	modify named objects
CLEAR	opcode	clear the display file
DISPLAY ON	opcode	initiate the display
DISPLAY OFF	opcode	terminate displaying
WINK	opcode	generate a DISPLAY ON/OFF sequence
ENABLE SEEK ON OBJECT	opcode N value 1 : value N	selective seeking
Interactive Device Control		
DEVICE ON	opcode N device$_1$ sub-device$_1$: device$_N$ sub-device$_N$	enable N input devices
SET MARKER MODE	opcode code	constraint on marker's movement
SET MARKER POSITION	opcode x y	set the marker at location (x,y)
KEYBOARD ACTIVATION CHARACTERS	opcode N char$_1$: char$_N$	set the specified N characters to be activation characters

Table 1 GRAPHICAL TASK INSTRUCTIONS (GTIs) (continued)

GTI SPECIFICATION	GTI STRUCTURE	REMARKS
VI Terminal generated (return)		
IDENTIFIER INTERRUPT DATA RETURN	opcode tag No. x y	return tag and (x,y) position information
MARKER POSITION DATA RETURN	opcode x y	return the current marker position
CHARACTER STRING RETURN	opcode N $char_1$: $char_N$	return the character string
PUSHBUTTON DATA RETURN	opcode button No.	return the pushbutton code
ERROR REPORT	opcode N $code_1$: $code_N$	return the set of error codes

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

DIALOGUE CONSIDERATIONS IN INTERACTIVE STATISTICAL GRAPHICS

Jane F. Gentleman
University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

ABSTRACT

Careful human engineering of the dialogue between program and user in interactive statistical computer graphics is encouraged. Six principles are presented, based on experience in the development of such programs. Some of the principles are applicable to the development of computer software in general.

Key words: Dialogue; human factors engineering; interactive statistical computer graphics.

1. INTRODUCTION

The purpose of drawing a graph is to increase a person's ability to perceive patterns, through use of visual, rather than just numerical, representation of data or formulas. In a sense, then, it is the limitations of human perceptive ability that cause graphs to be needed at all. Thus, developers of graphics software should consider these limitations when engineering those portions of the software of which the user will be directly aware. This is especially important in software designed for interactive computing, as the confrontation of the user is immediate and quick. This paper discusses the importance of careful human engineering of the dialogue between program and user in interactive statistical computer graphics.

2. SIX PRINCIPLES

Imagine yourself sitting at a computer graphics terminal, using an interactive statistical graphics program. Or perhaps you are using an ordinary printing terminal to produce crude character plots. You are generating a sequence of plots. Somehow, before each plot, the program asks you what you want, and somehow, you tell the program what you want.

Six general principles are suggested below to govern the design of this sequence of questions and answers. The ideas are based on experience gained from developing the ST Interactive Statistical Plotting Package. (For discussion of the use of this package in data analysis and in teaching, see Gentleman (1976a) and Gentleman (1976b), respectively.) As work on these programs proceeded, it was found that an unexpectedly large amount of time was being spent in polishing the dialogue between user and program.

PRINCIPLE I. Program query sequences should evolve based on user feedback.

The developer should literally stand behind, looking over the shoulder of, the user, i possible without being identified as the developer. Confusing terminology and othe problems can then be identified. The users will submit numerous suggestions to th developer, who has to learn to sort them out and say no to the right ones.

One justification for saying no is that the suggested change is beyond the desire scope of the program. There are roughly three reasons for using interactive graphics:

(1) Exploration
(2) Education
(3) Publication.

By exploration is meant a free-and-easy approach to data analysis, in which the use "wanders" through his or her data, following up old ideas as well as new hunches suggeste in the process. The educational use of interactive graphics can be subdivided into tw areas: the use of interactive graphics by students doing assignments or for reinforcing concepts, and the use of it by a lecturer, either for teaching students or for giving a seminar. For publication in journals and reports, hard copy output from a graphics termina is often adequate; it is produced more quickly (once the program is written) and more accurately than by a draftsman.

Difficulty in defining the scope of an interactive graphics program occurs because th above three types of users sometimes have conflicting needs. In fact, one user can easil fall into all three categories - e.g. a professor who teaches, who analyzes data, and whos results are published. The developer must somehow decide on whom to please when conflict arise.

PRINCIPLE II. Seek a balance between detailed user control and speedy plotting.

The explorer of data wants the plot to appear quickly, without being bothered by rela tively trivial details such as axis titles and number of tic marks. Axis limits, however, are more important. Designed primarily for data exploration, the ST Package initially did not ask the user for axis limits, but determined them automatically from the data. This was found to be an insufficient amount of control, for reasons given in the discussion below of Principle V. The programs now ask

USER CONTROL OF AXIS LIMITS?

once, at the beginning of execution. A user who responds affirmatively to the above question is then asked separately about X-axis and Y-axis limits, e.g:

OF X-AXIS LIMITS? no
OF Y-AXIS LIMITS? yes

The user is then asked before each plot to provide the selected axis limits. (Automatic limits are still available if no limits are specified.) This minimizes the number of questions the user is asked before each plot.

The lecturer's audience may not need to see the questions and answers at all. Assuming that the sequence of plots is not completely predetermined (in which case, transparencies could just as well be used), the commonly used method of permitting multiple answers, separated by semi-colons, in anticipation of future questions can appreciably reduce the number of queries and speed up the generation of plots. For example, the above three questions can be answered all at once:

USER CONTROL OF AXIS LIMITS? y;n;y

(where "yes" and "no" are abbreviated as "y" and "n"). This device is favored by any type of user, once he or she is familiar with the query sequence.

Users generating plots for publication are more interested in control than in speed. They often know ahead of time exactly what plot is desired, and they are very concerned with details such as axis titles and tic marks. They are thus in conflict with those in a hurry to explore or present their data. Perhaps they should use different programs (although the effort required to learn to use two different programs may be considerable), and perhaps interactive graphics does not offer them significant advantages over batch computing.

Theoretically, all three types of users could be satisfied if the program contained any optional detailed controls over plotting and if these options could be ignored when speedy plots were wanted. But at the present state of computer technology, the program then tends to grow to the point that its core requirements slow down the response time so much that interaction is not worth the human waiting time.

PRINCIPLE III: Seek a balance between terseness and understandability.

If the question-and-answer sequence is too cryptic, the statistician will look upon it as a new computer language to learn, and may be sufficiently intimidated never to get started. There are important uses for graph-generating languages, but there is also a strong need for interactive programs that use English so that users do not have to sit at terminals with manuals on their laps. The ST package therefore uses ordinary English for its program queries and answers, although this sometimes may be slightly long-winded. Much effort has been expended in making the questions as short as possible while still keeping the meaning clear.

It is also best to avoid abbreviations and notation when possible, e.g. to mention the INDEPENDENT VARIABLE rather than the IND VAR or X.

PRINCIPLE IV: Some things are unesthetic (e.g. coding).

This principle is rather generally stated, but much of the decision making in program development is based on subjective and/or stylistic considerations. An embarrassing example can be found in an early version of one of the ST programs, as it was originally written by a student for a class project. The program draws a selected probability density function (p.d.f.) and optionally shades the tails. For instance, the user might want to see a normal(0,1) p.d.f. with the left tail shaded to the left of -1.96 and the right tail shaded so as to achieve a shaded area of .05. The query sequence was originally as follows:

```
TYPE DISTRIBUTION NAME: nor
MU, SIGMA-SQ: 0 1
IS SHADING DESIRED? y
THERE ARE 2 MODES OF SHADING UNDER PDF:
  1 SHADE SPECIFIED TAIL AREA
  2 SHADE TAIL OUTSIDE SPECIFIED LIMITS
  (CARRIAGE RETURN IMPLIES NO SHADING).
FOR LEFT TAIL, ENTER MODE OF SHADING, AREA (OR LIMIT): 2 -1.96
FOR RIGHT TAIL, ENTER MODE OF SHADING, AREA (OR LIMIT): 1 .05
```

There are too many instructions here, and the coding of possible answers as 1 and 2 is awkward and confusing. Today, the program uses the following query sequence:

```
TYPE DISTRIBUTION NAME: nor
MU, SIGMA-SQ: 0 1
SHADE LEFT TAIL? y
  UPPER LIMIT OF SHADING: -1.96
SHADE RIGHT TAIL? y
  LOWER LIMIT OF SHADING: (carriage return)
  AREA TO BE SHADED: .05
```

The carriage return means not applicable, not interested, or no, depending on the context.

This query sequence can be further improved: The word TYPE in the first query can be omitted, and MU, SIGMA-SQ would be better phrased as MEAN, VARIANCE. Because the query AREA TO BE SHADED is somewhat concealed, only appearing if the previous query is not answered, the program has been known to confuse a user who wanted to specify a tail area and not an abcissa value. But without providing a "menu" of choices, this type of tree structure of options may be unavoidable.

As another example, some programmers do not like to use the word "you" in prograi queries, just as some writers avoid the use of the words "I" and "we" in formal technica articles; e.g. SHADE LEFT TAIL? is preferred to DO YOU WANT THE LEFT TAIL SHADED? (aside from the fact that the former is shorter).

PRINCIPLE V: "Pretty numbers" are not enough.

Axis limits can be determined in three ways: (1) Limits are automatically selected to b "pretty numbers," e.g. values of the form $10^m nr$, where r is 1, 2, or 5, and m and n are integers; (2) Limits are automatically selected by the program to be the maximum and minimu coordinate values (or values slightly outside these); and (3) the user is asked to specif axis limits.

Many packages use only method (1) or only method (2). See Lewart (1973), Malcolm, an Thayer and Storer (1969) for examples of algorithms for determining pretty numbers. The S Package now uses a combination of (2) and (3), having initially used just (2). Many user had requested (1) or (3) for publication purposes, but it was to satisfy the data analyst that the change was made. Pretty numbers alone do not provide enough control because tw different plots with pretty axis limits do not necessarily have the same scale, which i often desirable for purposes of comparison. For example, consider performing probabilit plots of two samples of similar data, one with minimum and maximum 0 and 400, the other wit minimum and maximum 0 and 401. An automatic routine to compute pretty limits for the "observed quantiles" axis might select axis limits of 0 and 400 for the first sample and limit 0 and 500 for the second. The two probability plots would have different scales, making them difficult to compare. The combination of methods (2) and (3) would allow the user t determine the limits of the data and then replot, using the same scale on both plots, and, if desired, with the same user-determined pretty axis limits on both plots.

This method has worked well for axis limits, but tic marks are still a problem. The automatic use of, say, four equal intervals on an axis can sometimes result in some very unpretty numbers as tic labels. Yet use of pretty numbers for tic marks but not for axis limits results in asymmetric positioning of tic marks. The user in a hurry would not usually want to specify how many tic marks should be used for each axis, and, as mentioned above, there is a practical limit to how many options an interactive program can have. For a publication-oriented program, tic mark control is desirable, even to the point of specifying whether the tics are to be long or short, inside or outside the axis.

PRINCIPLE VI. To generate a particular type of statistical plot, the questions that need to be asked are, to a large extent, independent of the program language and plotting device.

Developers of statistical plotting programs can learn and borrow from one another. If a particularly nice query sequence to obtain a certain kind of plot is discovered by one programmer, it can be copied and, if necessary, appropriately revised by another programmer.

For example, to obtain a histogram, the program must obtain the data and determine the interval boundaries. A typical query sequence using the ST histogram program would be as follows:

(Assume that a sample of data has already been accessed. The sample size, minimum and maximum data values, and range have been displayed to aid the user in making subsequent decisions.)

```
DESCRIBE THE INTERVALS (FROM LEFT TO RIGHT).
  LOWER LIMIT OF LEFTMOST INTERVAL: 0
  INTERVAL WIDTH, NUMBER OF INTERVALS: 1 5
  NEXT INTERVAL WIDTH, NUMBER OF INTERVALS: 2 2
  NEXT INTERVAL WIDTH, NUMBER OF INTERVALS: (carriage return)
```

The resulting histogram will have seven intervals, the leftmost five of width one and the other two of width two. The major program packages which produce histograms seem, without exception, to require equal interval widths. This is an unnecessary and inconvenient restriction. Bar heights can be computed by dividing frequency by interval width (as well as by sample size). The resulting ordinate scaling is then appropriate for superimposition of a probability density function.

3. CONCLUSION

The six principles above are given in order to share some of the ideas gained from the experience of developing one program package. Other developers will undoubtedly have more to add, and future technological advances will cure some of the currently unsolved problems.

4. REFERENCES

GENTLEMAN, J. F. (1976a). Interactive Statistical Graphics as an Aid to Data Analysis. Transactions of 30th Annual Technical Conf., Am. Soc. for Quality Control, 267-275.

GENTLEMAN, J. F. (1976b). Interactive Graphics in a Terminal-Equipped Classroom. Comm. in Statistics, A5(10), 949-967.

LEWART, C. R. (1973). Algorithms SCALE1, SCALE2, and SCALE3 for Determination of Scales on Computer Generated Plots. Algorithm 463, Comm. A.C.M., 16, 639-640.

MALCOLM, Michael A. Unpublished manuscript. Dept. of Computer Science, University of Waterloo.

THAYER, R. P. and STORER, R. F. (1969). Scale Selection for Computer Plots. Algorithm 21, Appl. Statist., 18, 206.

BIOGRAPHY

Jane F. Gentleman is an Associate Professor in the Department of Statistics at the University of Waterloo, and has a cross appointment with the Department of Computer Science. She received a Ph.D. in Statistics from Waterloo in 1973.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

HUMAN FACTORS AT THE GRAPHICS INTERFACE

CAPTAIN A. SIMANIS
DIRECTORATE OF AVIONICS AND ARMAMENT
SUBSYSTEMS ENGINEERING
NATIONAL DEFENCE HEADQUARTERS
OTTAWA, CANADA

The interactive computer graphics package, Pierce, used for data display and analysis was designed with human factors criteria being applied to the functional aspects of the man-computer interface. The human factors are described, and examples from Pierce are used to show how they were applied.

THE HARDWARE SYSTEM

The hardware on which Pierce was implemented is shown in Figure 1. The PDP-9 computer handled all computer functions except for curve-fitting calculations, which were done using existing software on a Xerox Sigma-7 via a communications link. The equipment was located at the Communications Research Centre in Ottawa.

HUMAN FACTORS

The human factors that apply to the functional aspects of the man-computer interface are not all amenable to precise definition, nor are they readily verifiable by experiment. Some of them are of an intuitive nature, and their application is more of an art than a science. Nevertheless, their use should be encouraged, if only in an attempt to discover whether or not the interface is improved by their application.

The factors that were applied to Pierce are:

(1) Iconic control cues. If semantic material (i.e., text) is used as light buttons for controlling system action then it is necessary for the user to perceive the word and encode it aurally before its meaning can be understood. However, if icons (i.e., pictures) are used, the intermediate aural encoding process is unnecessary and the meanings of light buttons are perceived more quickly, especially in the initial learning stage of system use. Figure 2 shows a part of the data entry sequence, where iconic cues are used to identify the input devices-magnetic tape, disk, keyboard, and paper tape.

(2) Short-term memory. Where a number of information entry or control actions are necessary, it is easy to forget the earlier actions as the sequence progresses. It should therefore be easy to check the complete state of the system to confirm the progress that has been made. Figure 3 shows how pertinent data is displayed after it has been entered. This tableau is always available for review.

(3) Man/computer allocation of tasks. In an interactive system, those tasks to which judgment or intuition can be applied should be assigned to the operator, whereas those which are essentially clerical in nature are best performed by the computer. In this case, sorting of data points into numerical order and calculation of the polynomial coefficients of a fitted curve are examples of the latter, while selection of the appropriate degree of curve to fit is an example of the former.

(4) Bandwidth of human information channels. Whereas the visual channel is capable

of receiving information at a high rate, there is a problem in enabling the operator to communicate at a high rate to the computer. Perhaps the fastest practical method is the keyboard, but this requires the learning of a concise command language with which to give instructions to the computer. However, if the range of possible meaningful commands at any stage of the man-computer dialogue is limited, then it is possible to display light buttons for each command. In this way, not only is the necessity of learning a command language avoided, but the possibility of giving illegal commands is avoided as well. Care must be taken however, not to display too many buttons at one time (which may be done by organizing them into a tree structure if necessary), so as not to overload the operator's input channel. (There may be interactive situations where system response requirements outweigh the desirability of limiting the bandwidth. In such cases all commands may be made available and the operator will have to familiarize himself with the command menu and learn to ignore what is not pertinent.) Figure 4 shows how the legal commands are all available around the periphery of the display.

(5) Spatial layout of commands. The light button for a particular display function should be integrated into the display in such a manner that it can readily be related to the function it controls. An example is shown in Figure 4 where the controls for the limits of the axes are placed in a label of the axis, rather than being in a list of limit commands in some unrelated part of the display. Selecting one of the limits numbers allows entry of a new number to replace it.

Each of the four sides of the screen can be associated with commands of a particular type. For example, commands on the left are used for going back to a previous state and those on the right for moving ahead to a new state. The locations of commands can be separated around the periphery of the screen, so that a particular light button can be remembered not only by its iconic or semantic content, but by its position as well.

(6) Borders. Borders can be placed around light-button items to draw attention to them, and around large areas having special significance. In Figure 2 a border is used to identify the area in which a selection must be made. (DATA SOURCE).

(7) Response time. Response time is the interval between an event and the system's response to the event. In an interactive system, response times must be fast. A response time greater than 15 seconds rules out interaction (although an operator may be content to wait some minutes if he knows that the processing he has requested involves a great deal of calculation -- an example is shown in Figure 5). A response greater than 4 seconds is too large for activities requiring retention of information in short-term memory; greater than 2 seconds is too long where a high level of concentration is required. Response must be less than 2 seconds where the operator has to remember information throughout several responses, and almost instantaneous to such actions as pressing a key or drawing a curve with a light pen. On the other hand, a too short response time may be harassing to a slow-thinking operator; some systems use a built in delay to make the minimum response time 1.5 seconds.

(8) Feedback. The system should always respond in some fashion, if only to acknowledge receipt of a command, to every operator action. The message in Figure 5, "WAITING FOR...", is provided chiefly for that reason.

(9) Errors and help. There should be a means for the operator to easily correct errors, and to obtain help in understanding a bewildering display. The system should be forgiving and understanding. Pierce allows any data item that has been entered to be changed, and, if necessary, changes the state of the system accordingly and asks for data that is no longer valid to be entered again. The "HELP" function, as can be seen from Figure 6, had not been implemented very elaborately at the time the project terminated.

(10) Security. The data base and system state created during a session should be

secure from session to session. The state of Pierce may be saved on disk or magnetic tape through use of the cues at the top of the display (Figure 7). A default file name for the saved information is provided which may be changed by the operator if he so desires. In that fashion a number of sequential states may be stored for rapid recall and comparison. System state is restored by use of the cues at the bottom of the display shown in Figure 2.

CONCLUSION

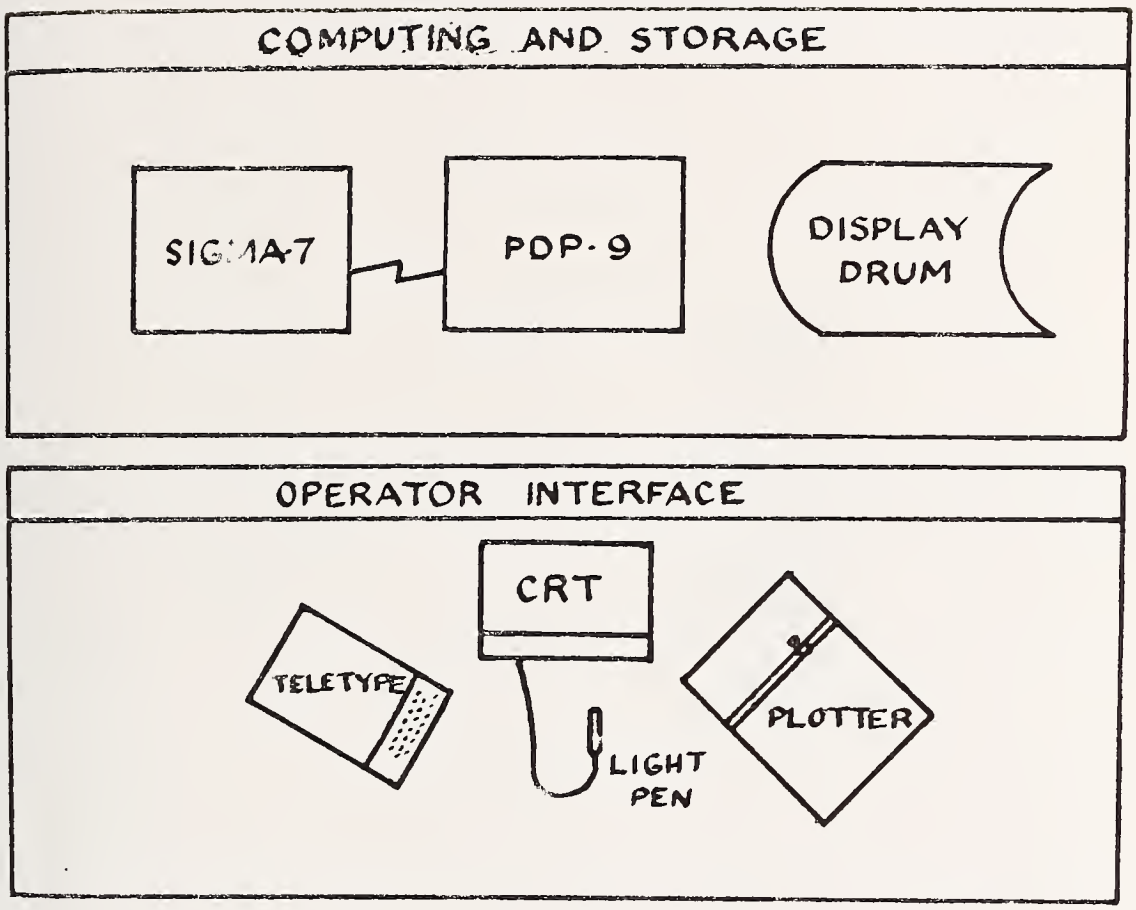
The factors described above do not constitute an exhaustive list. Nor are they universally applicable. Catering to them puts extra demands on the hardware resources of the system and on the time required for design and programming. Nevertheless, these features do enhance the ease of use of systemsand may be essential if a system is to find acceptance in a broad market. An iconic language will undoubtedly develop over the next few decades as the cost and complexity of equipment necessary to "write" in the language is reduced. The interactive devices that are being developed for television receivers are a step in this direction. The application of human factors to the design of these systems can be expected to speed development and user acceptance in this area.

REFERENCES

James Martin, Design of Man-Computer Dialogues, Prentice-Hall, 1973.

W.M. Newman and R.F. Sproull, Principles of Interactive Computer Graphics, McGraw-Hill, 1973.

E.J. McCormick, Human Factors Engineering, third edition, McGraw-Hill, 1970.



PIERCE HARDWARE SYSTEM

Figure 1

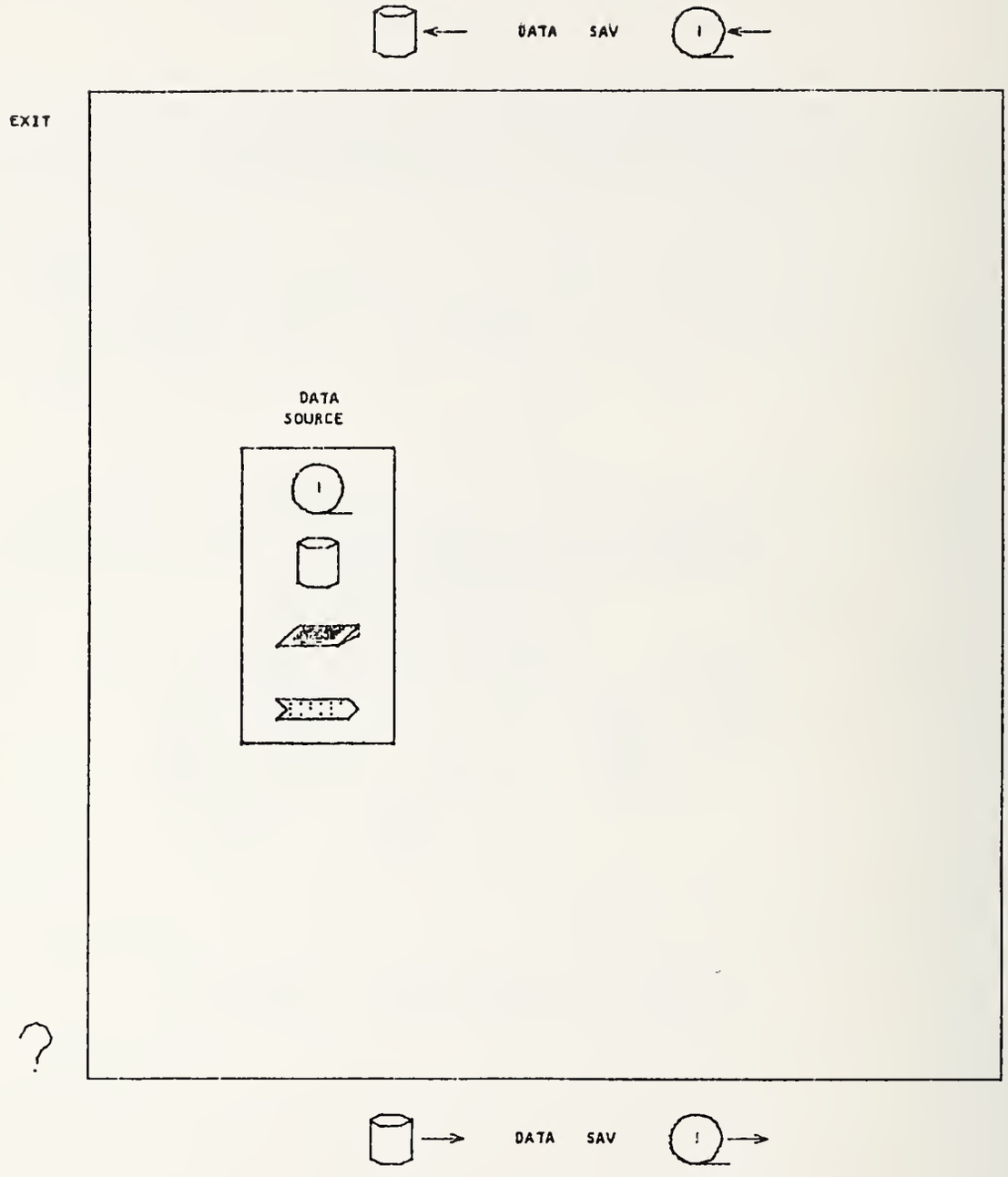


Figure 2

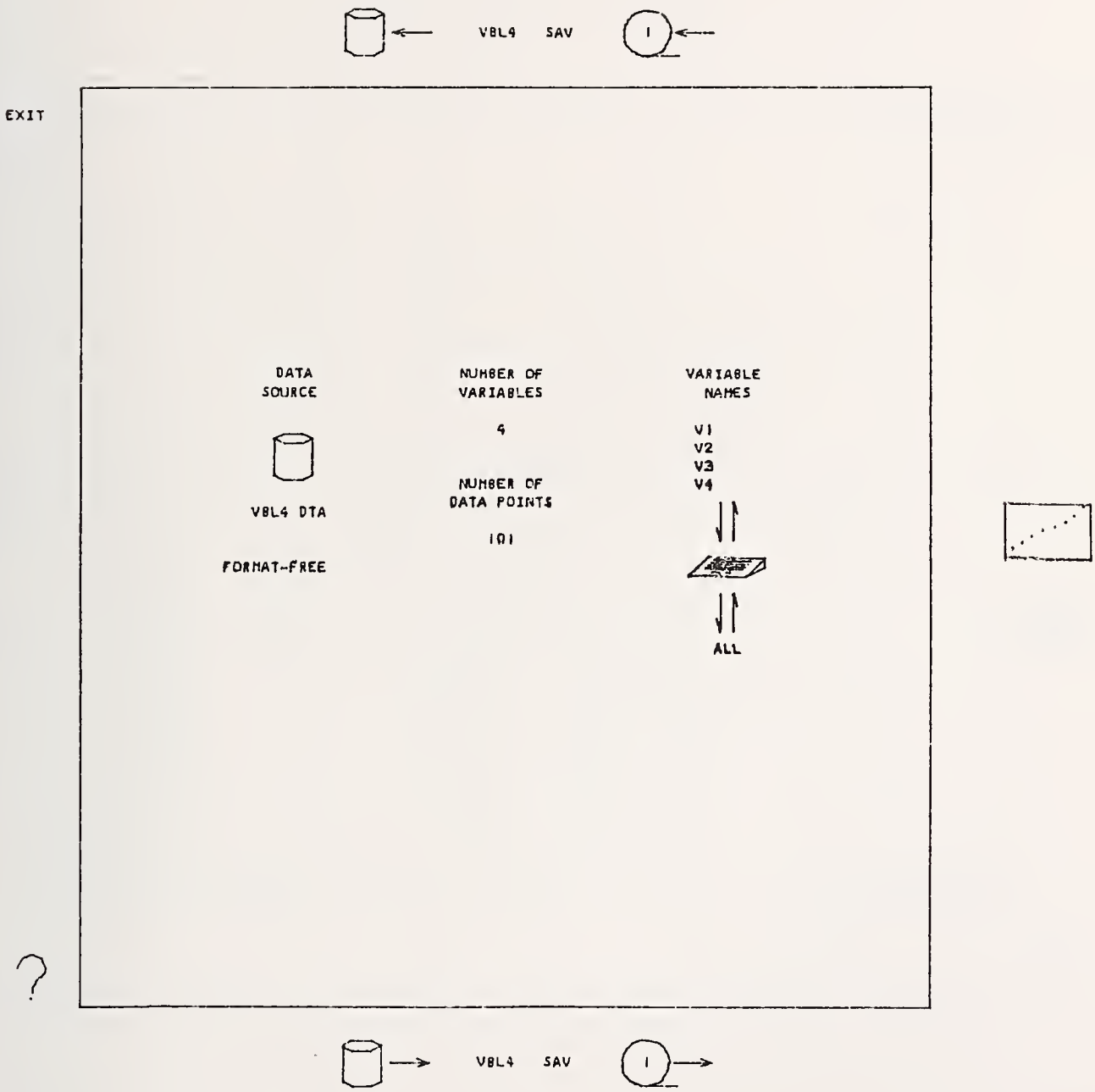

VBL4 SAV
1
EXIT
DATA SOURCE
VBL4 DTA
FORMAT-FREE
NUMBER OF VARIABLES
4
NUMBER OF DATA POINTS
101
VARIABLE NAMES
V1
V2
V3
V4
ALL
VBL4 SAV
1

Figure 3

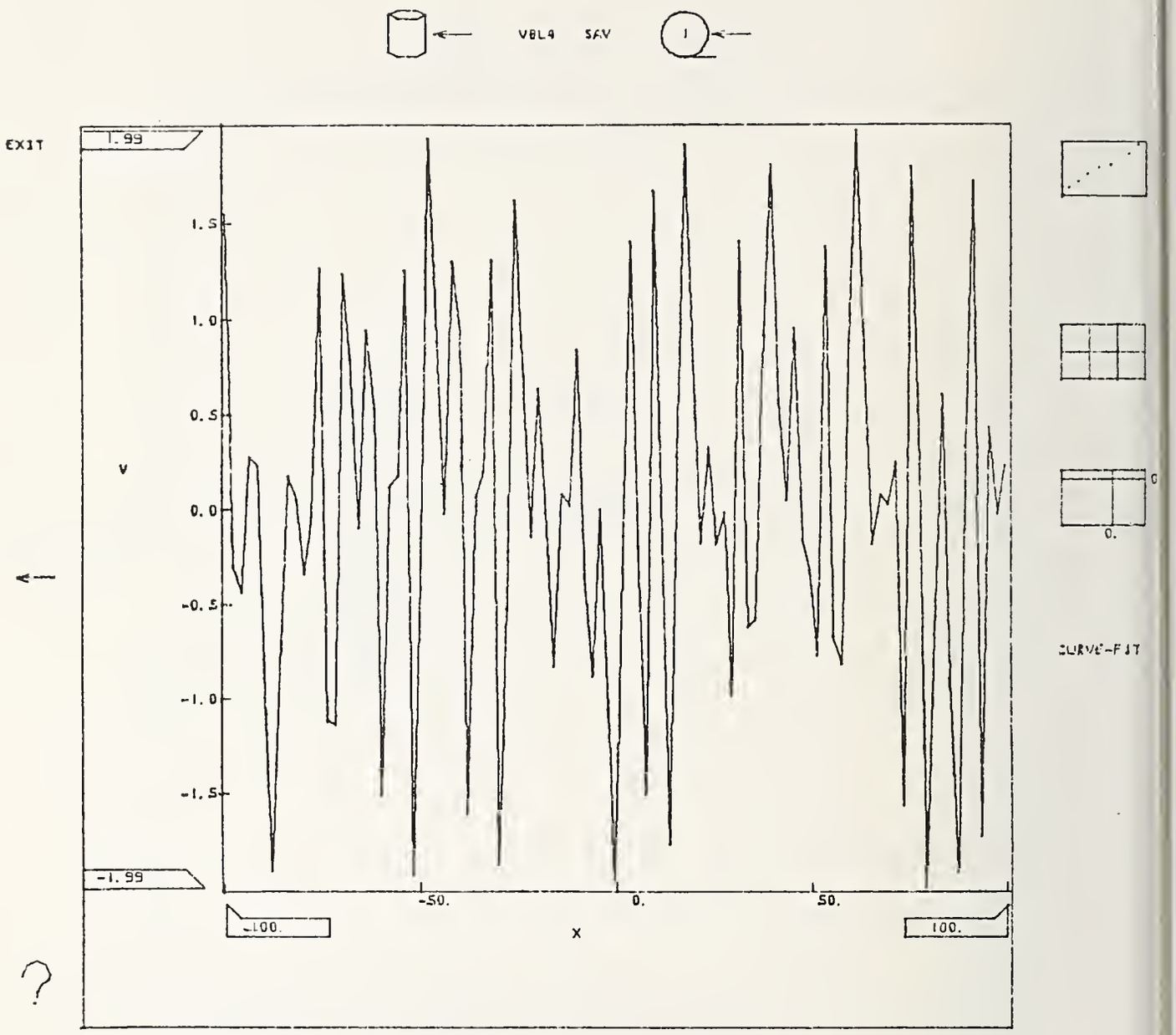

VBL4
SAV
1
EXIT
1.99
1.5
1.0
0.5
V
0.0
-0.5
-1.0
-1.5
-1.99
-50.
0.
50.
-100.
100.
X
CURVE-FIT

Figure 4

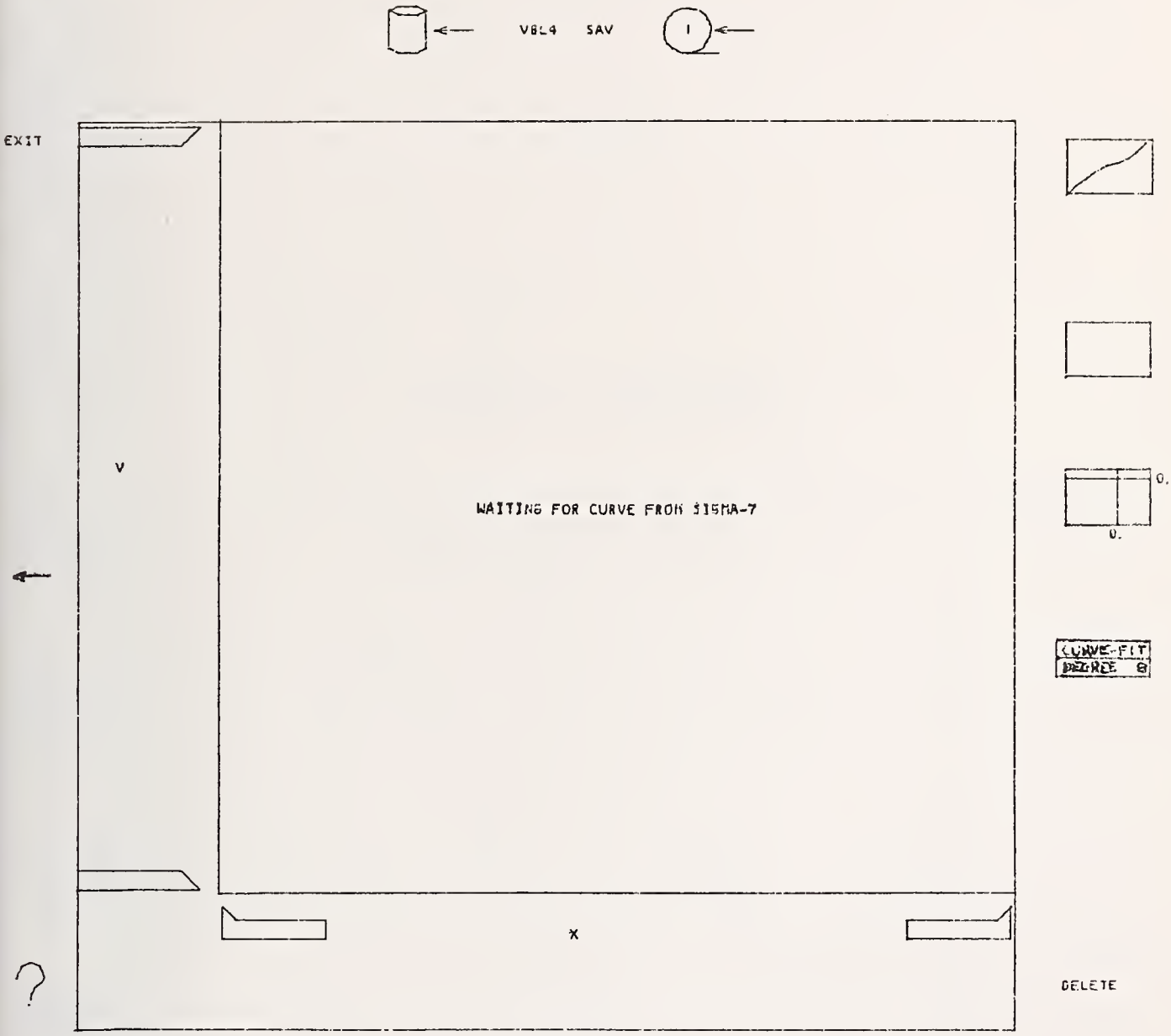

VBL4
SAV
1
EXIT
V
WAITING FOR CURVE FROM SISMA-7
0.
0.
CURVE-FIT
DEGREE 8
X
?
DELETE

Figure 5

PLEASE CONSULT STAFF FOR ASSISTANCE

Figure 6

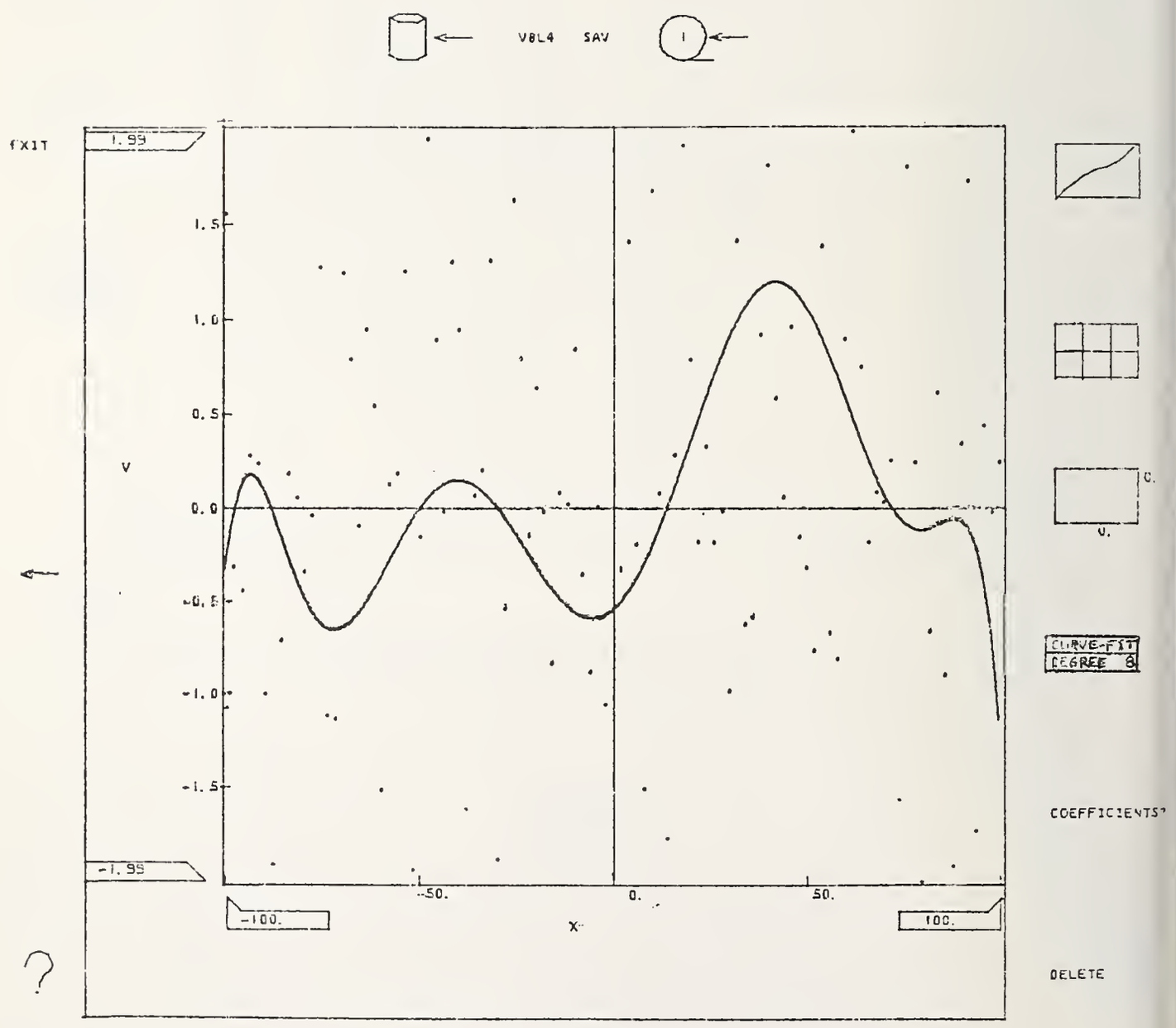


Figure 7

LARGE DATA FILES WORKSHOP

Gordon Sande, Jr., Chairperson

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

LARGE SCALE CLINICAL TRIALS OR HOW DO WE ANSWER THIS

Gary R. Cutter
The University of Texas School of Public Health, Houston, Texas 77030

ABSTRACT

Large scale clinical trials generally pose difficult problems in the area of data analyses. Although purely methodological issues often arise, data analyses for papers produce a host of problems from large volume to inappropriateness of the data set to answer certain questions. The Hypertension Detection and Follow-Up Program (HDFP), a major large scale clinical trial of antihypertensive therapy is discussed. Three examples of problems are given: one related to large volume requests; one on post stratification based on treatment response and a third which combines the stratification problem and selection via truncation.

Key words: Cooperative trials; HDFP; hypertension; large data files; post-stratification; truncation.

1. INTRODUCTION

This mornings' workshop, while focusing on methodological issues of large data files, are really issues of "Data Analysis". Although statistical machinery is a mathematical problem, the actual techniques we use are often a minor note in the problems we face. In a moment I shall return to a few issues of Data Analysis, but I'd like to give you a little background in one of the clinical trials with which I am involved.

The Hypertension Detection and Follow-Up Program (HDFP), one of the largest randomized controlled trials ever undertaken, was initiated with pilot studies by the National Heart, Lung and Blood Institute in 1971. The primary goal of this program is to determine whether systematic antihypertensive therapy, compared with customary medical care, can effectively reduce mortality in a wide spectrum of individuals, aged 30-69, with elevated blood pressure. This program will also permit assessment of whether intense community efforts to identify and treat hypertensives in special programs can improve control of hypertension for those previously undetected, untreated, or uncontrolled in the general population.

Defined populations in 14 communities of varied composition across the United States were enumerated and screened from February, 1973, through May, 1974. Most individuals were first screened for elevated blood pressure in their homes. Suspect hypertensives then underwent a second screen in HDFP clinics. Based on random allocation, participants were given treatment at HDFP clinics, or were referred to an existing source of medical care in these communities.

2. METHODS

Investigators at 14 clinical centers followed a common protocol to assure maximum standardization and comparability of data within the Program. Each center chose its own target population according to local conditions, accessible data, and HDFP requirements. The total Program population was planned to consist of men and women of varied socioeconomic status and racial background, with a broad age range. Sampling frames were census tracts,

robability samples of defined areas, residents of housing projects and in one center orkers employed by selected organizations. Enumeration and screening were aimed at complete coverage of the target population.

2.1 Household enumeration and first (home) screen. The purpose of enumeration was o obtain demographic data on household residents for description of the target population. t consisted of listing the name, age, and sex of all residents and their relationship to he head of each household or dwelling unit. The first screen for those aged 30-69 followed numeration immediately, or as soon after as could be arranged. It consisted of a 15 minute nterview on demographic and health-related topics and three consecutive casual blood ressure readings taken near the end of the interview. If the mean fifth diastolic blood ressure (DBP) of the last two readings was 95 mm Hg or higher, the participant was conidered a first screen hypertensive and eligible for a second screen at the HDFP clinic.

2.2 Second (clinic) screen. Persons with elevated pressures at the first screen who ame to the HDFP clinic for the second screen rested for 5 minutes before blood pressures ere taken. Individuals with a mean fifth phase DBP of 90 mm Hg or higher at the clinic isit were considered eligible for the Program and counted as participants, regardless of heir subsequent actions regarding the HDFP. A cut-point of 90 mm Hg was used in the clinic ather than the 95 mm Hg used at the home screen partially to offset losses anticipated from ower pressures on repeat screening. Participants selected were randomly assigned after tratification by blood pressure level at the second screen to one of two groups: Stepped are (SC) or Referred Care (RC). The results of this assignment were revealed after a econd clinic visit for the collection of additional baseline data including medical history, hysical examination, chest x-ray, electrocardiogram, blood and urine tests. All participants were evaluated for possible secondary hypertension by history and physical examination; ab results and chest x-rays and a more extensive work-up was initiated when indicated by linical criteria.

Referred Care participants were referred to their usual sources of care, frequently heir own physicians, Stepped Care participants were offered free a standardized program of ntihypertensive therapy in HDFP clinics. These clinics differ from most traditional ambuatory care facilities in a number of ways. The participants have been actively and intenively recruited. Uninterrupted antihypertensive drug therapy is attempted as far as posible using techniques presently believed to enhance compliance. Emphasis is placed on linic attendance and adherence to medication schedules. Economic barriers to compliance re removed as much as possible with drugs, clinic visits, laboratory tests, and, if necesary, transportation provided at no cost to the participant.

The Stepped Care drug protocol consists of a standardized program of stepwise, defined ose increments and/or addition of specified drugs until a predetermined level of blood ressure control is achieved. The objective is to provide effective long term control of lood pressure with minimal side-effects. Participants who entered the program with DBP of 00 mm Hg or more, had goal reduction in BP to 90 mm Hg and for those who entered with DBP 0-99 mm Hg a 10 mm reduction in DBP was set as goal. Participants who were already reeiving antihypertensive medicine at baseline were assigned a goal DBP of 90 mm Hg. articipants are seen at least every two months and more frequently when necessary. All ata is collected using a common Manual of Operations and sent to the Coordinating Center hich receives approximately 400 forms per day.

3. RESULTS

3.1 Target population and enumeration, by center. Over 178,000 households were in he Program's target areas. Of these, 84% were enumerated, resulting in the listing of 42,000 individuals of all ages, of whom 178,000 were 30-69 years old and eligible for the irst screen.

3.2 First (home) screen. Screened population: Of 178,000 people aged 30-69 at enu eration, 159,000 (89%) completed the first screen consisting of blood pressure measurement properly recorded and a six page form consisting of 81 items of data. Over 22,000 persons were found to have elevated pressures and of the over 17,000 that came to the second scree over 11,000 were confirmed hypertensives. These participants have generated nearly one-ha a million study forms including clinic revisits, annual revisits, ECGs, x-rays, lab report and other miscellaneous study forms.

As can be gleaned from this brief discourse, an extremely large volume of data is ava able and on file. Because of its availability certain issues have arisen. A paper writin committee recently preparing a paper on the prevalence of high blood pressure innocently r quested a series of tables each displayed by multiple combinations of characteristics. Th request was cutdown but the output still resulted in over one half of a box of computer paper. The overwhelming volume led the committee to ask for regression methods to solve their problems, rather than digging through the initial results.

Regression analyses are how everyone analyzes data of this type, remarked one member the committee. Another noted, regression will tell us what's significant and then we can seek the appropriate cross-classifications.

The problem was to explain the difficulties of interpretation in analyzing regression analyses, involving over 150,000 cases. The usual concepts of utilizing a regression mode when the number of variables are large relative to the sample size is often done because there are too few observations to adequately analyze cross-classifications. Our approach was to take a 10% stratified random sample on the characteristics hypothesized as of inter est, perform the regression analyses and then design the appropriate displays of the cross classifications, and compute various covariate adjusted rates.

A second very common methodological issue is the difficulty in translating questions into an appropriate form for analyses within the data set. The extremely large volume of data encompassing varied and multidisciplinary topics allows not only the most straightforward questions to be conjured up, but also those that appear to be much more subtle in nature. The biggest problem with the subtlety is that although the questions are reasonab and sometimes of great importance, the data, which appears to the investigator requesting the analyses to be appropriate often suffers from limitations in the design. These restri tions do not prohibit one from looking, they only hinder interpretations and in fact may raise more questions than the proposed analysis could answer. For example, certain controversial questions have arisen in the HDFP, such as: Do diuretics alone or in combination have adverse affects on serum cholesterol or potassium? It appears to be well known that body potassium is depleted through the use of diuretics, but what is of interest are the long term effects of the depletion. Obviously such long term effects require some stratif cation either by blood pressure or medication group. It is clear in the minds of the clinicians that to request data solely on all participants is really not an appropriate type of analysis. Different drugs have different actions on these responses. However, unlike a randomized controlled drug trial where the drugs are allocated randomly or patients are allocated randomly to drugs: the HDFP has a stepwise procedure that is a incremental approac increasing dosage when a desired response is not met. Therefore attempts to try and compar various biochemical or treatment responses stratified by what drugs a person is taking at a particular point in time, prohibits any reasonable interpretation of drug induced responses This is because drugs and dosages are increased or decreased only on the achievement of a particular response and that response is generally related to all these other factors that are being measured. This results in comparing persons who are different and one would reasonably expect differences in their responses. It is a continuing problem to identify the requests where it is the response upon which the stratification is being requested and not the initial characteristics. It is often said by the persons requesting the analyses "Oh this is what is done all the time". That may well be true, however, it must also be brough to light that with the thousands of participants we are working with, such analyses inevitably will produce differences amongst the various stratifications. This is particularly true since small correlations among variables will be identified as significant.

Another slightly different issue is the problem of truncation. There is an analytical solution to this problem provided we are insightful enough to recognize those situations where the solution must apply. As I mentioned earlier, the HDFP had a two-stage screening

rocess whereby participants were selected on two occasions. Each time with elevated ressures with respect to a specific and different cutpoint on each occasion. This pro- edure of double truncation has inherent in it some problems that, in part, are well known, ut also very often overlooked in analyses. In many studies the use of a control group in ts purest sense, eliminates some of the concern in the comparison of the variable upon nich selection was based, but not entirely. The HDFP is very interested in the course of lood pressure control. Since a portion of its comparison group is also under therapy with different approach to drug treatment and on fewer persons, the problem is one of attempt- ng to measure the treatment portion of the reduction in blood pressures in both groups. It s of interest to separate the portion due to selection, that is, regression toward the mean, nd that which is due to the effectiveness of treatment in both the stepped and referred are groups. The use of mathematical models is fairly straightforward and has been discuss- d by James (1973) for single truncation, Cutter (1976) and Stinnett (1977) from a multi- ariate point of view which deals with two-stage selection. If noticed as a straightforward roblem, there are tractable solutions that provide reasonable and consistent results. ften there is a more basic problem or question more difficult to recognize as related to nis truncation problem. For example, in a particular series of issues in the HDFP, it was esired to ascertain whether reduction in a particular parameter increased the reduction in lood pressure. The analysis proposed was to stratify by changes in the parameter of inter- st and compare changes in the blood pressure. This seemed to be a reasonable question and nalysis. In that situation there were several problematic and methodological issues which urfaced after utilizing some dice tossing experiments by which we were able to produce cer- ain results based upon selection that mimicked the results we were getting in the analyses hat had been proposed to study the relationship of the two variables. The problem appeared o be in formulating a situation where due to the correlation between measurements, that ortion of regression to the mean that is actually part of the random biological variation nd not measurement error, was contributing not only to the reduction in blood pressure but ifferentially affected the reduction within stratification by the second parameter's reduc- ion. You can see this truncation problem is quite closely entwined with the problem of ost stratification based upon a response. As in the above example both the changes are esponses of interest. In addition, that analysis was further confounded by the use of vary- ng drug regimens. Diuretics for example, were felt to have greater effect on the second arameters reduction as well as blood pressure than the other classes of drugs. The stepped are protocol maintains persons on proportionately more diuretics if they were successful in chieving the goal reduction of their blood pressure than if they were not. As a solution, he approach to this problem has been a stagewise regression model adjusting for various ovariates, changes in certain variables and drug treatment as dummy variables; then assess- ng the impact of the parameters of interest on blood pressure reduction.

In summary, I have presented a few inter-related examples of methodological problems hich related to the extreme volume of data, post stratification by a response and a com- ined truncation and stratification problem. During the course of the requests and dealing ith the actual questions, the problems were not always so obvious or straightforward. side from appropriate techniques, it was very difficult to identify the problem, convince hose requesting the data of the problem and the inappropriateness of our data set to actu- lly answer certain questions. These are but a few of the issues we all face. As this work- hop continues, there will be discussion of what is an outlier in a large data set; how to ffectively use multi-response data; how to assess error, bias and falsification; how to pre- ent the use of statistical techniques for "sanctification" of the analysis and how to mini- ize the proliferation of new data; to mention a few.

It is because of the difficulty in dealing with these kinds of problems that we are ere. When the design has been established for measuring some overall effect and yet ex- reme expenditures of time, effort and money have been put into the collection of data in assive detail, the requirements of data analyses dictate that we must improve our abilities or recognizing and handling these methodological problems. Because the looks are relative- y free, the data available, and that we should not be constrained by a lack of models; we hould look. However, we do not know the controlling factors in many of the responses, we ust be very cautious in the production of results.

4. REFERENCES

CUTTER, G. R. (1976). Some examples for teaching regression toward the mean from a sampling viewpoint. The Amer. Statist., 30, 194-197.

HYPERTENSION DETECTION AND FOLLOW-UP PROGRAM COOPERATIVE GROUP (1977). Blood pressure studies in 14 communities: a two stage screen for hypertension. J.A.M.A., 237, No.22, 2385-2391.

JAMES, K. E. (1973). Regression toward the mean in uncontrolled clinical studies. Biometrics, 29, No.1, 121-130.

STINNETT, S. S. (1977). Regression to the mean in sequential measurement of blood pressure. Master's Thesis, Univ. of Texas, Health Science Center, School of Public Health.

BIOGRAPHY

Gary Cutter received a Ph.D. in Biostatistics from the University of Texas School of Public Health in 1974. He is an assistant professor of Biometry at the School of Public Health and for the past three years he has been Assistant Director for Data Management of the Coordinating Center for the Hypertension Detection and Follow-Up Program and the Principal Investigator of the Coordinating Center for the Impact of Hypertension Information in Selected U.S. Cities.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

SALVAGING EXPERIMENTS: INTERPRETING LEAST SQUARES IN NON-RANDOM SAMPLES

Albert E. Beaton

Office of Data Analysis Research
Educational Testing Service, Princeton, New Jersey

Department of Statistics
Princeton University, Princeton, New Jersey

ABSTRACT

The test statistics in a regression analysis may be interpreted as measures of goodness-of-fit in data sets that are arbitrarily collected. The regression equation is a parsimonious representation of an aspect of the data. The goodness-of-fit is established by demonstrating that the residuals are so small that changing their signs and permuting their positions could not affect the value of the regression coefficients very much. No assumptions outside of the data set need be made.

1. INTRODUCTION

Fitting simple functions to complex data sets is a very basic procedure in all quantitative sciences. A simple--and I think sufficient--reason for fitting is parsimony: if a simple mathematical function can adequately represent a large and complex set of measurements, then the scientist has a hope of reducing the phenomenon under investigation to manageable size. Furthermore, the knowledge that a particular simple function does not fit a set of data is also important.

The theory of statistical inference has added greatly to the potential of some fitting procedures. Statistical theory allows the researcher to generalize beyond his sample to a population of similar objects or events. However, statistical theory typically requires adherence to sampling procedures and to assumptions about the population from which the sample is drawn. The usual assumptions are that elements of a sample are drawn independently from identically distributed elements of a population and, for significance tests, some form of distribution, such as Gaussian, is also assumed. The power of statistical inference, therefore, comes at some cost.

Surely, in any cases where these assumptions are plausible, the logic and therefore the power of statistical inference should be used. But sometimes only a non-random sample is feasible, or the returns from a well designed survey are so biassed that formal statistical inference is no longer plausible. Also, in studies where many statistical models are fitted from a single sample (e.g. stepwise regression), statistical theory either breaks down or is too complicated for practical use. Despite these problems, many researchers proceed to use statistical theory with--it would seem--an intuitive notion that the numbers computed by a fitting routine such as a standard regression program are useful despite the inapplicability of the theory by which the numbers are given meaning.

The purpose of this paper is to discuss the process of fitting mathematical models to data without making the usual assumptions of statistical inference. The focus will be on the size of the residuals relative to the fitted values. The paper will show that each of the statistics associated with regression analysis can be interpreted in terms of the goodness of fit. No assumptions of random sampling nor of Gaussian distribution in a population will be made.

2. LEAST SQUARES FITTING

The commonest form of fitting is (unweighted) least squares, that is, fitting a mathematical model in such a way that the sum of squared residuals from the fit is at its minimum. The theory and derivations of least squares are too well known to cover here (see Daniel and Wood (1971) or Draper and Smith (1967)). The definitions used in this paper are shown in Table 1 and the basic calculations of standard regression programs are shown in Table 2. Except for the matrix C, which is used here for notational purposes only, all other statistics in Table 2 have well known interpretations in statistical estimation.

Some of the statistics in Table 2 have meaning without reference to a population. The mean, $\bar{y}$, is the center of distribution in the sense that the sum of the deviations of the y_i from that point are zero. s_y^2 is a measure of variance, although dividing by N instead of N-1 might seem more appropriate. The vector $\hat{b}$ contains the coefficients of the least squares fit. The vectors $\hat{y}$ and $\hat{e}$ are the fit and the residuals, respectively. The standard error of estimate is clearly a measure of goodness-of-fit, but the simpler root mean square of residuals would suffice. The squared multiple correlation is also easily interpreted as a measure of goodness-of-fit.

The covariance and standard errors of the regression coefficients have no obvious interpretation in the sample nor do the test statistics nor their associated probabilities. In the next sections it will be shown that these statistics can be also interpreted as measures of goodness-of-fit for a particular model and a particular set of data.

Table 1

Notation

N	number of observations
m	number of regressors
$i,i' = 1,2,\ldots,N$	indices of observations
$j,j' = 0,1,2,\ldots,m$	indices of regressors
$\underset{\sim}{y} = \{y_i\}$	Nth order vector of values to be fitted
$X = \{x_{ij}\}$	Nx(m+1) matrix of regressors. All elements $x_{i0} = 1$. The rank of X is m+1

Table 2

Standard Least Squares Calculations

$\bar{y} = \Sigma_i y_i/N$	mean value of the y_i
$s_y^2 = \Sigma(y_i-\bar{y})^2/(N-1)$	variance of the y_i
$\hat{b} = \{\hat{b}_j\} = (X'X)^{-1}X'y$	regression coefficients
$\hat{y} = \{\hat{y}_i\} = X\hat{b}$	fitted values
$e = \{e_i\} = y - X\hat{b}$	residuals
$\bar{e} = \Sigma_i e_i/N = 0$	mean residual = 0
$s_e = \sqrt{e'e/(N-m-1)}$	standard error of estimate
$R^2 = \frac{\Sigma(\hat{y}_i-\bar{y})^2}{\Sigma(y_i-\bar{y})^2}$	squared multiple correlation
$F = \frac{\Sigma(\hat{y}_i-\bar{y})^2/m}{\Sigma(y_i-\hat{y}_i)^2/(N-m-1)}$	test statistic for $\beta_1=\beta_2=\ldots=\beta_m=0$
$p(F)$	probability associated with F
$\text{cov}(\hat{b}) = \{\text{cov}(\hat{b}_{ij})\} = s_e^2(X'X)^{-1}$	covariance of $\hat{b}$
$SE(\hat{b}_j) = \sqrt{\text{cov}(\hat{b}_{jj})}$	standard error of $\hat{b}_j$
$t_j = \hat{b}_j/SE(\hat{b}_j)$	test statistic for $\beta_j=0$
$p(t_j)$	probability associated with t_j
$C = \{c_{ij}\} = X(X'X)^{-1}$	Nx(m+1) matrix of catchers or generalized inverse of X

3. SIGNED PERMUTATIONS OF RESIDUALS

As suggested in the introduction, a parsimonious representation of a complex relationship among a set of variables is a sufficient reason for least squares fitting. Any set of numeric data for which X'X has an inverse can be fit by least squares, although not necessarily very well. The regression coefficients are the basic summary statistics for, given the values of a row of X, we can use the regression coefficients to approximate the corresponding value of y. The question of goodness-of-fit asks how well can the values in the vector $\underset{\sim}{y}$ be reproduced from the values in the matrix X, or, how small are the residuals when compared to the fit. Clearly, if the residuals are all zero, the fit is perfect, but in almost all real data sets there will be some nonzero residuals. We would like the residuals to be small enough such that if we removed them - threw them away - there would be no particular effect on the least squares coefficients which are the important summary of the data. We need, therefore, to develop a metric for measurement of goodness-of-fit.

Although a researcher may accept the regression coefficients if the standard error of estimate is small enough to suit his liking, the statistician can tell him more about the properties of the fit. Basically, accepting a fit means throwing away--at least singling out for special study--the residuals. To do so, the researcher should know whether or not the residuals are small or irrelevant enough that they can be ignored. We will consider the residuals small enough to be ignored if the fit does not change much whether we rearrange the residuals or change their signs.

Let us consider a simple, contrived set of data such as in figure 1. The residuals seem small compared to the fit. If the residuals are small, then we can rearrange them at will with little effect on the fit; for example, we might swap the first and second residuals. The value of the first element of a new vector of the reconstructed values of the regressand would be $\hat{y}_1 + e_2 = 5-1 = 4$ and the second element $\hat{y}_2 + e_1 = 7+2 = 9$. Such a permutation of the residuals would typically affect the regression coefficients if we were to recompute the regression analysis using the new vector as the regressand. There are N! different ways the residuals could be permuted. My colleague, Paul Holland, suggested changing the signs of the residuals which results in 2^N possible vectors of residuals with with signs changed. Combining the sign changes and permutations, there are $2^N N!$ possible different signed permutations of the residuals.

We cannot, of course, reasonably compute the effect of each possible signed permutation of the residuals, for a sample of just five would require 3,840 regression computations, a sample of six would require 46,080, and a sample of 50 would require 3×10^{64} regression analyses. We can, however, calculate the mean and variance of the effect of these signed permutations on the regression coefficients and the regression plane.

The notation to be used in the rest of this paper is shown in Table 3. σ^2 is the mean square of the residuals. The matrix P_k is an NxN permutation matrix which denotes the transformation from the original residual vector e to the kth signed permutation, i.e.

$$\underset{\sim}{e}_k = P_k \underset{\sim}{e}$$

Each row and column of P_k may have one non-zero element which is +1 or -1 depending on whether or not the sign of the residual is changed. The modified values $\underset{\sim}{y}_k$ are defined as the original values of the regression surface, $\hat{\underset{\sim}{y}}$ plus a signed permutation of the residuals. Given the vector $\underset{\sim}{y}_k$, the calculations of the modified regression coefficients, $\hat{\underset{\sim}{b}}_k$, and regression surface, $\hat{\underset{\sim}{y}}_k$, follow from least squares.

A summary of the net effects of the signed permutations are shown in Table 3b. (Proofs are in Beaton (1977)). We see that:

a. The average of all $2^N N!$ different sets of regression coefficients is the original set of regression coefficients;

b. the covariance of all the sets of regression coefficients is $\sigma^2(X'X)^{-1}$;

c. the average value of all $2^N N!$ fits of the regression surface is the original regression surface;

d. the covariance of all sets of points on the regression surface is $\sigma^2\, X(X'X)^{-1}X'$; and

e. the average squared distance of each set of regression coefficients from the original set is m+1.

We note, first, that these summaries are exact, not estimates or approximations. Secondly, we note that the values are similar, almost identical, to the statistics derived from sampling theory for estimating population values, the only difference being substitution of σ^2 for s_e^2 in the covariances of the $\underset{\sim}{b}_k$ and of the $\hat{y}_k$. This finding allows us to make a new interpretation of the results of standard regression programs.

Knowing the exact variance over all signed permutations of a single regression coefficient, b_j, say, gives us an opportunity to say something about the standard error of a regression coefficient and its associated t and p statistics. The different values of the jth regression coefficient on the kth signed permutation, $b_{j(k)}$, say, are symmetrically distributed about the average value $\hat{b}_j$, since, for every positive deviation, there is an identical negative deviation. Since any regression coefficient is a weighted sum of the regressands, the central limit theorem leads us to expect the distribution of $b_{j(k)}$ over all k to approach the Gaussian distribution in reasonably large samples. (Since the sample data are finite, the Gaussian distribution will never actually be reached.) Thus we have the exact mean, exact variance, and approximate distribution of $b_{j(k)}$.

We may now ask the question: how many of the signed permutations of the residuals affect the regression coefficients so much that the coefficient of X_j changes sign? In other words, what proportion of the $b_{j(k)}$ are of different sign from $\hat{b}_j$? We can answer this question exactly in small problems by computing all values of $b_{j(k)}$, then calculating

the proportion with different signs. In large samples, we may approximate the proportion through the Gaussian distribution. The first step is to measure the distance of b_j from the point where its sign would change, i.e. zero, in terms of the standard deviation of the $b_{j(k)}$. Since the standard deviation is

$$\sigma_{bj} = \sigma\sqrt{(X'X)^{jj}}$$

where $(X'X)^{jj}$ is the jth diagonal element of $(X'X)^{-1}$, we may write

$$t_j^* = \hat{b}_j / \sigma_{bj}$$

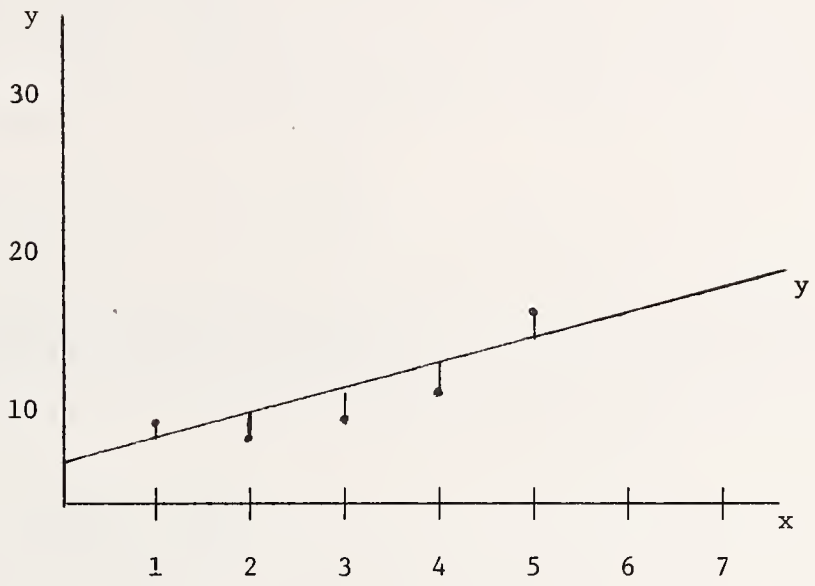
as that distance. If we are willing to assume that the Gaussian distribution can be used a an approximation to the actual distribution, then the value t_j^* may be referred to probability tables (one tailed) for the approximate proportion (p_j^*) of $b_{j(k)}$ that would differ in sign from $\hat{b}_j$.

We stress here that the values of σ_{bj}, t_j^*, and the associated proportion p_j^* are almost the same statistics computed in standard regression analyses. The σ_{bj} differs from SE_{bj} only in the subtraction of degrees of freedom in the denominator, making σ_{kj} slightly smaller. The value t_j^* is thus slightly larger than t_j. Thus the proportion of $b_{j(k)}$ changing sign will be almost the same as the sampling theory probability of finding a sample value as large as b_j when the actual population value is zero.

Therefore, the standard error, t, and p statistics in a regression output have an approximate meaning even in nonrandom samples. Note that the Gaussian distribution was used here as a mathematical approximation, not as a statistical assumption about an underlying distribution from which a random sample was taken. Although there may be cases in which the approximation is poor, it should be fairly accurate in models with well-behaved residuals.

Another question we may ask is: how many of the signed permutations affect the regression coefficients so much that all of them change sign? Putting this question another way, we may ask: how many of the vectors $b_{(k)}$ are as far away from $\hat{b}_j$ as the point where all elements of the vector change sign (i.e. the origin). (Note: we will ignore the intercept as is usually done in regression programs.) We state without proof here that the F statistic computed in a regression analysis is an approximate measure of the distance of $\hat{\underset{\sim}{b}}$ from $\underset{\sim}{0}$, and that the p statistic associated with that F is the approximate proportion of $\hat{\underset{\sim}{b}}_k$ that are as far away from $\hat{\underset{\sim}{b}}$ as the origin. This statement is contingent upon the approximate multivariate Gaussian distribution of the $\hat{\underset{\sim}{b}}_k$, but not on population values. Thus, if the p statistic is small, then the residuals are small enough that they seldom affect the signs of all the coefficients in the $\hat{\underset{\sim}{b}}$.

Figure 1



$$\underset{y}{\begin{bmatrix} 7 \\ 6 \\ 7 \\ 10 \\ 15 \end{bmatrix}} \quad \underset{x}{\begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \end{bmatrix}} \quad \underset{e}{\begin{bmatrix} +2 \\ -1 \\ -2 \\ -1 \\ +2 \end{bmatrix}} \quad \underset{\hat{y}}{\begin{bmatrix} 5 \\ 7 \\ 9 \\ 11 \\ 13 \end{bmatrix}} \quad \underset{b}{\begin{bmatrix} 3 \\ 2 \end{bmatrix}}$$

$$\hat{y} = 3 + 2X$$

$$\underset{CP}{\begin{bmatrix} 459 & 45 & 155 \\ 45 & 5 & 15 \\ 155 & 15 & 55 \end{bmatrix}} \quad \underset{(X'X)^{-1}}{\begin{bmatrix} 1.1 & -.3 \\ -.3 & .1 \end{bmatrix}}$$

$$s_y^2 = \frac{54}{4} = 13.5$$

$$s_e^2 = \frac{14}{3} = 4.67$$

Table 3

a. Signed Permutation Definitions

$\sigma^2 = \Sigma_i e_i/N$	Mean square of residuals
$N_k = 2^N N!$	number of possible signed permutations
$k = 1,2,\ldots,N_k$	index of signed permutations
$P_k = \{p_{kii'}\}$	kth NXN signed permutation matrix. Each row and column has exactly one nonzero element which may be either +1 or -1.
$\underset{\sim}{y}_k = \hat{\underset{\sim}{y}} + P_k \underset{\sim}{e}$	Modified values of $\underset{\sim}{y}$ for kth signed permutation
$\hat{\underset{\sim}{b}}_k = (X'X)^{-1}X'\underset{\sim}{y}_k = \hat{\underset{\sim}{b}} + C'P_k\underset{\sim}{e}$	Regression coefficients for kth signed permutation
$\hat{\underset{\sim}{y}}_k = X\underset{\sim}{b}_k = \hat{\underset{\sim}{y}} + XC'P_k\underset{\sim}{e}$	Fitted values for the kth signed permutation

b. Summary of Statistics for Signed Permutations

$$\text{ave}\,(\hat{\underset{\sim}{b}}_k) = N_k^{-1}\,\Sigma_k\,\hat{\underset{\sim}{b}}_k = \hat{\underset{\sim}{b}}$$

$$\text{cov}\,(\hat{\underset{\sim}{b}}_k) = N_k^{-1}\,\Sigma_k(\hat{\underset{\sim}{b}}_k - \hat{\underset{\sim}{b}})(\hat{\underset{\sim}{b}}_k - \hat{\underset{\sim}{b}})' = \sigma^2(X'X)^{-1}$$

$$\text{ave}\,(\hat{\underset{\sim}{y}}_k) = N_k^{-1}\,\Sigma_k\,\hat{\underset{\sim}{y}}_k = \hat{\underset{\sim}{y}}$$

$$\text{cov}\,(\hat{\underset{\sim}{y}}_k) = N_k^{-1}\,\Sigma_k(\hat{\underset{\sim}{y}}_k - \hat{\underset{\sim}{y}})\,(\hat{\underset{\sim}{y}}_k - \hat{\underset{\sim}{y}})' = \sigma^2 X(X'X)^{-1}X'$$

$$\text{ave}\,(d_k^2) = N_k^{-1}\,\Sigma_k(\hat{\underset{\sim}{b}}_k - \hat{\underset{\sim}{b}})'\,[\text{cov}(\hat{\underset{\sim}{b}}_k)]^{-1}(\hat{\underset{\sim}{b}}_k - \hat{\underset{\sim}{b}}) = m+1$$

4. REFERENCES

BEATON, A.E. (in process). Least Squares in Non-Random Samples.

DANIEL, C. AND WOOD, F.S. (1971). Fitting Equations to Data. John Wiley and Sons, N.Y.

DRAPER, N.R. AND SMITH, H. (1967). Applied Regression Analysis. John Wiley and Sons, N.Y.

BIOGRAPHY

Albert E. Beaton was born in Boston, Massachusetts in 1931. He received a Doctor of Education degree in Educational Measurement and Statistics from Harvard University in 1964. He has worked in the areas of statistics, educational measurement, and computer science. He is now the Director of the Office of Data Analysis Research at Educational Testing Service and Visiting Lecturer in Statistics at Princeton University.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

RECORD LINKAGE BY BIT PATTERN MATCHING

David Blaxell, Ph.D.
Consumer and Corporate Affairs, Ottawa, Canada K1A 0C9

ABSTRACT

Record linkage is non-trivial when records share no common access key. One application is searching for records most similar to a query record; another is bridging two independent files covering similar universes.

Bit pattern matching technique and experience are reported. Every record is preanalyzed into a fingerprint bit pattern of 60 bits. The ones population count of the logical product of two such patterns scores similarity between two records.

Bit pattern generation guidelines and tolerances of data errors and of non-ideal design are considered using unit hypercubes and unary numbers in base one. Use of the similarity scores for linkage criteria depends on the application philosophy.

Key words: Best matches; bit pattern matching; bit sum; boolean arithmetic compatibility; interpretation; name searching; ones population count; similarity evaluation; unary numbers; unit hypercube.

1. INTRODUCTION

1.1 Linkage Applications.

How to find it when you don't exactly know what you are looking for.

"The licence was ARN 287 or ANR 237, I think, and the car was green or bluish." Such information illustrates the problems of linkage between records. These clues constitute a query record and we must use them to find the most likely one or few records it might be in a file of data.

Retrieval of the most probable record in a file, given an incomplete or faulty query record, is needed to follow up clues to a crime or errors detected in a database. Retrieval of the most similar records to a query, is needed for legal research on new trademarks and corporate names [1,2]. Linkage of exceptionally close records, is needed in bridging together two files; that is attempting to find corresponding records in each file for every record. It is also needed in trademark surveillance or watch-dogging for close record pairs.

Retrieval of unlinkable records is needed because they represent gaps in, or extensions to, another file. Linkage of questionnaire responses with the best of a set of typical profiles, is needed for automatic classification or cluster recognition.

Files of corporations and of trademarks frequently exceed 100,000 records and efficient, effective and automated record linkage techniques are much wanted.

1.2 Evolution of bit pattern matching. Searching for trademark records most similar in sound, design, meaning and products began simply. Those records within a Boolean subset of a group of product classes, a design class if any and some spelling criteria as specified in a formulated request, were retrieved. [3]

Retrieved records, too many of course rather than too few, were automatically evaluated against the query to give a similarity value key for sorting out the most similar by whatever criteria. Numerous small criteria, the letters used to spell the name and the groups to which a product class belonged, all contributed to a similarity score and were thought of as elements since the score was the criterion to keep or drop a record found.

The evaluator became efficient and effective enough to be used for searching on the whole file rather than on just the subset retrieved: when the similarity elements were ultimately bits, zeros and ones, in a pattern whose size was large enough to ensure elimination of enough unwanted pairs but small enough to match in only a handful of computer instructions. Then no query formulation was needed, just the query itself, entered like an update record. That is to say the searching became automated.

1.3 Use of the technique. The bit pattern matching technique involves:

- definition of bit pattern generation appropriate both to the data at hand and to the recall objective of the proposed linkage, and

- linkage program containing the bit pattern matching routine (which may be hardware dependent for the sake of efficiency) to compute linkage values for record pairs of desired retrieval precision, which in turn may use a feedback formula or algorithm.

The records are all first preanalyzed generating bit patterns and bit sums as two extra fields in each record, possibly dropping other fields unnecessary for linkage. Then the linkage processing is done producing a subset of linked record pairs with link values. These may then be sorted as appropriate, descending link value giving recall with precision adjustable after linkage!

The preanalysis is done only once to form the bit pattern for every record on a file and for every update record, however many times that file is subsequently subjected to a search, bridging, or other linkage processing.

The linkage processing may be part of some merge-and-match process and be called upon only for problem record pairs. It may be called for every possible pair of a batch of 1 to 1000 records in central memory each paired with every record on a database file. It may be used for browsing between two roughly ordered files. It could be used in an associative array processor like STARAN to exhaust all possible trillion record pairs between two files each of a million records.

2. SIMILARITY EVALUATION

2.1 Scalar values as criteria.

"Does she or doesn't she?"
Trademark Canada Reg. No. 124509

An algorithm takes information from two records and computes a score, on a scale of a hundred, say, which score is a similarity value. The similarity value result is to be used to decide whether these two records are closer than one of them compared with a third record (relative value) and, if so, whether the similarity value is high enough to warrant record linkage (absolute value). These decisions are yes or no binary choices. The similarity value must be a scalar quantity, that is simply a number and *not* a multi-valued set of numbers, in order to be used as the basis, that is criterion, for a decision. [4]

If one scores the two records against each other, field by field, some fields will agree, others not: same in name, place, date but different in initials, class and size, for instance. One can assign statistical weights to each field and total the weights of all fields different to give another number. Then one needs a formula to combine these two numbers into an explicit scalar result, the similarity value.

2.2 Formula justification.

You all look alike to
me, you humans

My similarity formula to give the similarity value between two information records can be argued for as follows:

The information of interest falls into one of three areas, calling one of the two records compared the query, the other the datum:

(a) information peculiar to the query,
(b) information peculiar to the datum, and
(c) information common to both records.

Call the total weights in each area A, B and C respectively. The similarity formula is to relate the similarity value, S, to parameters A, B and C as a formula into which one inserts the values of A, B and C to calculate S.

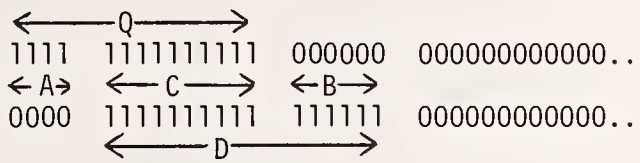
Now similarity will increase with any increase in C, but will decrease as A or B increase. Both A and B contribute to dissimilarity. One might naively add together A+B to produce a dissimilarity value but a better measure of distinction would reflect the fact that one of the records may be an incomplete, abbreviated or partial version of the other (and conversely that the other is a fuller, extended or elaborated version of the first one). This is achieved by multiplying A times B to give A*B where * is the multiply operator. This distinction value, A*B, vanishes to zero if one of the records has no information in it *other than* information which occurs in the other record.

Combining A*B with C, that is a distinction value with a commonality, can be done as the algebraic difference, that is subtracting one quantity from another if the two quantities are measured in the same units. To do this, the commonality can be expressed in weight-units squared, like the distinction value, to give C^2-A*B. One could alternatively have reduced the distinction to the weight-units by using C-$\sqrt{A*B}$ but

the squared version has more analogs in physical science as a parallel to intensity rather than amplitude and gives a simpler algorithm to compute. [5]

2.3 Universal and absolute. A similarity formula may process slogans, say, differently from acronyms [6]. Indeed it would be hard to define a "same way" of processing two such different fields. However, the possible values generated by two formulas for the A, B, C values on which each formula applies, define sets of contours in ABC-space. Irregularities occur at boundaries of formula applicability conditions. Each formula may suffice for shortlisting similar records relative to one query or to another query but the values in the cases of the two queries are not absolutely comparable and one cannot judge which query has the closer matches.

Therefore, the similarity formula should be universally applicable and designed to give a result which always falls in the same finite range, zero to one hundred, say, no matter how little or how much information is in the records compared. This absolute (or normalized) similarity value is then comparable itself from one record pair to another record pair with no record common to the two record pairs. With more information in the records, the number of similarity values possible will be greater and, in a sense, the similarity value will be more accurate in such cases than with sparse information.



Since the possible range of C^2-A*B is

from a low, when C=0 of -A*B or -Q*D
to a high, when A=B=0 of $+C^2$ or +Q*D

where Q=A+C the total information (peculiar or common) in the query
and D=B+C the total information (peculiar or common) in the datum

the similarity formula to give a similarity value on a scale of 100 (whatever the weight-unit is, large or small) is

$$S = 50 + 50*(C^2 - A*B)/(Q*D) \qquad \#1$$

which simplifies, eliminating A, B in terms of Q, D which are precalculable, to

$$S = C*(50/Q + 50/D) \qquad \#2$$

This formula is a pragmatist's delight! The two variables, 50/Q and 50/D, can be precalculated once for each record no matter how many comparisons will be made. The files can be sorted or merely relaxed a little to keep (50/Q+50/D) constant for thousands of consecutive comparisons. C, the common information, is readily computed if bit patterns represent the salient information in each record.

Variation of the 50-50 balance between query and data is appropriate if the query and data are not of equal reliability. It only affects pairs of records with different amounts of information in each.

3. BIT PATTERN MATCHING

3.1 Computer algorithm.

To find a needle in a haystack
an attractive tool is a magnet

Bit patterns can be matched in two steps:

1. Forming the logical product of two bit patterns (*B*oolean AND). This result is a bit pattern having ones for those corresponding bits one in both the patterns operated on. 0011, 0101 give 0001.

2. *C*ounting the population of bits one in that result (bit sum). This result is a binary integer number of how many bits were one in the pattern operated on. 1111 unary is 100 binary is 4 decimal.

These steps can be coded simply - only two assembly language instructions [7] for 60-bit patterns on CDC 6000 or CYBER series computers, for instance. The two instructions could be the following:

```
BX3   X1*X2      (X3 is AND(X1,X2))
CX4   X3         (X4 is Bitsum(X3))
```

Thus, a simple routine can be written to make a bit pattern match operation available in any high-level language. The programming is quite easy.

The bitsum or ones population count of a bit pattern on a computer lacking the count ones instruction (also called the population instruction) can be calculated in other ways [8,9]. Each 8-bit byte of the pattern can be used to index its bitsum in a table of 256 bitsums ranging from 0 to 8 and the bitsum of the whole pattern is the sum of the bitsums of each byte. Another method, most appropriate for bitsums of logical products which tend to have few ones, is faster the fewer ones there are to sum. That is to form the logical product of the pattern M and the pattern M with the rightmost one removed (zeroed) - simply by M.AND.(M-1). These steps can be repeated until a zero results. Each repeat of the loop counts another one in the original pattern.

The use of this simple technique for record linkage is a harder part to understand. Some interpretation is helpful.

The *matching process* takes as its input the bit patterns, which represent the salient information in two records, and produces as its output a number, which is a scalar value that can be used for comparing this match (of two records) with some other match. It condenses multiple factors into a single, net criterion.

3.2 Models for a bit pattern.

Any bit pattern can be regarded as a set of binary answers to some set of corresponding questions, that is to a questionnaire. The binary answers are each yes-or-no (or true-or-false) and the pattern could be thought of as a truth-matrix. Although the bits can define a row it is not necessary to think of the leftmost as the most significant - it is indeed misleading to do so in the context here, where each bit is equal in significance to, and statistically independent of, every other bit in the ideal bit pattern. A bit pattern in a set of patterns whose every bit is independent and equally probably one (not the other bit value, zero) is interpretable as the coordinates of some corner of a unit hypercube in hyperspace having as many (orthogonal) dimensions as there are bits in the bit pattern. [10]

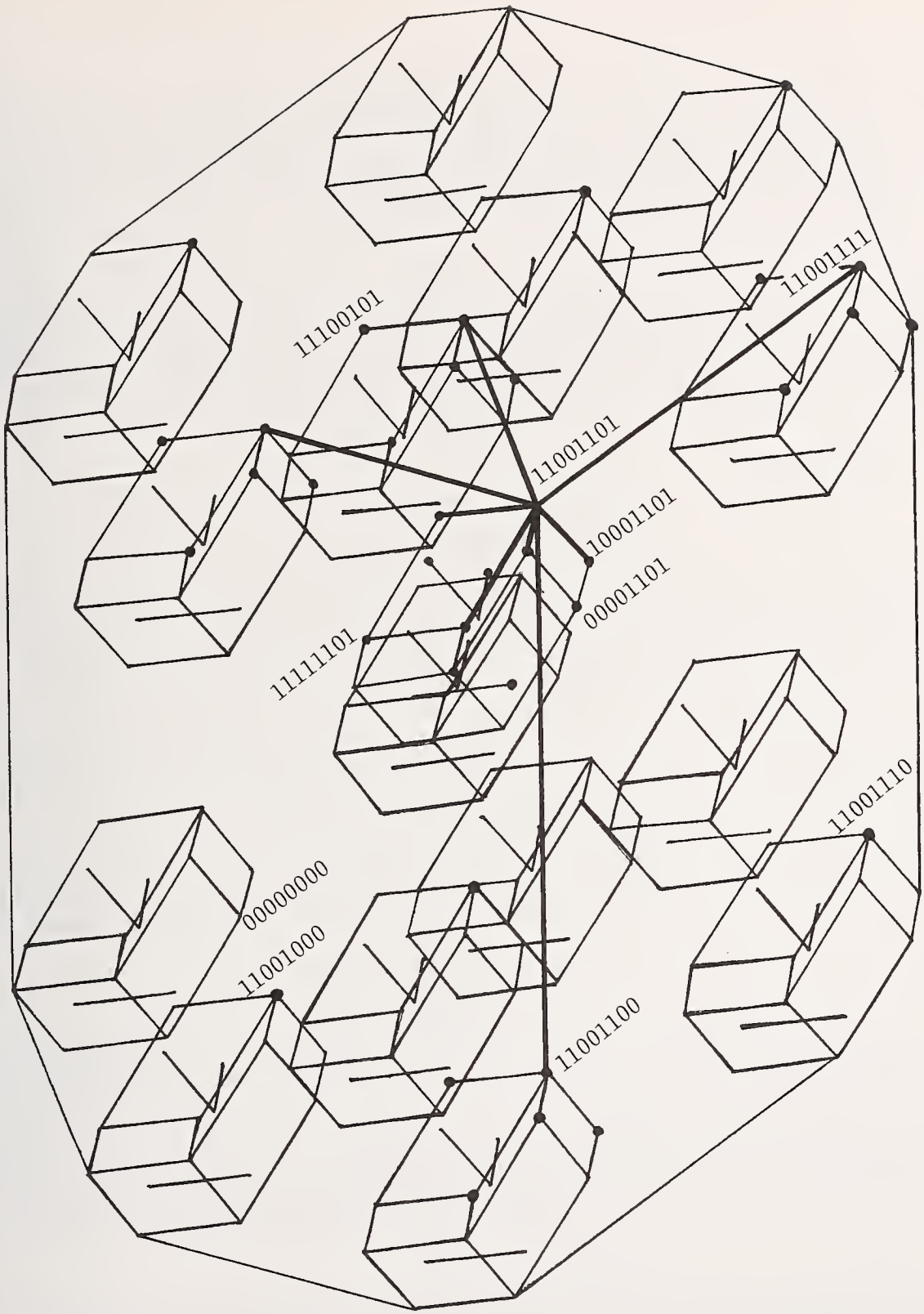


Figure 1: All 256 possible 8-bit patterns as vertexes of a unit 8-cube: near 11001101 are 8 one bit or edge away, 28 two away (•)..

Instead of a binary tree, variants of a query being similar routes up alternate branches, the bit pattern matching explores a framework, corners of a hypercube close to the corner representing the query. Further, instead of checking numerous possible permutations of a query, the bit pattern matching checks all data, measuring how similar it is. This is an approach capable of finding all records most similar however dissimilar they may be to, or distinctive is, the query.

This interpretation is a useful model for considering bit pattern matching. The layman may find it adequate and preferable to think of a bit-pattern just as the "fingerprint" of a record. Particular patterns may be represented by checkerboard-like diagrams with black and white squares shuffled and 60-bit patterns may be defined by 20-digit octal numbers in coding programs or printing out data fields. They can also be represented by code descriptions using one symbol, a letter of the alphabet for instance, for each position in which there is a one. Thus,

```
111000000000011100
ABCdefghijklMNop
```

is ABCMN. (The converse of this is one method of generating a pattern for a field whose value is ABCMN.)

3.3 Interpreting matching. *Counting the population* of ones in a bit-pattern transforms a logical quantity into an arithmetic one. The logical product (bit pattern) and the population count of ones (or "bitsum") are merely different ways of representing the same integer number. The bitsum is a binary integer, the form required for integer arithmetic instructions and quite familiar on all digital computers. The logical product bit pattern represents that integer number not in binary but, like Roman numerals I, II, III, IIII and like dominoes and like dice and like playing cards, by the number of spots (ones in the computer may be called spots in this sense). This is unary or base one and is simpler than binary or decimal representations of numbers. The population instruction or count-ones, as it is called, is thus a base conversion which provides compatibility between boolean instructions preceding and arithmetic instructions following. (It is the inverse of the CDC mask generator instruction, MXi jk, which converts the binary number, jk, into that many one-bits leftmost in the Xi-register with all zeros trailing.)

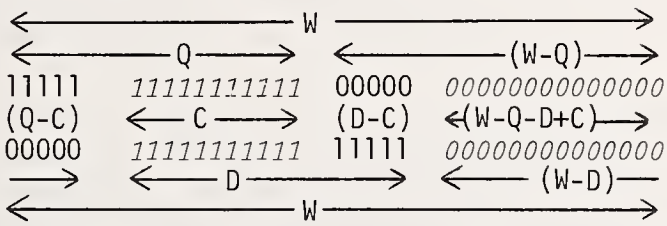
The logical product bit pattern represents why the records are similar. Its bitsum only represents how much similarity there is between them. Bit patterns for the inputs to a similarity formula force pattern matching rather than pattern recognition. Other techniques [11] constrain some of the bits in the pattern and allow some of the others to vary but in this bit pattern matching technique - none are constrained and all may vary. The number of bits differing between two patterns is variable too. The number of possible 60-bit patterns is 10^{18} which is 10^{12} for each of 10^{6} data records in the base. So for every record recorded, there are 10^{12} empty corners nearby in the hypercube (up to 12 edges away). The top similarity value expected is thus 80 percent. It is less work to check the 10^{6} patterns there are than the 10^{12} there might be by techniques such as indexing a direct access structured file.

3.4 Performance. *The logical product*, or "AND", thus gives the unary number of dimensions in truth-space, using the hypercube model, shared by the two bit patterns matched. Since it is a unit hypercube, this number is the Pythagorean sum of squares giving the square of the diameter (longest hyperdiagonal) of the product bit-pattern hypercube. In the hyperspaces used, typically about 40 dimensions, this length is extremely insensitive to departures from orthogonality (independent questions) or from unity (equally important questions) of the hypercube. The orthogonality perturbation is only a cosine variability in each of all dimensions and the non-unity perturbation asymptotically vanishes as high similarity of patterns

reaches identity. [12] This means non-ideal questions work quite successfully in practice.

A similarity value computed from the match result can be defined so that it is independent of variations in the amount of information available in the patterns matched (section 2.3 above). In practice, the factor (50/Q+50/D) can be made constant for 99% of tests and it is the integer result, C, which is tested to discriminate between good and bad matches.

High values are rare and this is the outstanding property that makes this bit pattern matching technique excellent for finding only the fewest best matches from a large file even when the best are not very good. 60-bit patterns, 50% ones, expect over 75% of the pattern to match only once in several thousand matches. Each bit better match is less likely still by a factor of more than an order of magnitude (ten times).



The number of possible different patterns with D ones among W bits is (W! denotes 1*2*..*W)

$$W!/D!(W-D)! \qquad \#3$$

The number of these which have C ones in common with a given pattern of Q ones also among W bits is

$$Q!(W-Q)!/C!(Q-C)!(W-Q-D+C)!(D-C)! \qquad \#4$$

Assuming any pattern with D ones as likely as any other the number of patterns expected for one with C ones common is

$$\frac{W!\ C!\ (Q-C)!\ (D-C)!\ (W-Q-D+C)!}{Q!\ D!\ (W-Q)!\ (W-D)!} \qquad \#5$$

Even when the patterns are not random the ratio of these numbers, or elimination ratio, is a hundred or more for every next common one as the effect of data variations which are still random. (Section 4.1, below, illustrates some numerical values.)

A valuable characteristic is that the more detail involved, the more efficient and selective the matching becomes. This is quite the reverse of the Boolean method which grows in cost exponentially with complexity.

4. BIT PATTERN GENERATION

4.1 Anagram masks.

To catch a butterfly on seeing it
flutter by one sets the set of ones
- it gives the method teeth

A name field and a number field comprise every record, say. The first letter of the name is one of 26 possible different values and is used to set a bit to one leaving 25 zeros in the bit pattern. The last digit of the number similarly sets a one among 10 other bits. We have generated a 36-bit pattern with 2 bits set one. One record in 260 would have a matching pattern if all letters were equally likely. Such a matching record could still differ from the first record in the spelling of the name and of the number.

If a one were set for every letter in the name and every digit in the number, we might get 10 ones from the name and 5 ones from the number. If every different letter and digit were equally likely, how many matches agreeing in 10 or more of the 15 ones might we expect?

90	matches	for	one	pair	with	10	common	ones
700	"	"	"	"	"	11	"	"
9,000	"	"	"	"	"	12	"	"
250,000	"	"	"	"	"	13	"	"
18,000,000	"	"	"	"	"	14	"	"
5,600,000,000	matches	for	identical	patterns	15	ones	in	36 bits

The trend is so obvious that it hardly matters that the figures are not very accurate and the assumptions are over-simplified! In an actual data situation the best match will be suspiciously linkable rather than an improbable chance event, in most cases.

4.2 Attributes interrelate classes. A class field with a numeric code in it denoting one of a number of classes may be better processed to help linkage when misclassification is the problem rather than transposed digits at data capture or mechanical loss of characters during storage or transmission. A group of related classes may be defined by some concept or attribute those classes share and which likely is a factor in correlation, confusion or misclassification within that group. Several such groups may be defined and a class may belong to several such groups. Suppose every class belongs to 3, 4 or 5 of about 16 groups, each group identified by some broad concept. It is a broad concept because it applies to 20-35% of all classes. Let each group concept define a bit to set one. Each class can now set a standard pattern for that class of 3 to 5 ones in a 16-bit component pattern and ones will be common to related classes. One can even interrelate classes from different classifications this way, using the group concepts as a questionnaire, coding a description pattern for every class and indexing the pattern by the class numeric code.

4.3 Combinations of identical field values. If the linkage is just to overcome noise in records then a 2-bit hash value, indexing component patterns 0001, 0010, 0100, or 1000, for every field of 10 to 20 fields can give an adequate bit pattern. [11] But if the linkage is to match records which *paraphrase* each other (an example might be two independent but conflicting patent claims) then it is the meaning rather than the format which must be represented in the bit patterns. The type of editing done to each field to standardize and to compress data is useful. Most values occurring in each field may then be recognized and indexed like classes. Keywords can be recognized and processed similarly. Phonetics can be represented when it is important to do so. Not only is redundancy eliminated the bit pattern may so compress the data that reconstruction of the record from the pattern cannot be done unambiguously.

Other methods of assigning patterns to field values are possible and give good results and are speedily implemented. If records have one or more numeric codes (e.g. parts numbers) of a large number of such codes possible and you want to link a given

combination of codes with the best records this is a useful method: generate a set of bit patterns with, say, 7 ones of 60 bits in each pattern so that each of the 60 bits is a one equally as often as the others are and so that there are as many patterns as codes possible, and assign one pattern to each numeric code. Then logically sum, that is "OR" together, all the numeric code patterns to generate the bit pattern for each record. [13] Then one can find, for example, which widgets best deplete a given inventory of parts. Note that there is no meaningful interrelation between the numeric codes and that linkage is based on identical codes and similar combinations. The number of ones, 7 above, is chosen to give enough variety of patterns to assign to the codes (even restricting them to subsets with minimum coincidence of ones) and to give not too many ones in the records with most codes (an average of 4 codes per record is OK, for 10 codes per record patterns with 4 ones would be better - even 3 ones for the more common codes, weighted less).

4.4 Application exigencies. Corporate names differ from trademarks statistically in the higher incidence of familiar words and word-processing techniques are thus much more useful for the former.

Linkage of a query with only one record, the best, demands better bit pattern definitions. This occurs in bridging applications. Even more demanding is linkage of an exceptionally good pair of records, linking a query rarely. This occurs in trademark surveillance. It depends, of course, on volumes of data processed - is one linking one pair in a thousand, a million or a billion? Most demanding is finding the most unlinkable records in a trillion potential pairs - it may be preferable to try to construct hypothetical data from missing bit patterns characteristic of holes in the data bit pattern hyperspace. False matches are intolerable in all these - precision as well as recall must be perfect by definition.

5. ACKNOWLEDGEMENT

World Searches Inc., Silver Spring, Md. 20910, USA, first developed and still uses trade name searching automated by bit-pattern matching.

6. REFERENCES

[1] D. BLAXELL & JOHN HOWARD "Automated Name Searches - Trademarks and Corporate Names" August 1973, Abidjan World Conference, World Peace through Law Center, Washington, D.C.

[2] D. BLAXELL, 11 articles in Document CTO/CE/I/2 Annex 3 pp. 8-41 English and French versions, Committee of Experts, World Intellectual Property Organization, Geneva, January 1975.

[3] BENNY BRODDA & HANS KARLGREN "Relative Positions of Elements in Linguistic Strings", Statistical Methods in Linguistics 3, p. 49, 1964. Discuss the Trade Name Problem.

[4] MARTIN GARDNER, TOM RANSOM, "Mathematical Games", Scientific American, December 1975, January 1976. The average is most extraordinary.

[5] S. REUBEN, et al, "Application of a Parallel Processing Computer in LACIE" International Conference on Parallel Processing, August 1976, Waldenwoods, Michigan. Adaptive clustering groups vectors similar as close in n-space, no a priori knowledge needed to "prime" the algorithm.

[6] FRITZ NEEB, "Buchstabenkorrelation Methoden zur praktischen Anwendung fur die maschinelle Wortmarkungprufung", March 1970, OGEFA, Austria. Derives formulas for long and short names.

[7] RALPH GRISHMAN, "Assembly Language Programming for the Control Data 6000 Series and the Cyber 70 Series", Algorithmics Press, Box 97, Prince St. Stn., New York, N.Y. 10012, January 1974, pp. 74, 115. Count ones instruction rarely used.

[8] D.E. KNUTH, "The Art of Computer Programming", Vol. III, "Sorting and Searching", Addison-Wesley, 1976 p. 9, ch. 5, ex. 20, answer on p. 576 finding all pairs of 30 bit patterns with 28 or more bits alike from 1000 patterns.

[9] GOSPER, et al, HAKMEM 140, i169, cited in DEC-10 manual. Algorithms to compute bit sum.

[10] Reference [8] p. 405, 6.1 ex. 21, adjacent vertices of n-dimensional cube, relating each to binary number representation. The next exercise 22, answer p. 667, gives a data structure useful in linkage programs.

[11] RIVEST (1971), ARISAWA (1971), Reference [8] p. 563. Combinational hashing, 10 fields contribute 1 one and 2 bits each to 20 bit pattern with 10 ones but use made is only to index the subset with 5 particular specified keys.

[12] VINCENT GEVERS, "The Problem of 'Noise' in Mechanical Research" Compu-Mark (formerly Documentation Gevers), Antwerp, No. 1973, p. 3 the problem of noise is really posed only for marks quoted which happen to be relatively distant.

[13] MOOERS (1971), Reference [8] p. 559, superimposed coding, the number of "false drops" can be statistically controlled, 2 ones among 10 bits. P. 560 Harrison suggests superimposing 49 letter pair patterns of 1 one among 128 bits to find identities in any position in text (see [3]). Such methods were used in 1968 by J.P. Vary (unpublished) commercially without an inverted file.

BIOGRAPHY

David Blaxell won an open Scholarship at Magdalene College, Cambridge, for 1957. He was elected a bye-fellow in 1963 and was awarded an M.A. in 1964 and a Ph.D. in 1967. In his career as a scientist he used Edsac computers and IBM, when he was a post-doctoral fellow at Yale University, for ion-beam electro-optical design and mixed-isotope radiochromatographic decay analyses. His interests in applications of computers in linguistics stems from 1959, in the Monte Carlo technique for hot-atom reaction theory from 1964 and in trademark searching from 1968. He has been working with private and government organizations since then on trade name databases, searching and related applications, using Control Data 3000 then 6000 now Cyber computers.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

A CLINICAL INFORMATION SYSTEM (ACIS) AND ITS APPLICATION TO CLINICAL TRIALS

Michael A. Fox
Departments of Biomathematics, City of Hope Medical Center, Duarte, CA. 91010
and University of California, Los Angeles, CA. 90024

ABSTRACT

With the increase in sophistication of statistical packages, manipulation of raw data prior to analysis has assumed tasks of great complexity.

ACIS is a file generating system in which a compiler accepts a description of the data and structures that are to be applied to this data and generates a series of PL1 programs. These programs are immediately available for use or can be user modified. This is far more powerful than providing the user with the conventional subroutine links.

Key words: ACIS, storage, retrieval, clinical information, file generator, data base, biomedical.

1. INTRODUCTION

ACIS has its antecedence in a data base system which currently maintains information on approximately 20,000 patients at the City of Hope Medical Center. Subsequent development of the system has been motivated by its use in clinical trials and other areas of biomedical research.

The system is a compiler designed to generate custom programs for a data base using a file description language composed of very simple elements provided by the user. The first generated program actually builds the data base and the second is used for retrieval.

The following description is couched in biomedical terms for it is in this field that the system has been used.

2. STRUCTURING THE DATA BASE

Although many of the procedures performed in a hospital eventually find expression in the patients' charts, these recorded sagas in many cases serve to entomb information rather than preserve it. It was with the conviction that a data base should be evolutionary not merely historical that a design was produced and implemented that not only would have a diverse appearance to different users but would permit growth and allow restructuring.

A physician is concerned with individual patients, while a chemist is making measurements on a series of bloods, and an admissions clerk is concerned with bed occupancy. The data pertinent for all of these facets of hospital activity can each be described independently in a rectangular or tabular manner. Different lines from each of these tableaux are, however, related, and even when one considers the individual patient the essential rectangular nature of the data is, though obscured, not lost. This follows from the axiom that all data can be written in a series of rectangular arrays with repeated groups as separate sub-rectangles. Fig. 1 depicts a series of such arrays covering some of the recordings that are ubiquitous to hospitals. These arrays will be referred to as sets, each row of which is then a member, and elements within a row are the items or tuples. Describing some of these sets, there is the general index of patients or the set P, which contains the chart number, name and certain demographic information. The set L shows the location of the patients within the hospital and parenthetically contains the occupancy. The set B produced by the technician drawing blood contains the patient chart numbers, acquisition numbers applied to the blood samples, together with the dates and times the bloods were drawn. The clinical chemist reports the results of his analysis as the set R, while the surgery performed on a patient appears as an element in set O.

Although each of these sets has utility in isolation, such use is generally of transient value. It is only when the relations that are inherent in the data are brought together that the power of the data base becomes evident.

Consider the members of each of these sets that are attributed to a particular patient, γ. Since the patient is unique there will be one member or row from the set P, i.e.$P(\gamma)$. There may be several different hospital stays for this patient $L(\gamma_1)$, $L(\gamma_2)$ ... and during each of these stays several bloods may be drawn $B(\gamma_{11})$, $B(\gamma_{12})$) ... $B(\gamma_{21})$, $B(\gamma_{22})$ ... Further, for each of these bloods a variety of tests may be performed producing results $R(\gamma_{111})$, $R(\gamma_{112})$ ... $R(\gamma_{211})$, $R(\gamma_{212})$ ... The notation is to add a suffix for each additional level of data, viewed in a hierarchical sense.

The representation of a patient by a hierarchical tree of strings is shown in Fig.2. This figure also shows links to operations, diagnoses and so forth. Additional primary sets, for example, services, can immediately be incorporated into this schema.

As information on individual patients is often required it was decided to maintain the structural form implied by Fig. 2 using embedded pointers rather than re-creating the structure implied by Fig. 2 from Fig. 1 each time these structures were required. There is, of course, no reason why all or part of Fig. 1 may not be maintained independently of Fig. 2. Moreover the various sets from which Fig. 2 is obtained do not necessarily come from the same physical file, neither are the lines or elements of these sets of the same length. A means of storing and retrieving variable length strings from different physical files is thus essential. This task is provided by a core management sub-system which will be described later.

Examination of any of the rows in any of the sets in Fig. 1 reveals that the items (tuples) can be considered as keys and/or descriptors or modifiers to these keys. For example, in an operation the operation can be one key, the surgeon a second key, whilst the date and anesthesiologist are descriptors to these keys. It is desirable to reference a series of patients who have in common either a separate key or multiple keys; thus further sets of structures are required. In these cases the structures are inverted or reference files. Here each inverted key is a member of a category of keys, e.g., a particular operation is a member of the sets of operations, and to this key is associated a set of patients which may be represented either by a list of patient pointers or by their individual chart numbers.

Three distinct types of data are being considered: the actual data itself, as shown in Fig. 1; those internal linkages of the data as implied in Fig. 2, which can if pointers or

reference numbers are used , be internally maintained as part of the data structure, and thirdly a structure which is external to the main body of information and can be maintained as a separate entity.

There are many instances where the information maintained on a patient exists, by design, in different types of files. He may be recorded as an inpatient, as an outpatient, as a patient who is part of a particular drug study proticol, or he may exist only as a blood or tissue sample sent to the institution for study. By creating a master file which upon reference gives the particular types of files in which this person is to be found an effective method for adding totally new types of files is available. Further, files of limited use can be maintained and deleted with no effect on the other files. This is essential in a research environment where many unusual tests may be performed on a select group of people, and the files holding this information, though linked to other information on the patient, have to be segregated from the main body of information.

3. STORAGE AND RETRIEVAL OF DATA

Examination of Fig. 2 reveals the enactment of three main operations, obtaining a free record from an appropriate free chain of records to hold the data items, linking this record to a chain of like records and in some cases providing a pointer to a chain lower in the hierarchy. Each of these functions is highly generalized and the specivic coding and call sequences required for these operations are generated by the compiler when the system is generated. The specific programming code to call the appropriate routines is generated by providing the compiler with the following information: the types of files that the input forms imply, whether forward or direct, inpatient or outpatient, the elements of the data that are to be associated with forward and inverted files.

Although there is a fundamental dichotomy between the data elements, which are stored, and the structures applied to them, which are also stored, the mechanism of storing and retrieving is universal to these two distinct quantities. It is useful to introduce the notion of internal and external structures to distinguish them. Thus the implied structure in Figs. 1 and 2 can be thought of as internal, while structures which reference the data in special ways, say inverted files, are defined as external structures. These external files are in general constructed to satisfy specific uses for the data. By manipulating external structures which only reference data, relations can be generated, which may be of temporary or permanent interest. Such relations will themselves be subject to an external storage class. Particular external files then provide candidates for analysis. For example the extraction of patients to provide statistically matched sets for drug protocol testing is not only facilitated by storing inverted files but by maintaing counts within them, the existence of comparable patients can be ascertained without file searches.

The core management sub-system accepts or provides variable length character strings and either writes these to or takes them from particular locations on records and it is this block which is moved to and from a variable number of buffers within the core and the appropriate direct access devices. During a run, blocks of records are transferred to these in core buffers as required. Hash tables provide the particular buffer and location in it of the string of interest. When all the buffers are filled an algorithm is used to determine the buffer which contains the records that are least expected to be used, and if any of these records have been updated, they are written to an external auxiliary file instead of their home files to prevent destruction of dynamic pointers in the case of machine failure. After a predetermined number of transactions have been made, a fail safe condition is involved, in which a reference matrix that maps from the auxiliary or working files to the main files is transferred to another offline device. Using this mapping matrix the main or permanent files can then be updated. Should machine malfunction occur during this period of transformation, the information being transferred and the mapping matrix are protected as they reside on offline devices. At the end of the fail safe period when all transfers have been made, processing can continue.

4. EXPERIENCE WITH THE SYSTEM

Apart from maintaining information on current patients and being a pool of knowledge for research purposes, the data base has a further important role. Increasing requirements are placed by both insurance and governmental agencies on hospitals for information that describes the activities within a hospital and to document variances from expected standards of care. Specific external structures have therefore been included to facilitate the production of these reports. They include summary statistics of length of stay by diagnosis and operation with detailed percentiles for specific age groups. Specialized report generators for these and similar reports have been found to be more economical to produce and run than a single more generalized generator.

An interactive retrieval program may also be generated for the data base. This program presents the user with information on the state of the file, consisting of the names of the inverse categories and the amount of information in each of them. A menu of options is offered together with the choice to limit the amount of data viewed on the interactive device and then to direct the complete set to either a hard copy medium or a pre-assigned data set. Thus the viewer is not subjected to a lengthy list of extractions he may not wish to examine, but rather a brief overview to decide if the complete set is worthy of further analysis.

This has been found useful for research purposes to ascertain whether enough patients are available with specific qualities, and if not, relatively close groups can then be combined to provide adequate numbers for comparable analysis. Since the generated retrieval program is written in PL1, code can be added to provide calculations to be performed by command on interactively determined groups of data.

An interactive system is only acceptable when the user is able to obtain enough information at a session without being subjected to a surfeit of irrelevant information.

5. OUTPUT OF STATISTICAL PACKAGES

Data held within an information system may differ in important ways from data presented to statistical packages. The latter often requires categorized information to be numerical: for example, 'male' is coded as 2, while the former excels in intelligibility when English is used. Recoding is then a required task.

Statistical packages are often oriented to accept case-wise information but because of the nature of some data a case can consist of a block of unrelated repeating groups; for example, several diagnosis and many laboratory findings. The program currently available allows selection of variable groups, and particular variables for these groups may be displayed and/or passed to an output medium. Development is on hand for collapsing data to case wise format, by a set of suitable commands, (means, etc. over specified variables). This will then be directly available to statistical packages.

All the programming for this system has been written in PL1, with much use of the preprocessor features of the language to write the expanding compiler sections. Care has been exercised in the design of the system to allow current development in inquiry languages to be acceptable adjuncts to the system.

6. THE LANGUAGE

The language has the following syntactic form:

```
VARIABLE_GROUP_NAME : COMMAND (    ) : COMMAND (    ) :.....::
```

The following is the File Definition Language that was used in an actual study. This linguistic description is composed of both definitions of data fields and the structures that are to be applied to the data. The command CONSISTS_OF describes the data elements, the fields they occupy, packing required and whether inversions are to be performed. The command CONTAINS structures the variable groups hierarchically. Many commands are implemented which do not appear in this example.

```
PROFILE:  CONSISTS_OF
    (FILE NO (1,4,N3), DATE (5,D), NAME (11,9), INIT (20,1), LAST (21,11),
     ADDRESS (32,21), CITY (53,10), STATE (63,2), ZIP (65,5,N3), TEL_AREA (70,7,N3),
     AGE *82,2,N1), SEX (84,1), RACE (85,1), MARITAL_STATUS (86,11) ):
CONTAINS:
     (MEASURES, PROBLEM TEST): KEY (FILE_NO) : FILE_POSITION (81):
     FILE (C): TYPE POSITION (80): TYPE (D): EXTERNAL LENGTH (90)::
MEASURES:  CONSIST_OF
     (HEIGHT (10,2,N1), WEIGHT (12,3,N2), OPTIMAL (15,3,N2), B_P_ARM_S (18,3,N1).
     B_P_ARM_D (21,3,N1), B_P_LEG_S (25,3,N1), B_P_LEG_S(28,3,N1), SMOKER (24,1,1,),
     CHOLESTEROL (31,3,1), URIC_ACID (34,4,1), PBI (38,4,), BUN (42,3), T4 (45,4),
     HB (49,4), GLUCOSE (53,3), WBC (56,5), URINALYSIS (61,1) )::       etc.
```

A recreation (for sake of typographic clarity) with annotations of a retrieval session using the data base generated from this language follows:

A particular patient in this data base has the logical structure

Profile/Demog

Measures Problem text

Encoded problems

```
THIS DATA BASE CONTAINS THE FOLLOWING VARIABLE GROUPS
PROFILE
MEASURES
PROBLEMS
PROB_TEXT
WHEN YOU ARE ASKED WHETHER YOU WISH TO DISPLAY A PARTICULAR VARIABLE THE FIRST TEN LETTERS
OF THAT VARIABLE WILL BE SHOWN. IF YOUR ANSWER TO THE QUESTION IS "ALL" ALL VARIABLES
PREVIOUSLY SELECTED WILL BE DISPLAYED. THE INITIAL DEFAULT IS ALL THE VARIABLES. BEFORE
ANSWERING A QUESTION WAITH FOR THE SYMBOL " " N.B. CR IS SYNONYMOUS WITH NO.
DO YOU WISH TO BROWSE THROUGH A CATEGORY? (PRINTS SORTED MEMBER CODES AND NO OF REFS.) Y/N
  Y.  THE FOLLOWING CATEGORIES ARE AVAILABLE:
1  MEASURES_B_P_ARM
2  MEASURES_SMOKER
3  MEASURES_CHOLESTEROL
4  MEASURES_URIC_ACID
5  PROBLEMS_T_FIELD
6  PROBLEMS_M_FIELD
7  PROBLEMS-E_FIELD
8  PROBLEMS_F_FIELD
9  PROBLEMS_P_FIELD
10  PROBLEMS_D_FIELD
WHICH CATEGORY DO YOU WISH TO EXAMINE? ANSWER WITH CATEGORY NO. TO EXIT PROGRAM ENTER Z.
TO RETURN FOR OTHER TYPES OF RETRIEVALS ENTER R.
 5
                                                    category 5 the topological sites
TXX000   2                                          has been selected for display
        XX      EYE AND EYE APPENDAGES
        EYE, NOS
        EYEBALL
TXX200   1
        CORNEA, NOS
TX0000   2
```

```
IF YOU WISH TO TAKE PRINT DEFAULTS ENTER Y
DO YOU WISH TO SEE PEOPLE ON THE TERMINAL?  ANSWER WITH Y/N
>  Y   ENTER MAXIMUM NO OF PEOPLE YOU WISH TO SEE DISPLAYED    5
THE FOLLOWING PRINT OPTIONS ARE AVAILABLE:  A) WRITE TO TERMINAL ONLY, B) WRITE TO PRINTER
ONLY, C) WRITE TO BOTH.  ENTER PRINT OPTION
>  C
AFTER PRINTING 5 SETS OF RECORDS ENTER PRINT OPTION AS ABOVE OR ENTER D FOR NO FURTHER
PRINTOUT    D.
LINES OF PRINT CORRESPONDING TO THE FOLLOWING ACRONYM ARE AVAILABLE
D M P T  ENTER WORD CONSTRUCTED FROM THESE LETTERS
>  DT
                     FOR THE VARIABLE GROUPS KNOWN          USING THE PROMPT MODE DECISIONS
                     TO THIS USER DEMOGRAPHY (PROFILE)      ON HOW MUCH TO PRINT ARE BEING
                     AND TEXT ARE TO BE DISPLAYED.          MADE. THESE,TOGETHER WITH THE
                                                            SELECTION MADE UNDER, WILL,FOR
                                                            THE RUN,BECOME THE DEFAULT.


FROM "PROFILE" IF YOU WANT THE VARIABLE PRINTED THEN ANSWER Y/N/P/All
FILE_NO    P
DATE  NAME  P  INIT   LAST        FROM THIS LIST THE VARIABLES FOR PROFILE/DEMOG ARE
                                                 TO BE DISPLAYED

REQUESTE D  VARIABLES FOR PROFILE      1 FILE_NO    3 NAME   12 AGE
                   13 SEX     14 RACE      15 MARITAL_ST
FROM "PROB_TEXT" IF YOU WANT THE VARIABLE PRINTED THEN ANSWER Y/N/P/ALL
NUMBER     P    Text  P        THE VARIABLES ARE SELECTED TO BE BOTH DISPLAYED
                              AND WRITTEN TO ON OUTPUT DATASET.

REQUESTED VARIABLES FOR PROB_TEXT   1 NUMBER  2  TEXT
THE FOLLOWING GROUP OF RETRIEVALS ARE AVAILABLE  A) INDIVIDUAL PATIENTS, B) INDIVIDUAL
CODES OR BOOLEAN EXPRESSIONS, C) ALL MEMBERS OF A CATEGORY, Z) QUIT.
PLEASE ENTER YOUR CHOICE OF RETRIEVAL WITH A/B/C/Z        WE ARE READY TO RETRIEVE NOW
>  B
DO YOU WISH TO RETRIEVE A SINGLE CODE OR THE INTERSECTION OF MULTIPLE CODES? ANSWER MUL/SIN
 MUL
WHEN REQUESTED SUPPLY  EITHER RETRIEVAL CODES OR "LOGICAL COMMANDS".  WHEN YOU HAVE FINISHED
ENTER "ALL".  ENTER FIRST OF CODES TO BE "ANDED".
> BP1                                              This is the code for normal blood press-
   CODE REQUESTED IS BP1                           ure
ENTER NEXT CODE OR COMMAND "OR/NOT/ALL"
> f7170.  CODE REQUESTED IS f7170.                 This is the code for hypertension
ENTER NEXT CODE OR COMMAND "OR/NOT/ALL
> OR.  ENTER CODE   f7640                          This is the code for heart murmur
  CODE REQUESTED IS f7460
ENTER NEXT CODE OR COMMAND "AND/NOT/ALL"
>  NOT
ENTER CODE  D 2350                                 This is the code for diabetes mellitus
  CODE REQUESTED IS D2350
ENTER NEXT CODE OR COMMAND "ALL"
> ALL

PROFILE      1712 JUNE      29 F C M
PROB_TEXT   01  OBESITY
PROB_TEXT   06  ANEMIA (PROBABLE IRON DEFICIENCY)  This patient is a 29 year old Caucasian
PROB_TEXT   05  HYPERTENSION                       married female and these are her medical
PROB_TEXT   04  VARICOSE VEINS                     problems
PROB_TEXT   03 BACKACHES
PROB_TEXT   02  CHOLECYSTECTOMY
```

```
ROFILE          748     LETA        38  F C M
ROB_TEXT    03  HYPERTENSION
ROB_TEXT    02  ANXIETY REACTION
ROB_TEXT    01  OBESITY

ROFILE          2663    HELEN       51  F N M
ROB_TEXT    05  DEPRESSIVE REACTION
ROB_TEXT    04  MUCOUS COLITIS
ROB_TEXT    03  BACKACHES
ROB_TEXT    02  EXOG OBESITY
ROB_TEXT    01  HYPERTENSION

ROFILE          2000    HELEN       37  F C M
ROB_TEXT    04  ANXIETY REACTION
ROB_TEXT    03  HYPERTENSION
ROB_TEXT    02  CHOLECYSTECTOMY
ROB_TEXT    01  OBESITY

ROFILE          3316    EDNA        53  F N D
ROB_TEXT    04  DEPRESSIVE REACTION
ROB_TEXT    01  RHEUMATIC FEVER AS A CHILD
ROB_TEXT    04  ANXIETY REACTION
ROB_TEXT    03  ARTHRITIS
ROB_TEXT    02  HYPERTENSION
ROB_TEXT    01  HEART MURMUR
ROB_TEXT    07  EXOG OBESITY
ROB_TEXT    06  BACKACHES
ROB_TEXT    05  PARTIAL HYSTERECTOMY
  NUMBER OF INTERSECTIONS = 5
DO YOU WISH A FURTHER BREAKDOWN OF THIS GROUP. ANSWER Y/N
ENTER "R" TO RETURN TO OTHER RETRIEVAL MODES
"Z" TO EXIT
"C" TO CONTINUE IN THE SAME MODE. "D" TO DECODE.
> D
INSERT CODE TO BE DECODED OR ALL
>f 7460
        746-747 HEART MURMURS AND ABNORMAL SOUNDS
        SYSTOLIC MURMUR, NOS
INSERT CODE TO BE DECODED OR ALL
> D2350
        235-237 ENDOCRINE DISEASES OF THE PANCREAS
        DIABETES MELLITUS NOS
INSERT CODE TO BE DECODED OR ALL
> ALL
ENTER "R" TO RETURN TO OTHER RETRIEVAL MODES
      "Z" TO EXIT
      "C" TO CONTINUE IN SAME MODE. "D" TO DECODE
>R.
IF YOU WISH TO TAKE PRINT DEFAULTS ENTER Y.

DO YOU WISH TO SEE PEOPLE ON THE TERMINAL?
ANSWER WITH Y/N.
```

These patients constitute the set

BP1 ∩ [F 7170 ∪ F 7460] ∩ D2350

that is

Note Diabetics who suffer from hypertension or heart murmur but manifestic normal blood pressure (on medication).

There were 5 such people

We now decode into English some of the encoded information.

Retrieval continues

Michael A. Fox received a Ph.D. in theoretical physics from London University in 196[illegible] and subsequently received a Postdoctoral Fellowship in Mathematical Biology from the USPHS, to study at the University of Chicago in 1965. He subsequently studied both computer science and statistics. He is currently director of the Biomathematics Department , City of Hope Medical Center, Duarte, California, and has an appointment in the Biomathematics Department, UCLA, Calif. His field of interest is biomedical information and the statistical analysis of biomedical data, including clinical trials and epidemiological studies.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

RELATIONAL DATABASE MODELS AND SOCIAL SCIENCE COMPUTING

ROBERT F. TEITEL
The Urban Institute
2100 M Street, NW
Washington, DC 20037

ABSTRACT

This paper discusses the applicability of the Relational Model of Data to the data collections in common use today in social science statistical computing. It reviews the structure of the commonly used data collections, presents the basic concepts of the Relational Model of Data, and applies the relational model to the description of contemporary social science data collections. The paper continues with a discussion of high level language concepts for use in social science statistical systems for large and complex data collections. The final section discusses the difference in the patterns of access to data by query and by statistical applications and the implementation implications.

Keywords: Complex data structures; database systems; high level language; relational model; social science computing; statistical systems.

(Opinions expressed herein are those of the author and do not necessarily represent the views of The Urban Institute or its sponsors.)

1. INTRODUCTION

During the past few years, E. F. Codd [1970, 1971a] and others [Date (1971, 1975), Heath (1971), Tsichritzis (1974), Date (1974)] have proposed a Relational Model as a user view of large stored data bases. A number of high level query languages for management information and bibliographic retrieval applications of relational data bases have been proposed [Boyce (1975), Chamberlin (1974), Zloof (1975), Codd (1971a)].

This paper discusses the applicability of the Relational Model to the data collections in common use today in social science statistical computing, and represents an updated report of a continuing study. [Teitel (1975, 1976)]. In the following section, it reviews the structure of the commonly used data collections and defines some of the terminology. The third section introduces the basic concepts of the Relational Model of Data and applies the Relational Model to the description of contemporary social science data collections. After demonstrating that the Relational Model is quite adequate for the description of social science data collections, the fourth section presents some high level language concepts for use in future social science statistical applications of relational data bases containing large and complex social science data collections. The final section discusses the difference in the patterns of access to data by query applications and by statistical applications and the resulting implementation implications.

2. SOCIAL SCIENCE DATA STRUCTURES

Surely we are all familiar with the earliest and most primitive of data structures--the matrix. It consists of a fixed maximum number of elementary data items arranged in rows (also called cases, observations, or "entities"), and columns (also called variables, or "attributes"). Several of the early social science statistical systems designed for this data structure are still in use today [BMD (1973), OMNITAB (1971)]. Next in evolution, and closely related, are the two rectangular structures. Rectangular structures, Figure 1, are simply matrix structures with a relaxation of the requirement that the total number of data elements be less than some fixed maximum.

With a limited number of columns and an "infinite" number of rows we call the structure "horizontal rectangular"--the traditional data model for observational or survey data collections; with a limited number of rows and an "infinite" number of columns we call it "vertical rectangular"--the traditional model of econometric time series. Most of the well known social science statistical systems process horizontal rectangular data [SPSS (1975), OSIRIS (1973), PSTAT (1975)]. Systems have also been implemented to process vertical rectangular structures [PLANETS (1975)]. So far we have described what are sometimes called "flat file" structures: no repeating groups, no multiple valued attributes, no nested segments, no longitudinal components.

The exclusions from flat file structures immediately suggest the next level of complexity of the data collections used in social science computing: Longitudinal and Hierarchical (also called "Tree Structure," and sometimes "nested", though that usually denotes a simple one directional hierarchy).

A longitudinal structure, Figure 2, is simply a "horizontal rectangular" with a time component or a "vertical rectangular" with a cross-sectional component. It can also be viewed as multiple attributes, each with both a temporal and cross-sectional dimension. Geometricians would remind us that it is a rectangular solid with orthogonal axis--attribute, temporal and cross-sectional--which we have simply sliced to present the forms above. The (Michigan) Panel Study on Income Dynamics-Family data collection [MPSID (1977)] is an example of a longitudinal file widely used in socio-economic research, containing

ırrently 8 years of data on about 5,000 households.

To define hierarchical structure we need first to define the concept of "segment." ıformally, a segment is a collection of attributes which are related in some physical or ·ganizational manner. For example, the National Travel Survey of 1972 [NTS (1972)] ɔnsists of 4 segments containing attributes of each HOUSEHOLD interviewed; of each PERSON ı the household; of each VEHICLE owned by the household; and of each TRIP taken by each ɛrson. The ierarchic structure for the NTS is shown in Figure 3 along with some sample occurrences. ɔte that this structure has two different segment types, VEHICLE and PERSON, at this same ɛgment level. Perhaps more common--in part because we cannot readily handle structures ɹch as the NTS with current statistical software--are the more limited cases of hier-rchical structures such as the Public Use Samples of the 1970 Census [PUS (1972)] shown in igure 4. The distributed PUS has three segments, NEIGHBORHOOD, HOUSEHOLD and PERSON, each n a different segment level (actual socio-economic research is frequently done at the family" level, but that is another matter). Several systems are available which allow ome limited processing of the PUS or similar hierachical data collection [TPL (1975), ENTS-AID (1976), SOS (1974), CENSTAT (1975)].

Finally, we have seen the creation and distribution of data collections which are both ongitudinal and hierarchical. The Panel Study on Income Dynamic-Person data collection ctually consists of two segments, HOUSEHOLD and PERSON, each of which contain attributes or, at this date, 8 years. Similarly, a data collection created from matched enumeration nits from 10 successive waves of the Current Population Survey [CPS (1977)] consists of wo segments, FAMILY and PERSON, containing attributes for 10 years.

3. THE RELATIONAL MODEL OF DATA

The following is but a superficial overview of the Relational Model of Data. The eader interested in further material is referred to the citations, especially Date (1975).

The Relational Model is rooted in the mathematical theory of relations and is pre-ented in set theoretic terminology. Once understood, however, the Relational Model ermits elegantly simple descriptions of complex data relationships. Given a number of ets (collections of possible data values), S1, S2, ..., Sn, a Relation, R, is a set of rdered "n-tuples": <s-1, s-2, ..., s-n> where s-1 is an element of a Set S1, s-2 an lement of a Set S2, ..., element s-n an element of a Set Sn. The sets, S1, S2, ..., Sn, eed not be distinct and are called the domains of the Relation R. The degree of Relation is 'n'--simply the number of domains in the Relation. Figure 5 illustrates the Relation ERSON, consisting of the domains ID#, AGE, SEX, RACE and INCOME, as a table with the lomains as the column headings and the occurrences or "n-tuples" as the rows.

A number of other properties, including the important concept of normalization [Codd 1971b)], to be satisified by relations need not concern us for the moment, except to note hat the order of the occurrences of the relation (i.e., "rows" of the table) must be nterchangeable and that each occurrence must be unique. Both are easily satisfied in ractice by including an identification (primary key) domain. Furthermore, if we assign nique names to each domain, we can ignore column order.

In essence, then, a relation is what social science statistical researchers would call a "flat file" consisting of simple data elements: no repeating groups, no multiple valued attributes, no nested segments, no longitudinal component.

Let us consider a Relational Model of a typical two segment, FAMILY and PERSON, data collection. Figure 6 illustrates the two separate and independent relations involved, FAMILY and PERSON; each could be stored, accessed and processed completely independently of the other. The identification domains insure that ordering is not imposed on the occurrences in either the FAMILY or PERSON relation--yet all information needed to relate a person to its household or a household to its persons is present..

One immediate advantage of the Relational Model description of a two segment FAMILY and PERSON data collection such as that shown in FIGURE 6 should be apparent: the model encourages each description of analysis of the domain values contained in, say, the PERSON relation without making any reference to the existance of a FAMILY relation.

Figure 7 illustrates a Relational Model description of the National Travel Survey. Each "segment" is described as an independent relation with appropriate identification domains as the primary key. It is again apparent that analysis may be made of TRIP occurrences, for example, without reference to any of the other segments.

A relation as described above is identical to the concept of segment introduced in Section 2. Since the latter is a more common term in social science computing it will be used in the rest of this paper interchangeably with relation.

We now turn our attention to some possible language concepts which may be the basis of a non-programmer, research user, high level language for a statistical database system based on the Relational Model.

4. LANGUAGE CONCEPTS FOR DATABASE USERS

A number of query languages for use with relation database systems have been proposed [Boyce (1975), Chamberlin (1974), Zloof (1975), Codd (1971a)]. Though there exists considerable overlap between the basic functions performed with a management information or a bibliographic system and those performed with a social science statistical system, there are some basic differences. The following paragraphs address some of the functions which appear to be necessary in a user language for a relational data base system oriented to statistical processing. It is difficult to discuss language function without language forms. Hence, examples will be in a command oriented language, similar in syntax to the command language of a modern operating system (CSTS (1975)].

For our purposes, we will define a database to be all independent segments of all data collections available to an individual or organization. Figure 8 depicts one possible database.

Usually the first activity of a researcher who is to perform some statistical operation on a database is the specification of the "unit of analysis" and the sampling criteria--which frequently is "all"--and the time period to be considered, if the data is longitudinal. We have seen how the Relational Model of, for example, the National Travel Survey, easily permits the specification of "unit of analysis"--which may be the trips, persons, vehicles or households contained in the TRIP, PERSON, VEHICLE or HOUSEHOLD segments, respectively.[1]

---

1. The relationship between units of anlysis and segments leads to the following conjecture: If a data collection is described by a set of 3rd Normal Form Relations, then those Relations represent the only possible units of analysis within that data collection.

The initial user "unit of analysis" specification consists of the segment or relation o be the unit of analysis, the sampling criteria and the time periods. Figure 9 contains everal sample USE statements which illustrate these functions.

Implicit in the specification of the unit of analysis is that every occurrence is ubject to the subsequent analysis requests. Alternatively stated, there is an assumption f iteration: each analysis request uses attributes from every occurrence of the defined opulation. This does not preclude the analysis request from restricting the population urther by means of a filter, or selection or rejection criteria. The very process of letermining the outcome of the filter will involve using attributes from every occurrence.

A fundamental function of a new database system will be the creation of data files rocessable by the many available statistical systems. Figure 10 illustrates a trivial xample of such an EXPORT function, together with an arithmetic transformation statement. full complement of arithmetic, logical, and functional transformations would be available n an actual system (including bracketing, recoding or category creation, dummy variable eneration, index scale creation and similar operations common in social science computing).

Data transformation would not be restricted to attributes of the segment specified as he unit of analysis. To use attributes in the specified unit of analysis, use of the name f the attribute should be sufficient for its full specification; for those attributes in ther relations, the relation name will be necessary. We will here use "@" to mean "of" so hat SEX@PERSON refers to the domain SEX of the PERSON relation. With the ability to pecify a segment as the unit of analysis and to attach other setments, simple inter-egment expressions may be constructed as in Figure 11.

Alternative forms of inter-segment expression have been proposed, usually employing eserved word operators [Kidd (1969), Mesnage (1972)]. One such form [Mesnage (1972)], is lso shown in Figure 11 and succeeding figures.

The use of the possessive operator "@" or "OF" is sufficient for one-to-one associa-tions between occurrences of the unit of analysis and those of other segments. That is, with reference to the National Travel Survey, if the unit of analysis is PERSON, there is only one HOUSEHOLD occurrence for any given PERSON occurrence. There exists, however, a one-to-many association between occurrences of PERSON and occurrences of TRIP ("many" actually means "varying" and includes zero and one). To utilize data from the "many" occurrences in the analysis, some form of reduction function [APL (1970)] is necessary. Among the more obvious reduction functions are a COUNT of the number of occurrences and the SUM, MAXIMUM, and MEAN of a domain expression. Reduction functions consist of three components: the expression whose values are to be reduced, the method of reduction, and the scope of the function. The latter consists of the segment name--which delimits the number of eligible occurrences--and the selection criteria--which selects occurrences from those eligible. Reduction functions may be nested. Figure 12 shows some sample inter-segment expressions using reduction functions.

If conditional choice is added to the operations permissible in expressions, the result is a very powerful descriptive capability for social science computing. Figure 13 contains several examples of the descriptive power of expressions containing conditional choice and reduction functions for the specification of data tranformations across segments.

5. QUERY AND STATISTICAL APPLICATIONS

There are significant differences between the design criteria of a database system for social science statistical research and those for management information or bibliographic

retrieval systems. The differences are not principally in the interrelationships of the data elements or logical data structure, but rather in the patterns of access to the data and in the operations to be performed on the data.

Somewhat oversimplified, the access pattern and operation performed with a management information system or a bibliographic retrieval system is to search for a particular occurrence (case, observation, etc.) in the database which satisfies a given condition and to display full information (all attribute or domain values) about that one occurrence. For example, "what widgets do we buy from ABC manufacturing?" would be a typical query of a management information system.

Similarly oversimplified, the access pattern and operation to be performed with a social science statistical system is to retrieve and manipulate very little information (few attribute or domain values) from every occurrence in a segment. Examples of such requests might be "display descriptive statistics, mean, variance, etc., of income and education," or "cross-tabulate education with race".

Figure 14 illustrates the target information of a typical query of a management information system and the target information of a research request of a social science statistical system.

The data access pattern of social science statistical requests suggest implementation strategies considerably different than those employed for management information systems, if efficient performance is to be realized. In an earlier paper [Teitel (1975)] the author proposed a detailed design for a relational database system including a procedural language (FORTRAN) interface. The design rests upon a substantial elaboration of the concepts and implications of transposed (not inverted) files used successfully in several single segment or flat files systems [PICKLE (1974), IMPRESS (1972), PLANETS (1975)]. Statistics Canada has placed their entire 1971 Population Census on-line using a specialized, more primitive form of such a data structure, and they have exploited it quite successfully with a geocode-based table generating system [Sandee (1976)]. Many aspects of the proposed design are currently being extensively revised and will be the topic of a subsequent paper.

6. SUMMARY

This paper has reviewed the structure of data collections used in contemporary social science statistical research, presented a very brief summary of the Relational Model of Data and applied that model to the description of social science data collections. The Relational Model appears to be a useful model for the description of social science data collections. Several language concepts have been presented which create a powerful descriptive capability for social science statistical researchers. And, finally, we have argued that the data access patterns of social science statistical research are different than those typically found in management information or bibliographic retrieval applications and have briefly discussed the implementation implications.

7. FIGURES

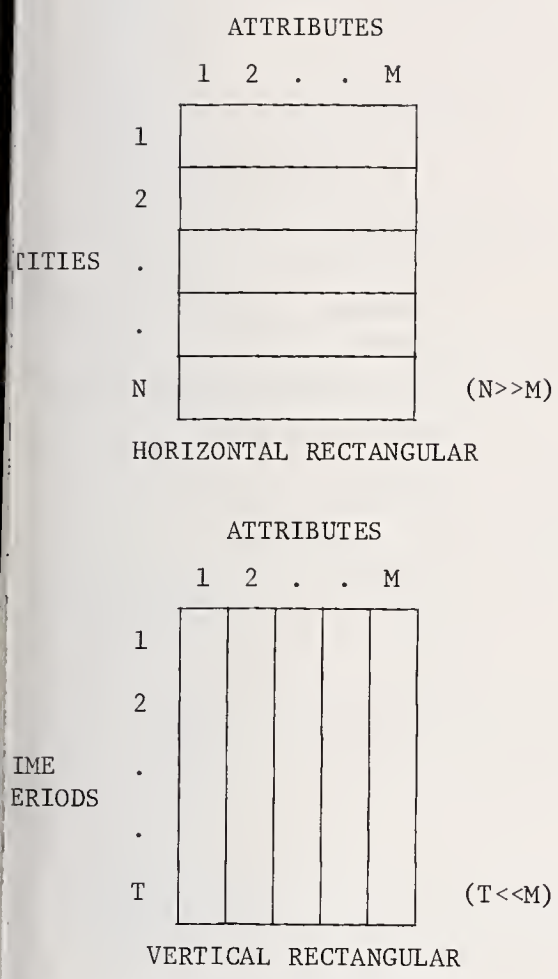


Figure 1: Rectangular Data Structures

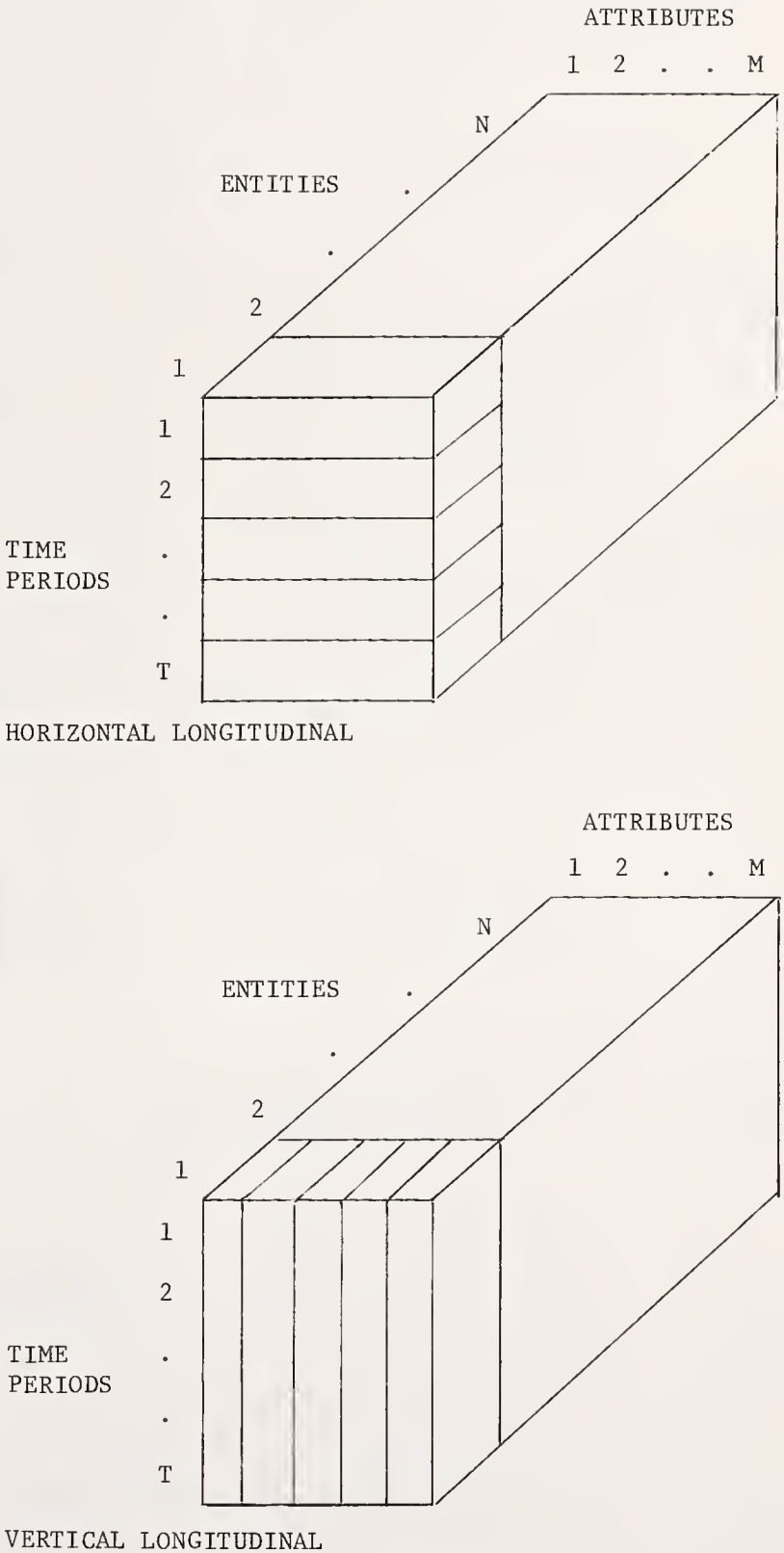
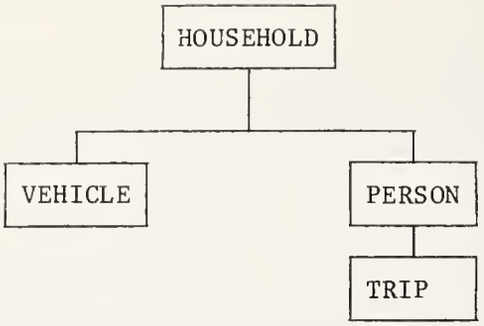
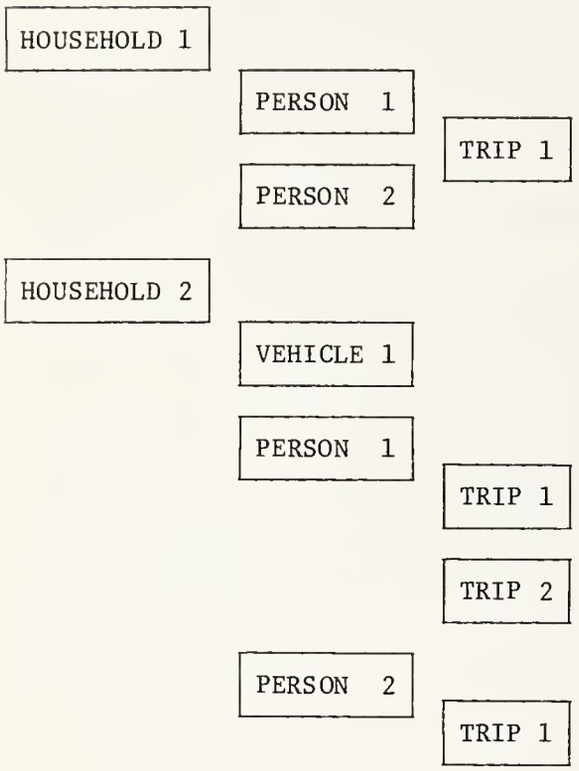


Figure 2: Longitudinal Data Structures

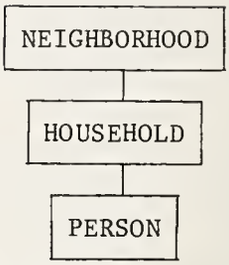


Hierarchic Structure

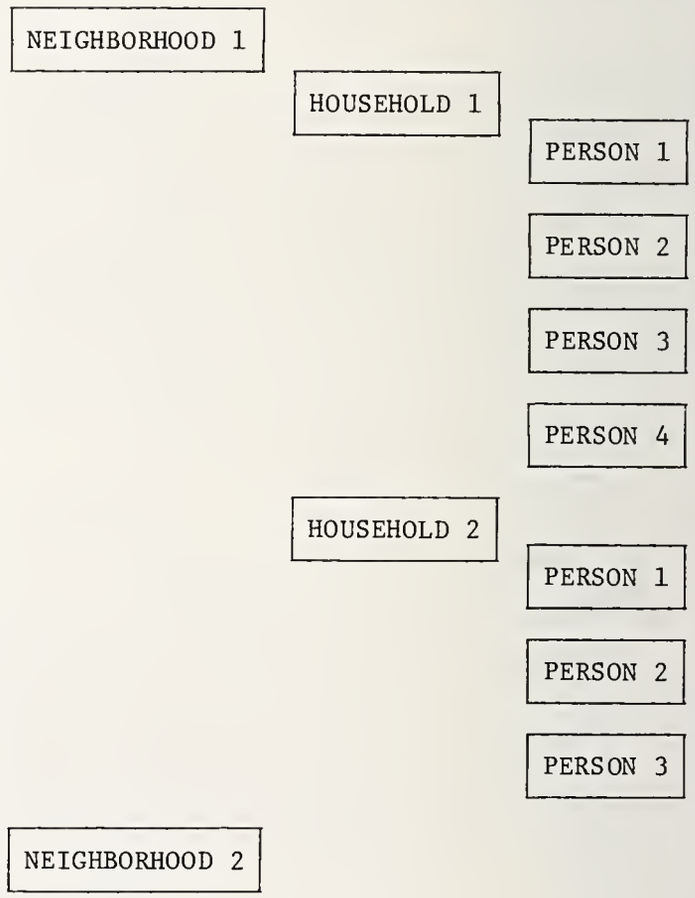


Sample Occurrences

Figure 3: National Travel Survey of 1972



(Simple or Nested) Hierarchic Structure



Sample Occurrences

Figure 4: Public Use Sample of the 1970 Census

THE RELATION:

PERSON <PID#,AGE,SEX,INCOME,...>
(PID# is the unique identification or key)

THE DOMAIN SETS:

AGE<0,1,...,98,99,*>
SEX<'M',"F",*>
INCOME<-99999,...,0,...,99999,*>
(* is used as a "missing" data code)

SAMPLE OCCURRENCES:

PID#	AGE	SEX	INCOME	...
1	37	M	17000	
2	52	M	13500	
3	29	F	18635	
4	18	F	0	
.	.	.	.	
.	.	.	.	
327	46	M	7625	...

Figure 5: The Relation PERSON, its Domain Sets and Sample Occurrences.

THE RELATIONS:

HOUSEHOLD <HID#,STATE,OWN-HOUSE,...>
PERSON <HID#,PID#,AGE,SEX,...>
TRIP <HID#,PID#,TID#,DURATION, COST,...>
VEHICLE <HID#,VID#,MODEL,YEAR,...>

(The identification domains or keys are: HID# for HOUSEHOLD,
HID#,PID# for PERSON
HID#,PID#,TID# for TRIP, and
HID#,VID# for VEHICLE.)

Figure 7: Relational Model of the National Travel Survey.

THE RELATIONS:

FAMILY <FID#,COUNTRY,OWN-CAR,...>
PERSON <FID#,PID#,AGE,SEX,INCOME,...>
("FID#" and "FID#,PID#" are the family and person identifications or keys, respectively.)

SAMPLE OCCURRENCES:

FAMILY:

FID#	COUNTRY	OWN-CAR	...
1	France	1	
2	USA	2	
.			
.			
8972	UK	0	

PERSON:

FID#	PID#	AGE	SEX	...
1	1	32	M	
1	2	29	F	
2	1	13	F	
2	2	41	M	
2	3	38	F	
.				
.				
3972	4	16	F	
1	3	2	M	

Figure 6: Relational Model of a FAMILY and PERSON Data Collection.

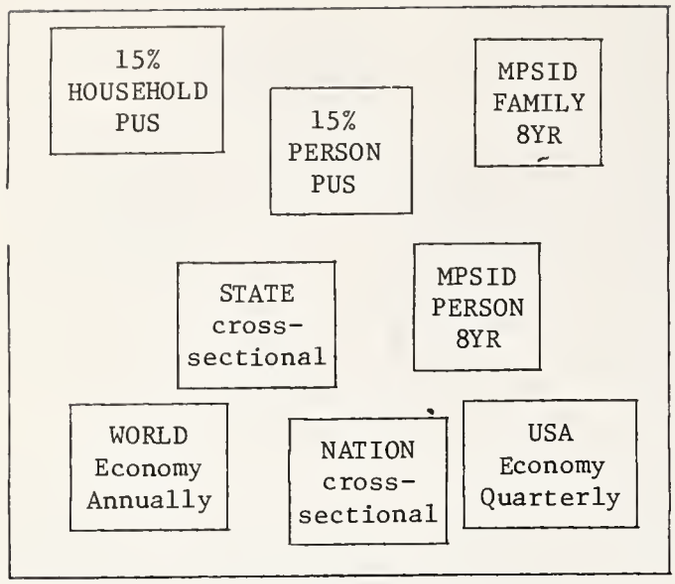
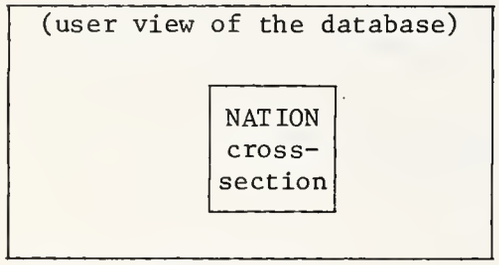
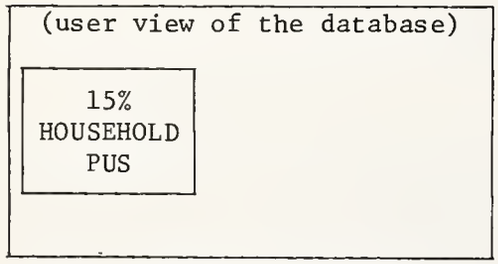


Figure 8: A possible Relational Database consisting of many independent segments.

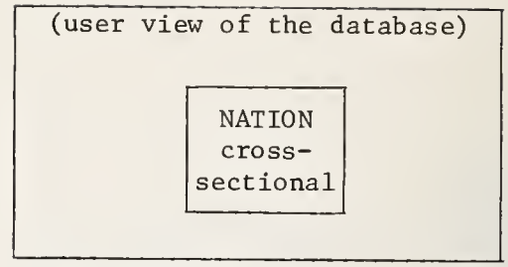



```
!USE  SEGMENT:NATION
```



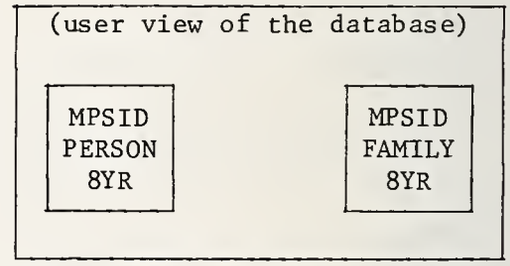

```
!USE  SEGMENT:PUS-HOUSEHOLD
      UNITS:FIRST,100
```

Figure 9: Specification of initial unit of analysis.




```
!USE  SEGMENT:NATION
!COMPUTE  DENSITY = AREA / POPULATION
!EXPORT,SPSS  DENSITY,UNEMP-RATE,
              MIN-WAGE,...
```

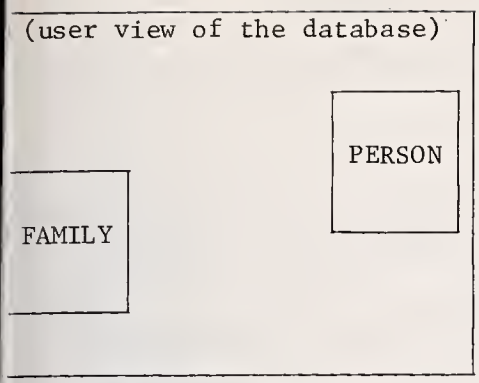
Figure 10: Illustration of an EXPORT capability.




```
!USE  SEGMENT:MPSID-PERSON  PERIODS:ALL
!ATTACH  SEGMENT:MPSID-FAMILY
!COMPUTE  PERCENT = INCOME / INCOME
          @FAMILY
!EXPORT,SPSS  PERCENT,SEX,AGE,
              STATE@FAMILY,...

alternative forms:
!COMPUTE  PERCENT = INCOME / INCOME
          OF FAMILY
!EXPORT,SPSS  PERCENT,SEX,AGE,STATE
          OF FAMILY,...
```

Figure 11: Simple Inter-segment Expressions.

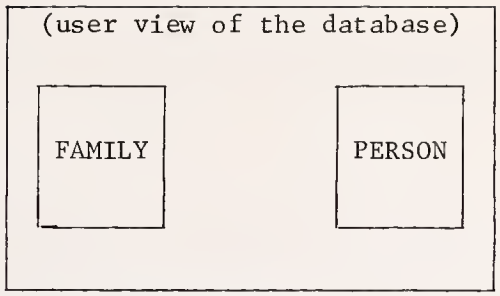
(Additional details of the attaching procedure have been ignored; they are beyond the scope of this paper.)




```
USE  SEGMENT:FAMILY
ATTACH  SEGMENT:PERSON
COMPUTE  Y = SUM(INCOME)@PERSON
COMPUTE  Y16 = MEAN(INCOME)@PERSON(AGE≥16)

lternative forms:
COMPUTE  Y = SUM(INCOME) OF PERSON
COMPUTE  Y16 = MEAN(INCOME) OF PERSON WITH
         AGE ≥ 16
```

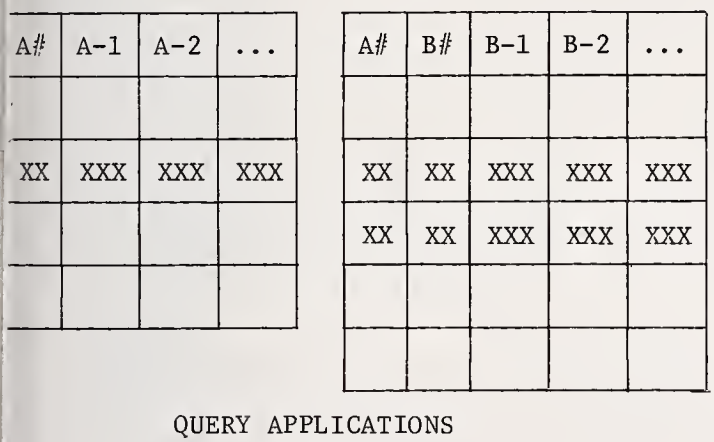
igure 12: Inter-segment Expression with Reduction Functions.

Additional details of the attaching procedure ave been ignored; they are beyond the scope f this paper.)

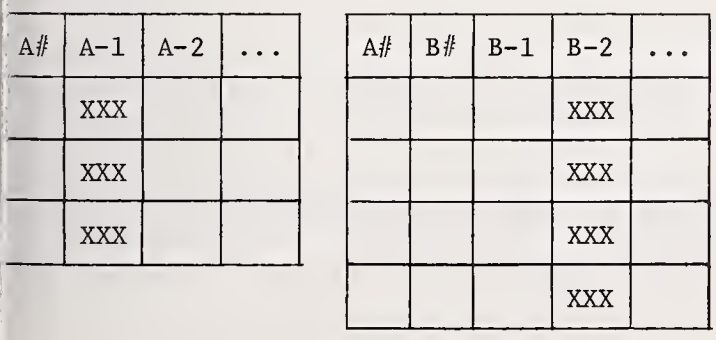



```
!USE  SEGMENT:FAMILY
!ATTACH  SEGMENT:PERSON

... an income calculation as may be made
    by a bank
!COMPUTE KIDS = NUMBER@PERSON(AGE<16)
!COMPUTE Y-MORTGAGE = SUM(INCOME)@
         PERSON IF KIDS=0 ELSE VALUE
         (INCOME)@PERSON(STATUS='HEAD')
         +.5*VALUE(INCOME)@PERSON
         (STATUS='WIFE')
... new family income if adult women's
    incomes increase by 20%
!COMPUTE  Y-NEW = SUM(INCOME IF SEX =
          'M' ELSE 1.20* INCOME)@PERSON
          (AGE≥16)

alternative forms:

!COMPUTE KIDS = NUMBER OF PERSON WITH
              AGE<16
!COMPUTE Y-MORTGAGE = SUM(INCOME) OF
          PERSON IF KIDS=0 ELSE VALUE
          (INCOME) OF PERSON WITH
          STATUS='HEAD' +.5*VALUE(INCOME)
          OF PERSON WITH STATUS='WIFE'
!COMPUTE  Y-NEW = SUM(INCOME IF SEX =
          'M' ELSE 1.20*INCOME) OF
          PERSON WITH AGE≥16
```

Figure 13: Inter-segment expression using conditional choices.



QUERY APPLICATIONS



STATISTICAL APPLICATIONS

Figure 14: Patterns of Access to Data Elements.

(The cross-hatched areas represent the data elements needed to answer a request for a query or statistical application.)

8. REFERENCES

APL(70) APL/360 - An Interactive Approach (1970). Gilman, L. and Rose, A., John Wiley & Sons.

BMD(73) BMD - Biomedical Computer Programs (1973). Dixon, W. J., Ed., 3rd Edition, University of California.

BOYCE(75) "Specifying Queries as Relational Expressions: The Square Data Sublanguage", Boyce, R. F., CACM, v18, n11, 1975.

CENTS-AID(76) CENTS-AID II - The Census Tabulating System Aid (1976). Hill, Gary L., Data Use and Access Laboratories.

CHAMBERLIN(74) "SEQUEL: A Structured English Query Language", Chamberlin, D. C. and Boyce, R. F., ACM SIGFIDET Workshop, 1974.

CODD(70) "A Relational Model for Large Shared Data Banks", Codd, E. F., CACM, v13 n6, 1970.

CODD(71a) "A Data Base Sublanguage Founded on the Relational Calculus", Codd, E. F., ACM SIGFIDET Workshop, 1971a.

CODD(71b) "Normalized Data Base Structure: A Brief Tutorial", Codd, E. F., ACM SIGFIDET Workshop, 1971b.

CPS(77) CPS - Current Population Survey, Bureau of the Census, continuing.

CSTS(75) CSTS General Programming System (GPS) Reference Volume 1: General, Computer Sciences Corporation, June, 1975.

DATE(71) "File Definition and Logical Independence", Date, C. J., and Hopewell, P., ACM SIGFIDET Workshop, 1971.

DATE(74) "The Relational and Network Approaches: Comparison of the Applications Programming Interfaces", Date, C. J. and Codd, E. F., ACM SIGMOD Workshop, 1974.

DATE(75) An Introduction to Database Systems, Date, C. J., Addison-Wesley, 1975.

GENSTAT(75) GRENSTAT- A General Statistical System, Nelder, J. A., et al, University of Edinburgh, 1975.

HEATH(71) "Unacceptable File Operations in a Relational Data Base". Heath, I. J., ACM SIGFIDET Workshop, 1971.

IMPRESS(72) The IMPRESS Manual, Project IMPRESS, Dartmouth College, 1972.

KIDD(69) "Incorporating Complex Data Structures Into a Language for Social Science Research", Kidd, Steven W., Fall Joint Computer Conference, 1969.

MESNAGE(72) "Organization for Storage, Input and Retrieval of Information in Statistics", Mesnage, M., Statistical Office of the European Communities, Brussels, 1972.

MPSID(77) A Panel Study of Income Dynamics, Survey Research Center, University of Michigan, continuing.

ITS(72) The 1972 National Travel Survey, U.S. Bureau of the Census, 1972.

OMNITAB(71) OMNITAB II - Users Reference Manual, Hogben, D., et al, National Bureau of Standards Technical Note 552, 1971.

OSIRIS(73) OSIRIS III - An Integrated Collection of Computer Programs for the Management and Analysis of Social Science Data, Institute for Social Research, University of Michigan, 1973.

PICKLE(74) PICKLE - Users Guide to the Berkeley Transposed File Statistical System, UC/Berkeley, 1974

PLANETS(75) PLANETS - Programming Language for the Analysi of Economic Time Series, Bracy, D. B., et al, The Brookings Institution, 1975.

PSTAT(75) PSTAT - A Computing System for File Manipulation and Statistical Analysis of Social Science Data, Buhler, Roald, Version 3.06, Princeton University, 1975.

PUS(72) Public Use Samples of Basic Records from the 1970 Census: Description and Technical Documentation, U.S. Bureau of the Census, 1972.

SANDEE(75) Sandee, G., (1975) personal communication.

SOS(74) SOS - A Statistical (Sub) System for the Public Use Samples, Levin, M., Data and Access Laboratories, 1974.

SPSS(75) SPSS - Statistics Package for the Social Sciences, 2nd Edition, Nie, N., et al, McGraw-Hill, 1975.

TEITEL(75) "Design Proposal for a Database and Transformation System for Arbitrarily Complex Social Science File Structures," Teitel, Robert F., Urban Institute Working Paper 0004-02, August, 1975.

TEITEL(76) "Database Concepts for Social Science Computing", Teitel, Robert F., Computer Science and Statistics: 9th Annual Conference on the Interface, April, 1976.

TPL(75) TPL - Table Producing Language, Version 3.5, Bureau of Labor Statistics, Department of Labor, July, 1975.

TSICHRITZIS(74) "Comments on Advantages of the Relational View", Tsichritzis, D., ACM SIGMOD Workshop, 1974.

ZLOOF(75) "Query by Example: The Invocation and Definition of Tables and Forms", Zloof, M. M., ACM SIGFIDET, SIGIR and SIGMOD Conference on Very Large Data Bases, 1975.

9. BIOGRAPHY

Robert F. Teitel is the technical advisor in social science computing at The Urban Institute, a nonprofit socio-economic research organization.

He has had technical and managerial responsibility for providing computing resources to academic communities generally and social science communities especially since the early 1960's.

NUMERICAL ANALYSIS IN STATISTICS WORKSHOP

Richard A. Tapia, Chairperson

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

KARL PEARSON WAS RIGHT

David W. Scott, Baylor College of Medicine, Houston, Texas 77030
Richard A. Tapia and James R. Thompson
Rice University, Houston, Texas 77001

ABSTRACT

A discussion is made of nonparametric versus parametric methods for the estimation of robability densities. A new algorithm for nonparametric density estimation is given and ts performance compared with state-of-the-art kernel estimation algorithms.

ey words: computational feasibility, maximum likelihood, Pearson family, kernel estimates, enalized maximum likelihood.

1. INTRODUCTION

Two major causes for poor (especially nonrobust) optimization theoretic techniques in tatistics are

(1) an inappropriate choice of a parameter (function) space
nd
(2) an inappropriate choice of a criterion function (functional).

"Appropriateness" is determined by a balance between computational feasibility and approximation to truth. It is to be expected that the advent of the high speed digital computer should drastically raise our pain threshold of computational feasibility. Consequently it is somewhat surprising that most standard statistical procedures have remained unchanged since the 1930's. Many of these involve the estimation of probability densities.

2. DISCUSSION

In 1922 Fisher [1] presented the concept of parametric maximum likelihood estimation. We recall that his development requires the functional form of the unknown density $f(x|\theta)$ be known. Given a random sample $\{x_1,x_2,\ldots,x_n\}$ from f, we seek that value $\hat{\theta}_n(\underset{\sim}{x})$ contained in appropriate parameter space $\Theta \subset R$ which maximizes

$$\log f_n(\underset{\sim}{x}|\theta) = \sum_{j=1}^{n} \log f(x_j|\theta) \ . \tag{1}$$

Then under very general conditions,

$$\hat{\theta}_n \xrightarrow{a.s.} \theta_o \tag{2}$$

and

$$\hat{\theta}_n \xrightarrow{d} N\left(\theta_o, \frac{-1}{nE\left(\frac{\partial^2 \log f(x|\theta)}{\partial \theta^2}\right)}\right) . \tag{3}$$

The latter result is particularly appealing, since it states that the parametric maximum likelihood estimator asymptotically achieves the Cauchy-Schwarz (Cramer-Rao) lower bound for $E[(\check{\theta} - \theta)^2]$, where $\check{\theta} \in \check{\Theta}$, the class of unbiased estimates for θ .

The optimality properties of parametric maximum likelihood algorithms are likely to be of little utility if (as is generally the case) we do not have a good idea as to the functional form of the unknown density. For example, if we assume the density is normal, a maximum likelihood estimator for the median θ_{me} is $\bar{x}$. If, in fact, the underlying distribution is Cauchy, $\bar{x}$ is no better an estimator for θ_{me} than any single one of the observations. In general, if we assume an incorrect functional form of the density and use any of the classical parametric techniques for estimating the density, we will find that

$$\lim_{n\to\infty} \int_{-\infty}^{\infty} E\left(f_{\text{est},n}(x) - f_{\text{true}}(x)\right)^2 dx > 0 \quad . \tag{4}$$

The pathology of parametric maximum likelihood estimation under real world conditions should not be unexpected. An optimization-theoretic technique designed to have good performance under very restrictive conditions (e.g., that the functional form of the density is known) is unlikely to perform well when we step outside the domain of these conditions. We need to devise algorithms which are "optimal" in a more general and realistic setting. This point was implicitly raised a quarter century before maximum likelihood by Karl Pearson [7]. (For a discussion of the Fisher-Pearson battle on maximum likelihood, the reader is referred to [13].) He considered a fairly large class of probability densities characterized by the differential equation

$$\frac{d \log f(x)}{dx} = \frac{x - a}{b_o + b_1 x + b_2 x^2} \quad . \tag{5}$$

The estimation of the four parameters is readily carried out via the first four sample moments. Unfortunately, although the Pearson Family contains many of the classical distributions, it has serious deficiencies. For example, it contains no multimodal densiti.

In order to obtain a practical extension of Pearson's concept to density estimation in the general setting where we know only that the underlying density is "smooth", we must develop an estimator where the number of characterizing parameters increases with the sample size. The simple histogram (dating back to John Graunt in 1662 [3]) has such a property but suffers from discontinuities. These may be eliminated quite readily by connecting midpoints with straight lines. The extreme "locality" of the histogram is less easily ameliorated.

Computationally more complicated but possessing better consistency properties than the histogram is the kernel density estimator (or "shifted histogram" [12], [6], [8]). Here, on the basis of a random sample $\{x_1, x_2, \ldots, x_n\}$ we have the estimator

$$\hat{f}_H(x) = \frac{1}{nh} \sum_{j=1}^{n} K\left(\frac{x - x_j}{h}\right) \tag{6}$$

where K is any probability density having

$$\int_{-\infty}^{\infty} |K(y)|\,dy < \infty \tag{7}$$

$$\sup_{-\infty < y < \infty} |K(y)| < \infty \tag{8}$$

$$\lim_{y\to\infty} |yK(y)| = 0 \quad . \tag{9}$$

To minimize the asymptotic integrated mean square error, we have the optimal

$$h = \left[\frac{9}{2\int (f''(x))^2 dx}\right]^{1/5} n^{-1/5}, \tag{10}$$

ich gives as asymptotic integrated mean square error

$$IMSE = 2^{-4/5}9^{-1/5}\,\frac{5}{4}\left[\int (f''(x))^2 dx\right]^{1/5} n^{-4/5} \tag{11}$$

fortunately, the design parameter h requires approximate knowledge of $\int (f''(x))^2 dx$. iterative algorithm for the estimation of h is given in [12]. Monte Carlo results dicate that a twofold overestimation or underestimation of h typically causes a two-ld increase of the IMSE over that shown in (11). A survey of other nonparametric nsity estimation techniques is given in [13].

A new approach motivated by a suggestion of Good [2] has been considered in [4], [5], 1], [13]. Here we seek that density $f \in H_o^s(a,b)$ which maximizes the criterion functional

$$L(f) = \sum_{j=1}^{n} \log f(x_j) - \sum_{k=0}^{s} \alpha_k \int_a^b (f^{(k)})^2 \, dx, \tag{12}$$

i.e.,

$$f^{(k)} \in L^2(a,b); \quad k = 0,1,\ldots,s$$

$$f^{(k)}(a) = f^{(k)}(b) = 0; \quad k = 0,1,2,\ldots,s-1$$

$$f \geq 0$$

$$\int_a^b f(x)dx = 1 .$$

e solution to (12) is referred to as the maximum penalized likelihood estimator. From [5] have

Theorem. The MPLE estimator exists and is unique. ∎

Recently, a discretized approximation to the solution of (12) has been algorithmitized d investigated by Scott [10], [11]. This work suggests

Theorem. If $\hat{f}_n(\cdot)$ is the solution to the MPLE criterion and $f_T \in H_o^s(a,b)$ then

$$\int_a^b E[(\hat{f}_n(x) - f_T(x))^2]\,dx \xrightarrow{n} 0 \tag{13}$$

where $f_T(\cdot)$ is the density f truncated to (a,b). ∎

From a practical standpoint, the performance of $\hat{f}_n(\cdot)$ is relatively insensitive to the lection of the design parameters $\underset{\sim}{\alpha}$. If we set all the $\alpha_j = 0$ except for α_2, it is ɔt unusual for a change of α_2 by a factor of 100 from the optimal to increase the IMSE by ɛss than a factor of 2 .

In Table 1, we compare the IMSE of the MPLE with that of popular Gaussian kernel estimator ɔr various densities and sample sizes. Of special note is the fact that although we have ɛed the optimal (and unobtainable) design parameter for the kernel estimator, we have used ıe suboptimal value of $\alpha_2 = 10$ throughout for the MPLE estimator.

TABLE 1

IMSE Values of the MPLE ($\alpha_2 = 10$) and Gaussian Kernel Density Estimation (with optimal h) for Various Distributions and Sample Sizes.

Density	n	MPLE IMSE	Kernel IMSE
N(0,1)	25	.0027	.0041
	100	.00079	.00129
	400	.00033	.00053
$\frac{1}{2}$N(-1.5,1)	25	.00159	.00128
$+\frac{1}{2}$N(1.5, 1)	100	.00054	.00052
t_5	25	.00282	.00475
	100	.00084	.00157

3. CONCLUSIONS

The supposed optimality of classical parametric density estimation procedures is frequently invalid because the true functional form of the density is unknown. Nevertheless, we can attack the more general and practical problem of estimating a density of unknown functional form. The maximum penalized likelihood density estimator has been algorithmitized and is now a part of standard statistical software [11].

4. ACKNOWLEDGEMENTS

The authors wish to thank the U.S. Office of Naval Research, the U.S. Air Force Office Scientific Research, the U.S. Energy Research and Development Administration and the National Heart, Lung and Blood Institute for their support of this work under grants NR042-283, AFOSR76-2711, E-(40-1)-5046 and NIH 17269 respectively.

5. BIBLIOGRAPHY

[1] Fisher, R.A. (1922). On the mathematical foundations of theoretical statistics. Philosophical Transactions of the Royal Society of London, Series A, 222, 309-368.

[2] Good, I.J. (1971). Nonparametric roughness penalties for probability densities. Biometrika, 58, 255-277.

[3] Graunt, John (1662). Natural and Political Observations on the Bills of Mortality.

[4] de Montricher, G.M. (1973). Nonparametric Bayesian Estimation of Probability Densiti by Function Space Techniques, Doctoral dissertation, Rice University, Houston, Tex

[5] de Montricher, G.M., Tapia, R.A., and Thompson, J.R. (1975). Nonparametric maximum likelihood estimation of probability densities by penalty function methods. Annals of Statistics, 3, 1329-1348.

[6] Parzen, Emmanuel (1962). On estimation of a probability density function and mode. Annals of Mathematical Statistics, 33, 1065-1076.

[7] Pearson, Karl (1895). Contributions to the mathematical theory of evolution. II. Ske variations in homogeneous material. Philosophical Transactions of the Royal Socie of London, Series A, 186, 343-414.

[8] Rosenblatt, Murray (1956). Remarks on some nonparametric estimates of a density function. Annals of Mathematical Statistics, 27, 832-835.

[9] Schoenberg, I.J. (1972). Notes on spline functions II on the smoothing of histograms. MRC Technical Report 1222.

10] Scott, D.W. (1976). Nonparametric Probability Density Estimation by Optimization Techniques. Doctoral dissertation, Rice University, Houston, Texas.

11] Scott, D.W. (1977). A software package for the nonparametric estimation of probability densities (subroutine NDMPLE). International Mathematical and Statistical Libraries.

12] Scott, D.W., Tapia, R.A. and Thompson, J.R. (1977). Kernel density estimation revisited. Nonlinear Analysis, 1, 339-372.

13] Tapia, R.A. and Thompson, J.R. (1977). Nonparametric Probability Density Estimation. The Johns Hopkins University Press.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

SOME PROBLEMS IN APPROXIMATION AND ESTIMATION

Murray Rosenblatt
University of California, San Diego, La Jolla, California 92093

ABSTRACT

A density function estimate based on cubic splines is introduced. Some asymptotic properties of the estimate are described. The relationship to a classical spline interpolation problem is noted.

Key words: Bias; bispectra; cubic spline; density function estimate; turbulence; variance.

1. INTRODUCTION

The object of this brief paper is to discuss some questions that may illustrate some aspects of the interface between statistics and related computational problems. Some probability density estimates are considered. The estimates were used in the analysis of some turbulent velocity readings.

A number of estimates of a probability density function have been proposed. Perhaps the most commonly used are the kernel estimates (see Rosenblatt (1970)). Recently another class of estimates based on splines have been proposed by Boneva et al (1971). It then seemed appropriate to investigate the large sample behavior of these spline estimates because of interest in the moderate sample approximations implied by such results.

2. DENSITY ESTIMATES BASED ON CUBIC SPLINES

Assume that f is a continuous density function on $[0,1]$. Let $X_1, X_2, \ldots, X_n$ be independent, identically distributed random variables with density f. Set $y_k = F_n\left(\frac{k}{N}\right)$, $k = 0,1,\ldots,N$, $N = 1/h$ where $F_n(x)$ is the sample distribution function and $h = 1/N$ is the bin size. Let $s_n(x)$ be the cubic spline interpolator of F_n with knots at the point $x_j = j/N$, $j = 0,1,\ldots,N$ and with boundary conditions $f(0) = s_n'(0) = y_0'$, $f(1) = s_n'(1) = y_N'$ These boundary conditions are just chosen for convenience. Comparable results are obtained with other (perhaps more plausible) conditions. See Alberg et al (1967) for a discussion of splines and Rosenblatt (1976) for an analysis of other boundary conditions. The derivative of the spline interpolator is then proposed as the estimate $f_n(x)$ of the density function

$$f_n(x) = s_n'(x)$$

$$= -M_{i-1}\frac{(x-x_i)^2}{2h} + M_i\frac{(x-x_{i-1})^2}{2h} - \frac{h}{6}(M_i - M_{i-1}) + \frac{1}{h}(y_i - y_{i-1}),$$

hen $x \in [x_{i-1},x_i]$ where $M_i = s_n''(x_i)$.

The mean square error of $f_n(x)$ is a measure of deviation and as usual can be expressed as the sum of the variance and squared bias of $f_n(x)$

$$E|f_n(x)-f(x)|^2 = \sigma^2(f_n(x)) + [Ef_n(x)-f(x)]^2$$

$$= \sigma^2(f_n(x)) + (b_n(x))^2 .$$

he mean square error can be gauged by separately getting asymptotic estimates for the bias nd variance. It's worthwhile noting that the question of dealing with the bias is a problem in numerical analysis. The mean $Es_n(x) = h_n(x)$ is the deterministic cubic spline nterpolator of $F(x)$ with knots at the points x_j and satisfying the boundary conditions oted above. The mean of $s_n'(x)$, $Es_n'(x) = h_N'(x)$. The object is then to estimate precisely o the first order the error

$$h_N'(x) - f(x) = h_N'(x) - F'(x) .$$

The desired result is given in the following theorem.

Theorem: Let $F \in C^4[0,1]$ *(continuously differentiable up to fourth order) with* $F'(x) = f(x)$. *Consider* $h_N(x)$, *the cubic spline interpolator of* $F(x)$ *with knots at* x_j, $j = 0,1,\ldots,N$, *satisfying boundary conditions* $f(0) = h_N'(0)$, $f(1) = h_N'(1)$. *Then if* $0 < x < 1$ *is fixed and* $x \in [x_{i-1},x_i]$ *(that is,* $x_{i-1} = [Nx]/N$ *where* $[y]$ *is the greatest integer less than or equal to* y*)*

$$h_N'(x) - f(x) = \frac{f^{(3)}(x)}{4!} h^3 \{(1-r)^4 - r^4 - (1-r)^2 + r^2 + o(1)\} ,$$

as $N \to \infty$. *Here*

$$r = \frac{1}{h} (x-x_{i-1}) .$$

Comparable results can be obtained for $h_N(x)$ and other derivatives of $h_N(x)$. It is curious that this result does not seem to appear independently in earlier literature on splines. The result implies that there is a local oscillation in the bias due to the binning.

Let $\sigma = \sqrt{3} - 2$. One can then also estimate the variance of the estimate.

Theorem: Let f *be continuous on* $[0,1]$. *The variance of the spline estimator* $s_n'(x)$ *of* $f(x)$ *is given by*

$$\frac{f(x)}{nh} A(r) + o\left(\frac{h}{n}\right) ,$$

if $0 < x < 1$ *is fixed and* $nh \to \infty$, $h \to 0$.

Here

$$A(r) = 1 - \frac{3(1-\sigma)}{2+\sigma}\left(2r^2 - 2r + \frac{1}{3}\right) + \frac{9}{4}\left(\frac{1-\sigma}{2+\sigma}\right)^2 \left[\left(2r^2 - 2r + \frac{1}{3}\right)^2 + \left\{\left(r^2 - \frac{1}{3}\right) + \sigma\left(\frac{1}{3} - (1-r)^2\right)\right\}^2 \frac{1}{1-\sigma^2} + \left\{\left(r^2 - \frac{1}{3}\right) + \frac{1}{\sigma}\left(\frac{1}{3} - (1-r)^2\right)\right\}^2 \frac{\sigma^2}{1-\sigma^2}\right].$$

Graphs of the bias and the function $A(r)$ are given in Figures 1 and 2. A more extensive analysis of the asymptotic behavior of such estimates can be found in Lii et al (1975).

Spline estimates of this type have been compared with some kernel estimates by some limited Monte Carlo simulations. Briefly, the spline estimates appear to be superior to the kernel estimates if f is quite smooth. However, if f is not sufficiently smooth a number of kernel estimates are seen to be superior to the spline estimates.

Readings of turbulent wind velocity derivative supplied by Wyngaard were used to get density estimates. Here, of course, the data is dependent. Nonetheless there are reasonable indications that similar techniques can be used here (see Rosenblatt (1970)). The data was sampled 3200 times per second for an hour and then binned. Two spline estimates and one kernel estimate were made of part of the left tail. The spline fit with cell-width equal to one bin is the light oscillatory curve in Figure 3. The spline fit with cell-width equal to 2 bins is the thick curve. The kernel estimate (using a triangular-like weight function) is given by the piecewise linear curve. The tail was fitted adequately by least squares by $f(x) = Ae^{-B|x|^C}$ with $A = 0.74$, $B = 4.2$ and $C = 0.41$. This appears to be consistent with an earlier fit by Tennekes and Wyngaard (1972) but in contrast with a suggested fit by Kolmogorov and Obukhov of the rate of energy dissipation in high Reynold's number turbulence by a log normal distribution.

3. SOME BRIEF REMARKS ON BISPECTRAL ESTIMATES

The sequence of readings of turbulent velocity derivative readings referred to earlier were used. As is usual, an initial calibration of the readings is made. Given the calibration, one wishes to estimate the bispectral density or Fourier transform of third order central moments so as to gauge the nonGaussian character of the readings and get some insight into the nonlinear character of turbulence. The assumption of stationarity seems to be a reasonable assumption for moderate time intervals (perhaps up to four or five minutes). The questions of statistical resolution and computational ease that arise here are, of course, related to those involved in a second order spectral analysis, but they are more complicated. At the very least one is now concerned with estimating a surface. Also, the variance properties of a bispectral analogue of the periodogram are much worse than in the second order case since the variance is proportional to the sample size of the data being processed. A detailed discussion of the theoretical and computational aspects of such bispectral analysis in the context of analyzing turbulent velocity derivative readings can be found in Lii et al (1976).

4. ACKNOWLEDGEMENT

The studies were supported in part by the Office of Naval Research.

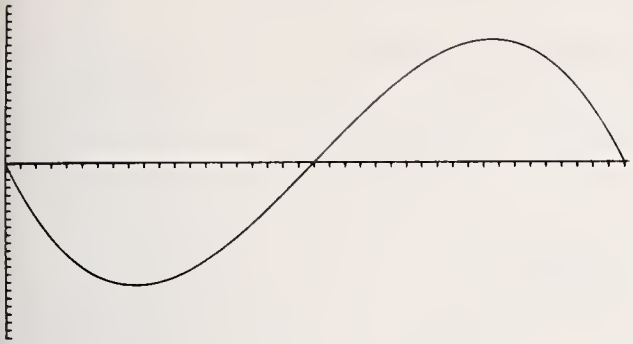


Fig. 1. Bias and the function $(1-r)^4 - r^4 - (1-r)^2 + r^2$.

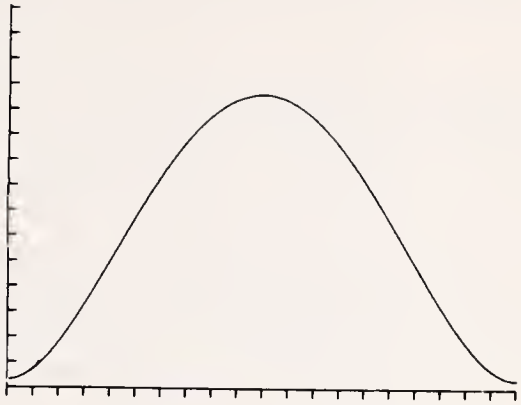


Fig. 2. Variance and the function $A(r)$.

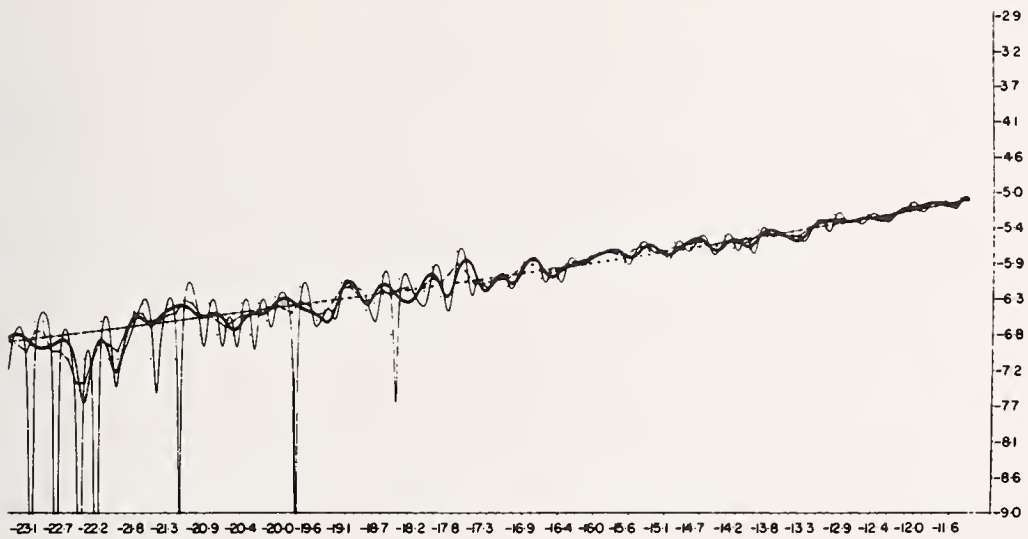


Fig. 3. Estimation of left tail of the probability density of turbulent wind velocity. Turbulent Reynold's number $\cong$ 8000. Histogram, kernel and two spline estimates.

5. REFERENCES

AHLBERG, J., NILSON, E., and WALSH, J (1967). The Theory of Splines and Their Applications. Academic Press, New York.

BONEVA, L., KENDALL, D., and STEFANOV, I. (1971). Spline transformations, J. R. Statist. Soc. Ser. B, 33, 1-70.

LII, K. S., and ROSENBLATT, M. (1975). Asymptotic behavior of a spline estimate of a density function, Comp. & Math. with Appls. 1, 223-235.

LII, K. S., ROSENBLATT, M., and VAN ATTA, C. (1976). Bispectral measurements in turbulence, J. Fluid Mech., 77, 45-62.

ROSENBLATT, M. (1970). Density estimates and Markov sequences, from Nonparametric Techniques in Statistical Inference (edited by M. L. Puri) 199-209.

ROSENBLATT, M. (1976). Asymptotics and representation of cubic splines, J. Approx. Theory 17, 332-343.

TENNEKES, H., and WYNGAARD, J. C. (1972). The intermittent small-scale structure of turbulence: data processing hazards. J. Fluid Mech. 55, part 1, 93-103.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

ORTHOGONAL TRANSFORMATIONS IN REGRESSION CALCULATIONS

G. W. Stewart
University of Maryland, College Park, MD 20742

ABSTRACT

The material sketched in this abstract will be presented in an expanded form elsewhere.

We are concerned with the regression problem of determining a vector β such that

$$\rho^2 = \|y-X\beta\|^2$$

is minimized. Here X is an $n \times p$ matrix of rank p and $\|\cdot\|$ denotes the usual Euclidean norm. The unique solution β satisfies the normal equations

$$(X^TX)\beta = X^Ty ,$$

from which most of the commonly used computational methods can be derived.

An alternative to the normal equations is furnished by the QR factorization of X. Specifically, there is an $n \times p$ matrix Q with orthonormal columns and an upper triangular matrix R such that

$$X = QR .$$

Since $X^TX = R^TR$, it is easily verified that

(1) $$R\beta = z$$

where

$$z = Q^Tr .$$

Thus a knowledge of the QR factorization of X reduces the solution of the least squares problem to that of forming Q^Tz and then solving the upper triangular system (1). It is also easy to see that $QQ^T = X(X^TX)^{-1}X^T$ is the projection onto the column space of X.

Methods based on the QR decomposition are numerically more stable than methods based on the normal equations in the sense that they can solve a wider range of problems at a fixed precision of computation. None the less, they are not widely used in statistical calculations. Three reasons are commonly advanced:

1. They are computationally more expensive;

2. They require more storage;

3. They do not provide the quantities required by statisticians and data analysts.

The first reason is true, although the difference is not great; a change in compilers can easily cause greater changes in computation time. The second reason is false. Although it is true that the Golub-Householder method of computing the QR factorization requires that all of X be present in main memory and then destroys X, there is another method by which X can be brought in row by row so that only storage for R must be allocated.

The third reason is also false; however, considerable ingenuity is required to perform the operations generally required in regression problems, and a large part of this paper is devoted to describing competitive algorithms for the following:

1. Adding an observation
2. Deleting an observation
3. Fitting arbitrary subsets of variables
4. Hypothesis testing
5. Forward stepwise regression
6. Backward stepwise regression

In comparing QR methods with methods based on manipulating X^TX (e.g. sweep methods), it is important to realize that neither class of methods has a clear superiority over the other. Sweep methods are efficient and simple. On the other hand they are numerically inferior in two respects. First, it is always possible to pose problems that can be solved at a given precision by QR techniques but require twice the precision to be solved by sweep techniques. A mitigating factor is that in double precision on most computers this phenomenon will not arise with statistically meaningful problems; however, on computers with a 32-bit floating point word it can cause trouble.

The second difficulty with sweep methods is that they form $(X^TX)^{-1}$ explicitly. If a highly colinear variable is added to the regression, $(X^TX)^{-1}$ will become large and numerically of rank unity. If subsequently the offending variable is removed by an inverse sweep operation, $(X^TX)^{-1}$ will consist largely of rounding error. In this case there is no choice but to recompute X^TX and start over.

The author's opinion is that if ten or more decimal digits are carried in the computations and precautions are taken to avoid adding colinear variables, then sweep techniques will solve virtually all meaningful problems and one should not be afraid to use them. On the other hand if one is designing portable software which must run on a variety of computers, then the increased cost and complexity of QR techniques is not too high a price to pay for their numerical stability.

Keywords: regression, least squares, sweep methods, QR decomposition, numerical stability, computational methods.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

AN APPROACH TO TIME SERIES PREDICTION

Marcello Pagano
State University of New York at Buffalo

ABSTRACT

A new method is presented for predicting stationary time series via the quantile function. The empirical regression distribution is smoothed, using Bernstein polynomials, to yield an estimator of the regression density function. This function, in turn, yields the prediction formulae. Numerical examples are presented.

Key words: Prediction; quantile function; Bernstein polynomials; regression function.

1. INTRODUCTION

Given a sample $Y(1),\ldots,Y(T)$ from a stationary time series $Y(\cdot)$, the objective is to predict the "future" observations, $Y(T+1), Y(T+2),\ldots$. The time series $Y(\cdot)$ is said to be stationary if for any positive integer n and integers h, $t_1,\ldots,t_n$, the joint distribution of $Y(t_1),\ldots,Y(t_n)$, is the same as that of $Y(t_1+h),\ldots,Y(t_n+h)$.

If one wishes to make specific assumptions about the form of the above joint distribution, or if one wishes to restrict one's attention to linear predictors, then the pioneering works of N. Wiener and A. Kolmogorov are well covered in the books by Whittle (1963) and Doob (1953).

Denote the predictor of $Y(T+1)$, based on the previous m observations, by $Y(T+1 \mid T,\ldots,T-m+1)$. If we wish to minimize the mean squared error of prediction, i.e., $\mathcal{E}\{Y(T+1) - Y(T+1 \mid T,\ldots,T-m+1)\}^2$, then

$$Y(T+1 \mid T,\ldots,T-m+1) = \int y \, dF\Big(y \mid Y(T),\ldots,Y(T-m+1)\Big) \quad , \tag{1}$$

where $F(\cdot \mid \cdot)$ is the distribution of $Y(T+1)$ conditional on $Y(T),\ldots,Y(T-m+1)$. If we wish to minimize the mean absolute error of prediction, i.e., $\mathcal{E}\left|Y(T+1) - Y(T+1 \mid T,\ldots,T-m+1)\right|$, then

$$Y(T+1 \mid T,\ldots,T-m+1) = M$$

where

$$.5 = \int_{-\infty}^{M} dF\Big(y \mid Y(T),\ldots,Y(T-m+1)\Big) \quad . \tag{2}$$

The spread of the predictor may be judged by evaluating either

$$\int \Big(y - Y(T+1 \mid T,\ldots,T-m+1)\Big)^2 \, dF\Big(y \mid Y(T),\ldots,Y(T-m+1)\Big) \quad , \tag{3}$$

and/or

$$\int_L^U dF\Big(y \mid Y(T),\ldots,Y(T-m+1)\Big) \quad . \qquad (\cdot$$

Of seemingly primary importance in equations (1) through (4) is the conditional distributi function. Indeed, if this function is known then the problem is solved. However, if one anticipates the requisite estimation to be performed, it is the functionals appearing in equations (1) through (4) which are of ultimate interest. Moreover, there is an attractiv alternative method of evaluating them.

Let $F_1(\cdot)$ be the distribution function of $Y(t)$, and define the quantile function

$$Q(u) = F_1^{-1}(u) = \inf\{x : F_1(x) \geq u\} , \quad 0 \leq u \leq 1 \quad .$$

The existence of the derivative $f_1(\cdot)$ of $F_1(\cdot)$ implies the existence of the derivative $q(\cdot)$ of $Q(\cdot)$. If $F_m(\cdot)$ denotes the joint distribution function of $Y(1),\ldots,Y(m)$, define the regression distribution function,

$$D(u_1,\ldots,u_m) = F_m\Big(Q(u_1),\ldots,Q(u_m)\Big)$$

and its derivative, the regression density function,

$$d(u_1,\ldots,u_m) = \frac{\partial^m}{\partial u_1 \ldots \partial u_m} D(u_1,\ldots,u_m) \quad .$$

Parzen (1977) introduced the distribution function $D(\cdot)$ in a regression context, and we propose to use it for prediction purposes. If $Y(T) = Q(u_1)$, $Y(T-1) = Q(u_2)$, $\ldots$, $Y(T-m+1) = Q(u_m)$, then equation (1) reduces to,

$$Y(T+1 \mid T,\ldots,T-m+1) = \int_0^1 Q(u) \frac{d(u,u_1,\ldots,u_m)}{d(u_1,\ldots,u_m)} du \quad , \qquad (5$$

and equation (2) to

$$.5 = \int_0^{F(M)} \frac{d(u,u_1,\ldots,u_m)}{d(u_1,\ldots,u_m)} du \quad , \qquad (6$$

with similar changes to equations (3) and (4). If $m = 1$ then a simplification occurs; th denominator in all four integrands is equal to one.

The advantage of this point of view reveals itself when we estimate the functionals in question. An obvious estimator of $F_m(\cdot)$ is the empirical distribution function,

$$F_{m,T}(y_1,\ldots,y_m) = \sum_{t=m+1}^{T} \prod_{j=1}^{m} e\Big(y_j - Y(t-j)\Big) \Big/ (T-m+1)$$

where

$$e(x) = \begin{cases} 1 & \text{if} \quad x \geq 0 \\ 0 & \text{if} \quad x < 0 \end{cases} .$$

Whence we can estimate $Q(\cdot)$ by

$$Q_T(u) = F_{1,T}^{-1}(u) = \inf\{x : F_{1,T}(x) \geq u\} \quad ,$$

and $D(\cdot)$ by

$$D_T(u_1,\ldots,u_m) = F_{m,T}\Big(Q_T(u_1),\ldots,Q_T(u_m)\Big) \quad .$$

Unfortunately $D_T(\cdot)$ is not differentiable and must thus be smoothed to yield an estimator of $d(\cdot)$. We propose to use Bernstein polynomials to smooth $D_T(\cdot)$ and review their

elevant properties in the next section.

2. BERNSTEIN POLYNOMIALS

A good account of Bernstein polynomials is given in Davis (1963), Lorentz (1953), and utzer (1953). We review them for two dimensional approximations. The extension to higher dimensions is clear. Let the binomial probability

$$b_n(x,j) = \binom{n}{j} x^j (1-x)^{n-j} \quad , \qquad \begin{array}{l} j = 0,1,\ldots,n \\ 0 \le x \le 1 \\ n = 1,2,\ldots \end{array}$$

Let S be the unit square, $S = [0,1] \times [0,1]$. For a function $f(\cdot)$, defined on S , define the Bernstein polynomial of degree (n_1,n_2) ,

$$B(f,x_1,x_2) = \sum_{j=0}^{n_1} \sum_{k=0}^{n_2} f\left(\frac{j}{n_1},\frac{k}{n_2}\right) b_{n_1}(x_1,j)\, b_{n_2}(x_2,k) \quad .$$

This can be written in kernel form as

$$B(f,x_1,x_2) = \int_0^1 \int_0^1 f(\lambda_1,\lambda_2)\, d_{\lambda_1} K_{n_1}(x_1,\lambda_1)\, d_{\lambda_2} K_{n_2}(x_2,\lambda_2)$$

where

$$\begin{aligned} K_n(x,\lambda) &= \sum_{j \le n\lambda} b_n(x,j) \qquad 0 < \lambda \le 1 \\ &= 0 \qquad\qquad\qquad\quad \lambda = 0 \quad . \end{aligned}$$

If $f(\cdot)$ is bounded on S then $B(\cdot)$ converges to $f(\cdot)$ at every point of continuity, as n_1 and $n_2 \to \infty$. Also, not only does $B(\cdot)$ approximate $f(\cdot)$, but its derivative approximates the derivative of $f(\cdot)$. If all the partial derivatives of $f(\cdot)$ of order $\le p$ exist and are continuous in S , then

$$\frac{\partial^p}{\partial x_1^q \partial x_2^{p-q}} B(f,x_1,x_2) \to \frac{\partial^p}{\partial x_1^q \partial x_2^{p-q}} f(x_1,x_2)$$

uniformly on S as $n_1,n_2 \to \infty$ in any manner. And, just as important in this context let

$$\begin{aligned} \Delta_{\epsilon_1,\epsilon_2} f(x_1,x_2) &= f(x_1 + \epsilon_1, x_2 + \epsilon_2) - f(x_1,x_2 + \epsilon_2) \\ &\quad - f(x_1 + \epsilon_1,x_2) + f(x_1,x_2) \quad , \end{aligned}$$

then, if for all nonnegative ϵ_1,ϵ_2 for which the function is defined $\Delta_{\epsilon_1,\epsilon_2} f(x_1,x_2) \ge 0$, then $\Delta_{\epsilon_1,\epsilon_2} B(f,x_1,x_2) \ge 0$. Note that $B(\cdot)$ is always differentiable and its derivative is nonnegative if $f(\cdot)$ has nonnegative first differences.

Unfortunately the convergence of $B(f,\cdot)$ to $f(\cdot)$ is slow, as exemplified by the one dimensional case when $f(x) = x^2$, then $B(f,x) - f(x) = x(1-x)/n$. This convergence is slower than can be obtained by other means. But, if one wishes to use the Bernstein polynomials in a stochastic setting then the size of this bias must be judged in the context of the standard deviation of the estimator being smoothed. It is usually of a smaller order.

Showing the formulae for the memory 1 predictor, we can estimate $D(\cdot)$ by, with $n = T - 1$,

$$\tilde{D}(u_1,u_2) = \sum_{j=0}^{n}\sum_{k=0}^{n} F_{2,T}\left(Q_T\left(\frac{j}{n}\right), Q_T\left(\frac{k}{n}\right)\right) b_n(u_1,j)\, b_n(u_2,k)$$

whence we obtain a nonnegative estimate of the nonnegative function $d(\cdot)$,

$$\tilde{d}(u_1,u_2) = \frac{\partial^2}{\partial u_1 \partial u_2}\tilde{D}(u_1,u_2)$$

$$= n^2 \sum_{j=0}^{n-1}\sum_{k=0}^{n-1} \Delta_{1/n,1/n} F_{2,T}\left(Q_T\left(\frac{j}{n}\right), Q_T\left(\frac{k}{n}\right)\right) b_{n-1}(u_1,j)\, b_{n-1}(u_2,k) \quad .$$

This formula is simplified because of the form of $F_{2,T}(\cdot)$. Let R_t denote the rank of $Y(t-1)$ amongst $Y(1),\ldots,Y(T-1)$, and S_t the rank of $Y(t)$ amongst $Y(2),\ldots,Y(T)$. Then

$$\tilde{d}(u_1,u_2) = n \sum_{j=0}^{n-1} b_{n-1}(u_1,R_{j+1}-1)\, b_{n-1}(u_2,S_{j+1}-1) \quad .$$

This function may now be inserted in equations (5) and (6). For example, if $Y(T) = Q(u_2)$, equation (1) may be estimated by

$$\hat{Y}(T+1|T) = \int_0^1 Q_T(u)\,\tilde{d}(u,u_2)\,du = \sum_{t=1}^{n} Y(S_t) \int_{\frac{t-1}{n}}^{\frac{t}{n}} \tilde{d}(u,u_2)\,du \quad , \tag{7}$$

and equation (2) by

$$.5 = \int_0^{F_{1,T}(M)} \tilde{d}(u,u_2)\,du \quad . \tag{8}$$

3. NUMERICAL EXAMPLES

Two data sets were chosen to display the above methodology. The first data set is Wolfer's annual sunspot data, and, the second is the daily electricity consumption of a large utility company. Both data sets have been mean corrected.

Figures 1 and 2 show the first 29 data points with a solid line and the result of equation (7) as circles. An improvement is seen in Figures 3 and 4 where the sample size has been increased to 59. Figures 5 and 6 refer to the sample size 59 but the circles represent the result of evaluating equation (8).

Increasing the sample size improves the pictures, as expected. A bigger improvement, not shown here, occurs when the memory length is increased from one to two.

4. ACKNOWLEDGMENT

Research supported in part by a grant from NSF whilst the author was a visitor at Stanford University. The author thanks Professor Gene H. Golub for his support and hospitality.

5. REFERENCES

UTZER, P. L. (1953). On two-dimensional Bernstein polynomials, Can. J. Math, 5, 107-113.

AVIS, P. J. (1963). Interpolation and Approximation, Blaisdell, Mass.

OOB, J. L. (1953). Stochastic Processes, John Wiley and Sons, Inc., New York.

ORENTZ, G. G. (1953). Bernstein Polynomials, University of Toronto Press, Toronto.

ARZEN, E. (1977). Nonparametric statistical data science: a unified approach based on density estimation and testing for "white noise," Statistical Science Division, SUNY at Buffalo, Report #47.

HITTLE, P. (1963). Prediction and Regulation, English Universities Press, London.

BIOGRAPHY

Marcello Pagano received a Ph.D. in statistics from Johns Hopkins University in 1970 and is presently an associate professor in the Statistical Science Division of the Department of Computer Science at State University of New York at Buffalo.

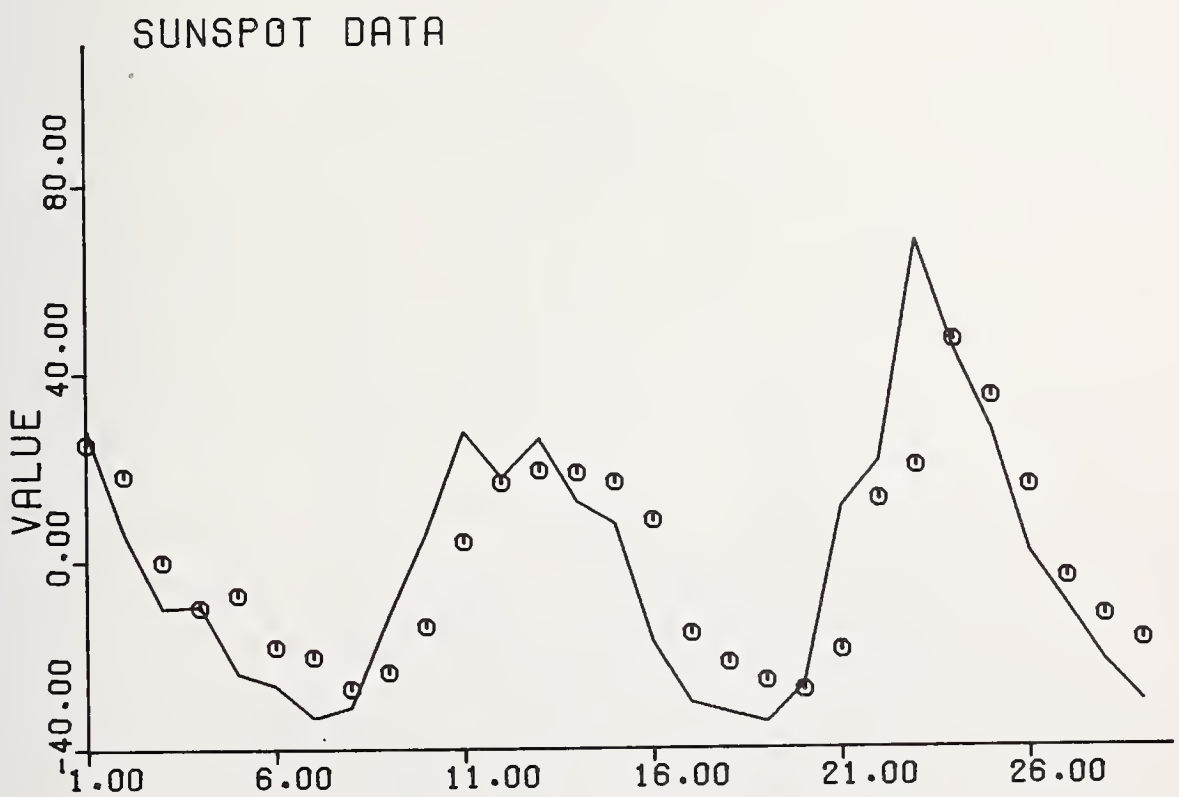


Figure 1

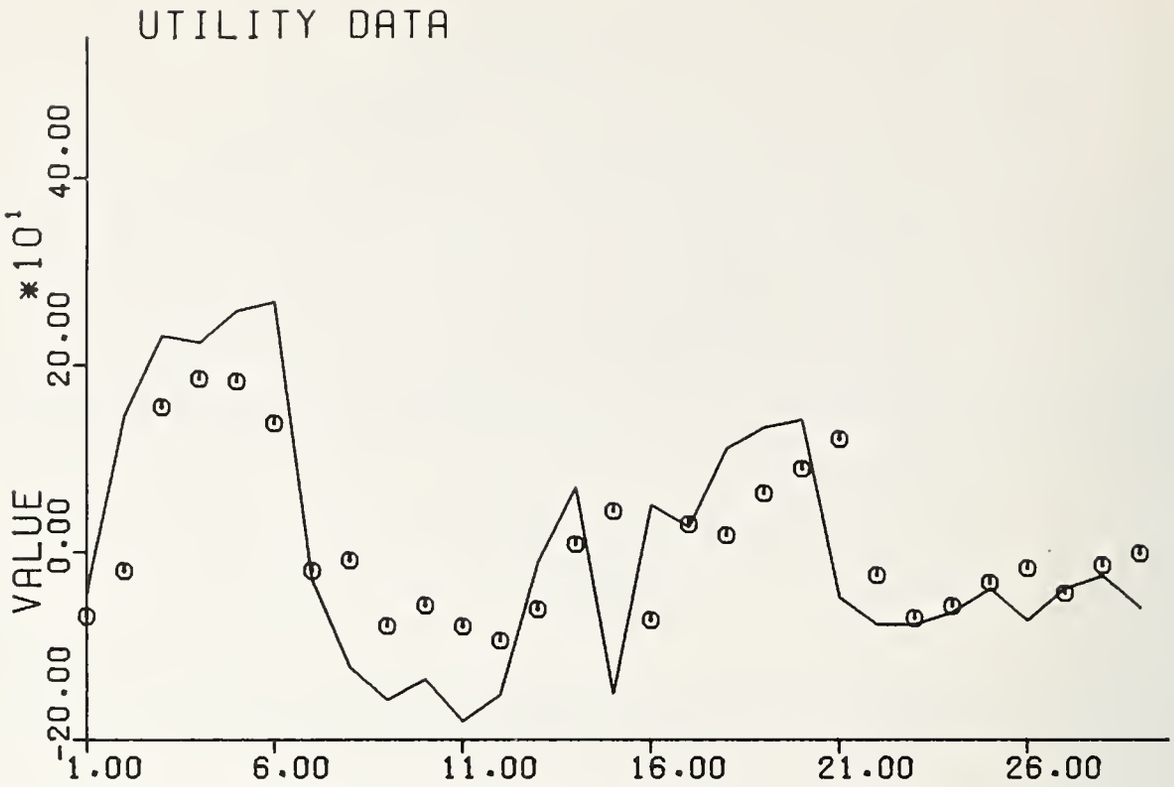


Figure 2

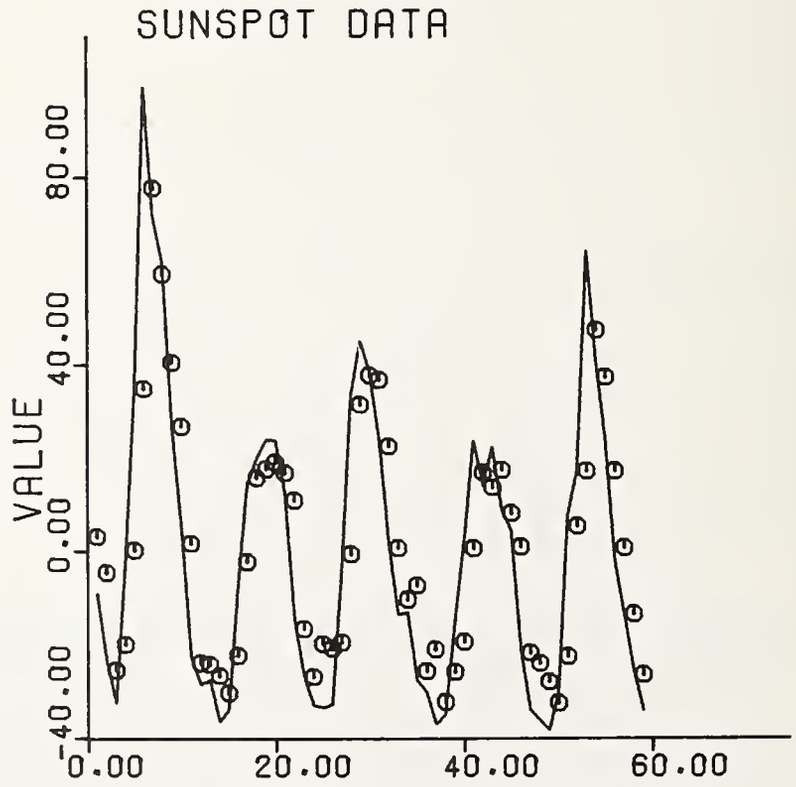


Figure 3

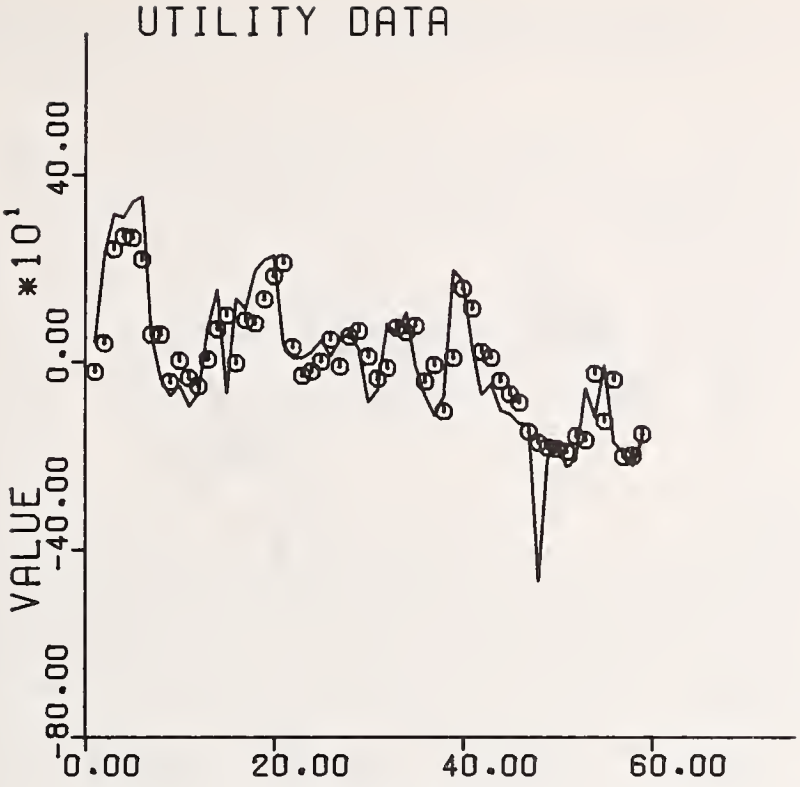


Figure 4

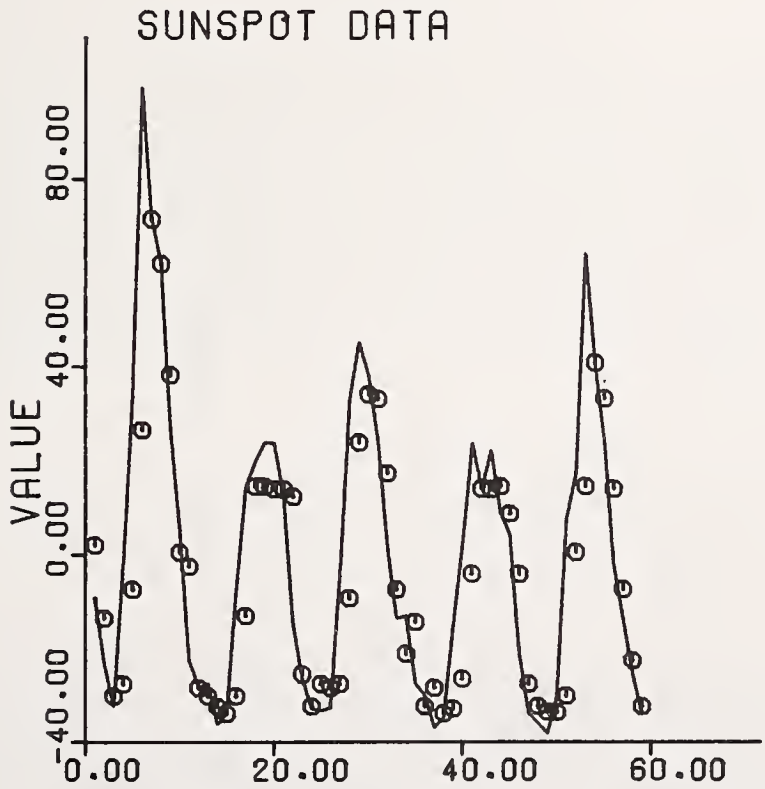


Figure 5

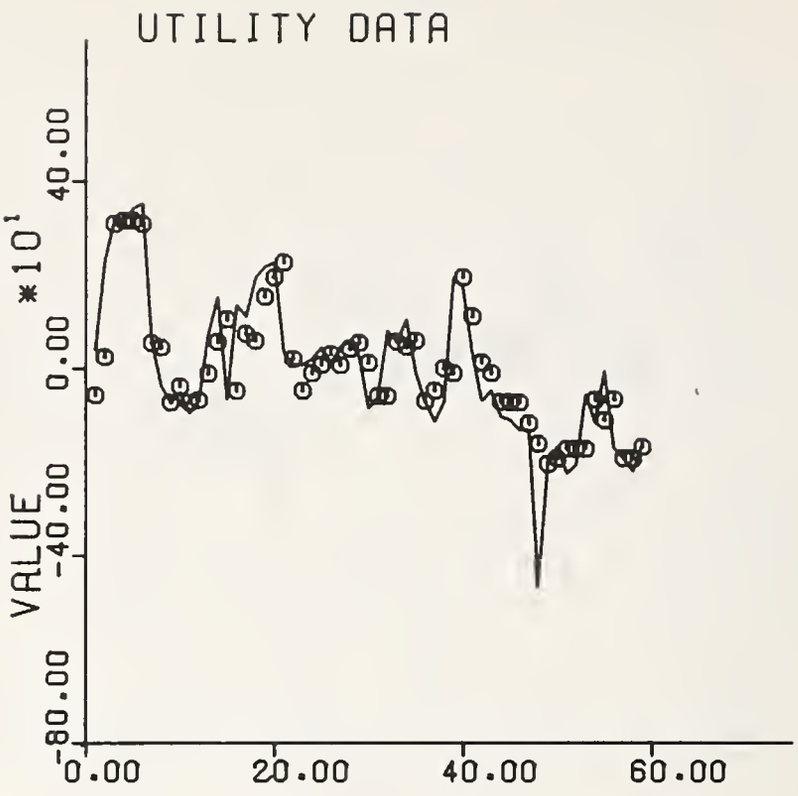


Figure 6

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
eld at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

SOME EXAMPLES OF THE INTERFACE

BETWEEN STATISTICS AND NUMERICAL ANALYSIS

C. P. Tsokos and J. J. Higgins
University of South Florida

ABSTRACT

The availability of increasingly sophisticated computer hardware and software is changing the nature of statistical research. Increasingly complex statistical models and methods are replacing the overly simplistic models which in the past had to be used because of their computational convenience. New statistical procedures are being developed and some old statistical theories are getting new emphasis because numerical implementation is feasible. Some recent problems and results in statistical methodology are discussed in which numerical methods are an integral part of the solution. Examples are drawn from the areas of Bayesian statistics, robust estimation, nonparametric methods, and stochastic differential equations.

1. BAYESIAN METHODS

Let $X_1, X_2, \ldots, X_n$ be a sample from a density function $f(x,\underset{\sim}{\theta})$ where $\underset{\sim}{\theta}$ is a vector of unknown parameters. Classical methods are concerned with the problem of making inferences about $\underset{\sim}{\theta}$ (e.g. confidence intervals, tests of hypothesis) on the basis of information contained in the sample alone. However the statistician may have additional information concerning the behavior of $\underset{\sim}{\theta}$ from scientists, specialists, etc., who have had experiences with similar sorts of data. This past experience is expressed in terms of the prior distribution of $\underset{\sim}{\theta}$, call it $g(\underset{\sim}{\theta})$. Note that $g(\underset{\sim}{\theta})$ is a multivariate probability distribution if $\underset{\sim}{\theta}$ is a multiparameter vector. The statistician combines the information of the sample with the prior information to form the posterior density

$$g(\underset{\sim}{\theta}|x_1,\ldots,x_n) = \frac{f(x_1;\underset{\sim}{\theta})\cdots f(x_n;\underset{\sim}{\theta})g(\underset{\sim}{\theta})}{\int\cdots\int f(x_1;\theta)\cdots f(x_n;\underset{\sim}{\theta})g(\underset{\sim}{\theta})d\underset{\sim}{\theta}} \tag{1.1}$$

Except in a small number of cases, the functional form of $g(\underset{\sim}{\theta}|x_1\ldots x_n)$ cannot be expressed in simple closed form. Thus, to carry out the Bayesian inferential solution to a problem, numerical methods are indispensable, and numerical problems especially in multiparameter cases can be formidable. It appears that the general implementation of Bayesian methods will proceed as rapidly (or as slowly) as the development of numerical procedures to handle the problems will allow. However, Bayesian procedures are used widely enough now to allow for the structuring of "canned" programs to handle the Bayesian analyses most commonly encountered. Such areas would include reliability theory and applications where Bayesian methods have been getting much attention recently, and the area of linear models in which errors are assumed to be normally distributed. Bayesian

methods in reliability are discussed extensively in Tsokos and Shimi (1977), and a number of interesting applications of Bayesian methods in education can be found in Novick and Jackson (1974).

A particularly interesting and important problem in Bayesian inference is finding estimates of the unknown parameter which will minimize expected losses. Typical loss functions for a univariate parameter θ are the squared error

$$L(\hat{\theta},\theta) = (\hat{\theta}-\theta)^2$$

and the absolute error

$$L(\hat{\theta},\theta) = |\hat{\theta}-\theta|.$$

where $\hat{\theta}$ is an estimate of θ. In general, if $L(\hat{\theta},\theta)$ represents the loss in estimating θ by $\hat{\theta}$, the problem is to find the value of $\hat{\theta}$ which minimizes the posterior loss

$$\int\cdots\int L(\hat{\theta},\theta)g(\theta|x_1,\ldots,x_n)d\theta \ .$$

Except for a few univariate or multivariate cases and a few simple loss functions, not much can be done with respect to closed form solution for $\hat{\theta}$. Numerical optimization, then, could play an important role in Bayesian estimation although to our knowledge research along these lines has been minimal.

One of the criticisms of the Bayesian methods is the subjectivity in the choice of a prior. To overcome this criticism, empirical Bayes methods have been proposed in which the prior distribution is structured on the basis of past data. A few cases along these lines yield simple, closed form answers. However, in general the prior must be structured numerically. Conceptually, it would be desirable to structure Bayesian problems in which the prior is empirically determined, the parameters are multivariate to give flexibility to the model, and the loss function is sufficiently realistic to reflect actual losses. Numerical complexities, however, may preclude the solution to such problems in their fullest generality.

The sensitivity of Bayesian methods to the underlying assumptions has been a topic of considerable interest recently. In particular, much attention has been directed to the effect that is produced by changing the prior distribution. In such studies, analytical results are rarely available because of the difficulties of the mathematics. However, computer simulation offers a relatively easy way to obtain answers. Thus, in Bayesian statistics and other branches as well, there continues to be the need for the development of efficient, user oriented, simulation techniques to solve analytically intractable distributional problems.

2. ROBUST ESTIMATION

A robust statistical procedure is one which, while possibly not optimal in any case, is nearly optimal in many cases. Robust procedures have been considered very extensively in recent years which can be attributed to the feasibility of handling such procedures numerically. Much effort has been directed to the development of robust estimates of location parameters. (A parameter μ is said to be a location parameter if the cumulative distribution function of the observations can be expressed in the form $G(x) = F(x-\mu)$ where F belongs to some well-defined class of functions. Typically, μ is the median.)

has long been recognized that the sample mean is a poor estimate of location in many ases, yet the properties of many other possibly desirable estimates of location could ot be investigated until large scale statistical simulation became feasible. One of ne most important simulation studies on robust estimates of location was undertaken Princeton University and the results are reported in Andrews, et. al. (1972).

Attention recently has been directed to robust regression methods. In the same way hat the sample mean has been shown to be deficient as an estimate of location, least quares methods have been shown to be deficient as estimates of univariate and ultivariate trends. Recent papers by Moussa-Hamouda and Leone (1977) and Denby and allows (1977) consider the problem of robust regression. Following along the lines of urrent research, the development and implementation of robust methods for multivariate nalysis can be expected to receive the increasing attention of statisticians and numerical nalysts in the near future.

3. NONPARAMETRIC STATISTICS

In general, a nonparametric treatment of any problem is one which expresses the ntended calculation in terms of operations on functions which satisfy a few side conditions. The class of functions with which we must work in performing these calculations will generally be contained in one of the standard function spaces of nalysis.

By far the most persuasive difficulty in nonparametric statistics is the necessity of finding representations of function spaces which are sufficiently rich to preserve the robustness of the procedure without being so large as to be computationally intractable. Spline functions seem to be the most flexible of such representations, but their properties are well understood only for spaces of functions defined on an interval of the real line. Since statistical problems generally involve approximation of density functions on multidimensional manifolds, there is a great need for empirical and theoretical work on the degree of approximation which can be expected from various classes of splines on n-dimensional regions. In practical computation, efficient algorithms for osculating interpolation (i.e., uniform interpolation of a function and its derivatives) using splines are urgently needed. The almost complete absence of such algorithms for spaces other than continuous functions on a finite interval is especially bothersome.

In many problems of nonparametric estimation and nonparametric curvilinear regression, we are able to characterize the functions involved as belonging to a separable Hilbert space. Representations of separable Hilbert spaces involving the use of complete orthonormal bases have the very desirable property of reducing many types of calculations to problems of matrix algebra. The degree of approximation attainable using a fixed, finite number of parameters is usually accurately predictable. Thus, a natural choice of a finite representation exists. In a Hilbert space representation, many operations of use in statistics (especially convolutions) are reduced from burdensome problems of integration in the original space to simple algebraic operations on the Fourier transform. For all of these reasons, such representations are very desirable. The success of algorithms based on complete orthonormal sets in such disciplines as quantum mechanics and electromagnetic theory has done much to substantiate their utility.

A practical problem is that of performing the appropriate generalized Fourier transform (and its inversion) once the problem has been approximated in terms of a finite-dimensional subspace of H. In the case of the conventional discrete Fourier transform, the special properties of the trigonometric functions have been fully exploited in the development of the well-known fast Fourier transform. Analogous high speed algorithms for other types of Fourier transforms would be extremely useful. For example, for constructing estimates of p.d.f. from knowledge of the moments the Fourier-Hermite transform is very useful and numerical inversion algorithms of high speed are needed. The existing algorithms are based on quadrature and are quite slow.

4. STOCHASTIC EQUATIONS

Another area in which efficient algorithms are lacking is that of approximate numerical integration of stochastic evolution equations. We single out for consideration here two particular problems, which arise in the solution of linear and nonlinear systems, respectively.

The study of Markov processes in particular leads to linear evolution equations on function spaces. The use of transform methods (especially Fourier, Laplace, and Z-transforms) to convert these systems to well-behaved partial differential evolution equations is widely used in seeking exact solutions, and in developing perturbation series for approximate solutions. A characteristic common to all such transforms is that most of the useful statistical information is contained in the fine structure of the solution of the transformed equation near the origin. The moments, in particular, are generally functions of the derivatives of the transform at the origin. Hence, we need numerical techniques which provide very high accuracy near the origin, even at the expense of global accuracy. It would seem that useful techniques could be based on recursive series expansion techniques or on the representation of the transform by suitable splines.

The study of nonlinear stochastic evolution equations presents especially severe computational difficulties. Perturbation methods are useful when the system is only weakly nonlinear, but highly nonlinear systems cannot usually be dealt with by such techniques. The problem, then, is to observe the evolution of the probability density function (p.d.f.) of the variables of interest over time. One approach could be the use of Monte-Carlo procedures, followed by the use of nonparametric density estimators to reconstruct the p.d.f. The other approach would be direct evolution, by numerical means, of the density function on a sufficiently dense grid of points in the (known) region of support of the p.d.f. Deterministic problems of this sort arise often in the study of hydrodynamic problems and many-body problems. The advent of Iliac-type multiprocessor devices holds great promise for increasing the speed of such computations. Thus, much could be expected from intensive research on the use of grid algorithms for numerical evolution of density functions. The fact that the integral of the density must be exactly 1 gives us an analogue of the continuity equation, which has been very useful in solving deterministic problems in hydrodynamics for stabilizing the solutions, and this property should be exploited. Strongly perturbed diffusion processes, and stochastic systems modeling problems of conflict and pursuit, seem particularly susceptible to such a treatment.

While this is by no means an exhaustive list of the areas in which contemporary statistical research places new demands on the theory and practice of numerical analysis, it is hoped that at least some of the most important problems of wide applicability have been identified.

5. ACKNOWLEDGEMENT

This research was supported by the United States Air Force, Air Force Office of Scientific Research, Under Grant Number AFOSR-74-2711.

6. REFERENCES

Andrews, D. F., Bickel, P. J., Hampel, F. R., Huber, P. J., Rogers, W. H., and Tukey, J. W. (1972). Robust Estimates of Location. Princeton University Press, New Jersey.

Dempster, A. P., Schatzoff, Martin, and Wermuth, Nancy. (1977). A Simulation Study of Alternatives to Ordinary Least Squares. JASA, 72, pp. 77-90.

nby, L. and Mallows, C. L. (1977). Two Diagnostic Displays for Robust Regression Analysis. Technometrics, 19, pp. 1-14.

ggins, J. J. and Tsokos, C. P. (1976). On the Behavior of Some Quantities Used in Bayesian Reliability Demonstration Tests, IEEE Transactions on Reliability, 4, pp. 261-264.

Montrickar, G. M. (1973). Nonparametric Bayesian Estimation of Probability Density Functions Space Techniques, Doctoral Dissertation, Rice University, Houston, Texas.

ıussa-Hamouda, Effat, and Leone, Fred C. (1977). The Robustness of Efficiency of Adjusted Trimmed Estimators in Linear Regression, Technometrics, 19, pp. 19-34.

ıvick, Melvin R. and Jackson, Paul H. (1974). Statistical Methods for Educational and Psychological Research. McGraw-Hill, Inc., New York.

ıst, A. and Tsokos, C. P. (1977). Nonparametric Probability Density Estimation with Appropriate Algorithms, to appear.

:ott, David W. (1976). Nonparametric Probability Density Estimation by Optimization Theoretic Techniques, Doctoral Dissertation, Rice University, Houston, Texas

tone, Charles J. (1977). Consistent Nonparametric Regression, Annals of Statistics, 5, pp. 595-645.

sokos, C. P. and Shimi, I. N. (1977). The Theory and Applications of Reliability with Emphasis on Bayesian and Nonparametric Methods, Vol. I and II, Academic Press, New York.

BIOGRAPHIES

Chris P. Tsokos is Professor of Mathematics and Statistics at the University of South lorida, Tampa, Florida. He received the B.S. degree in Engineering Sciences and the M.S. egree in Applied Mathematics from the University of Rhode Island in 1961 and 1963, respectvely, and the Ph.D. degree in Probability Theory and Statistics from the University of onnecticut in 1967.

Professor Tsokos served on the Mathematics faculty of the University of Rhode Island rom 1963-1969 and at the Department of Statistics of Virginia Polytechnic Institute and tate University from 1969-1972. He has served as a statistical consultant for several industrial firms and governmental agencies. He is the author of several textbooks and ver 100 research publications in his broad areas of interest.

Professor Tsokos is a Fellow of the American Statistical Association, a member of the Institute of Mathematical Statistics, listed in Outstanding Educators of America, American Men and Women of Science, among others.

James J. Higgins is Associate Professor of Mathematics at the University of South Florida, Tampa, Florida. He received the B.S. degree in Mathematics from the University of Illinois, the M.S. degree in Mathematics from Illinois State University, and the Ph.D. degree in Statistics from the University of Missouri-Columbia. He served for four years on the faculty of the University of Missouri-Rolla before joining the faculty of the University of South Florida in 1974.

Dr. Higgins' current research interests are in the areas of robust methods for time series, statistical methods in educational research, and reliability theory and its applications. He has served as a statistical consultant for researchers in education, sociology, biology, engineering, medicine, and government.

MAINTENANCE AND DISTRIBUTION OF STATISTICAL SOFTWARE WORKSHOP

Mervin E. Muller, Chairperson

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

MAINTENANCE AND DISTRIBUTION OF STATISTICAL SOFTWARE: SATISFYING DIVERSE NEEDS

Mervin E. Muller
World Bank*, Washington, D.C. 20433

ABSTRACT

Reasons for a computer science statistics interface workshop on the maintenance and distribution of statistical software are presented, i.e. a means for 1) fostering the sharing of statistical software among a community of users, 2) promoting a dialogue among computer scientists and statisticians and among users, developers and distributors of software, 3) presenting and promoting significant technical ideas in the presence of constraints and divergent interests, and 4) discussing unmet needs. A formal definition of maintenance is given in order to show the many aspects related to this problem. From the perspective of the workshop organizer, several important considerations for effective maintenance and distribution of software are presented; namely, types of technical documentation, aspects of testing, performance evaluations, and management commitments. The paper concludes with some other relevant technical considerations, such as user-created extensions, and raises some issues about future directions, including minicomputers.

1. BACKGROUND; REASONS FOR COMPUTER SCIENCE AND STATISTICS INTERFACE WORKSHOP ON MAINTENANCE AND DISTRIBUTION OF STATISTICAL SOFTWARE

During the meeting of the 9th Computer Science and Statistics Symposium on the Interface, David Hogben, Chairman of this Symposium, and I discussed possible Workshop topics that might be of interest to computer scientists and statisticians at this 10th symposium. During the last several years there has been renewed emphasis on the portability of software and on the performance evaluation of software. However, with few exceptions (for example Buhler (1975)), little attention had really been given to the question of how design features might aid in the distribution of statistical software.

There are significant technical and managerial considerations related to the effective maintenance and distribution of software. The speakers invited to present papers on maintenance are people who have technical and administrative interests in improving the maintainability of software which is to be shared among multiple installations, possibly among multiple machine types and computer environments. There are also important questions related to how best to distribute such software. Consequently, distributors of some of the most widely available statistical packages have been invited to a roundtable, along with several users, to discuss and share their experiences in the distribution of such packages as BMDP (1975), COCENTS (1976), OMNITAB (1971), SAS (1976), SPSS (1975), and STATJOB (1973). It is hoped that this workshop will provide a means for:

- fostering the sharing of statistical software among a wider community of users,

*Comments made here do not represent the official views of the World Bank.

- promoting a dialogue among computer scientists and statisticians, and among users, developers, and distributors of software,
- presenting and promoting significant technical ideas,
- recognizing the constraints and divergent interests of those engaged in developing, using, or distributing software, and
- identifying unmet needs.

2. FORMAL DEFINITION OF "MAINTENANCE"

The term "statistical software", is used here in the broadest possible sense, to encompass facilities which may consist of a procedure (module), a program, a package of programs, a system, a language, or other combinations.

"Maintenance status": formal definition. Let us consider a piece of statistical software to be in maintenance status if it has been tested and distributed on the assumption that it can provide the capabilities specified in the User's Manual. Maintenance work can be spent on changing the actual programming, performing tests related to programming changes, changing the documentation, or providing assistance to those using this software.

The reason for presenting a formal definition of maintenance is to emphasize that it includes more than changes to the programs. The documentation must be "maintained", and the users must be able to obtain assistance in resolving any difficulties that they may encounter when trying to use the software. Difficulties with the software can arise for many reasons, some of which the developer or distributor could not have anticipated. These might involve: 1) unexpected uses of controls and control procedures, 2) invalid data that were not foreseen by the designer of the package, 3) misunderstanding of the user documentation, 4) errors in documentation, 5) programming errors (correction of errors is usually the activity that most people associate with maintenance work), 6) malfunction of either the equipment, the operating system, or the control program for the particular statistical software, and 7) errors made by the computer operator. Under ideal conditions the programs and documentation have been designed to facilitate maintenance; otherwise the work of maintenance can be unnecessarily complex and costly for all concerned.

3. MAINTENANCE CONSIDERATIONS FROM THE PERSPECTIVE OF THE WORKSHOP ORGANIZER

Five aspects will be mentioned here, because it has been my experience that these are aspects of maintenance that tend to receive the least attention.

3.1 Documentation requirements. The term "documentation" for statistical software and most other types of software usually brings to mind some type of user's manual or guide, possibly even an abstract or brochure. However, this description fails to take into account the background and experience of the intended users. Depending on the user's background and experience, it may be necessary to include primers on the statistical aspects of the capabilities or on the associated software. In addition, many other types of documentation that affect both the maintainability of the software and its distribution are sometimes needed.Muller and Wilkinson (1976),in working with the ISI Committee on Statistical Computation, have identified a variety of other types of documentation that are relevant to the maintenance and distribution of software. In particular, one may require detailed installation or operating instructions, flow charts, program logic diagrams, descriptions of algorithms and references to them, samples of input and output, descriptions of data structures of both input and output, description of facilities (if provided) to enable users to make extensions, and descriptions of test data. Within the framework of documentation it

would also be desirable to have results of performance evaluations. Also, it is necessary to know whether or not the documentation is being kept current with the software. This may be difficult to determine. If the documentation is not current, there is reason to suspect the long-term value of the software to the user.

3.2 Testing considerations. From the points of view of the maintainer of the software, the distributor, or the user, it is vital that test data be considered an integral part of the design of the software so that it can be available to aid in the maintenance effort and enable the user to determine that in fact the software is performing as intended. The design of the software should take into account the need for testing; otherwise it is very likely that the user or maintainer will be unable to validate the software when this becomes necessary. However, this entire subject could be a separate topic and time does not permit going further into it here. The inclusion of test aids and test data is certainly a very important and relevant issue, and the designer of the software should include them as a design consideration. With such a design, the software should include capabilities for executing, timing, and performing test conditions whenever there is concern about the correctness of the software, say, following any modification. Another aspect being emphasized here is the need for test data and instructions for using or creating test cases and interpreting the results of tests. Under the best of conditions there would also be special software to aid in comparing test results from the point of view of the user and the developer or distributor of the software. It has been my experience that if the software is not well documented and the test cases are not well developed and documented, then much of the value of having software available is lost. Testing and documentation are very important and expensive tasks in the development of software, which accounts for the prominence given to these two items in this brief consideration.

3.3 Performance evaluations. One reason for desiring information on performance evaluation is that this can provide a good indication of whether or not the software is performing correctly without requiring a potential user to make a large investment to evaluate the software. Such information is also necessary to enable the user to make a rational selection from available software and determine under what conditions to use the software. It is desirable to have some of the performance evaluations done by impartial observers (other than those developing, maintaining, or distributing the software). Attempting to use unevaluated software can be dangerous and costly. In this regard, it is encouraging to see the recent activities of the ASA Section on Statistical Computing on the evaluation of software. This is an effort that is expensive to do and beyond the means of most individuals.

3.4 Management commitments. In considering management commitments, one can adopt the points of view of the developer, maintainer, or user of the software. From the point of view of the user, there needs to be an assurance that the developer and maintainer are prepared to service the product, once distributed (or at least advance knowledge that such service is not to be provided). The user and his management need to determine that the software will in fact perform as promised, or be corrected to protect the users who have made the investment to learn to use the package. In this regard, some of the important items of commitment by the maintainer are to: 1) keep track of all reported problems as they are received, to ensure required follow-up; 2) make concurrent corrections to programs, modules, systems, and to their documentation; 3) maintain up-to-date test data to evaluate the correct operation of the software; 4) make the necessary changes to the documentation; 5) test individual modules and the entire package; 6) maintain adequate storage and distribution of both the program and the documentation; and 7) notify the distributor, if different from the maintainer, who will in turn notify the end users, of the implications of reported errors, error corrections or maintenance changes.

The distributor will want to be assured that the above-mentioned maintenance activities are in fact being done well, and that there is a proper information exchange between the user and the maintainer. Another user concern which ought to be the respon-

sibility of the distributor is to ensure that there is some way of providing assistance or consultation to the user in the event of difficulties in using the software because of mis information or misinterpretation as well as actual problems with the software or equipment Another consideration is to determine the likelihood that the software and documentation will continue to be maintained. That is, after investing in learning and adapting to the particular software, is it likely to be available and maintained, and for how long?

3.5 Consulting and Training. From the perspective of the user and the user's management, it is important to know not only that the software and documentation are being maintained, but also that consulting or training are available if needed. Otherwise, the investment made in acquiring the software could be in jeopardy.

4. SOME TECHNICAL CONSIDERATIONS (PORTABILITY, TIMING, USER EXTENSIONS)

As noted in Muller (1975), the question of portability of software is very much a question of time and cost. Therefore, portability is a question to be kept in mind by the user when he desires to obtain software that runs on a particular machine, environment and machine type. This is an important issue because the user may eventually want to change to a larger or different machine. He should be in a position to understand whether or not the software is portable and capable of being used on different machines without requiring large investments in software modifications or staff retraining. Some of the difficult aspects of portability relate to documentation.

Another technical question is the presence of a capability to insert within the software special routines to obtain "timings" of the software or the generation of test cases to foster adequate testing. One of the technical considerations is how one can be assured that the software includes adequate test aids which can, when desired, be bypassed when using the software for production purposes. The technical design should also provide for the insertion of user-developed extensions. If user extensions are permitted, then there must be techniques to control the extensions so that they cannot inadvertently modify in undesired ways those parts of the software provided by the distributor.

5. CONSTRAINTS

As statistical software has improved, it has become easier to obtain specific types of statistical computations and analysis. Furthermore, some of the recently developed techniques and programs are far more complex than their predecessors. In the early days of software availability, it was usually free and there was a rather generous exchange of software.However, as noted in Muller (1976),this changed in the late 1950's and early 1960's, and now there are almost antagonistic points of view. Those who have developed proprietary software cannot look kindly on free software, which may directly compete with their own software, particularly if it is distributed without the attendant responsibility of future availability or assurance that it is free of errors. Furthermore, there must be safeguards to protect the investments of these individuals who developed proprietary software, to encourage them to develop additional types of software. It is my feeling that because of the incredibly large development investment now required, the free distribution of major statistical packages will be the exception in the future.

With respect to statistical algorithms, and with the continued interest in developing them and announcing them in various statistical publications, there is hope that such types of exchange can continue to take place without large costs. However, the larger packages certainly create conflicts and constraints for many. Undoubtedly, the users would like as much documentation available as possible. Moreover, the developers and distributors could find this discouraging if it would enable competitors to undercut

packages in price with competitive products by completely avoiding the large development cost, because they were able to exploit the advantages of the current packages. I mention these various constraints because they are real and should no longer be ignored by professionals of computer science and statistics. In this regard, the International Association for Statistical Computing (IASC) through its proposed work, may encourage both the exchange of techniques and programs and the development of necessary safeguards.

6. FUTURE DIRECTIONS AND NEEDS

Better methods are needed for the evaluation of software to enable better decision-making. Only time and research can help here. There is also the need for better dissemination of what is already available. Here again the planned work of IASC may be of some help as well as the activities being done by the ASA Section on Statistical Computing. See Muller and Wilkinson (1976), as an example of what is involved in establishing an information exchange on statistical software.

Finally, the future is already at hand in the sense that much is made of the opportunity of using minicomputers. It is not at all clear exactly what is meant by a minicomputer, but what is clear is that there is an incredible number of different manufacturers and even larger number of different types of such small computers. The current state of mini's is very similar to the situation of 15 years ago with large computers that had the same computing power as the mini's of today. There is dire need for good portable software for applications that would make sense on such types of computers. Hand-held calculators which are programmable or have circuit chips to perform statistical calculations need to be given adequate attention too. It seems to me that one of the real challenges in the area of computer science and statistics interface is to obtain better insight into what type of applications should go onto a particular computer--whether it be hand-held, mini, or otherwise.

7. REFERENCES

BMDP (1975). Biomedical Computer Programs, edited by W.J. Dixon, University of California Press.

BUHLER, Roald (1975). P-STAT Portability. Computer Science and Statistics. Proc. 8th Ann. Symposium on the Interface, 13-14 February 1975, pp. 165-172, UCLA.

COCENTS (1976). COBOL Census Tabulation System. International Statistical Programs Center, Bureau of the Census, U.S. Department of Commerce, Washington, D.C.

MULLER, Mervin E. (1975). Portability standards for software. Computer Science and Statistics. Proc. 8th Ann. Symposium on the Interface, 13-14 February 1975, pp.173-176 University of California, Los Angeles.

MULLER, Mervin E. (1976). Challenges in using computers in statistical applications in developing countries. Int. Stat. Rev., 44, No. 2, pp. 249-263.

MULLER, Mervin E. and WILKINSON, Graham N. (1976). A standardized form for describing programs, packages, and systems for statistical applications. Int. Stat. Rev., 44, No. 3, pp. 349-353.

OMNITAB (1971): OMNITAB-II. User's Reference Manual. National Bureau of Standards, U.S. Department of Commerce, Washington, D.C.

SAS (1976). A user's guide to the Statistical Analysis System, by Barr, A.J., Goodnight, J.H., Sall, J.P., and Helwig, Jane T., SAS Institute Inc., P.O. Box 10066, Raleigh, North Carolina.

SPSS (1975). Statistical Packages for the Social Sciences, Second Edition, by Nie, N.H., Hull, C.H., Jenkins, J.G., Steinbrenner, K., and Bent, D.H., McGraw Hill.

STATJOB (1973). STATJOB Series, Reference Manual. Academic Computing Center, The University of Wisconsin, Madison.

BIOGRAPHY

Mervin E. Muller studied at UCLA, receiving a Ph.D. in Mathematics, Director of Computing Activities Department, World Bank since 1971 and previously at the University of Wisconsin as Professor of Computer Sciences and Statistics, and Director of the Computing Center; at IBM as Department Director, Senior Corporate Statistician, and Control Planning Associate; at Princeton University as Senior Research Scientist, Departments of Statistics and Electrical Engineering, and as Research Associate, Mathematics Department; at Cornell University as Instructor, Mathematics Department; and at Institute for Numerical Analysis, NBS as Fellow; has publications on information systems, statistics, uses of computers, mathematics, simulation, and sampling.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

SOME TESTING AND MAINTENANCE CONSIDERATIONS IN PACKAGE DESIGN AND IMPLEMENTATION

James R. Allen
Academic Computing Center, University of Wisconsin, Madison 53706

ABSTRACT

Some approaches taken at the University of Wisconsin to minimize errors prior to distribution of STATJOB are discussed. Included are descriptions of design concepts which nelp programmers avoid errors and notes on procedures followed to minimize errors during implementation and maintenance. Also discussed are internal accounting methods used to provide STATJOB users with protection against the consequences of serious software errors -- those that result in plausible but incorrect results.

Key words: Development; documentation; implementation; maintenance; package design; reliability; reporting errors; statistical package; STATJOB; testing.

1. INTRODUCTION

This paper describes experiences related to the maintenance of the statistical package STATJOB developed at the University of Wisconsin. STATJOB is described in the STATJOB Series of reference manuals available from the Madison Academic Computing Center (MACC).

STATJOB was first implemented on the CDC 1604 computer in 1965 and shortly thereafter was installed on the CDC 3600. In 1968-69, the package was converted to the Univac 1108 computer. The CDC version was not maintained after installation of the Univac version.

There are two functions of maintenance. One is to alter a package to reflect changes in what is viewed as correct statistical computing. The other is to correct flaws introduced during design and implementation.

This paper deals with the latter function, which we will call corrective maintenance. In particular, we look at ways of reducing maintenance requirements by adopting appropriate standards and practices during design and implementation.

2. CORRECTIVE MAINTENANCE, RELIABILITY, AND APPROACHES TO DEVELOPMENT

Before getting into details of design and implementation, some remarks about package reliability are appropriate.

If perfection in design and implementation (given the knowledge and tools available at the time) were attainable, then there would be no need for corrective maintenance. The amount of such maintenance performed on a package, then, reflects the extent of flaws in design, implementation, or both, and is probably a good indicator of the overall reliability of the package.

In a well-conceived and carefully implemented system, each corrective action should bring the system closer to perfection, and within a short time after the release of a new component, the need for corrective maintenance should be very rare. We feel that STATJOB, which consists of about 20 major components, is such a system. For example, in the approximately one year between releases of versions 9 and 10, corrective maintenance was needed on only four of the components, and errors in two of those were very minor (the errors are listed in "Introduction to STATJOB, Version 10"). Some components have never needed "serious" corrective maintenance (regression and STJbank file-handling system) and others only once (tabulation, factor analysis) (by "serious" we mean maintenance to correct an error which caused plausible but incorrect results to be printed). We feel that the performance record of STATJOB entitles it to be called a "highly reliable" package. Furthermore, our policy of personally notifying affected users of "serious" errors, when feasible, makes the overall reliabiltiy of STATJOB computing at Wisconsin difficult to surpass.

That STATJOB is a "well-conceived and carefully implemented" system is in large part attributable to the "global system view of data analysis" taken by its principal designer, Dr. Mervin E. Muller. In that approach, there is considerable interaction and compromise

between the user, the designer, the implementor, and the maintainer; see Muller (1969). Of course, this type of interaction must occur in the development of any system. But the quality of the final product depends on how the interaction is controlled and focussed. For example, if a user goal is "reliability", the designer will ask the implementor to explain how that goal is to be achieved. Clearly, the response, "we'll be very careful" is not adequate, and a process of seriously considering methods of achieving reliablity takes place at the outset, and hopefully is repeated as each new component of the system is designed.

The remainder of this paper discusses some of the standards and methods which came out of development processes like those just described. While most things considered are simple, straight-forward, and owing to common sense (such as the subroutine naming convention) rather than scholarly insight, many of them are overlooked in other systems we are familiar with.

3. CONSIDERATIONS AT THE DESIGN STAGE

The designer of a package is responsible for determining the nature of all user interfaces: the control language, input, output and documentation. The designer can make the work of the maintainer easier by considering how the design might ultimately affect the reliability of the package. Here are some considerations:

(a) structure of design

Should many related capabilities be handled together in one set of specifications, or should they be broken into several? The general approach to design structure used for STATJOB is to treat as a single component all capabilities related to one general type of statistical analysis. Then the "global system view" can be applied to each major component of the system.

(b) documentation of computational methods

At an early stage in the design of any analysis component, all computational formulas and algorithms should be fully documented. The document can, and probably should, be in the form that eventually will be included in the user's manual. It will serve as copy for review; as specifications for implementation, testing, and maintenance; and as an indispensable reference for cautious users who frequently check results and thereby constantly test the reliability of the package.

(c) computational accuracy in numerical algorithms

The designer should decide whether computational accuracy is a design or implementation problem. If it is to be handled at the design stage, appropriate specifications must be prepared. In any event, the designer should raise the issue and see it to a conclusion.

(d) review process

The designer, after preparing adequate documentation for a new component of the system, should distribute review copies to key users, staff, and experts in areas related to the type of analysis to be performed. This review procedure, which should be repeated if substantial changes are later made to specifications, will reduce the need for future improvements (and thus maintenance), and will verify that computational methods are correct.

(e) standardization in design

As much as possible, control cards should have a consistent syntax, not only to facilitate documentation and use of the system, but to enable a small set of utility processors to do most of the work to interpret and store the control information. While other opportunities to standardize are obvious, some aren't. For example, we should have adopted uniform standards for handling missing data, then implemented some of the checking in the I/O system, not separately in each program. Also, we devised some powerful machinery to handle a wide variety of scale specifications for the tabulation program. This machinery should have been designed as a general system capability.

4. CONSIDERATIONS DURING IMPLEMENTATION

When STATJOB was converted from the CDC 1604 to the Univac 1108 in 1968-69, the control processing and I/O components of the system were redesigned. The new implementation took advantage of lessons learned in the first implementation. Procedures and techniques used in the new system reduce implementation, testing and maintenance requirements and contribute to the overall reliability of the system.

) use of utility routines

As many system processes as possible should be incorporated in utility routines. The overall reliability of the system can be enhanced (and maintenance needs reduced) by concentrating programming talent and other resources on the utility routines early in the implementation stage. Utility routines are used in STATJOB for interpreting, storing and retrieving control information, allocating dynamic storage, input handling, most vector and matrix output, most card and file output, and many other processes.

) control information storage and retrieval

The CDC version of STATJOB used common blocks to store control information, as most packages now do. To expand a common block, one must check every routine in which that common block is used to avoid name conflicts. Moreover, code added to the system must carefully avoid use of names that are in a common block that may be included in the routine.

These time-consuming and error-prone procedures were replaced with a "tagged storage" scheme. To reserve space to store control information, a call is made to a subroutine, giving the amount of space needed and a "tag" to be entered in an index. To retrieve control information, a subroutine is called either to retrieve the location of the first word of reserved space for a tag, or to retrieve directly the value stored in that first word.

Some scratch arrays for I/O, such as buffers and error flags, are also stored in tagged storage.

c) internal documentation

With the Univac implementation in 1968-69, a series of internal memoes was begun to document individual "utility" routines and other components of the system. In recent years, memoes have been incorporated in the code as comments. The memoes have, of course, been very useful both in performing maintenance and expanding the package.

d) system test modes

STATJOB can run under three test modes: program test, system test, and detailed system test. In the program test mode, intermediate computational results are printed. In the system test mode, limited information is printed about control information storage, dynamic storage allocation, intermediate results of the transformation compiler, and results of the collection of the analysis phase. In the detailed system test mode, complete contents of control storage areas are printed, detailed output is generated by the transformation compiler, and detailed results of the collection of the analysis phase are printed. Detailed system tests are avoided when possible because of the large volume of printed output.

f) subroutine naming convention

All STATJOB subroutine names begin with "Sn" ("Dn" for double precision routines), where n is 1 for system routines and otherwise is a number unique to each program. This simple convention has helped avoid a few conflicts with names of library subroutines and user subroutines, and has been an important convenience during implementation and maintenance. For example, routines associated with one component appear together in various listings.

(g) dynamic storage allocation

All scratch arrays that vary in size depending on the application are allocated dynamically (i.e., at execution time). As in some other systems which dynamically allocate storage, the arrays are stored in blank common, beginning at an address computed from control information. STATJOB differs from other systems in that all dynamic allocation is made at one time, in one place, thereby minimizing chances of miscalculation of addresses and making it easier to find errors in storage allocation. Furthermore, dynamically computed addresses are passed through a subroutine calling sequence, so all references to array elements are relative to the beginning of the array, rather than relative to the beginning of blank common, making the source code easire to write initially and easier to understand later.

(h) modification procedures

To modify temporarily an analysis program, all a programmer need do is store the compiled relocatable elements of the routines to be modified in TPF$, the temporary program file automatically assigned to each run. STATJOB is then executed in the normal manner; the analysis program is re-collected and the modified routines included in a manner transparent to the user (unless the system is in test mode). This simple

procedure makes it convenient to set up debugging runs. It also allows users to interface easily special routines of their own with STATJOB.

(i) protection against user-supplied routines

Occasionally problems brought to our attention involve user-supplied routines which have been interfaced with STATJOB. We have begun to install traps to catch routines which, for example, exceed the boundaries of arrays made available to them. Such traps will save debugging time in the future as more users interface their own routines with STATJOB.

(j) testing procedures

Three levels of testing are done during STATJOB maintenance. "Standard" tests, which check only a few of the capabilities of each program, are run during a release. "Detailed" tests are run for a program when significant changes are made to that program. "Special" tests are included with each release to check changes made in that release.

5. REPORTING ERRORS TO USERS

In our experience developing and maintaining STATJOB, relatively few programming errors encountered were of a "serious" nature; i.e., few of the errors would have lured a user into believing (and publishing, perhaps) incorrectly computed results. However, to protect users against such errors, in 1974 we implemented in STATJOB an internal accounting system which recorded enough information about each run to permit identification of each user affected by a serious error. Information recorded includes the user's account number, the date and time of the run, the procedure used, the size of the data set, and the form of the input. Additional information is recorded depending on the procedure used. To facilitate notification of users, software was written to extract records and, through an interface with the center's billing system, print mailing labels.

The STATJOB accounting and user notification system can provide other useful information. For example, statistics on the size of data sets were useful in designing the internal file (STJbank) system for STATJOB. An important potential use of the system is the identification of users who might assist in preparing specifications for new development or contribute in other ways to the support of statistical software.

The most recent STATJOB installation manual contains instructions for installing the accounting system at other sites, although billing system incompatibilities preclude use of the mailing label program.

Protection against the relatively infrequent "serious" errors is an important responsibility of package distributors. Our experience with STATJOB shows that the protection can be provided at a small cost (unless, of course, the package contains many serious errors). It should be the goal of distributors to maintain an account and an on-line file at sites using their package. This is already being done by the distributor of a new interactive statistical package, SCSS, although the file is kept for billing purposes rather than to permit direct communication with users of the package.

6. ACKNOWLEDGEMENT

Many of the concepts presented in this paper were proposed by Peter Wolfe, former Manager of Applications Programming at the Madison Academic Computing Center, who implemented much of the STATJOB system.

7. REFERENCE

Muller, M. E. (1969). Computers as an Instrument for Data Analysis. Technometrics, Volume 12, Number 2, 259-293.

BIOGRAPHY

James R. Allen received a Master of Science degree in Computer Science from the University of Wisconsin in 1967. He was Coordinator of Statistical Programming at the Madison Academic Computing Center from 1970 through 1976, and now is an independent consultant.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

THE DISTRIBUTION AND MAINTENANCE OF SAS

Anthony J. Barr, SAS Institute Inc.
P.O. Box 10066, Raleigh, N.C. 27607

ABSTRACT

The SAS system has been optimized for a single family of computers and operating systems. This has reduced the size of our universe of users, although it is still large. New portability problems arise out of our efforts to adapt more closely to the environment than is possible in FORTRAN or COBOL. We have tried to avoid requiring users to compile or link-edit SAS. Identical tape copies of the system are sent to all SAS users. A SAS program copies the tape to disk, optimally blocking the SAS library. The system is then configured by another SAS procedure which writes the installation-dependent configuration data into the disk copy of the program. Information relating to the environment that can be obtained from the operating system is discussed. The SAS communication mechanism between procedures and the supervisor is such that user-written procedures need not be re-linkedited or compiled for new releases of SAS.

Key words: Dated software; diagnostics; distribution; installation parameters; load modules; maintanence; portability; versions.

1. INTRODUCTION

Ease of installation and reliability are two prime goals of our installation procedures. To achieve these goals, we want to reduce the number of steps needed for installation and to make the installation process immune to the differences between installations. We assume that the person installing SAS has minimum knowledge of the system.

A system's portability domain can greatly affect its implementation. We have chosen as the portability domain for SAS all computers running variations of the IBM 360/370 Operating System. This domain includes IBM 360/370, Amdahl, Itel and Ryad computers. Other possible portability domains could be computers supporting the ANSI COBOL compiler or computers supporting a FORTRAN IV compiler.

Since we have chosen our domain to be operating-system-dependent instead of compiler-dependent, we are free to use the most appropriate features and languages supported by that operating system. For example, our group uses mostly PL/I for mathematical and statistical applications. Assembly language is used for data management, compiler writing and report generation features. Most of our users program in FORTRAN when they augment SAS with their own procedures.

2. LOAD MODULE DISTRIBUTION

SAS is distributed as a load module library on tape. The installatio process consists of copying this library to disk, where it can ru immediately. There is no need to compile or link-edit.

One reason we distribute SAS in load-module form is to reduce problem that arise from installation differences. Distributing source program results in many such problems:

1) Different compilers for the same language have different restrictions. long DATA statement may compile under our FORTRAN Gl compiler but not under user's FORTRAN H compiler.

2) Different releases of the same compiler have different bugs.

3) The optimizing compilers interpret the language rules more strictly than do the simpler compilers.

4) Sometimes a program requires a large amount of memory for compilation. On a small system, memory restrictions may make it impossible to compile such a program.

5) An optimizing compiler may not be available at the user's installation. We can compile the program with an optimizing compiler and he can run the optimized object program.

6) As it is usually expensive to compile a large system from source, users tend not to accept updates to programs that are not being used or that are running to their apparent satisfaction. Thus enhancements and fixed bugs are not available to users on a timely basis.

Load module libraries also have disadvantages. The IBM utility program IEHMOVE uses space inefficiently due to a poor choice of blocking factor. Thus, tape reels containing unloaded libraries must be large enough to hold the inefficiently blocked library. Copying costs are also increased. The newer program IEBCOPY attempts to solve these problems, but only works on virtual storage operating systems.

Another problem arises because load module libraries are link-edited with a specific block size. When a library is optimally blocked for an IBM 3330 disk unit, it can't be installed on an IBM 2314 disk since the 2314 has a smaller blocking capacity. If a library that is optimally blocked for the 3330 is installed on an IBM 3350 disk, which has a larger track size, considerable disk space is wasted and the program load time is more than if the library were link-edited for the 3350.

To solve these problems of distributing load modules, we have written our own program, PDSCOPY. PDSCOPY overcomes wasteful use of tape and automatically adjusts load modules to the blocking factor of the device on which they are written.

Added benefits are reductions in disk space and copy time. For example, we have reduced the number of tracks SAS uses on the IBM 3330 from 330 tracks to 272 tracks simply by using PDSCOPY to copy the program library. At the current North Carolina State University on-line storage rates, this would reduce the disk storage costs by $254 per year.

Our aim for PDSCOPY was to save tape and to eliminate the link-edit step for installation of SAS. It turned out that our blocking strategy was far superior to that of the IBM linkage editor, and we combined some program records in overlay programs that the IBM linkage editor did not. It appears from our 5 tests that we can save between 13% and 18% of disk space over the IBM linkage editor.

3. INSTALLATION PARAMETERS

Although our installations all have similar operating systems, their hardware differs considerably. Consider just line printer characteristics:

1) The line size may vary from 72 to 132 characters.
2) The number of lines per page varies widely.
3) The position of the paper when at "top of forms" varies.
4) The printer may only print the 48 most common characters.
5) Some printers do not have the overprint feature.
6) Some printers go faster if the lines are printed left justified.

SAS has 6 options that cover all of these printer differences. Other parameters give the suggested block size for SAS data sets, the installation rate charged for disk storage, the source statement length, the memory allocation scheme to use for the system sort, and the names of all I/O units for SAS to use, a total of 28 parameters.

An installation may run SAS in many environments, each of which requires a separate set of parameters for optimal execution. For example, in batch mode a large blocksize and the 48-character set might be needed. In time-sharing mode, the 60-character set may be available; memory limitations may dictate a smaller blocksize. This same installation may also support a high-speed autobatch mode for execution of short student jobs, in which some limitation on pages printed and time used may be imposed. SAS has a different set of installation parameters for each of these environments, plus 6 user-defined parameter sets.

With many software systems, it is necessary to change some source statements, then compile and link-edit them to adjust installation parameters. This can be awkward to document and hard to modify once the system has been installed. We have a special module that compiles the system parameter definitions. The module gives good diagnostics about misstated parameters and is easy to document. The SAS procedure SETINIT is used to write the parameters into the load module copy on disk. At any time, after thought and new considerations, changes can be easily incorporated into the installation parameters stored with the load module library.

Our company has an agreement with each installation that the system will only be used at one installation, and that the system will only operate if the yearly service agreement is in effect. To assure compliance, we want each copy dated and personalized with the installation's name. The procedure SETINIT was modified to set the expiration date and name in the load module. For example, the following sets the expiration date for SAS INSTITUTE to January 1, 1978:

```
PROC SETINIT NAME='7800123153SAS INSTITUTE INC.';
```

The digits 23153 are produced by computing a cyclical redundancy check on 7800lSAS INSTITUTE INC. The SETINIT procedure will not operate if anything is altered in the name or expiration date because the check digits will not compute correctly. Thus we do not have to send every user a unique copy of

SAS and we can make our tapes in large batches and inventory them.

4. ERROR MESSAGES

We give considerable thought to producing useful error messages Printing an error message in layman's language on the computer output wil often eliminate the need for a user to call us for help. A good example o the range of possibilities can be found in the evolution of our "memor exceeded" message.

1966-1967 COMPLETION CODE- SYSTEM=80A USER=0000

This was the IBM-supplied error message. To a novice computer user, it information content was almost nonexistent. He would certainly need t consult someone for help.

```
1967-1975  THE PROCEDURE NEEDED 183572 BYTES OF CORE
           ONLY 101324 BYTES WERE AVAILABLE.
```

Although this message contained more information, users still phoned us to ask what it meant. We would explain that the user needed to run SAS in a larger region. Then the user would ask what region size to use. We would ask him to check his cataloged procedure listing to see what had been used. Then we would add 183572 less 101324 to that given in the cataloged procedure and tell him to use that for the region size. The user would then ask how he could tell SAS to use the larger region. Then we would tell him to look for the EXEC card in his deck and change it.

```
1975-1976  NOTE: MORE MEMORY IS NEEDED TO COMPLETE TASK.
                 TRY THE FOLLOWING EXECUTE STATEMENT.
                 // EXEC SAS,REGION=212K
```

We now give the JCL statement the user should code to allow SAS to run the task. The format of the message changes depending on whether SAS is running under the operating systems OS/MFT, TSO, or another variation of the Operating System. The type of operating system and the memory size is determined by looking at system control blocks. The other formats of the above message are:

```
(MFT)      NOTE: MORE MEMORY IS NEEDED TO COMPLETE TASK.
           TRY A PARTITION SIZE OF AT LEAST 212K.

(TSO)      NOTE: MORE MEMORY IS NEEDED TO COMPLETE TASK.
           TRY SIZE(212) AT THE END OF YOUR LOGON LINE.
```

1977-Present (same as above)

We now change the name of the cataloged procedure in the message. Many users use different names for the cataloged procedures and support several cataloged procedures for SAS. Our message was not fully accurate except when the user's cataloged procedure was named SAS.

5. INTERFACE BETWEEN SYSTEM AND PROCEDURES

No supervisor code is link-edited with SAS procedures. This means that any changes we make to the supervisor are reflected in all SAS procedures without having to re-linkedit. A new release of SAS can be installed without impacting existing user-maintained libraries of procedures. These user-written procedure libraries are simply concatenated by JCL to the library we distribute. We use a branch vector technique to communicate between the supervisor and the procedure. It is similar to the technique used by the IBM overlay supervisor.
Below is an example of this technique.

A) The code in the SAS procedure:

```
     CALL INPUT(IEND)
```

B) The code link-edited with the SAS procedure:

```
BRANCHV  EQU   *
entryl   B     SAVE-*(15)
         ...
INPUT    B     SAVE-*(15)    called by SAS procedure
         ...
SAVE     LR     0,15         calculate offset from BRANCHV.
         S      0,=A(BRANCHV)
         L      15,LINKADDR  address of LINK.
         BR     15           go to LINK
LINKADDR DC     A(0)         address of link
```

The code in the supervisor:

```
LINK     LR     15,0         offset from branch vector.
         L      15,VECTOR(15) address of INPUT subroutine.
         BR     15           go to INPUT
         ...
VECTOR   DC     (entryl)
         ...
         DC     A(INPUT)     address of INPUT subroutine.
```

6. VERSIONS OF SAS

Special versions of SAS are used in the operating system LPALIB and i the Autobatch mode of execution. LPALIB is a library of programs kept i virtual memory by the operating system. When the SAS supervisor and compile are stored in LPALIB, concurrent users of SAS can share the same copy of SA in memory. To access the SAS supervisor, the system loader does not have t be called. If the module is already in main memory, it can be entere directly without any I/O. If it is not in memory, it will be read in from th virtual memory swapping device, a very efficient operation.

Only reentrant programs (programs which do not modify themselves) can b put in LPALIB. Use of this library reduces the amount of I/O and memor required to run SAS. It is especially advantageous when running SAS i timesharing mode.

Many universities have a mechanism, called autobatch, for running shor student jobs. The operating systems collects several SAS jobs into a batc which is then fed into the SAS processor. For this application, SAS i generated in three versions:

Case 1. Minimum memory of 150K is available to autobatch SAS.

Case 2. At least 200K memory is available to autobatch SAS. In this case the SAS compiler is kept in memory for the entire batch of jobs.

Case 3. SAS has been installed in the LPALIB system library. In this case the autobatch supervisor uses the code that is in virtual memory.

7. FIELD MAINTENANCE

Most errors in SAS are corrected by new releases. Between releases, we send corrections to the load module library as patches. These are short one-to-ten-word changes to the object code, which are applied by an IBM utility program. Most assembly language errors can be corrected this way, and some critical problems in our PL/I programs have been corrected by patches.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

RECENT DEVELOPMENTS IN THE MAINTENANCE AND DISTRIBUTION OF BMDP

James W, Frane
Health Sciences Computing Facility, UCLA, Los Angeles, Ca. 90024

ABSTRACT

Health Sciences Computing Facility at UCLA distributes the BMDP series of biomedical computer programs as FORTRAN source and as load modules to IBM 360 and 370 OS facilities. New releases are made approximately twice yearly. The in-house version undergoes constant improvement. Our concerns include error reporting, selection of improvements and new features, extensive testing after modifications have been made, update notices and newsletters, changes in user documentation, interface with other packages, portability and implementation on non-IBM computers, reliability of tape copies, delivery of tapes by the Postal Service and United Parcel, installation documentation, and monitoring actual usage. Our chief concern - beyond correct results -- is monitoring the use of our programs to be sure good analysis is being done.

Keywords: Errors, improvements, installation, portability, testing

1. INTRODUCTION

Health Sciences Computing Facility maintains and distributes two statistical packages: the BMD and the newer BMDP. The BMD series is now rarely updated, although the manual is reprinted about once a year. Improvements are constantly being made to the BMDP series and a tight quality control detects possible errors that may be introduced by the improvements.

In the last year, 522 copies of BMDP and 237 copies of BMD were distributed by Health Sciences Computing Facility. The most recent version of BMD is dated 1975 and the latest version of BMDP is dated April, 1977.

2. MAINTENANCE OF BMDP

Reports of errors and suspected errors are logged into a computer file with details of what caused the problem. The errors are then checked out and corrected. Conditions causing known errors are checked by the program, and the program terminates with an error message if these conditions are met. A list of restrictions in each program is included in the heading of the output; the heading is also used to describe any features not included in the BMDP manual (Dixon, 1975). Roughly twice a year the distributed version of BMDP is updated. Official recipients are sent an update notice describing the changes made.

Suggestions for improvements are also logged into a computer file. As time permits, assignments are made to staff members to implement improvements. Error correction, of course, takes precedence over implementation of improvements.

Updates of programs are refereed; refereeing is both human and mechanical. Visual inspection of computer output is tedious and prone to error, so programs have been written to compare output from the proposed new version of the program with the current output, or with output stored in a library of official output, or with output produced by the load modules that are distributed outside HSCF.

The value of such comparison procedures is highly dependent on how extensively the test problems exercise the program in question. A highly successful strategy proceeds as follows:

a. Create test problems that exercise all options (but not all combinations of options).

b. Add or modify test problems to constrain the program (e.g., zero variance, near singularity, all cases with one or more missing values, etc.).

c. Feed the entire collection of tests into Gordon Sande's FORTRAN execution profiler. This profiler now reveals sections of coding that are only executed when certain options are used in combination or that are executed depending on the data values.

d. Add tests and try the profiler again.

Although this strategy of creating test decks has been used on some BMDP programs, it has not yet been used on all of them. However, the library of test problems contains several tests for each program.

We maintain several libraries (with separate versions for in-house use and outside distribution): source, load modules, manual sample input and output, overlay structures, and update notices. In-house, we have additional libraries for test version of load modules, extensive test input and output, updates, update procedures, and software tools for output comparison, execution profiling and comparison of different versions of the source.

From time to time new programs are added. Each new program is reviewed by a committee and by consultants outside UCLA. The committee include HSCF staff members and members of other departments at UCLA.

Maintenance of any package includes maintenance of the documentation. The newsletter, BMD Communications, is a primary vehicle for updating the BMDP manual with respect to existing BMDP programs.

In August, 1976 we reissued BMDP1F - Two-Way Contingency Tables - and began to distribute four new programs:

BMDP2F - Two-Way Contingency Tables -- Empty Cells and the Identification of Departures from Independence

BMDP3F - Multiway Contingency Tables

BMDPAM - Description and Estimation of Missing Data

BMDP9R - All Possible Subsets Regression

Complete writeups for these programs can be purchased from HSCF. Abstracts of the programs were printed in the newsletter, BMD Communications, which is distributed without charge - a note to us will add your name to the mailing list.

A completely new edition of the BMDP manual will be ready for the publisher in July, 1977 and for distribution in November, 1977. It will include thirty-three program descriptions (the 1975 BMDP manual has twenty-six) and an index. The new style contains more discussion of the options. Discussion of options is tied to annotated computer output.

One of the most important aspects of program maintenance is observing the way the programs are used. Many improvements are made (and notes added to the output) as a result of monitoring usage. While several programs are written for data screening, we find that many users skip immediately to regression or multivariate analysis, so we have included data screening as an integral part of advanced techniques. In our April, 1977 update, we include computation of residuals in BMDP2V, separate variance ANOVA in BMDP7D, and Cook's (1977) measure of the influence of each case on a set of regression coefficients in BMDP9R.

3. DISTRIBUTION

Health Sciences Computing Facility at UCLA distributes the BMDP series as FORTRAN source and as load modules to IBM 360 and 370 OS and OS/VS facilities. At the present time, we do not have a version that is as easy to install for IBM DOS since we do not yet include the job control language for DOS. Conversions of the FORTRAN source for other computer types have been made by special redistribution centers. The non-IBM versions are not all kept completely up-to-date with the IBM version. Non-IBM versions include CDC, Honeywell, Univac, PDP 10, PDP 11, HP 3000, Riad, Hitachi, etc.

The basic distribution tape for IBM OS and OS/VS consists of load modules for all the programs, the FORTRAN source, input and output for the examples in the BMDP manual, overlay structures, and the procedure for running the programs. These are written as partitioned data sets onto tape with the IBM utility routine IEHMOVE. Distribution of BMDP in load module form began with the August, 1976 release. The FORTRAN H compiler with OPT=2 is used in generating the load modules.

For DOS and non-IBM facilities (and some OS facilities), we write the FORTRAN source sequentially with a variety of options regarding block size, tape marks between programs, number of tracks, etc.

Since we distribute several hundred tapes each year, an experiment was performed with three different brands of tapes. We found Memorex were best. Most problems that we have had with tapes have been related to the clarity of the installation instructions, treatment by the Postal Service (we prefer United Parcel), and gross errors by the recipient such as writing on the tape.

We distribute BMD and BMDP for the cost of handling, which includes a new tape, writing it, telephone calls, correspondence, update notices, etc. At the present time, the charge is $100 for each package each time a tape copy is made. Requests should be made on one of our special forms. The tape copy request brochure includes a list of redistribution centers for non-IBM computers.

BMDP programs are not portable in the sense of being originally written completely ready to run on a variety of computer types, but they have been converted to a large number of computer types. The Bell Laboratories FORTRAN portability verifier is a major help in reducing machine dependencies.

The IBM 360 H-compiled versions of the BMDP programs require about 160K bytes including buffers. All arrays are dynamically determined as subarrays of a single array of 15,000 words. Almost all of the programs can run with 5000 or fewer words of array storage. A version that runs in 120K bytes can be made by relinkediting with two small modified subroutines. Reduction of the array storage has also been used in conversions to smaller computers such as PDP-11/45 and HP-3000.

Implementation on smaller computers is facilitated by the modularity of BMDP. While the programs are integrated through a common control language and self-documented save files, not all programs need be kept on line and a subset can be easily converted to non-IBM computers.

HSCF has a 360/91 computer; its FORTRAN subroutine library differs from that of other 360 and 370 models. When the load module library for distribution is created, the programs are linkedited with the standard FORTRAN library. Output from these load modules is compared with the in-house version.

4. ACKNOWLEDGMENT

This work is supported by NIH Grant RR-3.

5. REFERENCES

COOK, R.D. (1977). Detection of influential observations in linear regression. Technometrics 19, 15-18.

DIXON, W.J. and M.B. BROWN (1977). BMDP Biomedical Computer Programs, 2nd edition. Berkeley, University of California Press.

BIOGRAPHY

James W. Frane is Supervising Statistician in charge of BMDP programming at Health Sciences Computing Facility, UCLA. He holds a Ph.D. degree in mathematics from the University of Kansas. He is the author of the BMDP programs for factor analysis, partial correlation, canonical correlation, all possible subsets regression, and description and estimation of missing data.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

PORTABLE STATISTICAL SOFTWARE - IN COBOL

J. Michael Hewitt
International Statistical Programs Center
Bureau of the Census, Washington, DC 20233

ABSTRACT

The paper discusses the search for a means of producing statistical software capable of executing efficiently on a wide variety of computers. The reasons for the selection of COBOL are cited and the suitability of COBOL as a system development language is covered. The paper includes details on maintenance of versions for 16 different mainframes. Mechanics of distribution of the system and updates to over 80 users in more than 40 countries are presented. The paper concludes with a retrospect on the success of the COBOL approach and plans for future COBOL-based statistical software systems.

Key words: Census tabulations; COBOL; COCENTS; large files; portable software; program generator; publication-quality tabulations; software distribution; software maintenance; tabulation system.

1. INTRODUCTION

The particular piece of software that will be discussed in this paper is called COCENTS, an acronym for COBOL Census Tabulation System. The title for this paper could be more accurately stated as 'Portable Tabulation Software ...', since COCENTS is a tabulation system, like TPL or CENTS-AID II, and not an analysis system, such as SAS or SPSS. It is 'generalized software' since it is completely driven by user parameters. And it is surely 'portable', currently being operational on about 20 distinct central processor architectures.

2. PAST

In 1972 the United Nations expressed to the Bureau of the Census the need for a tabulation package for census use that would operate on a wide variety of non-IBM 360 computer equipment. The International Statistical Programs Center (ISPC) of the Bureau, funded by the Office of Population of the U.S. Agency for International Development, at that time was distributing a tabulation package. This system, called CENTS, was already being used in many countries for the tabulation of censuses and surveys. CENTS was conceived in the late 60's by Howard G. Brunsman, formerly Chief of the Population Division at the Bureau, and now a consultant to the Bureau and USAID. The CENTS system, as implemented by Brunsman and Bureau computer technicians, was a parameter-driven interpretive system written in IBM 360 Assembly Language. Minimum memory requirement was 32K bytes, and 3 tape drives were usually used for the sorting phases. This system worked well, but being written in an

assembly-level language, was obviously not portable to other machine architectures. Since at that time many developing countries, the recipients of technical assistance from ISPC, had computing equipment with only 16K characters of main memory, the CENTS system was not appropriate from this standpoint either.

In June of 1972 a prototype tabulation program, written in COBOL, was benchmarked against the data tabulation portion of the CENTS system. This hand-coded prototype ran in about 30 per cent less time than its counterpart section of the CENTS system. This benchmark proved that it would be worthwhile to develop a tabulation system written in COBOL. A very limited COBOL language subset was chosen, compatible with the known target computers, and a design goal of execution in 16K characters of memory was specified.

In February of 1973, after about 6 man-months of work, the first operational installation of COCENTS was made on a 16K IBM 1401 in San Jose, Costa Rica, at the Department of Statistics and Census. The documentation and some revisions to the system delayed general release of the system until September, 1973. Since that time COCENTS has been widely distributed in the international statistical community, and to a lesser extent, domestically.

3. PRESENT

The COCENTS system is a complete tabulation package for producing the results of censuses and surveys. Its parameter language is rather primitive when compared with TPL and some other systems (see Languages and Programs for Tabulating Data From Surveys by Francis, Heiberger, and Sherman in the proceedings of the Ninth Interface Symposium). It does have a number of characteristics, however, that can make it the best tabulation package for producing census results on small- to medium-scale computers.

- The parameters are easy to use, even if their format is not intuitively obvious. A complete tabulation can be specified in an hour or two, and no data dictionary is required.

- It can process the type of hierarchical files found in censuses of population and housing.

- It requires very little memory, and can produce many tabulations with one pass of the data file.

- It executes extremely quickly - estimates range from 4 to 10 times as fast as some other widely used tabulation and statistical packages.

- Tabulations can be produced in small runs if desired, and later consolidated for publication by a standard system module. Computer equipment in many locations can not be counted on for longer than one or two hours at a time, so this can be a vital attribute.

- Most importantly for censuses, it produces publication-quality tables, ready to be photographed for a plate and then published as the final result of the census effort.

COCENTS currently has over 80 known users in more than 40 countries around the world. These are conservative figures based on information in ISPC files. COCENTS is also distributed by other divisions of the Bureau of the Census, and by the United Nations in Bangkok and Santiago. ISPC has recent correspondence from over 60 separate users.

The COCENTS system for any single computer is composed of about 6500 lines of code and comments. Versions of the system exist for about 16 different CPU's in ISPC's files, and individual users have converted it to a number of other configurations.

In its present form COCENTS is an effective tabulation tool, and for some computer systems, the only one available!

4. FUTURE

COCENTS (as a package with that name) is not scheduled for any further enhancements. Since its inception in 1973 one major revision has occurred which essentially doubled the minimum memory requirements (to 32K characters). A few more such improvements would negate some of the original package features. It currently fulfills the goals for the package as specified by ISPC and USAID, and most National Statistical Offices using it for census processing.

COCENTS has a continuing and substantial effect on the domestic statistical community through its offspring. The CENTS-AID II tabulation package from DUALABS is an enhancement to the COCENTS system that provides an easier to use specification language for the COCENTS internals. This has obviously been a successful effort since DUALABS has distributed over 50 copies of CENTS-AID II, mostly through the National Technical Information Service.

COCENTS has been used in many divisions at the Bureau of the Census, and the System Software Division there has developed an enhanced version of COCENTS for use by the Bureau. This system, called GTS, for Generalized Tabulation System, again is a development of COCENTS with a language for the user that is intended to be easier to learn. The package is to be the basis for a complete tabulation system for all Bureau divisions.

Finally, COCENTS has had a considerable influence on its users in how they produce software. COBOL has gained respectability for speed of processing, and many users report that they are writing similar software systems using COBOL and the COCENTS techniques for many different tasks!

5. WHY COBOL?

COBOL was chosen as the computer language in which to write COCENTS for one reason only: the availability of relatively similar COBOL compilers on a wide range of small- to medium-sized general-purpose computer equipment.

There have been a number of very relevant pluses to the use of the language:

- More or less self-documenting code can be written.

- If the data is carefully specified, and the proper verbs are used, very 'tight' and efficient machine code is possible. Overall, it is our opinion, in the light of our experience with tabulation systems written in both COBOL and assembly language, that computing efficiency equal to that possible with a large assembly language system is readily attainable in COBOL.

- The COBOL code is easier and quicker to write than an assembly language, and contains fewer program 'bugs' both initially and throughout the software life-cycle of fixes and enhancements.

- The various implementations of COBOL appear nearly identical, if only a subset of the full COBOL language is adhered to. The result is that if the COCENTS programs compile correctly on a given computer (after the necessary program changes for the new system), they give the correct answers. The only deviations from this have been in the case of provable 'bugs' in the target COBOL compilers. These have occurred on only three compilers in ISPC's experience. This situation is quite different from that for transporting FORTRAN programs, where a clean compile is only the beginning of the task! FORTRAN source code is known to often yield different results from different compilers.

6. SUITABILITY OF COBOL

There are a number of problems with using COBOL as a development language for large generalized software systems. All of them can be overcome or programmed around, using standard COBOL, with various impacts on the effectiveness of the code. The common result of these difficulties is a significant penalty in execution speed over equivalent code in assembly language. In the discussion that follows, remember that the chosen COBOL language must be common to all of the target COBOL compilers. There may be special constructs in some compilers that solve the problems, but were not usable because of a lack of widespread adoption.

It should be pointed out at this time that the COCENTS system contains two types of programs: interpreters and compilers. An interpretive program deciphers the requested function from the user input and immediately executes it. The deciphering process is repeated each time the interpreter is asked to execute the function. A compiler deciphers the requested function and composes a sequence of instructions which can later be executed, as many times as necessary, without referring to the original request. Compilers are typically much faster than interpreters in the actual performance of the requested function, and COCENTS recognizes this by containing a compiler for the portion of the system that processes the user's data file. The other two primary phases of the system process the user specifications interpretively since the smaller files they work with make speed of execution a less significant factor. The term 'generator' is more common than 'compiler' when the code produced is not machine language, but COBOL, as in COCENTS. Thus one speaks of the COCENTS 'generator' where the function is compilation into COBOL rather than into machine code. This use of a program generator solves some of the following problems normally encountered with COBOL as a language for programming generalized systems. The price paid for the solution is the time necessary for the COBOL compiler to produce the final executable machine code. This is a reasonable trade on large files since the compile time is trivial compared to the time required to process the data.

The COBOL language does not provide for modularity. Indeed, some of the target machines do not even have a linkage editor to collect program modules into an executable unit. The entire executing program has to be presented as one piece. The result is that all names are global to the entire program. This makes it very difficult for more than one person to work on the development of a single program. The COCENTS system was not impacted by this problem since a memory limit of 16K bytes (with no overlays permitted) does not allow for large programs. This limitation has been addressed in other situations by having each programmer use a special prefix on all names - this is workable, but not really satisfactory.

In addition to the global name problem, COBOL also does not provide parameter list facilities for the in-program subroutine calling verb, the PERFORM statement. Parameters must be moved to special data-names used only by the subroutine. In the initial COCENTS research it was determined that in many cases it took as much memory space and execution time to merely pass the parameters as it took to actually replicate the required code in-line.

The most significant problem with using COBOL for generalized systems is that the sizes of all data areas and data items must be fixed at the compile-time of the executable program unit. When writing an interpreter this means that all internal arrays and storage areas have fixed sizes - the interpreter is not able to decide dynamically, based on the content of the parameter cards, how much memory to allocate to each item or function. Since in a tabulation system the user must specify the size of the final tables, this is a crucial problem. Fortunately it can be resolved by using the generator approach. This technique delays the compile of the executable unit until the user-specified table sizes are known. (It can also be solved with an interpreter, at some expense in execution time, if data areas and table areas large enough to handle most tasks are reserved - this of course greatly increases the memory requirements for the program.)

When writing generalized software in assembly languages, arbitrary fields of arbitrary length are dealt with by referring to the address of the field. The length factor is then used with this address to determine the proper data item to move. Both the address and the

length can be modified at execution time in this assembly language reference scheme. No data names need be used for the desired items in the assembly code other than for the base addresses.

Contrast this situation with that in COBOL where a data-name is defined at compile-time to have both a fixed address and a fixed length. There is no direct way to use the machine's addressing facilities from within a COBOL program. These facilities can only be simulated, at a considerable penalty in execution speed, by the COBOL subscripting mechanism. With subscripting, any field necessary can be moved - but only one character at a time. A 7 character field would require 14 subscripted references, since both the sending and the receiving fields must be subscripted. Each of these 14 subscripted references involves from 4 to 50 or more machine instructions, whereas in assembly language the same move would require a total of three or four machine instructions. It is easy to see that the COBOL subscripted move will take from 16 to 200 or more times as long as the assembly language version. The solution, of course, is to generate a program - the parameters are known, so a data name with the correct address and length attribute can be specified, and the COBOL compiler can emit the proper code to accomplish the move in less than the three or four instructions of ISPC's assembly language interpreter CENTS - probably in one machine instruction.

Finally, COBOL contains no verbs designed to aid in writing compilers, such as a SEARCH, or a SCAN-TO-DELIMITER, or a TRANSLATE-AND-TEST. The aspiring COBOL programmer is necessarily reduced to subscripting character-by-character through the user's parameter cards. This works satisfactorily, but it certainly is slow!

Despite this depressing list of deficiencies, the compiler (or generator, if you prefer) for the COCENTS system works pretty well. The time for the COCENTS generator is only a fraction of the time taken by the COBOL compiler to compile the generated program, and the total of this time is not significant when compared to the processing time for other than very small data files. All phases of the COCENTS system, in fact, execute with very satisfactory timings on most computers. The single variant not controlled by the COCENTS system is the quality of the code output by the COBOL compiler. This does vary greatly, and some quite competent computers are crippled by inadequate COBOL compilers that generate subroutine calls for simple functions, rather than straightforward in-line executable code.

7. MAINTAINING THE SYSTEM

The subject of this session, maintenance of the software, is really the stage COCENTS has been at since 1973. It was decided very early in the development of the system that it would be an impossible task to maintain different source decks for every computer version. With all the best intentions, the multiple sets of code will tend inevitably to diverge. Patches or modifications made to one system will just not get into the the others. When enhancements are made they turn out slightly different for each version, perhaps involving some compiler-specific feature or defect. The end result is the maintenance of as many quite different software packages as there are versions for different computer systems.

The solution chosen for COCENTS was to have only one source deck for each program. This deck will contain all the variations necessary for the different versions as comments. The non-comment code is the master version, currently that for the IBM 370 using OS. A text editor is used to maintain and extract the different COCENTS versions from each single source file. The text editor is used to make the proper mass changes on a program that will deactivate the master S360/OS version, and make active the target version, say for an ICL 1903A to be used in Jakarta. Statements are made active or inactive by changing the character in column 7 of each COBOL line. An asterisk there makes the line a comment, while a space makes it valid executable code. Additionally, the text editor is used to perform some other clean-up operations with its mass-change facilities.

There are four basic types of changes that can be required to obtain a different computer version from the master decks. The first operation required is to make the necessary changes on each occurrence of a particular character string. The primary target for this in COCENTS is the USAGE clause which describes the format of each data item (for example, as binary or internal decimal or external decimal). For the ICL 1900 series it is

necessary to change all occurrences of COMPUTATIONAL-3 to the words COMP SYNC RIGHT, while for the IBM SYSTEM/3 or the Control Data 3100 each occurrence of COMPUTATIONAL-3 needs to be changed to COMPUTATIONAL.

The second type of change is to substitute code for certain broad classes of computers. Some computer systems allow the use of the COBOL feature called 'indexing' in addition to subscripting for array reference. Two of the COCENTS programs use indexing, when available, because of a considerable decrease in execution time requirements. (Indexing is only advantageous when stepping through an array, or when making multiple references to the same element of the array.) In these two programs there are sections of code using indexing, marked by INDEXING in positions 73 through 80 of the source line, following sections of substitute code for subscripting, marked by SUBSCRPT in positions 73 through 80. Since the IBM 370 COBOL compilers support indexing, the master version has column 7 blank when INDEXING is in columns 73 to 80, and column 7 is an asterisk (*), indicating a comment line, when 73 to 80 contains SUBSCRPT. To prepare a distribution version for the ICL 1900 series, which does not support indexing, the statements for indexing must be eliminated and those for subscripting must be activated. The third type of change that is required is to substitute lines of code that are unique for each computer system. The prime example of this for a COBOL program would be the preparation of the SOURCE-COMPUTER and OBJECT-COMPUTER paragraphs - the entries are unique for each different computer version. Uniqueness is not the usual case, however. More often the entries for a few systems differ from the pattern for all the others. Whether the ACCEPT or READ verbs are used to input user parameters typifies this kind of change.

The only tedious aspect to this type of change is that if a maverick computer is encountered, and there is one notorious example in COCENTS, then those unusual lines must be duplicated in their standard form for all of the other computer versions. Once this is completed, however, it requires no extra steps in the version preparation, and only increases the space on the disk used for program storage. It is tiresome to set up the extra lines originally, however.

The Fujitsu FACOM 230-15 computer system does not contain the ALTER verb in its vocabulary - an omission which in general is praiseworthy, but which was a very painful discovery during the initial COCENTS conversion for that machine. (The ALTER was used in COCENTS to save memory and reach the 16K target configuration.) It was necessary to work out some method of simulating the ALTER statement for the F230-15. This required replication for each computer version of all lines in the source file containing the ALTER verb.

The ALTER problem was virtually the only non-common code necessary in the PROCEDURE DIVISION other than that for some input and output statements and the INDEXING substitutions. Use of a sufficiently small COBOL subset prevented most other dissimilarities in the PROCEDURE DIVISION. This system of different lines for different versions is used extensively in the DATA DIVISION, however. Different REDEFINES patterns and different item lengths resulting from machine architectural differences cause considerable variation between versions.

The final type of change required is for character sets. Some computers, such as the IBM systems, use single quotes (') to bound alphanumeric literals; other systems, such as the Burroughs or Digital Equipment compilers, use double quotes (") to bound these literals. Some systems require more extensive character set changes. If the computer system uses BCD (Binary-Coded Decimal) or the special ICL character set, the required changes can be made with the text editor. This was found to be too costly, though, for more than a one character change. A program was written, in COBOL, using IBM's TRANSFORM verb that generates a machine-level TRANSLATE instruction, to do the character set conversions. This put the version preparation cost back to a reasonable amount.

One other modification is usually made to each program. This can be done with the text editor, but is more often performed on the target computer system. Many internal tables are marked in positions 73 to 78 with the identifier EXPAND. These table sizes need to be expanded or contracted, as required, according to the memory available on the target computer system and the needs of the installation. The advantage of keeping all the change-required symbols within an 80 position record is that this type of change is possible to recognize and perform at the installation site.

Source line positions after column 80 are used in the master program files, however. When a correction or enhancement is made to the system, the indicators

ADDED date
or REPLACED date

are suffixed to the end of the line, starting in column 81. This permits the determination of whether a problem is a new one, just added, or an old bug hanging around waiting for discovery. Knowing this, the releases and versions which require notification of the problem can be pinpointed.

The above techniques give us a method of maintaining different versions of the COCENTS system along with the master version. When corrections or enhancements are made, it is to common code. There have not been any instances of incompatibilities developing between versions because all code is kept together. As far as testing of changes, it is of course not guaranteed, but because the code is mostly common code, if it works on the 370 version normally used for production at ISPC, it will probably work for all the versions.

8. DISTRIBUTION

ISPC is in a unique position as a distributor of software because the costs of distribution are not a direct factor. ISPC is funded by the USAID Office of Population to distribute COCENTS throughout the developing world for use in population-related projects. No charge is made to the recipient for systems distributed by ISPC; all costs are covered by the contract between the Bureau of the Census and USAID. (ISPC developed software is available domestically, for a nominal copying charge, through the Data Users Service Division of the Bureau of the Census.)

The software is usually distributed in source program form on magnetic tape. Also included in the distribution package is a listing of the entire system and the appropriate COCENTS manuals. The documentation (also maintained using the text editor) can be supplied on the tape if requested.

The format of the distribution tape is important because it contains two copies of the COCENTS system. It is desirable to send two copies since in many remote locations ordering another tape because of a read error in a file can cause a lengthy transportation delay. These two copies of the system are not identical, however. The first file on the tape contains all of the COCENTS material, concatenated together. If the user intends to punch the system out on cards (and most small installations do) then only this file needs to be punched. Following this first file all of the separate programs and test-data groups follow as separate files, one item to a file. If the recipient has a text editor or a source program library facility, these separate files can be used without conversion to punch cards.

Distribution of software fixes and enhancements is nearly as important as the distribution of the original system. Virtually all software systems of any complexity contain errors and omissions that require correction or at least notification. The method used by ISPC to address this continuing maintenance problem is known as the COCENTS PROBLEM REPORT (CPR). This is a single sheet of yellow paper containing four sections. The first section discusses the COCENTS version and release dates that the CPR applies to. The second section describes the characteristics of the problem. The third section addresses the solution to the problem, or advises on the lack of a solution. The last section lists any supporting documentation which may be attached. These sheets, with the supporting documentation, have been sufficient for all problems encountered to date.

The key to the success of the CPRs is that the source code for the COCENTS system is always distributed. In addition, this source code is relatively straight-forward COBOL, with every paragraph name and data name having a unique numeric prefix. It is therefor very easy to state in a few sentences which source code statements are to be replaced, or where new lines of code are to be added. The COBOL sequence numbers in positions 1 to 6 are not

used in this process since they are changed as new computer versions are added.

One important point is that the user is never told to delete records from his source file. If a line needs to be removed from the program, the user is told to replace it with a card that is blank in columns 8 through 72. Of course the card in the master deck contains the indication that it was replaced along with the date in the columns after column 80. This ensures that a trail of changes is left in the files, and reduces the possibilities for unsuspected error on the part of the user.

A major enhancement to the system was successfully distributed in printed form. The user was given a listing of source records to add with detailed instructions for adding them. The release letter included procedures for compiling the programs and testing the enhancements. This was a satisfactory distribution procedure since the COBOL compiler aided the user by detecting most transcription errors - compiler syntax errors are the usual result. Of course, new copies of the system, for any computer version, are available on magnetic tape. Most enhancements are, in fact, distributed in this manner.

9. EVALUATION

Looking back after nearly five years of working with the COCENTS system, and with COBOL, it must be concluded that the approach was far more successful than was expected at the time. Original plans included only the IBM 1401, 360/20, and SYSTEM/3, and perhaps the ICL machines. Instead, rather than being merely an adjunct to the assembly language tabulation system CENTS, COCENTS has generally replaced it, even for use on the IBM 370s using DOS and OS. The expectations that were raised at that time that COBOL programs were too large and too slow have not been proved valid.

The COCENTS system, in COBOL, was produced much more quickly than its assembly language counterpart, and has turned out to have had far fewer bugs. The transferability from computer to computer has exceeded what was thought possible. Finally, enhancements have been easily integrated into the system.

Five years ago many had hopes that a new, more modern and more complete programming language would be developed and gain acceptance on a wide variety of computers. Today it appears that COBOL and FORTRAN are more entrenched than ever.

History does repeat itself: today ISPC is developing a generalized tool for editing statistical data called CONCOR. But this is not a new system - it is a COBOL version of a previously developed assembly language system. COBOL still offers the best avenue for the development of portable statistical software!

10. REFERENCES

Francis, I., Sherman, S. P., Heigerger, R. M., (1976). Languages and Programs for Tabulating Data from Surveys. Proc. 9th Interface Symposium on Computer Science and Statistics. D. C. Hoaglin and R. E. Welsch, Editors. Prindle, Weber and Schmidt, Inc.

U. S. Bureau of the Census (1977). COCENTS: COBOL Census Tabulation System, Series ISPC 4, No. 2, 1.3.

U. S. Bureau of the Census (1977). CONCOR: Editing and Imputation System for Censuses and Surveys, Series ISPC 4, No. 4 (draft).

BIOGRAPHY

J. Michael Hewitt is Chief, Computer Methods Laboratory, International Statistical Programs Center, Bureau of the Census. He developed the COCENTS tabulation system, and is currently directing the development of the CONCOR generalized statistical editing system.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

DISCUSSION: WORKSHOP ON MAINTENANCE AND DISTRIBUTION OF STATISTICAL SOFTWARE

William J. Hemmerle
University of Rhode Island, Kingston, R. I. 02881

A university environment with its diversity of interests and objectives is not well suited to software maintenance. The faculty are apt to be interested in research--new algorithms, techniques, languages, systems constructs. Most of the graduate students are concerned principally with completing their degree requirements and obtaining a permanent position. No one is particularly interested in maintenance and there is a high turnover of personnel at the programming level.

There is one advantage with respect to documentation, however, when the student must prepare an acceptable thesis. Many of our theses in statistics as well as computer science involve writing a program or programs in support of the research. The major professor can insist that a computer listing of the program be incorporated as an appendix to the thesis. At least in this manner, you retain a copy of the program and you can also get included some auxiliary documentation on how to use the program. But, by and large, graduate students, particularly at the Masters level are not as thorough as they should be in: a) testing their program; b) making the program user oriented; c) documenting their program or; d) having other people use the program from the documentation.

In years past, I tried to insist that programs appearing in thesis appendices be written in standard FORTRAN for possible use elsewhere. I would question the student on the care that he had taken to do this and his confidence that this was the case. I never received a reply that was fully satisfying. The development of FORTRAN verifiers such as PFORT [4] have, more or less, eliminated this problem. Timing of algorithms was another problem. If you determined analytically, by counting operations, that an alternate approach produced a speed up by a factor of 4 in some part of the algorithm, you wanted verification that this was in fact true. Software monitors are now available which permit obtaining reasonably accurate timings. These analytical aids, verifiers and monitors, are perhaps most valuable when one is dealing with relatively inexperienced programmers or software developers.

For several years now we have received support from NSF on a project to develop new algorithms for statistical computation. Various new algorithms have been developed analytically and implemented computationally. Emphasis has been placed upon iterative A.O.V. algorithms for unbalanced data, algorithms for variance component estimation for the general mixed model, and biased estimation procedures (see for example [1], [2], and [3]). Although primary interest is in the algorithm development, we would nevertheless like to have a transportable (correct) program available for anyone who is interested in applying or experimenting with these algorithms. Furthermore, the general computer implementation of an algorithm, with some attention to usability, is frequently a very suitable topic for a Master's thesis. (I have always been troubled with use of the word algorithm. If Pete Nitney programs Euclid's algorithm, do we call Pete's program "Nitney's Algorithm"?)

Three successive M.S. students have worked on different phases of development of the iterative A.O.V. algorithm, 2 successive students on the mixed model algorithm, and 2 successive students on biased estimation procedures, each borrowing upon the work performed in the previous implementation. (In addition, they usually had some rudimentary program that had been written to confirm the analytical work.) There are some problems associated with apportioning the development and implementation of general application programs over a

succession of graduate students. For one thing, the programs seem to mushroom since each student is apt to build his extension on the previous base. Another problem is that of emphasis and the last extension may tend to unnaturally dominate the overall effort. We were embarrassed when we found out that the version of the iterative algorithm we were distributing which included a covariance extension would not handle an analysis of variance because of control information. The student whose thesis project was to implement the covariance extension diligently checked out all sorts of covariance problems but apparently neglected to run an analysis of variance. To prevent such things from happening, it helps to give someone who has not been involved with the project the assignment of being an outside recipient of the programs, isolating him as much as possible from the developers. If he starts from scratch with the program tapes and documentation to run a host of examples, then it has been our experience that both the usability of the programs and the quality of the documentation are materially improved as a result. However, the appropriateness of this use of research resources at a university is perhaps somewhat questionable.

We have a problem inasmuch as many of the algorithms are more suited to interactive use than batch and the programs for the most part are developed interactively. The interactive version is definitely non-standard so it must be suitably modified or converted into a transportable batch program. It is unfortunate that you still have problems in preparing transportable interactive algorithms. I really do not think that there is much of a problem anymore with transporting batch programs provided that you are willing to be restrictive with your language (e.g., standard FORTRAN) and do not do such things as code machine dependent items, such as one or two line random number generators, in the higher level language. We have used the PFORT verifier and have successfully transported large verified batch programs as far away as CSIRO, from IBM to CDC equipment. Things have improved tremendously. I can remember back in the early 60's at Iowa State--we had an IBM 7074 (not an IBM 7094) with a non-standard 20k memory and a non-standard compiler which permitted an intermix of FORTRAN and Assembler statements. I do not think that you will find very many installations today that are willing to be that "different".

REFERENCES

[1] Hemmerle, W. J. (1975). "An Explicit Solution for Generalized Ridge Regression," Technometrics 17, No. 3, pp. 309-314.

[2] Hemmerle, W. J. (1974). "Nonorthogonal Analysis of Variance Using Iterative Improvement and Balanced Residuals," JASA 69, No. 347, pp. 772-778.

[3] Hemmerle, W. J. and Hartley, H. O. (1973). "Computing Maximum Likelihood Estimates for the Mixed A.O.V. Model Using the W Transformation," Technometrics 15, No. 4, 819-831.

[4] Ryder, B. G. and Hall, A. D. (1975). "The PFORT Verifier," Computing Science Technical Report No. 12, Bell Laboratories, Murray Hill, N. J.

POSTER SESSION CONTRIBUTED PAPERS

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

PLOTTING BINARY TREES

Kurt J. Schmucker
Department of Defense, Washington, D.C. 20755

ABSTRACT

The results of a number of statistical procedures can be summarized in terms of binary trees. This paper describes an algorithm for plotting these trees on electrostatic or incremental plotters. This algorithm has been implemented in IMP, a higher-level language designed for the CDC 6600, and used in conjunction with PEP-1, a hierarchical cluster analysis program included in the Guttman-Lingoes Nonmetric Program Series.

Key Words: Cluster analysis; Guttman-Lingoes Program Series; IMP; linked lists; multidimensional scaling; plotting algorithms; PEP-1.

1. INTRODUCTION

Often the results of exploratory data analysis techniques (i.e. Multidimensional Scaling, Cluster Analysis, etc.) can be summarized in terms of oriented tree diagrams. These diagrams allow the viewer to gain the immediate insight that a graph provides before performing a lengthy analysis of the actual numerical data produced by these statistical procedures. While a number of authors have discussed the methods of representing certain data relationships as trees, none have been concerned with the actual drawing of the tree [Carroll(1976), Hartigan(1967, 1975)]. The concrete realization of the abstract tree structure is left to the subjective influences of the individual researcher. Often the same tree structure, graphed by different individuals can lead to trees which give vastly different impressions to the viewer. Since the tree diagrams are used to provide the viewer, in a glance, with the overall structure of the data, it is most disturbing that this impression can be so drastically affected by the actual drawing of the tree as opposed to the abstract tree structure. As an example of this problem, consider the tree diagrams in figure 1. While both represent the same tree structure, the diagram in figure 1b leads the viewer to "feel" that the relationships between the objects represented by the leaves in the tree (labeled A, B, C, ...) are not as "strong" as those of the tree diagram in figure 1a. The data appears to be more "strung out". Since it is true that both diagrams represent the

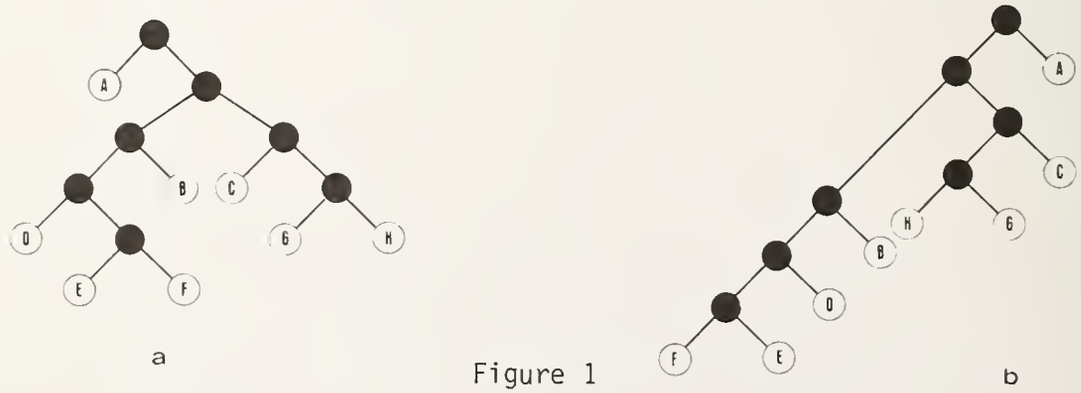


Figure 1

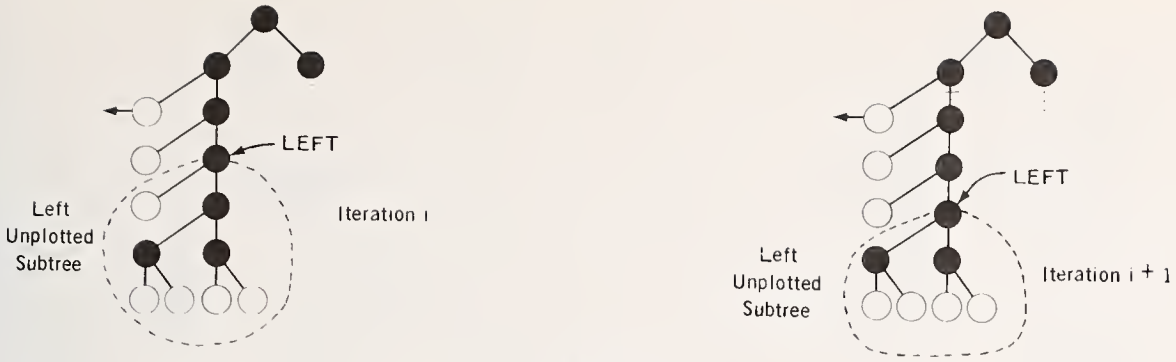


Figure 2

same tree structure, any difference in the viewers immediate interpretation is due only to the particular realization chosen. In this paper, an algorithm for graphing binary trees is given. This algorithm plots a binary tree in a deterministic, reproducible fashion, thereby making the first steps toward a standard graphical representation.

2. THE PLOTTING ALGORITHM

The particular algorithm used here to plot the binary tree assumes that the tree structure is stored in a linked list with two links, LLINK and RLINK, for each node. The algorithm simultaneously plots the left and right subtrees, but for ease in describing the workings of the algorithm, only the left side will be considered in detail. As the algorithm proceeds down the branches of the tree plotting a representation, two pointers, LEFT and RIGHT, are established. The pointer, LEFT, points to the root of the left UNPLOTTED subtree. If the unplotted subtree has a "simple enough" structure, the algorithm (1) generates a portion of the plot, (2) establishes new LEFT and RIGHT pointers, and (3) loops. Figure 2 is an example of the results of one such iteration. Otherwise the algorithm divides the left subtree into ITS left and right subtrees and continues plotting only the right subtree. When the plotting of this subtree is finished, the algorithm is recursively called with the left subtree as a parameter. A new page is plotted for this tree alone. Figure 3 is an example of this more complex case.

The actual plotting in the algorithm is simplified by allowing nodes or leaves to be drawn only at discrete points. This simplification virtually eliminates any possibility of inadvertently overlapping portions of the final plot.

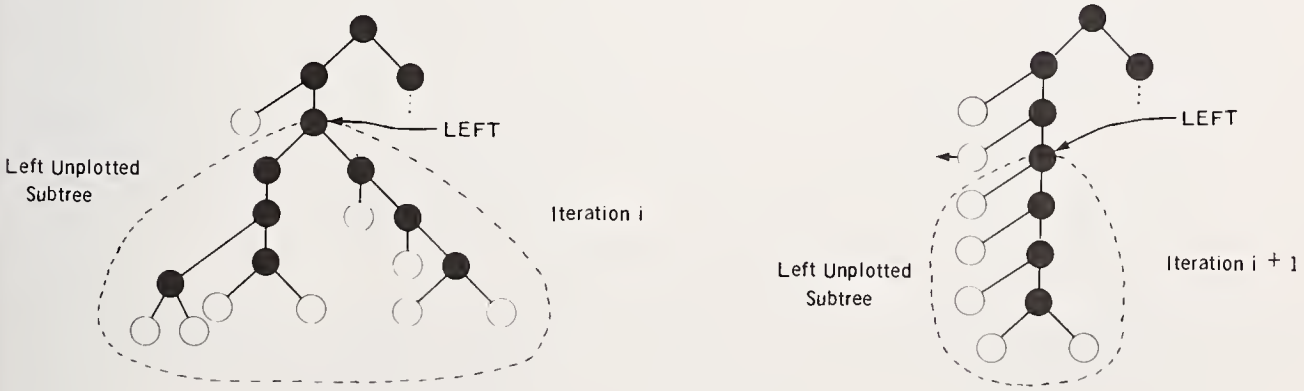


Figure 3

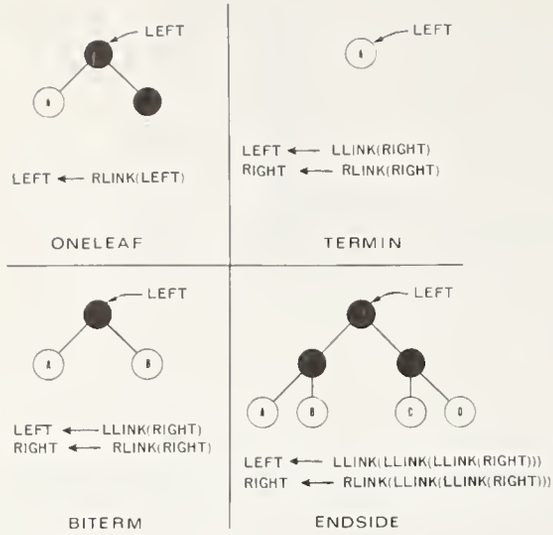


Figure 4

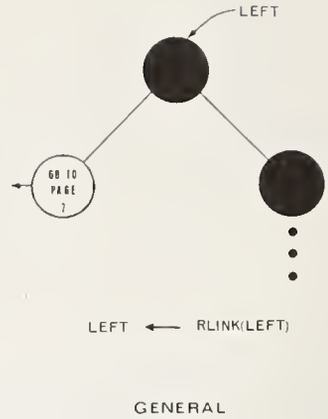


Figure 5

The algorithm recognizes four cases as simple enough to plot and these are represented and named in figure 4. These diagrams also assume, for simplicity's sake, that only the left unplotted subtree is being considered. Beneath each diagram, the relationships used to re-establish the LEFT and RIGHT pointers are given. If none of these cases are applicable, then the algorithm continues by plotting only one of the two subtrees of the LEFT pointer. The other subtree is plotted by recursing, after the plotter has advanced to a new page. This default case is diagrammed in figure 5. At any point the algorithm tests for these four cases by calculating the length of the path from the current node (either RIGHT or LEFT) to its deepest leaf.

With this background, the algorithm can now be stated consisely. Assume that (1) MAXLENGTH(A) is a function whose value is the length from the node A to its deepest leaf (Note: MAXLENGTH(x) > 0 for all x), (2) PAGELIST is an integer array (of sufficient size) initialized to zero, (3) HEAD is a pointer to the root of the tree to be plotted, and (4) PAGE ← 1. The plotting algorithm is:

ALGORITHM A

A0 [Initialize]

1) Draw root of tree
2) CURRENTPAGE ← PAGE
3) LEFT ← LLINK(HEAD)
4) RIGHT ← RLINK(HEAD)

A1 [Test Special Cases]

1) If LEFT is the null pointer, go to A3

2) If LEFT is a leaf, then
   a) Call TERMIN
   b) LEFT ← LLINK(RIGHT)
   c) RIGHT ← RLINK(RIGHT)
   d) Go to A1

3) If MAXLENGTH(LEFT) = 2, then
   a) Call BITERM
   b) LEFT ← LLINK(RIGHT)
   c) RIGHT ← RLINK(RIGHT)
   d) Go to A1

4) If MAXLENGTH(LLINK(LEFT)) = 2
   and MAXLENGTH(RLINK(LEFT)) = 2, then
   a) Call ENDSIDE
   b) TEMP ← LLINK(LLINK(RIGHT))
   c) LEFT ← LLINK(TEMP)
   d) RIGHT ← RLINK(TEMP)
   e) Go to A1

5) If MAXLENGTH(LLINK(LEFT)) = 1
   or MAXLENGTH(RLINK(LEFT)) = 1, then
   a) Call ONESIDE
   b) LEFT ← RLINK(LEFT)
   c) Go to A1

A2 [Default Case]

1) Call GENERAL
2) CURRENTPAGE ← CURRENTPAGE + 1
3) PAGELIST(CURRENTPAGE) ← LLINK(LEFT)
4) LEFT ← RLINK(LEFT)
5) Go to A1

A3 [Recursive Step]

1) Advance plotter to the next page
2) If PAGELIST(PAGE + 1) ≠ 0, then
   a) PAGE ← PAGE + 1
   b) HEAD ← PAGELIST(PAGE)
   c) Call A

A4 STOP

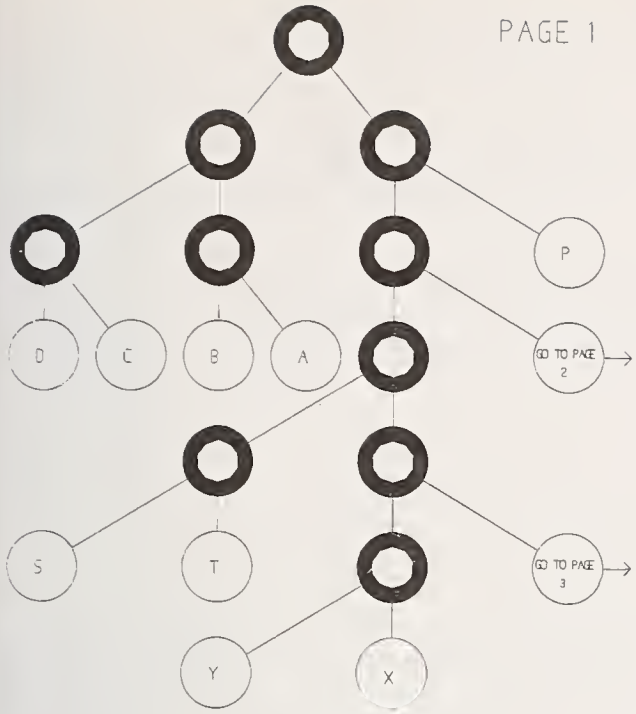


Figure 6

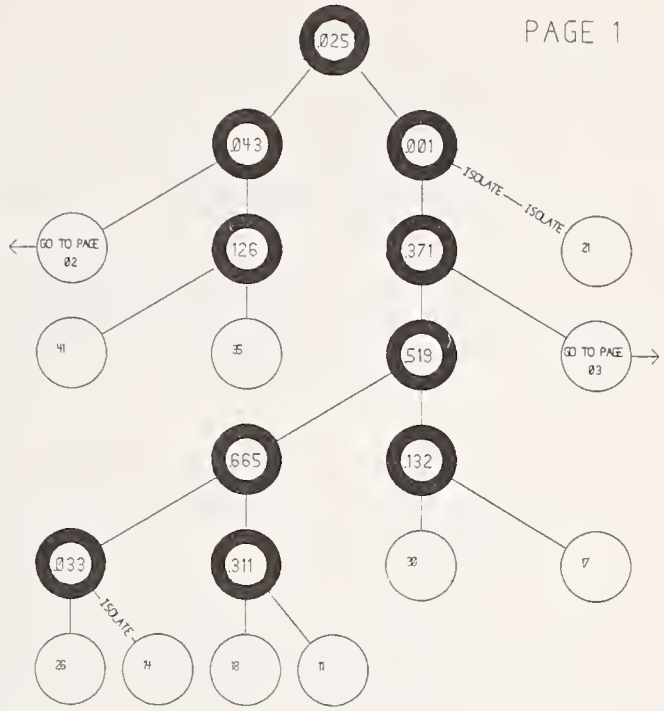


Figure 7

3. IMPLEMENTATION

In order to easily code this algorithm in a higher-level language, the language chosen must have list processing capabilities, graphics (either access to a system subroutine library or language primitives) and recursively callable procedures. IMP, a higher-level language designed for the CDC 6600, meets all these requirements and this plotting algorithm was coded in IMP [Irons(1970)]. Normal output is on an electrostatic printer/plotter although an incremental plotter can also be driven. A sample output, from data contrived to show all the algorithm's capabilities, is shown in figure 6.

4. USES

This algorithm has been used in conjunction with the CDC 6600 version of PEP-1, a hierarchical-divisive cluster analysis program contained in the Guttman-Lingoes Nonmetric Program Series [Lingoes(1973)]. The output of PEP-1 has been modified to include a plot of the cluster structure found. This plot is used in conjunction with the often unwieldly numerical output produced by the routine. Since the cluster structure found by PEP-1 can not always be represented by a binary tree, a few minor changes were necessary in the plotting algorithm. A sample of the PEP-1 output, which includes some of these anomolies, is shown in figure 7.

5. ACKNOWLEDGMENT

The author wishes to thank Dorothy A. Kalb for invaluable assistance in the preparation of this manuscript.

6. REFERENCES

CARROLL, J.D. (1976). Spatial, non-spatial, and hybrid models for scaling. Psychometrika, 41, Number 4, 439-463.

HARTIGAN, J.A. (1967). Representation of similiarity matricies by trees. J. Amer. Stat. Assoc., 62, 1140-1158.

HARTIGAN, J.A. (1975). Clustering Algorithms, John Wiley & Sons, New York.

IRONS, E.T. (1970). Experience with an extensible language, Comm. of the ACM, 13, Number 19, 31-41.

LINGOES, J.C. (1973). The Guttman-Lingoes Nonmetric Program Series, Mathsis Press, Ann Arbor.

BIOGRAPHY

KURT J. SCHMUCKER has been employed as a mathematician at the Department of Defense in Washington, D.C. since 1974. He received an M.S. in mathematics in 1974 from Michigan State University and an M.S. in computer science from Johns Hopkins University in 1977. Mr. Schmucker is also currently an Advanced Special Student in the computer science department at the University of Maryland. Mr. Schmucker is a member of Phi Beta Kappa, the MAA, and the ACM.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

THE STATISTICAL ANALYSES OF MONTE CARLO SIMULATION DATA USING THE TECHNIQUES OF DISCRETE MULTIVARIATE ANALYSIS

J. Jack McArdle
Hofstra University, Hempstead, New York 11550

ABSTRACT

A paradox is posited which suggests that most statisticians do not appropriately analyze their simulation data. This paper deals with the structural, systematic analysis of Monte Carlo frequencies and associated contingency tables using the techniques of Log-Linear Modelling. Emphasis is on practical problem areas and implications for simulation design and the Monte Carlo research process.

Key words: Contingency tables; data analysis; Log-Linear Modelling; Monte Carlo experiment; multivariate frequency.

1. INTRODUCTION

1.1 A Monte Carlo paradox. "Monte Carlo" (MC) simulation techniques are widely used to approximate mathematico-statistico solutions when exact answers are too complex. This is especially true in "robustness" studies where the behavior of statistical methods are examined under assumption failure with known population parameters. Simulated datasets are created which represent a random series of events from such configurations and methods are used to estimate the parameters from the random data. In the frequency domain probabilistic estimation is phrased in terms of "acceptance/rejection" at a specified alpha (α) level. This procedure is repeated for a specified number of trials (t) and observed frequencies (f) or "Percentage Exceedence" rates ($PE=f/t$) are noted for all population configurations. In Neyman-Pearson terms, in the presence of a true null hypothesis (H_0) the PEs represent Type I errors. The evaluation of such data is directed at determining which $PE=\alpha$. When $PE=\alpha$ the evaluation of a false H_0 is a Type II or "greatest power" examination of the largest (among several) PEs.

A statistical problem develops because the PEs are only estimates of the true long-run behavior of the statistic. When the MC researcher tries to objectively make statements and quantify such qualitative terms as "too large", "too small" and/or "most", the accuracy and precision of these estimates must be taken into account. The paradox that has developed at this analytic phase is that *most statisticians do not statistically analyze their data!* Many choose to ignore this phase completely and subjectively explore their contingency tables. Others subject these data to a wide variety of inappropriate, unstructured analyses. A minority of studies systematically attempt to account for the inferential effects, but most times fail to report this information. It is reasoned (See McArdle, 1976) that this paradox has developed out of the unavailability of theoretical methodology rather than out of any bias about this analytic estimation phase. The purpose of this paper is to propose the application of known statistical theory to the unknown of MC studies.

1.2 Previous approaches. Three statistical methods have been used: 1) *Standard Error* (SE) estimate based on binomial *PE*, 2) *Goodness-of-fit* χ^2 tests based on expected values, and 3) *Variance Stabilization Transformations* (VST) followed by *MANOVA* methodology. Each approach is problematic in some regard. The SE approach ignores overall experiment-wise error rate and design structure. This is akin to examining correlation matrices by testing each correlation (ignoring the past forty years of multivariate work) and true, sometimes latent, structure will be overlooked. The usual χ^2 analyses centers on the fit of the full distribution yet the crucial information is in the extreme percentiles. Global χ^2 tests represent misuse of χ^2. With VST the *PE*s can be used in *MANOVA* framework (Olson, 1974). However, this methodology was originally based on the fact that there were no alternative methods. Systematic investigation is available from the realization that the output behavior of MC work is in the form of *f* and the population structures are usually multifactor. The appropriate analysis of such data is termed Discrete Multivariate Analysis.

1.3 Log-Linear Models. Many (See Kleijnen, 1977) have suggested that MC experiments have all of the relevant characteristics of usual research studies. The design and analysis of such experiments should therefore be based on similar statistical and methodological concerns. Measurement in much MC work has a unique characteristic in that it is *f* or *PE*, termed binomial or discrete data. In most MC studies the population parameters (IVs) are also nominal or ordered categories so these proportions can be described as a multinomial form. The questions of interest here are the relations between independent population parameters and dependent outcome frequencies, the "meta-model". The state-of-the-art techniques for handling such datasets are best given by Bishop, Fienberg and Holland (1975) and Bock (1975, Chap. 8) and termed "Log-Linear Modelling" (LLM). The following sections show how these techniques can be logically and efficiently applied to MC data. It is believed that this is the first statement of such an application in the research literature. Emphasis here is on the practical computing of such models and much of the theory of LLM will not be discussed.

2. QUALITATIVE DESIGN IN MONTE CARLO EXPERIMENT

2.1 The choice of a dependent variable. The first issue that may be faced is the determination of the variable(s) to be studied. In much MC data the binomial parameter of "Acceptance/Rejection Frequency" is of major interest. The "*PE* of Rejection" is a direct function of *f* and t. Because most MC experiments use the same t for all experimental conditions either side of the binomial parameter *PE* perfectly describes the full binomial estimate (unequal t can be handled by fitted estimates). The *PE* is the DV.

In many MC studies more than one statistical method (DV) is observed on the same series of population parameters. This is done for the reduction of unnecessary CPU waste and, more importantly, the DVs are calculated from the same dataset (i.e. blocked) so that comparisons between IVs are less effected by random fluctuations in the data generation (Olson, 1974, p. 898). The DVs are now *repeated measures PE* because they are all calculated from the same dataset each t. While the simple χ^2 has repeated measurements analogs (e.g. McNemar's test) this was not generally true of multivariate frequency models (Bock, 1975, p. 552; Smith, 1976, p. 494) until recent advances offered by Koch, et al. (1977). However, this is not the tack taken here. First, due to the use of the one-sided *PE* parameter the marginal *f*s are not constrained (within the 0 to t range). This design consideration allows more freedom on these *f*s and they may be viewed as "different items from the same set" rather than "the same set measured at more than one time". A second argument could be the suggestion that the design factor that is considered repeated measure-

ments be broken up into special single degree of freedom questions. This can easily be done by independently examining all simulated statistical procedures against expected values. This could be tested by setting up a "dummy" dataset with all $\delta=\alpha t$ and tested in the usual "observed versus expected" framework. However, an even better approach would be to contrast an unknown procedure with a theoretically exact procedure which was simulated as part of the study. This increases the precision because it accounts for the random error due to the data generation technique. The systematic evaluation of specific contrasts does not mathematically rule out repeated measures problems and it is unknown if these must still be considered repeated measures or if the contrast questions minimize the problem. In any case the DV of interest is now a comparison factor termed TEST (or T).

2.2 Selection of models. The primary concern here rests on the choice of effects on interest and the elimination of certain unnecessary factors. The α factor should not be considered a factor of the LLM. Differences found between δ at different α levels yield no added information and, in fact, have different interpretative meaning. This also improperly increases the total degree of freedom for the LLM (α interacting with every other factor). The δs for different α levels should be treated as separate models which can later be compared for general fit.

Variance reduction techniques in the design stage would suggest that all combinations of all IVs are not required. This quite naturally leads to fractional factorials or unbalanced designs, all handled by LLM. Estimates of IV effects are not calculated because of the peculiar costly (CPU) nature of MC experiment. On the contrary, in the analysis of MC data *not all effects are of interest*. There is no logical reason to collapse over the T factor. This merely evaluates the effects of combinations of IVs and obscures T differences, usually the purpose of MC evaluation. The only effects of importance are the IV effects that interact with the T factor (or DV). This is exactly the conception of LLM offered by Bock (1975). The T is considered a "Response" factor and the IVs are considered "Sample" factors. The only estimates that may be made are the overall "Response" and all the interactions of "Response" and "Sample" factors. This appropriately limits the amount of models that are to be tested. The IVs can also be separated into specific contrasts of interest and take the form of polynomial trends when the IV categories are ordered in some fashion.

The selection of models should not be a haphazard run through every possible combination of effects but a careful evaluation of specific models that may provide useful information. The choice of a small set of theoretically important effects increase the chances of finding underlying structure as well as in computing these solutions at all.

3. QUALITATIVE ANALYSES OF MONTE CARLO EXPERIMENT

3.1 Hierarchical models. The mathematical formulation of LLM is best schematized by Brown (1976, p. 38). LLMs are termed "hierarchical" when the presence of a higher order interaction implies the presence of all effects whose factors are subsets of that interaction. This hierarchy also suggests that the evaluation of adequacy of fit of such models use the *Maximum-Likelihood* χ^2. χ^2 is identical to the minimum discriminant information statistic, is additive under subset partitioning, and has good behavior for all size δ. This is important when comparing α models.

3.2 Successive association. The study of MC behavior may be characterized under the same general rules that Brown (1976) proposes for Census type

data. The analogy to the examination of large sample population estimates between Census and MC data is not to be understated. Brown suggests the examination of two tests of association, marginal and partial. The marginal association of an effect tests whether or not the addition of a single higher order effect significantly increases adequacy of fit. The partial association of an effect tests whether or not the addition of an effect of the same order produces a significant increase in adequacy of fit. If both comparisons of successive fit are significant the effect is required. Simply, there is significant difference between T proportions, or between T proportions on IV Factor 1, and so on. In the spirit of parsimony a *forward selection* testing scheme is probably the most useful for MC experimentation. In this framework the T is first tested. Then each first order interaction between T and each IV is successively added to the model. All one-way interactions are evaluated before any two-way effects are estimated, etc. This gives a parsimonious answer to the global questions and assures computability.

3.3 Computer programs. Many LLM programs yield answers to MC problems. But by far and away the best and most flexible routine for MC studies is MULTIQUAL (Bock and Yates, 1975). MULTIQUAL theory is exactly the MC conception offered here and it yields tests of virtually any hypothesis of interest (i.e. polynomial fits, etc.). Also the T factor can be extended to simultaneous global multivariate tests. While the C-TAB algorithm (Haberman, 1973) is easier to use (especially in BMDP3F from) it will collapse over DV and print effects for IV interactions. Only the expert modeller is able to use C-TAB appropriately and still cannot test all contrasts of interest without great diffuculty. Many other programs use different minimization criteria for convergence and fit. It is unknown if these will have any effect on MC problems specifically. This is doubted because MC tables are *not different from any other contingency table!* A practical problem is encountered in fitting estimates past about a 5-way table. Algorithms usually *cannot converge* or are extremely costly. This limits MC IVs to 4 factors (T being the other). MC architects would be wise to note such computational limitations.

3.4 Unanswered questions. Great advances in knowledge on LLM theory have taken place in the last few years. There are still many questions of importance to MC researchers such as: 1) interdependence of probability estimates, 2) post-hoc procedures, 3) strength of association, 4) minimization criteria, and 5) computer algorithms. In fact virtually any item of statistical importance to contingency analysts will also be important to MC researchers who produce contingency tables. For example, a measure of strength of association (e.g. phi) can be used to compare the LLMs of different α for a specific DV-IV interaction. Discrete theory is not advanced in this area but the future looks bright and MC analyses will benefit.

4. IMPLICATIONS FOR MONTE CARLO RESEARCH

4.1 Objectivity. The tabular display of all MC data has recently been the only fashion in which results could be presented. The *amorphous* types of formal analyses offered are usually of haphazard, piecemeal variety which tend to negate, rather than enhance, good design. In such multiway tables it is rather *difficult to visually determine* what is actually going on. Simple visual alterations are not always possible in studies with many IVs. Problems of overestimation and underestimation may be in large part due to the nature of such visual display. The systematic and structured analysis of MC data can only lead to a more objective framework, a vitally important point for MC research. The results and recommendations given by statisticians are too often taken on *faith* by the applied research community.

4.2 Design. This objective approach also leads to possibilities that are not readily available from the usual tabular presentations (i.e. polynomial trends, etc.). Somewhat exact probability statements may be made about *structural hypotheses* and carried out through the use of these special contrasts and effects. The general MC design considerations should now include the usual research issues of the number of parameters and effect sizes. This should lead to very carefully planned considerations about the parameters of investigation and (theoretically) should improve MC research.

Another interesting idea is the application of expected effect size (e) and desired power levels (Cohen, 1969) to the determination of the *run-length* (t). Most MC researchers use variants of several thousand t for accuracy to specified significant digits. With hypothesized e, t could be significantly reduced. LLM may bring MC research into the mainstream of knowledge in research design methodology and at the same time cut down on costs associated with large t.

4.3 Analysis. Initially, the MC researcher has the opportunity to *reanalyze* almost any published data when tabular information is presented. The information required for LLM is probably a useful publishing requirement in itself. This gives the new MC experiment a chance to more fully investigate real problem areas that may have been overlooked, not dealt with, or a chance to weigh the practical necessities of utilizing one technique over another (e).

An important feature of the LLM given here is that the techniques may be used to *crossvalidate* new information with previously published MC or mathematical results. A component of good design is the inclusion of previously studied population parameters. These effects can be compared for fit in the spirit of the T contrast.

Perhaps the primary benefit obtained from the LLM approach is the dramatic systematic solution of complex MC issues. LLM allows structured design to evolve into systematic, structured analysis that might not be possible by any other perspective. In fact, a large complex dataset stimulated this paper and provides several application examples (McArdle, 1977).

4.4 Conclusion. The analysis of many MC studies require some form of LLM. However, there are many others which can utilize the more advanced theory of optimum operators (Kleijnen, 1977) or quantitative analyses (Bock, 1975). A significant problem may arise in the misuse of LLM in such studies. Of course, there is good reason to believe that MC scientists can easily learn both statistics and computer programming (Hope springs eternal).

The structured analysis of multivariate qualitative data by the systematic, objective methods of LLM is a transition that MC researchers must make. The paradoxical tendency to take the subjective summary statements of MC analysts on *faith alone* is heretical to the ideals of scientific research. The conceptual framework offered here is only an initial guide for the application of an emerging field in data analysis to old problem areas. The message is clear; *Analyze your simulation data!!* An answer to "How?" is provided by the methods of Log-Linear Modelling.

5. ACKNOWLEDGEMENTS

I wish to thank Michael S. Goldberg (Hofstra U.) for both the time and computer capability and Philip H. Ramsey (Hofstra U.) for important contributions to all statistical issues discussed. I also wish to thank R. Darrell Bock, Paul W. Holland, Richard S. Lehman, and I. E. Chas. Woodson for their helpful comments.

7. REFERENCES

BISHOP, Y.M.M., FIENBERG, S.E., and HOLLAND, P.W. (1975). Discrete Multivariate Analysis: Theory and Practice. Cambridge: MIT Press.

BOCK, R.D. (1975). Multivariate Statistical Methods in Behavioral Research. New York: McGraw-Hill.

BOCK, R.D. and YATES, G. (1975). MULTIQUAL: Log-Linear Analysis of Nominal or Ordinal Qualitative Data by the Method of Maximum Likelihood. A Fortran IV Program. International Educational Services, Chicago, Ill.

BROWN, M.B. (1976). Screening effects in multidimensional contingency tables. Applied Statistics, 25 (1), 37-46.

COHEN, J. (1969). Statistical Power Analysis for the Behavioral Sciences. New York: Academic Press.

HABERMAN, S.J. (1973). C-TAB: Analysis of Multidimensional Contingency Tables by Log-Linear Models. A Fortran IV Program. International Educational Services, Chicago, Ill.

KLEIJNEN, J.P.C. (1977). Design and analysis of simulations: Practical statistical techniques. Simulation, 28 (3), 81-90.

KOCH, G.G., LANDIS, J.R., FREEMAN, J.L., FREEMAN, D.H., Jr., and LEHNEN, R.G. (1977). A general methodology for the analysis of experiments of categorical data. Biometrics, 33 (1), 133-158.

McARDLE, J.J. (1976). Empirical tests of multivariate generators. Proceedings of the Ninth Annual Symposium on the Interface Between Computer Science and Statistics, Department of Statistics, Harvard University. Boston: Prindle, Weber, and Schmidt, 263-267.

McARDLE, J.J. (1977). An Applied Monte Carlo Study on the Type I Behavior of Univariate and Multivariate Strategies for Repeated Measures Hypotheses. Unpublished Doctoral Dissertation, Dept. of Psychology, Hofstra University, Hempstead, New York 11550.

OLSON, C.L. (1974). Comparative robustness of six tests in multivariate analysis of variance. JASA, 69 (348), 894-908.

SMITH, J.E.K. (1976). Analysis of qualitative data. In Rosenzweig, M.R. and Porter, L.W. [Eds.]. Annual Review of Psychology, 27, 487-499.

BIOGRAPHY

J. Jack McArdle received the Ph.D. in Applied Research Psychology at Hofstra University in April 1977. The content of this paper represents part of his dissertation research. For the past four years McArdle has worked as the statistical data analyst at the Academic Computing Service at Hofstra. Starting this year he holds the position of senior biostatistician at Hillside Psychiatric Research, Long Island Jewish Hospital. Previous publications include work on multivariate simulation methodology and mathematical modelling of human behavior. Starting September McArdle will be at the Department of Psychology, University of Denver, Denver, Colorado 80208.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

DESIGN AND ANALYSIS TECHNIQUES FOR LARGE DATA FILES: THE CODAP SYSTEM

Eduardo N. Siguel, Ph.D. and Sidford F. Sand
National Institute on Drug Abuse, Rockville, MD 20852

ABSTRACT

This paper describes issues related to the design and analysis of large data files, and indicates how one set of large data files, the Client Oriented Data Acquisition Process (CODAP), is currently maintained and analyzed.

Key words: Client Oriented Data Acquisition Process; CODAP; data collection; large data files; statistical analysis; statistical issues; systems design.

1. SYSTEM DESIGN CONSIDERATIONS

The major steps in the design of a data system are: (1) determine objectives, (2) decide what kinds of issues one wishes to deal with, (3) describe the questions one wants to answer, and (4) design a data system so that it will provide the answers (research or system design). Ideally, data systems should be designed with specific objectives, and the data to be collected should be able to meet those objectives. Unfortunately, these objectives are rarely met when large data systems are designed. In practice, one may find that the design of a large data system is characterized by: (1) general, non-specific objectives such as "support of management decisions", (2) general issues, such as "We want to improve planning, management, evaluation, etc...", (3) failure to define in advance of the system design effort the questions which are to be asked, and (4) system design being executed on the basis of what seem to be "interesting" questions, subject to constraints imposed by money, time, administrative "clearance" requirements, and the willingness of respondents to provide the information.

If one may assume that objectives were clearly stated, that issues and questions were defined in operational terms, and that the data elements to be collected are necessary and sufficient to answer the questions posed, then it is useful to consider the area of system design which directly effects the analyst's ultimate products: data collection. (For purposes of this discussion, availability of internal data control and processing resources which are adequate to handle collected raw data is also assumed.) There are two major aspects to consider when designing the data collection instruments and processes:

A. Substantive Attributes: (1) The complexity of the questions asked and the ease of formulation and expression of the answers. (2) The likely availability of respondents' knowledge and informational materials (records, logs, interviewees, etc.) which facilitate determination of correct answers. (3) The degree of interrelatedness of questions and answers, and the "intensity" of the requirement that answers be internally consistent. (4) A host of environmental and attitudinal aspects which inevitably influence all of the above. The amount of self-discipline which the data acquisition process imposes may be

---

The views presented in this paper are those of the authors and do not necessarily reflect the opinion and views of the D.H.E.W.

realistic or absurd depending on attitudinal and role factors. The most important determinant of the success of the respondents' activities is usually the answer to his/her question: "What's in it for me?"

B. Attributes of Form Design and Instructions: A wide variety of "structural" techniques for increasing the viability of a form are available. Usually attributes such as arrangement of items on the page and the coding structures employed are belabored at length. Then a professional forms preparer adds a few additional niceties such as compatibility with typewriter spacing, different printing fonts, and color or shading for emphasis.

Most frequently overlooked or badly rendered are aspects having to do with "data control", such as use of carbon copies, preprinted serial numbers, aids to batching, logging, transmitting and filing of forms, turnaround and feedback documents or printouts, and machine-sensible forms. Even if the data to be acquired is easily encoded and training of respondents is sound, data control is crucial. In a large system, one frequently deals with a geographically distributed population of respondents who vary greatly in education and motivation and whose internal record-keeping arrangements vary from immaculate to nonexistent.

More difficult are problems of "followup" in systems which "track" an activity of some kind in which a second, third, or nth transmission of data is related to previous data transmissions and provides additional information or corrects or updates previous data. Here problems of missing or duplicate items in a series of transmissions, failures to properly associate a transmission with its related predecessors, incorrect "transaction types" and resulting imbalances between types of transactions can result in buildups of records which cannot be disposed of properly within the rules that govern the system.

For every such problem there are potential solutions. These may include manual and computerized logging, validity and consistency checks and a variety of feedback mechanisms, "turnaround" documents and a host of other techniques. The problems which defy solution usually stem from human factors of motivation, staff turnover and conflicting priorities or are problems whose genesis is a flawed, unreasonable or obsolete aspect of the system design itself. In the former situation the respondent and his motives, methods, and priorities are at least partially beyond the reach of the system maintainers, and even where the respondent's errors, inconsistencies, and omissions can be identified, usually only the respondent himself can provide the correct answers. Since the respondent's performance fell short the first time, the chances that he will ignore or compound the errors are quite high. There is thus a considerable difference between being able to detect errors and being able to get them corrected. The latter situation frequently stems from the indicipline, alluded to above, of the systems designers themselves. Such problems may ultimately destroy the system itself by the simple process of yielding a data base of questionable usefulness. The cost in human terms of a system based on flawed concepts is immeasurable, and serves to reemphasize the importance of formulation of the system's basic concepts and objectives.

C. Compromises Between Substantive and Technical Issues: In the final analysis, for each system a balance is struck between substantive and technical issues. Each has a limiting effect on the other. The most perfect, elegant expression of the designer's data "needs" will probably require a respondent population of psychic Ph.D.'s and a 20-page input form, while the data processing technician can easily design an almost infallible form and instructions, but one whose infantile oversimplifications and omissions will yield data which is clean, complete, and of almost no use to a statistician or program manager.

During the system design and testing process a large number of compromises are reached to ensure that, firstly, the data gathered will actually answer most of the important questions it is designed to answer in a meaningful, relatively undistorted manner. Secondly, the information must be obtainable and expressible for the respondent, and the form to be completed must make rendering of such answers as easy as possible.

When the mechanics of the information-gathering process are finally defined, a variety of training requirements and strategies will have been identified and instructional materials prepared, usually including manuals which tell a respondent how to fill out the forms involved. The strategies will reflect the designers' emphasis of various factors: minimization of errors in specific items, minimizing the time required to fill out the form, restrictions on coding space, simplification of questions and instructions, increased probability of legibility or successful transmission of completed forms, etc.

2. STATISTICAL ISSUES IN THE ANALYSIS OF LARGE DATA FILES

There are a wide variety of issues to be considered when one attempts to analyze the data in a given file. We will name a few of the most important ones that have particular impact upon large data files.

The first step in data analysis is to define the problem and the model or framework used to consider it. The objectives, issues, problem or question under consideration must be stated in operational terms, and phrased in the form of questions or hypotheses to be tested. In addition, there must be a model which serves as a framework within which to answer questions and a context within which to test hypotheses. Data, by itself, has no meaning, and must be interpreted within the context of a model. Therefore, design, issues and questions make sense only within the framework of a model of the situation under consideration. The statistician's role is to define the model which best describes the issues. Within the model, the statistician must phrase the questions in such a manner that a researchable, objective answer is possible.

Once the problem and the model are operationally defined, a methodology is developed which takes into account the nature of the data. Factors which the statistician may consider include: (1) How the data were collected. (2) The nature of errors. Usually emphasis is placed on sampling errors, but non-sampling errors may actually be much larger than sampling errors. Non-sampling errors include such errors as respondent errors, poor instrument reliability, measurement errors of other kinds, transmission errors, data processing errors, etc. (3) Methods useful in the analysis of the data. There are a variety of multivariate methods available. When large amounts of data are involved, efficient use of computer time becomes a necessity. Computer efficiency begins with the use of efficient software and proper file design. Unnecessarily large record sizes or inadequately grouped records may greatly increase computer processing time. When one uses standard software packages, such as SPSS or BMDP, and not all cases are to be considered (for example, when one instructs the program to consider only females 18-20 years old), it is important to phrase a complex sequence of conditional statements in such a manner that conditions are tested according to the likelihood that they will fail, conditions with a higher probability of failing being tested first. This procedure reduces processing time because fewer records need to be processed completely. (4) Interpretation of the results. It is important to distinguish between statistically significant differences and differences that are not large enough to be meaningful in terms of policy and program decisions, management issues, etc. One often finds that relationships between two variables, X and Y (or the difference between X and Y) are analyzed testing for no relationship (or no difference between two distributions) using a chi square statistic (or similar statistic). With a large data file, a crosstabulation of almost any two variables is likely to have a very high chi square value. Two empirical distributions are likely to be found different even though the differences between them may be very small. Two alternative approaches can be used: (a) report the data with an appropriate confidence interval, or (b) determine, "a priori", a particular relationship that is meaningful (or a particular difference that is meaningful) and then test the hypothesis that the difference is greater than the pre-established value (rather than the null hypothesis), or that the relationship is stronger than the pre-established value (using non-central chi square).

Another aspect requiring consideration involves the complications arising from the use of many variables. A relationship between two variables may change direction when a third variable is used as a control variable. When the data file consists of many observations (cases) and many variables, it is possible to obtain apparently contradictory findings according to which variables are included in the analysis. Inclusion or exclusion of subpopulations may change relationships. The availability of many cases and many variables encourages alternative approaches to data analysis and potential apparent inconsistencies in the interpretation of findings.

3. CODAP -- AN EXAMPLE OF A LARGE DATA FILE

A. Description of CODAP: The Client Oriented Data Acquisition Process (CODAP) is a data collection system developed and operated by the National Institute on Drug Abuse (NIDA) in treatment facilities (clinics) that receive federal funds. Its purpose is to provide current information which describes clients and the treatment provided to them in order to aid in planning, management and evaluation activities. Reports from between 1,500 and 1,800 clinics are received each month. Fifty states participate in data collection. About 40,000 admission and discharge reports describing clients admitted to and discharged from treatment are processed each month.

B. How CODAP Data are Analyzed: A large data file coupled with many demands for analysis requires automated procedures for table generation and a variety of approaches to satisfy user demands. The Division of Scientific and Program Information, NIDA, has developed several approaches: (1) Periodic, usually quarterly, reports are prepared which present close to 100 tables. (2) Special issues are addressed in the Statistical Series, which describes applications to management of drug abuse programs, evaluation of treatment outcomes, and studies of patterns and factors associated with the development of drug abuse (epidemiology of drug abuse). (3) Data files are available less than five months after the data are collected. These files are provided to the Single State Agencies which coordinate drug abuse programs, and to an outside organization which in turn makes the files available to requestors or prepares tables upon request (at cost). (4) Technical assistance is provided to the states on how to use CODAP data. (5) Reports unique to each clinic/program are sent to those clinics/programs, together with comparable state/national data and suggestions for interpreting the data. (6) Special analyses are prepared upon request from federal government agencies.

In order to handle the large amounts of data involved, special analytic software has been developed which allows the following tasks to be performed automatically: (1) SPSS output is sent, via magnetic tape, to disk files for manipulation by text-editing software which produces camera-ready copy of tables. (2) Tables with a large number of variables (of the form A vs. B vs. C vs. D vs...) are stored on magnetic tape. Another program reads those tables and produces summaries (collapsed over the categories of a given variable). In addition, for continuous, time-related variables, the output of both programs can be plotted using a CALCOMP plotter. (3) Depending on the nature of the analysis, users can utilize extract files consisting of 20% and 1% samples of the data file, and also special subpopulations (such as daily heroin users) which have been found to be of specific interest. (4) A file of all tables computed from several of the larger files (such as the 100% sample) is kept as a reference. Requests are often answered from that reference system at a considerable savings in time and money.

ACKNOWLEDGEMENTS

We are grateful for the assistance and support of Dr. William H. Spillane and Mr. Neil Sampson, and the valuable comments provided by the staff of the Division of Scientific and Program Information, NIDA.

REFERENCES

SIGUEL, E. N. and SPILLANE, W. H. (1976). The epidemiology of drug abuse; a new data base, new techniques and new findings. Proceedings of the Social Statistics Section of the American Statistical Association.

SIGUEL, E. N. and SPILLANE, W. H. (1977). The client oriented data acquisition process (CODAP-77). Amer. J. Drug & Alcohol Abuse, IV(2).

NIDA Statistical Series (1975 through 1977). A series of reports which primarily concern admission and discharge activity, client characteristics, types of drugs abused, and patterns of drug abuse; these variables are examined in relation to each other and to calendar quarter of admission, size of SMSA, and geographical region.

BIOGRAPHIES

E. Siguel received his Ph.D. in Mathematical Psychology from the University of Michigan. He has masters degrees in Physics and in Computers & Statistics and more than 16 years in statistical analysis with emphasis on large data files. His major interest is mathematical models applied to biological and social systems and he has written extensively on health care delivery and the epidemiology of drug abuse.

S. Sand received his B.A. in Political Science and Government from the American University. He has 9 years of systems analysis experience in mental health and drug abuse related data processing. His major interests are data base maintenance and documentation standards and techniques.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

VEHICLE ROUTING WITH PROBABILISTIC DEMANDS

Bruce L. Golden and William Stewart, Jr.
College of Business and Management, University of Maryland at College Park

ABSTRACT

The vehicle routing problem has been receiving a great deal of attention recently in the operations research and computer science literature. The basic problem is to design a set of vehicle routes of minimal total distance leaving from and eventually returning to a central depot, which satisfies capacity constraints and customer demands that are known in advance. It is generally assumed that a new set of routes will be generated if the demands at the delivery points are varied. In this paper, we treat the more complex problem of determining a fixed set of routes in the case where demands are probabilistic in nature, rather than deterministic. Potential applications include schoolbus routing, municipal waste collection, and daily delivery of dairy goods. We assume that the demands at each node i can be modeled by a Poisson distribution with mean λ_i. We describe two types of error situations which we seek to avoid and point out the close relationship they bear to Type I and Type II errors. The objective is to minimize expected distance traveled subject to the restriction that the probability of a primary error is sufficiently small. Computational results are discussed in detail.

Key words: Vehicle Routing, Probabilistic Demands.

BACKGROUND

The vehicle routing problem, sometimes referred to as the truck-dispatching problem, is frequently encountered by management in both the public and private sectors. In recent years, this problem has attracted widespread attention for a number of reasons. First of all, increased oil prices and truck drivers' salaries have brought into focus the complexity and importance of this distribution problem. Secondly, sophisticated implementation techniques and data structures (see Fox [4]) enable us to approach large-scale problems of this kind which we simply could not, previously. Finally, the determination of good heuristic approaches to computationally refractory real-world problems has become a more respectable avenue of research.

The vehicle routing problem in its simplest form, is to find a set of delivery routes from a central depot to a large number of demand points each of which has known requirements, in such a way that the total distance covered by the fleet is minimized. We will assume that all vehicles have the same capacity and that these vehicles depart from and return to the central depot. Extensions and generalizations to this model are mentioned in Golden, Magnanti, and Nguyen [5].

The Clarke-Wright "savings" approach is the heuristic algorithm which is most widely used in solving vehicle routing problems. Suppose that, to begin with, each demand node is served individually from the central depot. Then, there are as many routes as there are demand points, clearly not a very cost-effective strategy. Now, if we link two nodes i and j (node 0 is the central depot) we incur a savings of $s_{ij} = d_{0i} + d_{0j} - d_{ij}$ (d_{ij} is the

distance from i to j). The algorithm requires that we first compute the matrix of potential savings $S = [s_{ij}]$ for $i, j = 1,2,\ldots,n$, where n is the number of demand nodes. Next, at each iteration, from among the feasible links we choose to link the nodes i and j which yield the greatest positive savings. See Clark and Wright [1] for clarification.

Golden, Magnanti, and Nguyen [5] present a new implementation of the Clarke-Wright algorithm which performs from one to two orders of magnitude faster than the traditional implementation. In their paper, the authors emphasize ideas from computer science such as heap structures and list processing. They consider savings only between nodes that are "close" to each other, eliminating the burden of computing the entire matrix S. Next, these savings are stored in a heap structure to reduce the number of comparison operations required. We will utilize this efficient computer code in our work here.

So far, demands have been deterministic. In this paper, we treat the more complex problem of determining a fixed set of routes in the case where demands are probabilistic in nature. Potential applications are numerous; for instance, consider a firm which makes daily deliveries of fuel oil to automotive service stations. Each route is fixed in advance, but the demand on any particular day is stochastic. Other examples are schoolbus routing, municipal waste collection, and daily delivery of dairy goods.

Tillman [8], in 1969, introduces a heuristic approach to a delivery problem with probabilistic demands and illustrates it with an example involving seven demand nodes. He assumes that demands at each node are generated from a Poisson distribution with a mean of two. Tillman's objective is to minimize the expected cost of operating the routes, which includes the cost of hauling an amount of commodity which is not needed and the cost of not hauling enough to satisfy the demands on a route. Analogous costs are associated with the collection problem.

Stewart [7] treats the stochastic vehicle routing problem from a different viewpoint. As motivation, he argues that even if a company had the time to determine different routes each morning depending on that day's demands, in many cases they would prefer to have their delivery routes fixed over time in order that the same driver make the same stops every day. This strategy promotes regularity of service. To avoid confusion, however, we will assume that the state of information is such that the driver does not learn a customer's demand on a particular day until he arrives for delivery. Stewart seeks to minimize total distance traveled; demands are Poisson distributed with mean λ. This work, although of a preliminary nature, provides valuable insight for the algorithm we develop in this paper.

As far as we can tell, there has been no additional research devoted to this very practical problem. We will be more ambitious than previous authors. First, we give a precise (yet non-mathematical) formulation. We model the demand at node i as a random variable from a Poisson distribution with mean λ_i. Next, we suggest an algorithm for solving the vehicle routing problem with probabilistic demands. Finally, we apply our method to a problem with 75 customers.

DISCUSSION

We consider a delivery problem where there is a central depot and n demand points. The demand at node i, denoted by d_i, is described by the independent Poisson distribution with mean and variance λ_i. As noted in Feller [3], the Poisson is a discrete distribution which arises in a great variety of problems. We have reason to believe that this modeling assumption is well justified. We must satisfy demands and we would like to do so in a minimum total amount of time or distance. There are two types of error situations which we seek to avoid.

A primary error occurs when a vehicle cannot satisfy the demands of the customers on the route to which it has been assigned. This means that an additional trip to the central depot must be made (incurring longer travel time and possibly overtime charges) while the customer experiences a service delay.

A secondary error occurs when a vehicle returns to the central depot after satisfying the demands on its route with more than $100(1-\pi)$ percent of its original load. Carrying an amount of the commodity when it is not needed is clearly a waste of loading and unloading time. In addition, the goods might be perishable. In any case, we might have been able to distribute some of the surplus elsewhere. Here, we suffer a holding cost. This mistake is not as serious as a primary error and our analysis will reflect this observation.

We present below a strategy for handling this probabilistic situation. Assume that all vehicles have the same functional capacity, c. Suppose we have a route which contains nodes $n_1, n_2, \ldots, n_k$ and has total demand $x = d_{n1} + d_{n2} + \ldots + d_{nk}$. Then $E(x) = Var(x) = \lambda_{n1} + \lambda_{n2} + \ldots + \lambda_{nk}$ on that route. By appealing to the Central Limit Theorem, we approximate this with a Normal distribution using $\mu = \lambda_{n_1} + \lambda_{n_2} + \ldots + \lambda_{nk}$ and $\sigma = \sqrt{\mu}$. This deserves some justification. Let random variable r be defined as the sum of n independent identically distributed random variables, each of which has known mean and variance. The Central Limit Theorem states that as $n \to \infty$, the CDF Prob $(r \leq r_0)$ approaches the CDF of a Normal random variable, regardless of the form of the PDF for the individual random variables in the sum. In this case, the distribution of r is Poisson with mean μ. But this can be represented as the sum of μ Poisson random variables each with mean 1. Thus, the Normal CDF gives an excellent approximation of the Poisson CDF for large μ.

Of course, we could have assumed originally that customer demands were Normally distributed, but then we would have to specify two parameters for each customer. Furthermore, it might make more sense to think of demands as integers, e.g., number of quarts of milk.

Using the Normal approximation we obtain:

$$\text{Prob } (x \geq c) = \text{Prob } \{\text{primary error on a route}\}$$

$$= \text{Prob } \left\{ z \geq \frac{c - \mu}{\sqrt{\mu}} \right\} \text{ and}$$

$$\text{Prob } (x \leq \pi c) = \text{Prob } \{\text{secondary error on a route}\}$$

$$= \text{Prob } \left\{ z \leq \frac{\pi c - \mu}{\sqrt{\mu}} \right\}.$$

Assume that μ is nearly the same for most of the r routes. We will view $\bar{\mu}$ (which will be defined shortly) as the artificial capacity of the vehicles and we will apply a Clarke-Wright algorithm treating λ_i $(i = 1, 2, \ldots, n)$ as demands and $\bar{\mu}$ as vehicle capacity to obtain a fixed set of routes. The problem we tackle then is of the following form:

Minimize (1) expected total travel distance

subject to (2) a fixed set of routes;

(3) customer demands are satisfied;

(4) vehicle capacity is obeyed;

(5) Prob {primary error on a route} $\leq \alpha$.

We will refer to the above problem (1) - (5) as the SVRP (stochastic vehicle routing problem). We must determine the routes themselves and their loads. Our approach is heuristic in nature. For each route, we want the probability of a primary error not to exceed α. Management should decide carefully on an appropriate value for α since there is a delicate tradeoff between customer satisfaction on one hand and extra trip distance and the cost of additional trucks on the other. We assume that almost all of the routes will load up to capacity and seek the optimal artificial capacity $\bar{\mu}$. We have

$$\text{Prob } \left\{ z \geq \frac{c - \bar{\mu}}{\sqrt{\bar{\mu}}} \right\} = \alpha$$

$$\Longrightarrow (c - \bar{\mu}) = z_{1-\alpha}\sqrt{\bar{\mu}}$$

$$\Longrightarrow (c - \bar{\mu})^2 = \bar{\mu} z_{1-\alpha}^2 \text{ which, after some algebra, yields}$$

$$\bar{\mu} = \frac{2c + z_{1-\alpha}^2 - \sqrt{z_{1-\alpha}^4 + 4cz_{1-\alpha}^2}}{2}. \qquad (6)$$

Notice that constraint (5) bounds the maximum variance in route demand. We also remark that if we let $\beta = \text{Prob}\left\{z \leq \frac{\pi c - \bar{\mu}}{\sqrt{\bar{\mu}}}\right\}$, our primary and secondary errors bear a close resemblance to Type I and Type II errors from hypothesis testing. Given $\bar{\mu}$ we can plot β vs. π.

From the analysis above, we have a safety stock (or extra inventory) of $c - \bar{\mu}$ units as a cushion against the occurrence of primary errors. In the case where a route has mean demand $\mu < \bar{\mu}$, let $\mu + (c - \bar{\mu})$ be the load on that route; constraint (5) will be satisfied easily.

In Table I, we illustrate the relationship between c and $\bar{\mu}$ for $\alpha = .10$. For instance, if $c = 100$ and $\alpha = .10$, then $z_{1-\alpha} = 1.28$ and $\bar{\mu} = 87.99$. We could equally well (because of integral demands) use an integral artificial capacity of 87 to set up fixed routes with "demands" of λ_i at node i. The safety stock would be 13.

c	$\bar{\mu}$
10	6.68
20	15.03
30	23.76
100	87.99
1000	960.33

Table I. Relationship between c and $\bar{\mu}$.

DESCRIPTION OF THE ALGORITHM

Suppose we are confronted with a stochastic vehicle routing problem where we know c, α, and λ_i $(i = 1,\ldots,n)$. We outline below a heuristic procedure for calculating a good solution to the problem SVRP.

Algorithm:

Step 0: Given c, α, and λ_i $(i = 1,\ldots,n)$, specify δ as a lower limit on the left-hand-side of inequality (5).

Step 1: Using equation (6) solve for $\bar{\mu}$ the artificial truck capacity.

Step 2: Let λ_i be the demand at node i. Construct fixed routes using the Clarke-Wright code mentioned earlier.

Step 3: Decrement α and repeat steps 1 and 2 if $\alpha > \delta$; otherwise go to step 4.

Step 4: Select the "best" set of fixed routes.

We will apply this solution procedure in the next section to a problem involving 75 customers. In addition, we will analyze its performance.

COMPUTATIONAL RESULTS

We have performed extensive computational experiments using a 75 customer problem as a test case for our approach. The data, taken from Eilon et al. [2], is shown in Table II. For each demand node the coordinates are given along with the mean demand at that node. Demands are Poisson distributed.

Since there are so many variables involved, we have chosen to analyze one test case thoroughly, rather than simulate a myriad of sample problems. We will try to make broad observations and recommendations based on our experience. However, we remark that this work is of an introductory nature; there are many additional questions relating to sensitivity analysis that should be investigated.

A secondary error occurs when a vehicle returns to the central depot after satisfying the demands on its route with more than $100(1-\pi)$ percent of its original load. Carrying an amount of the commodity when it is not needed is clearly a waste of loading and unloading time. In addition, the goods might be perishable. In any case, we might have been able to distribute some of the surplus elsewhere. Here, we suffer a holding cost. This mistake is not as serious as a primary error and our analysis will reflect this observation.

We present below a strategy for handling this probabilistic situation. Assume that all vehicles have the same functional capacity, c. Suppose we have a route which contains nodes $n_1, n_2, \ldots, n_k$ and has total demand $x = d_{n_1} + d_{n_2} + \ldots + d_{n_k}$. Then $E(x) = Var(x) = \lambda_{n_1} + \lambda_{n_2} + \ldots + \lambda_{n_k}$ on that route. By appealing to the Central Limit Theorem, we approximate this with a Normal distribution using $\mu = \lambda_{n_1} + \lambda_{n_2} + \ldots + \lambda_{n_k}$ and $\sigma = \sqrt{\mu}$. This deserves some justification. Let random variable r be defined as the sum of n independent identically distributed random variables, each of which has known mean and variance. The Central Limit Theorem states that as $n \to \infty$, the CDF Prob $(r \leq r_0)$ approaches the CDF of a Normal random variable, regardless of the form of the PDF for the individual random variables in the sum. In this case, the distribution of r is Poisson with mean μ. But this can be represented as the sum of μ Poisson random variables each with mean 1. Thus, the Normal CDF gives an excellent approximation of the Poisson CDF for large μ.

Of course, we could have assumed originally that customer demands were Normally distributed, but then we would have to specify two parameters for each customer. Furthermore, it might make more sense to think of demands as integers, e.g., number of quarts of milk.

Using the Normal approximation we obtain:

$$\text{Prob}\ (x \geq c) = \text{Prob}\ \{\text{primary error on a route}\}$$

$$= \text{Prob}\left\{ z \geq \frac{c - \mu}{\sqrt{\mu}} \right\} \text{ and}$$

$$\text{Prob}\ (x \leq \pi c) = \text{Prob}\ \{\text{secondary error on a route}\}$$

$$= \text{Prob}\left\{ z \leq \frac{\pi c - \mu}{\sqrt{\mu}} \right\}.$$

Assume that μ is nearly the same for most of the r routes. We will view $\bar{\mu}$ (which will be defined shortly) as the artificial capacity of the vehicles and we will apply a Clarke-Wright algorithm treating λ_i $(i = 1, 2, \ldots, n)$ as demands and $\bar{\mu}$ as vehicle capacity to obtain a fixed set of routes. The problem we tackle then is of the following form:

Minimize (1) expected total travel distance

subject to (2) a fixed set of routes;

(3) customer demands are satisfied;

(4) vehicle capacity is obeyed;

(5) Prob {primary error on a route} $\leq \alpha$.

We will refer to the above problem (1) - (5) as the SVRP (stochastic vehicle routing problem). We must determine the routes themselves and their loads. Our approach is heuristic in nature. For each route, we want the probability of a primary error not to exceed α. Management should decide carefully on an appropriate value for α since there is a delicate tradeoff between customer satisfaction on one hand and extra trip distance and the cost of additional trucks on the other. We assume that almost all of the routes will load up to capacity and seek the optimal artificial capacity $\bar{\mu}$. We have

$$\text{Prob}\left\{ z \geq \frac{c - \bar{\mu}}{\sqrt{\bar{\mu}}} \right\} = \alpha$$

$$\Longrightarrow (c - \bar{\mu}) = z_{1-\alpha}\sqrt{\bar{\mu}}$$

$$\Longrightarrow (c - \bar{\mu})^2 = \bar{\mu} z_{1-\alpha}^2 \text{ which, after some algebra, yields}$$

$$\bar{\mu} = \frac{2c + z_{1-\alpha}^2 - \sqrt{z_{1-\alpha}^4 + 4cz_{1-\alpha}^2}}{2} . \qquad (6)$$

Notice that constraint (5) bounds the maximum variance in route demand. We also remark that if we let $\beta = \text{Prob}\left\{z \leq \frac{\pi c - \bar{\mu}}{\sqrt{\bar{\mu}}}\right\}$, our primary and secondary errors bear a close resemblance to Type I and Type II errors from hypothesis testing. Given $\bar{\mu}$ we can plot β vs. π.

From the analysis above, we have a safety stock (or extra inventory) of $c - \bar{\mu}$ units as a cushion against the occurrence of primary errors. In the case where a route has mean demand $\mu < \bar{\mu}$, let $\mu + (c - \bar{\mu})$ be the load on that route; constraint (5) will be satisfied easily.

In Table I, we illustrate the relationship between c and $\bar{\mu}$ for $\alpha = .10$. For instance, if $c = 100$ and $\alpha = .10$, then $z_{1-\alpha} = 1.28$ and $\bar{\mu} = 87.99$. We could equally well (because of integral demands) use an integral artificial capacity of 87 to set up fixed routes with "demands" of λ_i at node i. The safety stock would be 13.

c	$\bar{\mu}$
10	6.68
20	15.03
30	23.76
100	87.99
1000	960.33

Table I. Relationship between c and $\bar{\mu}$.

DESCRIPTION OF THE ALGORITHM

Suppose we are confronted with a stochastic vehicle routing problem where we know c, α, and λ_i (i = 1,...,n). We outline below a heuristic procedure for calculating a good solution to the problem SVRP.

Algorithm:

Step 0: Given c, α, and λ_i (i = 1,...,n), specify δ as a lower limit on the left-hand-side of inequality (5).

Step 1: Using equation (6) solve for $\bar{\mu}$ the artificial truck capacity.

Step 2: Let λ_i be the demand at node i. Construct fixed routes using the Clarke-Wright code mentioned earlier.

Step 3: Decrement α and repeat steps 1 and 2 if $\alpha > \delta$; otherwise go to step 4.

Step 4: Select the "best" set of fixed routes.

We will apply this solution procedure in the next section to a problem involving 75 customers. In addition, we will analyze its performance.

COMPUTATIONAL RESULTS

We have performed extensive computational experiments using a 75 customer problem as a test case for our approach. The data, taken from Eilon et al. [2], is shown in Table II. For each demand node the coordinates are given along with the mean demand at that node. Demands are Poisson distributed.

Since there are so many variables involved, we have chosen to analyze one test case thoroughly, rather than simulate a myriad of sample problems. We will try to make broad observations and recommendations based on our experience. However, we remark that this work is of an introductory nature; there are many additional questions relating to sensitivity analysis that should be investigated.

No.	x	y	λ	No.	x	y	λ	No.	x	y	λ	No.	x	y	λ
1	22	22	18	20	66	14	22	39	30	60	16	58	40	60	21
2	36	26	26	21	44	13	28	40	30	50	33	59	70	64	24
3	21	45	11	22	26	13	12	41	12	17	15	60	64	4	13
4	45	35	30	23	11	28	6	42	15	14	11	61	36	6	15
5	55	20	21	24	7	43	27	43	16	19	18	62	30	20	18
6	33	34	19	25	17	64	14	44	21	48	17	63	20	30	11
7	50	50	15	26	41	46	18	45	50	30	21	64	15	5	28
8	55	45	16	27	55	34	17	46	51	42	27	65	50	70	9
9	26	59	29	28	35	16	29	47	50	15	19	66	57	72	37
10	40	66	26	29	52	26	13	48	48	21	20	67	45	42	30
11	55	65	37	30	43	26	22	49	12	38	5	68	38	33	10
12	35	51	16	31	31	76	25	50	15	56	22	69	50	4	8
13	62	35	12	32	22	53	28	51	29	39	12	70	66	8	11
14	62	57	31	33	26	29	27	52	54	38	19	71	59	5	3
15	62	24	8	34	50	40	19	53	55	57	22	72	35	60	1
16	21	36	19	35	55	50	10	54	67	41	16	73	27	24	6
17	33	44	20	36	54	10	12	55	10	70	7	74	40	20	10
18	9	56	13	37	60	15	14	56	6	25	26	75	40	37	20
19	62	48	15	38	47	66	24	57	65	27	14				

Table II. Vehicle routing problem with probabilistic demands. Central depot has coordinates (40, 40). Vehicle capacity is 250 units.

In our experiments, we have varied the vehicle capacity c from 100 to 300 units by increments of 50 in order to study the effects. In addition, α takes on the values .01, .05, .10, and .15. Then, for each (α,c) pair, we perform steps 1 and 2 from the algorithm developed in the previous section. Once a fixed set of routes is formed, we simulate a 50 workday period in order to evaluate the effectiveness of the fixed routes. Each day, new demands are generated at each customer location according to the specified Poisson distribution.

The difference $c - \bar{\mu}$ becomes our safety stock, so that when $\mu \leq \bar{\mu}$ is the mean demand on a route, we load the truck assigned to that route with $\mu + (c - \bar{\mu})$ units. This insures that constraint (5) will not be violated. Our approach will be to contrast the performance of the fixed routes against a Clarke-Wright solution which is computed each day after demands d_i $(i = 1,...n)$ are known. The distance for the fixed routes is calculated in the following manner. First of all, distances are Euclidean in the problem under consideration, although they certainly need not be in general. If a route with mean demand μ has a demand which exceeds $\mu + (c - \bar{\mu})$, then the truck assigned to the route will have to return to the central depot in order to finish its route. Again, assume it carries a safety stock of $c - \bar{\mu}$ units for the remainder of the trip or, more logically, assume the demands become known exactly. The distance for the route is the total distance covered by the truck, including the return round trip to the central depot.

For a given day, the ratio of the distance for the fixed set of routes to the distance for the Clarke-Wright routes will be our principal performance measure. Since for each (α,c) pair the simulation produces fifty days of random demands, we focus attention on the average ratio and the worst-case ratio. Table III displays our findings. In general, as c increases the ratios decrease (we will come back to this point later). Furthermore, we should point out that for the original problem, where c = 250, an α level of .10 yields an excellent set of fixed routes. The average ratio is 1.024 while in the worst case the ratio is still only 1.107.

We remark that our computer code currently sets the initial load on each truck to c rather than $\mu + (c - \bar{\mu})$. This is being remedied; we expect the alteration to have a negligible effect on our conclusions.

α \ c	100		150		200		250		300	
	A	WC	A	WC	A	WC	A	WC	A	WC
.01	1.175	1.269	1.080	1.141	1.099	1.143	1.052	1.112	1.035	1.080
.05	1.123	1.183	1.086	1.157	1.066	1.143	1.027	1.101	1.052	1.092
.10	1.118	1.240	1.087	1.178	1.055	1.130	1.024	1.107	1.049	1.146
.15	1.107	1.171	1.088	1.176	1.034	1.153	1.040	1.149	1.041	1.106

Table III. Average and worst-case ratios. The column heads A and WC denote average and worst-case ratios respectively.

Relating to the same fifty days of random demand, we report on additional measures of performance in Tables IV and V. In Table IV we display the average percent of unused truck capacity and the average proportion of routes which incur a primary error (demand exceeds $\mu + (c - \bar{\mu})$). We notice that as α increases for a fixed c, the average percent of unused truck capacity tends to decrease, and the average proportion of routes which incur a primary error (this will usually be a lower bound for α) increases.

α \ c	100		150		200		250		300	
	A	B	A	B	A	B	A	B	A	B
.01	28.54	.004	24.57	.005	24.57	.009	22.4	.006	24.57	.007
.05	24.04	.023	17.14	.035	14.55	.033	21.87	.034	24.04	.043
.10	19.40	.071	16.96	.104	14.36	.098	21.7	.089	8.65	.128
.15	19.80	.12	17.37	.127	14.79	.12	9.11	.15	9.11	.148

Table IV. The column heads A and B denote average percent of unused truck capacity and average proportion of routes which incur a primary error.

In Table V, we show for each (α,c) pair the corresponding value of $\bar{\mu}$, the number of routes in the fixed set of routes, and the average number of routes more than is actually needed (that is, if demands were known in advance). We see that the entries in columns B and C decrease as truck capacity is increased for a fixed level of α.

α \ c	100			150			200			250			300		
	A	B	C	A	B	C	A	B	C	A	B	C	A	B	C
.01	79	19	4.24	124	12	2.14	169	9	1.60	215	7	.98	262	6	1.0
.05	84	18	3.12	131	11	1.04	178	8	.56	225	7	.98	272	6	1.0
.10	87	17	2.00	135	11	1.04	182	8	.5	230	7	1.0	278	5	0.0
.15	90	17	2.06	137	11	1.06	185	8	.52	234	6	0.0	282	5	0.0

Table V. The column heads A, B, and C denote $\bar{\mu}$, the number of routes in the fixed set of routes, and the difference between the number of fixed routes and the average number of routes needed if a new solution is generated each day.

OBSERVATIONS AND RECOMMENDATIONS

We have now solved a sample SVRP for various (α,c) combinations. In this section, we try to reach some conclusions based on the computational results reported in the previous section. We discuss several below.

(i) For a fixed level of α, the efficiency of the routes will improve as c increases (see Table III). The reason for this is that for larger values of c the standard deviation in demand for a route is small relative to the mean demand. This means that the ratio $\frac{\bar{\mu}}{c}$ will increase as c increases and that the fixed set of routes will be "fuller" for large c than for small c. For instance, for $\alpha = .01$, the ratio $\frac{\bar{\mu}}{c}$ increases from .79 to .873. These arguments are verified in Tables IV and V.

(ii) We have not found that a particular level of α is best, in general. Rather, it seems that, roughly, for $.87 \leq \frac{\mu}{c} \leq .93$ the average and worst-case ratios are minimized although these ratios are fairly insensitive to small changes in α. For example, with c = 250, we expect $\alpha = .05$ or $\alpha = .10$ to be best strategies for α from among the four choices considered. This finding is consistent with previous work reported by Stewart [7].

(iii) The customer demand distributions remained fixed during our computational experiments. If we were to vary these distributions, the average number of stops on a route would become an important parameter. Our algorithm performs better when a truck is capable of handling more demand points. Because our results apply to only one set of demand distributions, we must be cautious in reaching conclusions. We underline this fact here.

(iv) For $c \geq 150$, our algorithm performs quite satisfactorily. The best average ratios for each value of c (150, 200, 250, 300) are all under 1.08. That is, on the average, we require less than 8% more travel distance than we would need if all demands were known in advance and drivers covered different routes each workday.

(v) Because of the additive properties of the Poisson distribution we were able to replace d_i with the parameter λ_i and apply the Clarke-Wright algorithm to obtain a fixed set of routes. We can proceed similarly if the demand at node i is:

(a) distributed binomially with mean $n_i p$,

(b) gamma distributed with mean $\theta_i b$, where θ_i is the shape parameter and b is the scale parameter,

(c) negative binomially distributed with mean $\frac{x_i (1 - p)}{p}$. Kao [6] discusses these same issues in the context of the stochastic traveling salesman problem where travel times are random variables with large variances.

In this paper, we have developed a framework for dealing with the vehicle routing problem with probabilistic demands. There are a host of additional, complicating considerations which should be examined in further work. The following questions come to mind: How does the geometry of the transportation network influence the effectiveness of routes? How sensitive are routing strategies to changes in the distribution of customer demands? Is our objective function realistic or appropriate? Can intercorrelation of demands be incorporated into our basic approach? What happens when both travel times and customer demands are probabilistic in nature? We would hope that a real situation could be studied in the near future to help address some of these questions. We feel that this is an important research area with great potential applications, which deserves much more research attention.

REFERENCES

1. Clarke, G. and Wright, J. W. "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points," Operations Research, 12 (4), 568-581 (1964).

2. Eilon, S., Watson-Gandy, C., and Christofides, N. Distribution Management, Griffin, London (1971).

3. Feller, W. An Introduction to Probability Theory and Its Applications, Vol. 1, 2nd ed., John Wiley & Sons, Inc., New York (1957).

4. Fox, B. L. "Data Structures and Computer Science Techniques in Operations Research," forthcoming in Operations Research.

5. Golden, B., Magnanti, T., and Nguyen, H. "Implementing Vehicle Routing Algorithms," Networks, 7 (2), 113-148 (1977).

6. Kao, E. P. C. "A Preference Order Dynamic Program for a Stochastic Traveling Salesman Problem," TIMS/ORSA Bulletin, No. 3, p. 169 (1977).

7. Stewart, W. R., Jr. "The Delivery Truck Routing Problem with Stochastic Demands," Management Science/Statistics Working Paper MS/S 76-005, University of Maryland at College Park (1976).

8. Tillman, F. A. "The Multiple Terminal Delivery Problem with Probabilistic Demands," Transportation Science, 3 (3), 192-204 (1969).

BIOGRAPHIES

Dr. Bruce L. Golden is an Assistant Professor of Management Science and Statistics at the University of Maryland. His research interests include network optimization, mathematical programming, and applied statistics, and he has published a number of articles in these fields. Dr. Golden received his BA from the University of Pennsylvania where he graduated summa cum laude with honors in mathematics. Later he earned his MSOR and Ph.D. in Operations Research from M.I.T. He is a member of ORSA, MAA, SIGMAP, the Mathematical Programming Society, and Phi Beta Kappa.

Mr. William Stewart is an Assistant Professor in the School of Business at the College of William and Mary. He is a candidate for a D.B.A. at the University of Maryland where he has been a lecturer in Management Science and Statistics for the last five years. Prior to that he was a Systems Analyst at the Westinghouse Defense and Space Center in Baltimore. He holds a Bachelors Degree in Mechanical Engineering from Tufts University and a Masters in Management Science from Johns Hopkins.

detail so they can be located easily. In SPSS there seems to be more emphasis on speed and efficient use of space and therefore less on data cleaning. If you are processing a file that is fairly 'clean' with one record per case and wish to do a single operation, there is no question that SPSS will be faster. If you wish to do several operations the differences will not be great. If you are processing a file with more than one record per case or with many keypunching errors, there is no question that P-STAT will catch more of these errors as it is processing the data and will report them in sufficient detail so that you may well be able to build a clean file in P-STAT with more efficiency than would have been possible in SPSS.

While the checking of row labels and sequence numbers when there is more than 1 record per case, and the exceptional amount of checking for mispunchings - as well as the way these errors are reported - are the most important aspects of the DATA program, there are several other features which can be extremely useful. In P-STAT, you do not need to have all the records for a case. You can tell the DATA program, for example, that while you have 9 possible records per case, it is all right if some of them are omitted for a given case. In this situation, you could also specify that each case must have at least 4 records including records 1 and 9. If a record does not fit these requirements it is not included in the file. This facility is particularly useful with medical data where patients may have different numbers of visits. With this facility, you do not have to supply dummy records for the missing visits. Thus it is possible to build a file with only the subjects who have at least a required minimum number of records. These features are possible only because of the use of row labels and sequence check fields, and also because the space is there for the extra code.

3. FILE MANIPULATION

The two systems differ in a fundamental way here. SPSS has a single system file as the file during a run. P-STAT's structure allows 20 P-STAT system files to be simultaneously accessible, and a number of P-STAT commands use three or four files at one time. This is a most basic design difference.

3.1 Usefulness of several system files. In P-STAT, one might correlate all but the demographic variables in a file, use those correlations (a second P-STAT system file) in regressions, get residuals (also a P-STAT system file), combine those with the demographic variables in yet another file and use it for crosstabulation, F tests, etc. It is all very smooth and natural. This is quite difficult in SPSS. The SPSS residuals can only be saved as raw card images. One must initiate another SPSS job to combine them with part or all of the original file in SPSS system file form. SPSS does provide some multi-file flow in this manner, but the system clearly was not designed to do it smoothly.

3.2 Combining files. In SPSS you can use MERGE FILES to combine variables in 2 to 5 existing SPSS system files, or use ADD VARIABLES to combine new raw input variables with an existing SPSS system file. These must be done at the start of an SPSS run to produce "the" SPSS system file for the run. Because of the lack of row labels, one must have case ID variables in all combined files, create new variables by subtracting (numeric) case IDs and produce a frequency of them to be sure that the correct cases were combined. P-STAT has a JOIN command that is comparable. It can be done at any point in a run. Checking on row labels or on designated variables is done automatically as the JOIN is being done.

3.3 Additional P-STAT file manipulation commands. Because P-STAT has a multi-file design, it was very natural to develop a series of file comparison and modification commands. MATCH finds the cases in two files whose row labels match, no matter what order the files are in. COLLATE can be used to join a mother's data with each of her children. SORT can be done at any point in a run and its result used immediately. There are several others.

3.4 Subfile structures. The SPSS design was to make their one file as good as possible. This is demonstrated in its subfile structure. The SPSS subfile structure is quite powerful, particularly if you are working with data, for example, for each of the 50 states and at some times wish to work with individual states or collections of states and at other times with the whole file. However, while you are not locked into a subfile structure, it is very awkward to change it. You must use 3 separate job steps, which cannot all be combined into one job. The first step sorts the input file into the new subset order. Because an SPSS sort must be last in an SPSS run, a second step is needed to do a frequency of the subset variable. It prints the counts of the members in each new subfile. The third step is to input those counts to SPSS so it can build the new SPSS system file incorporating this subfile structure.

In P-STAT, when you have the type of data appropriate for SPSS subfiles, you would either build several small files and then dynamically concatentate them when you wished to use more than one, or you would build a large file and select appropriate subsets. This is more flexible but not as convenient as the SPSS subfiles operation when there are a large number of subfiles. A P-STAT user frequently uses MACROS of P-STAT commands in situations where an SPSS user makes use of subfiles. A P-STAT MACRO is a series of P-STAT commands which, once defined, can then be invoked repeatedly to process different files or subsets of files.

If you are working with a large file which falls naturally into a single subfile structure with many subfiles, the SPSS approach may be very convenient. If, on the other hand, you are working with files that are updated frequently or which require changing the subfile structure for different analyses, the P-STAT approach is more flexible. The use of multiple files also makes the saving of correlation matrices or factor scores a trivial process in P-STAT. In SPSS you can write these files as data on a scratch unit, but unless you supply separate JCL for each array saved, they will be written one behind the other and it is up to you to write a program to recover them.

4. REPRESENTATION OF MISSING DATA

Both systems allow 3 missing values for a variable. P-STAT system files use 3 explicit values, -123456.E20, -123457.E20, and -123458.E20 to indicate missing data. SPSS allows you to define the values for each variable which are to be considered missing, but does not recode them to general system missing values. This may not seem to be an important difference but it has a number of subtle effects.

4.1 Unique versus original-score representation of missing. As the P-STAT DATA command makes a system file, all ways on all variables of indicating missing, blank or invalid data are automatically recoded into one of the three unique values. This makes it very easy for both users and P-STAT itself to notice missing data. The SPSS file, on the other hand, contains a table of the three different missing values for each variable. It remembers, for example, that 9 on SEX was defined as missing when the file was built. Cases with missing data 1 on SEX therefore continue to have a value of 9. P-STAT prints such a value as literally 'M1'. SPSS prints a 9 and it is up to you to remember that on SEX a 9 means missing. (This is not too had for the variable SEX, but may be more difficult with a variable like EDUCATION or AGE.)

P-STAT treats missing data as special in all circumstances. Any computation involving missing data automatically produces a missing result, thus the user is quite well protected. SPSS treats missing data as normal unless a calculation is involved, and even then (see below) it is possible for an SPSS user to be careless and erroneously use the missing value in an unwanted calculation. This all boils down to a system design-time trade-off; SPSS felt that it was important to retain the original code value that meant missing, we (in 1962) decided that a unique value was better.

4.2 Problems with missing values in crosstabulation. Because SPSS does not have unique internal representation for missing values, we have had numerous little problems setting up runs that were nearly identical in both P-STAT and SPSS. If you wish to have a count of your missing data in SPSS CROSSTABS, you can do it but only in integer mode, which requires giving the range of scores for each variable. This range must include the missing values or they will be omitted. If you have simplified entering the missing data and solved the problem of remembering which scores are missing by using as the missing value a score that is higher than any of the actual scores in the file, for example 99, you suffer a penalty in doing CROSSTABS which allocates enough core to hold all the values from the lowest to the highest.

The crosstab for AGE by SEX, if AGE is coded 0-9 and 99 and SEX is coded 1-2 and 99, will need 9,900 cells for a single table. If you assign 3 as the missing score for SEX and 10 as the missing score for AGE, you must constantly remember what those values are. This is no problem in P-STAT because of its unique missing values. A similar table in P-STAT would allocate space for 0-9 plus missing by 1-2 plus missing, a total of 33 cells.

4.3 Problems with missing values in transformations. SPSS's lack of system missing value settings causes awkwardnesses in the transformation language. Consider the following arithmetic effects in SPSS....

```
COMPUTE         A = B + C
```

In this example, if you do not recode C to a defined missing value and it is blank on an input record, you will get in effect, A = B + 0. On the other hand, if you do recode blank to 9 on variable C and define 9 as missing for variable C, you must remember to supply an 'ASSIGN MISSING' card or you will get A = B + 9. The same trap exists when you say....

```
IF (some test) X = B
```

Suppose X is a new variable and the test is not true. If you do not explicitly specify an 'uncomputed' value for X using an ASSIGN MISSING card or a MISSING VALUES card, the value for X is zero. In P-STAT it automatically is Missing Value 1, which is quite a bit safer for the user.

5. CONCLUSIONS

The issues described here are areas that we think are important and have always thought so, which is why we believe our design in these areas was good. SPSS, it should be said, does numbers of things in social science computing extremely well, and has some capabilities that we will never have. P-STAT, for the reasons cited above, can handle some areas more smoothly than SPSS. There are benefits to social science computer users in having a variety of tools at hand, particularly when they have somewhat differing strengths. The increasing availability of interfaces between system files should be helpful in this respect.

BACKGROUND

The authors received B.A. degrees from Oberlin in the early 50's and began P-STAT development in the early 60's. Both are on the Technical Staff of the Computer Center at Princeton University.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

INSTRUCTIONAL USE OF STATISTICAL PROGRAM PACKAGES: BMD, IMP, OMNITAB II, AND SPSS

Ronald E. Wyllys
Graduate School of Library Science, University of Texas at Austin, Austin, TX 78712

ABSTRACT

An introduction to inferential statistics forms the major part of a research-methods course taught for students whose backgrounds are predominately non-mathematical and non-scientific. Course objectives include developing the student's confidence in his or her ability to solve practical, library-oriented problems (1) through statistical techniques and/or (2) with the aid of computers. Both objectives are served by the emphasis in the course on using computer program packages that perform statistical tasks. Students begin with OMNITAB II and IMP. The former is available at UT-Austin in the original batch-mode package and also in a somewhat condensed interactive version prepared locally. IMP, based on and very similar to OMNITAB II, was locally written specifically for interactive use. After acquiring moderate facility in interactively manipulating columns of observed data in OMNITAB II and IMP, and after some experience in batch-mode use of OMNITAB II, students are introduced to the more formal approach required in SPSS, progressing from examples with detailed explanations to the point of setting up their own problems. An exercise using a BMD regression routine introduces the students to this package. Throughout the course the students are made to realize that most of them will be working in environments in which they will have access to a computer with one or more of these statistical packages, and that solutions to on-the-job problems will be "only a keyboard away."

Key words: BMD; IMP; OMNITAB II; SPSS; statistical program package; statistics, teaching of.

1. INTRODUCTION

Still very much in evidence in today's world is the stereotype of the librarian as a "little old lady in tennis shoes" mainly concerned with shushing the visitors to her library or, unfortunately, according to television concerned with giving advice on laxatives. Those who cling to such stereotypes may be surprised to learn that today's library school students are typically enthusiastic and forceful young advocates of making libraries effective institutions for social change and individual growth. (Incidentally, some 25%-35% of these students are men, and all the students could hardly care less about audio levels in libraries.)

The strong tendency to view libraries and librarianship as a social force is reflected in current education for librarianship. Increasingly, library science has come to be considered one of the social sciences, the one whose domain is communication among people, with emphasis on those communications that are recorded in written, graphic, electromagnetic, or other semi-permanent forms. As a social science, library science recognizes its need of the research tools of the other social sciences. Accordingly, increasing numbers of library schools are offering courses in research methods.

The Graduate School of Library Science (GSLS) of the University of Texas at Austin

(UT-Austin) not only offers, but also requires all students to take, a course called "Research in Library Science." The aim of this course is to provide the students with a basic knowledge of standard tools of research including, in particular, statistics and the use of computers. The faculty recognize the need to overcome the problems presented by the fact that the great majority of the students took their undergraduate work in fields outside the sciences--social or physical, including mathematics and computer science. This is reflected, for example, in GSLS students' scores on the Graduate Record Examination; the students have a mean of about 550 on the Quantitative Aptitude test compared with a mean of about 640 on the Verbal Aptitude test. Remarks by beginning students frequently evidence fear of, or at least hostility toward, mathematics and/or computers.

The purposes of the GSLS research-methods course include, therefore, overcoming these attitudinal and cognitive handicaps on the part of the students. An objective of the course is to develop each student's confidence in his or her ability to solve practical, library-oriented problems (1) through statistical techniques and/or (2) with the aid of computers. Since assuming responsibility for the research-methods course in 1972, the author has tried to serve both these objectives by emphasizing in the course the use of computer program packages to perform statistical tasks.

2. FACILITIES

Among the program packages available at the UT-Austin Computation Center (UTACC) are BMD, IMP, OMNITAB II, and SPSS. OMNITAB II is available in two batch-mode versions, known locally as OMNITAB L, which has the original 12,462-cell worksheet, and OMNITAB, which has been modified to have a 1000-cell worksheet. The latter is also available in an interactive version, in which some of the output is condensed. IMP is an interactive adaptation of OMNITAB II, written at the UTACC by G. Scott Harris specifically for fast response under the time-sharing algorithm employed by the UTACC's CDC 6600/6400 system (Swanson et al., 1975). It has since been installed in other computing centers. IMP is also available interactively through the UTACC's DECsystem-10. BMD is installed only on the CDC 6600/6400. SPSS is available on both systems.

The 250 full- and part-time students and the 14 full-time faculty of GSLS can work with these computers via a Texas Instruments model 733 hard-copy terminal, a TI 745 portable hard-copy terminal, and an Ontel model OP-1 cathode-ray-tube terminal, all in the School's quarters. Communication channels consist of hard-wired lines to the CDC 6600/6400 and the DECsystem-10, and dial-up connections to both computers. A keypunch is provided by the UTACC in a remote job-entry and -output site on the floor below the School, one of several such sites on the UT-Austin campus.

Although this report deals with instruction in computer-based statistical analysis, it should be mentioned that students are also required to do several exercises to become familiar with some of the more sophisticated electronic calculators. Among the exercises are non-elementary ones in analysis of variance and chi-square analysis. Currently, the School makes available for student use Hewlett-Packard models 67 and 45 and a Commodore model S-61 (Statistician). The students are encouraged to use their own calculators in class and for quizzes.

3. INSTRUCTIONAL REFERENCE MATERIALS AND EXERCISES

Space does not permit reproduction here of the more than 70 pages of notes and exercises used by the students in the research-methods course. Therefore, this discussion will attempt to summarize the contents of these materials, any or all of which are available upon request to the author.

3.1 Reference materials. At the beginning of the course, the students receive three basic handouts on using computers. "Talking to Taurus" tells the students how to use the CDC 6600/6400 interactively. "Dealing with the DEC-10" does the same thing for the

DECsystem-10. Both of these begin by telling the student what "interactive use" means, and describe the necessary steps down to the level of when to turn what switch on or off. Common problems in transferring typing habits from typewriters to terminals are discussed, typical system difficulties are described, and even the Computation Center hours are included. "Keypunching Simplified" tells the student in similar fashion how to use a keypunch. These materials are intended to introduce the UTACC facilities to students, some of whom have never before used any computer. Fortunately, the proportion of such students is decreasing.

Also given to the students is an introduction to OMNITAB and IMP, "OMNITAB-IMP Notes," with appendices that provide the students with a quick-reference guide to the commands available in these packages. Currently IMP lacks several of OMNITAB II's most important statistical commands (e.g., STATISTICAL ANALYSIS and CORRELATION). At the author's request the UTACC is working on the addition of a number of these commands to IMP.

The students are urged to purchase the SPSS Primer (Klecka et al., 1975), not only as a manual for SPSS but also as a very readable introduction to computers and to statistics. It would be most helpful if there were a comparable primer for OMNITAB II. The existing OMNITAB II User's Reference Manual (Hogben et al., 1971) is a reference manual, useful for experienced computer users but very difficult for novices to learn from.

3.2 Computer-based statistical exercises. The aim of the computer-based statistical exercises is to develop the students' skills and confidence in using computer assistance to handle statistical problems. The rest of this section consists of comments on the exercises, presented in the order in which they are assigned to the students.

3.2.1 Introductory Manipulations.

OMNITAB-IMP Problem I. Gives the student 12 numbers. Asks the student to use IMP or OMNITAB interactively to find the mean of the numbers and then their standard deviation, considering them first as a sample and second as a population. Familiarizes the student with the idea of manipulating columns of data and with basic commands.

OMNITAB-IMP Problem II. Gives the student 11 three-digit numbers and asks the student to supply a twelfth from the last three digits of his or her Social Security Number. Introduces batch-mode usage by requiring the student to prepare the data and program cards to perform the OMNITAB command STATISTICAL ANALYSIS on the twelve numbers. This very powerful command yields a large number of results: e.g., mean, median, mid-range, 25-percent trimmed mean, standard deviation, standard error of the mean, range, mean deviation, variance, coefficient of variation, 95-percent confidence intervals for the population mean and standard deviation, minimum, maximum, the t-score testing the hypothesis that the population mean is zero, linear trend statistics, tests for non-randomness of the observations and of their deviations from the mean, and lists of the observations in original and in sorted sequence, with their ranks in both sequences. In class the author provides a full discussion, based on Ku (1973), of the output from STATISTICAL ANALYSIS, using its features as a springboard for reinforcing various concepts already introduced in the lectures and for looking ahead at ideas to be treated later in the course.

OMNITAB-IMP Problem III. Introduces the use of large tape- or disk-based files as the source of data, by asking the student to use a tape file, GRADS, that contains sex, age, verbal score on the Graduate Record Examination (GRE), and quantitative score on the GRE for 135 randomly selected former students. These data are used because of their familiarity to the student. The exercise begins with the creation of histograms using various class sizes. Following the histograms, the student is asked to apply STATISTICAL ANALYSIS to the verbal and quantitative GRE scores and to examine the results of the various tests of non-randomness. Then the verbal-quantitative pairs are sorted on the verbal scores, resulting in a complete ordering of the verbal scores and a partial ordering of the quantitative scores. The role of the correlation between the verbal and quantitative scores is pointed out to the student. Then the student again applies STATISTICAL ANALYSIS to both sets of scores, and the student is asked to compare the new results of the non-randomness

tests with the original results.

SPSS Problem I. Introduces the student to SPSS. Treats the preparation of two-way frequency tables, provides a first glimpse of chi-square analysis, and displays the excellent capabilities of SPSS for handling missing data and formatting output. Uses a copy of the tape-based data-file, GRADS, with sex data removed from two cases and replaced by a missing-value code.

3.2.2 Note on Tests of Statistical Hypotheses.

After SPSS Problem I the lectures take up the theory of testing statistical hypotheses. In all subsequent exercises, the students are required to formulate, in words relevant to the situation described in the exercise, both the null hypothesis being tested and the resulting acceptance or rejection decision.

3.2.3 Analysis of Variance.

OMNITAB-IMP Problem IV. Treats a two-population single-classification ANOVA problem that is discussed in the textbook (Hardyck and Petrinovich, 1976) for the course, so that the student may see how the ONEWAY command in OMNITAB II displays the results. In class the author draws the students' attention to the calculation of the significance level of the observed F-ratio and to some of the attractive features of ONEWAY, especially the incorporation of the Kruskal-Wallis rank test and the Newman-Keuls and Scheffé techniques. The discussion also compares this ANOVA problem with what the students have learned earlier about the t-test for the difference of population means.

OMNITAB-IMP Problem V. Treats a five-population single-classification ANOVA problem from the course textbook. The class discussion touches on the F-ratio for the slope of the group means, another attractive feature of ONEWAY, and reinforces the use of the Newman-Keuls and Scheffé techniques.

OMNITAB-IMP Problem VI. Applies single-classification ANOVA to the tape-based file of data, GRADS, which the students have already examined in OMNITAB-IMP Problem III and SPSS Problem I. The students are asked to determine whether it appears that men and women differ with respect to (1) verbal GRE scores and (2) quantitative GRE scores.

OMNITAB-IMP Problem VII. Applies double-classification ANOVA without replication, since OMNITAB's TWOWAY command carries out only this kind of two-way ANOVA. The problem is a 4x2 table in which only the possible column differences are of interest. The student's attention is drawn to the fact this situation is analogous to those for which the student has used the t-test for the difference of means of independent and non-independent groups.

SPSS Problem II. Treats double-classification ANOVA with replication as performed by SPSS, using a problem discussed in the course textbook. Also introduces the student to the use of data in punched-card form in SPSS.

3.2.4 Chi-Square Analysis.

Memorandum on "Using OMNITAB-IMP for Chi-Square Analysis." A comment rather than an exercise, this memorandum explains the use of stored commands in OMNITAB II, in a problem concerned with the chi-square test of association.

SPSS Problem III. Applies the chi-square test of association to the tape-based file, GRADS, that the students have already examined in terms of histograms in OMNITAB-IMP Problem III,

frequency tabulations in SPSS Problem I, and ANOVA in OMNITAB-IMP Problem VI. The point is, of course, to compare the analyses of one set of data via various statistical procedures (one more is yet to come in Correlation Problem 4).

Three other chi-square problems are provided. The students are allowed to choose whether to work them using a computer or using an electronic calculator.

3.2.5 Correlation.

It should be emphasized here that the students are required to handle the correlation problems, like the other problems in the course, as tests of statistical hypotheses. In the correlation problems the only null hypothesis discussed is that the population correlation coefficient (whether Pearson or Spearman) is zero.

Correlation Problem 1. Treats a small (but tape-based for convenience) data file via the CORRELATION command in OMNITAB II. The data are assumed to be suitable for the Pearson product-moment correlation coefficient. The discussion touches on the use of the significance-level and confidence-interval computations displayed on the CORRELATION printout.

Correlation Problem 2. Shows how the CORRELATION command calculates the Spearman rank-order correlation coefficient. Uses a tape-based file of rank data.

Correlation Problem 3. Applies CORRELATION to a tape-based file, using the Pearson correlation coefficient. It turns out that $r = .98$, and this provides a springboard for discussing the coefficient of determination.

Correlation Problem 4. Introduces the use of partial correlation coefficients. The student is asked to use CORRELATION to analyze, from the viewpoint of correlation, the same tape-based file, GRADS, examined earlier in OMNITAB-IMP Problems III and VI and SPSS Problems I and III.

3.2.6 Regression

Regression Problem I. Introduces regression and exposes the student to BMD, using BMD05R. The problem provides a small sample of pairs of heights of brothers and sisters. The sample is too small for the correlation to be significant. The students' attention is called to the discrepancy between their knowledge that sibling heights do tend to be similar and the failure here to reject the null hypothesis of no correlation. This discrepancy affords an opportunity to reinforce their understanding of the role of sample size in interpreting the significance of an observed correlation.

Regression Problem II. Applies the OMNITAB II command FIT to a tape-based file of data on the value of the dollar from 1947 through 1976. (The problem is updated annually.) The use of the PLOT command is introduced, and a UTACC-written link produces output from OMNITAB II on a CalComp plotter. In class the capabilities of FIT for curvilinear and multiple regression serve as the basis for a brief discussion of these techniques.

4. SUMMARY

The exercises discussed above lead the student from elementary arithmetic manipulations of data to the use of powerful statistical commands in three major statistical program packages. In all but the initial lectures and statistical exercises, the emphasis is on how to set problems up for computer solution and on the student's interpreting the results

provided by the various programs.

Throughout the course the student is repeatedly reminded that he or she will very likely be working in a library or other information agency with access to a computer system in which a statistical program package is, or can be, installed. The total cost for computer time and supplies for all the exercises averages about $15 per student. The low costs of the individual problems are brought to the student's notice as further evidence of the practicality of computer-based statistical processing.

5. REFERENCES

HARDYCK, C. D., and PETRINOVICH, L. F. (1976). Introduction to Statistics for the Behavioral Sciences, second edition. Philadelphia, PA: W. B. Saunders. (This is currently being used as the statistics text for the research-methods course.)

HOGBEN, D., PEAVY, S. T., and VARNER, R. N. (1971). OMNITAB II User's Reference Manual. NBS Technical Note 552. Washington, DC: National Bureau of Standards.

KLECKA, W. R., NIE, N. H., and HULL, C. H. (1975). SPSS Primer. New York: McGraw-Hill.

KU, H. H. (1973). A User's Guide to the OMNITAB Command "STATISTICAL ANALYSIS". NBS Technical Note 756. Washington, DC: National Bureau of Standards.

SWANSON, J. M., RIEDERER, S. A., REYNOLDS, E., and HARRIS, G. S. (1975). IMP and SHRIMP: Small, interactive mimics of OMNITAB designed for teaching applications. Proceedings of Computer Science and Statistics: 8th Annual Symposium on the Interface, p. 84. Edited by J. W. Frane. Los Angeles, CA: Health Sciences Computing Facility, University of California, Los Angeles.

BIOGRAPHY

Ronald E. Wyllys is Assistant Professor, Graduate School of Library Science, University of Texas at Austin. He was formerly Chief Systems Analyst, University Libraries, and Lecturer, Computer Sciences Department, University of Wisconsin--Madison. He also worked as a computer systems analyst for the System Development Corporation, and as a mathematician for the Planning Research Corporation and the Department of Defense. He holds a B.A. in mathematics from Arizona State University and a Ph.D. in information science from the University of Wisconsin--Madison.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

A ROBUST PROCEDURE FOR ESTIMATING THE TREND-CYCLE COMPONENT OF AN ECONOMIC TIME SERIES

Edward L. Frome
Oak Ridge Associated Universities
Oak Ridge, TN 37830

Ronald D. Armstrong
University of Texas
Austin, TX 78712

ABSTRACT

Economic time series are often represented as a composite of trend-cycle, seasonal, and irregular movements. We propose that the cubic spline regression method be used to estimate the trend-cycle component and that parameters be estimated by minimizing the sum of the absolute values of the deviations. If there is a seasonal component present, the regression model can be extended using dummy variables. In both cases, least absolute value estimates are obtained using a special purpose linear programming algorithm. An example of the application of the cubic spline smoothing procedure to monthly Texas construction data is discussed.

Key words: Cubic spline; least absolute values; time series; robust; data analysis; linear programming; trend-cycle; L_1 norm.

1. INTRODUCTION

Suppose that we have observed values of a variable y at equidistant time points. A problem of considerable practical interest is to obtain a new sequence of "smoothed" values whose terms differ "as little as possible" from the terms in the original sequence. The smoothed sequence is referred to as the trend-cycle component. If the trend-cycle component is subtracted from the data, then the residuals are called the "noise"--or possible noise plus seasonal component of the time series. One approach to this problem of time series decomposition has been developed by the Bureau of the Census, and their computer program Census X-11 has been widely used in government and industry--see Shiskin, Young, and Musgrave (1967). Cleveland and Tiao (1976) have proposed a stochastic model for which the linear filter version of the Census X-11 program is nearly optimal and have discussed its relationship to the Box-Jenkins' approach to time series analysis.

In this paper, we propose that the trend-cycle component be represented with an "empirical function" composed of polynomial pieces called cubic splines--see Section 2. The application of spline functions in data analysis has been considered by Wold (1974); and Buse and Lim (1977) have shown that when the least squares principle is used to estimate the parameters, the cubic spline regression method is a special case of restricted least squares.

We propose that the least absolute value principle be used to estimate the unknown parameter. Consequently, the procedure is "robust" with respect to model specification and the method of estimation. The least absolute value estimates are obtained using a special purpose linear programming algorithm, and an efficient starting procedure is described.

2. DEFINITION OF THE MODEL

Consider the problem of estimating the parameters of a polynomial, $y = f(t)$, where the parametric structure varies over t. The domain of t is divided into a set of $(k + 1)$ intervals which are defined by the knots $(t_j^*;\ j = 1,\ldots,k)$ and within each interval

$$y = f_j(t) = a_j + b_j t + c_j t^2 + d_j t^3. \tag{2.1}$$

We assume that the knots are known, that they are in order, and that the polynomials are joined together at the knots by the following continuity restrictions:

$$\begin{aligned} &f_j(t_j^*) = f_{j+1}(t_j^*),\ f_j'(t_j^*) = f_{j+1}'(t_j^*), \text{ and} \\ &f_j''(t_j^*) = f_{j+1}''(t_j^*),\ j = 1,\ldots,k. \end{aligned} \tag{2.2}$$

These restrictions specify that the level and first and second derivatives of the polynomials at the knots are equal.

Suppose that we have observed values of a variable y_t at equidistant time points $t = 1,\ldots,n$. An equivalent expression to (2.1) and (2.2) is

$$f(t) = \Sigma_{j=1}^{4} \beta_j t^{j-1} + \Sigma_{j=5}^{k+4} \beta_j (t - t_{j-4}^*)_+^3, \tag{2.3}$$

where $(t-t^*)_+^3 = (t-t^*)^3$ if $t \geq t^*$, and otherwise is equal to zero.

In (2.1) there are $4(k + 1)$ parameters, but the continuity restrictions (2.2) reduce the dimensionality of the parameter space to $k + 4$. The cubic spline (2.3) is a smooth function that represents the trend-cycle portion of the time series.

In many situations, there may also be a seasonal component in the time series. We assume that the seasonal component is additive and that there are s observations per season (i.e., $s = 12$ for monthly data). The seasonal terms are represented using dummy variables, and the combined seasonal plus trend-cycle model is

$$y_t = \beta_1 x_{t1} + \ldots + \beta_s x_{ts} + \beta_{s+1} x_{t,s+1} + \ldots + \beta_m x_{tm} \tag{2.4}$$

where

$$x_{tj} = \begin{cases} 1 \text{ if } t - s[(t-1)/s] = j \\ 0 \text{ otherwise} \end{cases} \qquad j = 1,\ldots,s\ ,$$

$$x_{tj} = t^{j-s},\ j = s+1, s+2, s+3, \text{ and } x_{tj} = (t - t_{j-s-3}^*)_+^3,\ j = s+4,\ldots,s+k+3.$$

In the above expression, $[x]$ denotes the integer part of x; and $m = s+k+3$ is the number of

parameters in the model. Note that when s=1, we obtain (2.3) as a special case of (2.4).

3. LEAST ABSOLUTE VALUE ESTIMATION

The least absolute value (LAV) curve-fitting problem can be stated as follows. Given $(y_1, x_{i1}, x_{i2}, \ldots, x_{in})$, $i = 1, \ldots, n$, in m + 1 dimensional Euclidean space, we wish to find $(\beta_1, \beta_2, \ldots, \beta_m)$ to minimize

$$\Sigma_{i=1}^{n} |y_i - (\beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_m x_{im})|. \tag{3.1}$$

The LAV, or L_1 norm, estimates have long been recognized as an acceptable alternative to least squares. Fourier appears to have been the first to consider the computational problem and formulated the solution in the form of what is now called a linear programming problem (see Harter, 1974). Until recently, the LAV estimation procedure has received little attention since the labor involved is considerable. Recently, Schlossmacher (1973) presented an alternative method for solving (3.1) using iterative-weighted least squares. Armstrong and Frome (1976) have shown, however, that the most realistic approach to solving the LAV estimation problem is to re-express (3.1) as a linear programming problem and then apply a special-purpose primal algorithm.

The LAV curve-fitting problem can be rewritten as a mathematical programming problem by setting

$$d_i^+ - d_i^- = y_i - (\beta_1 x_{i1} + \ldots + \beta_m x_{im}),$$

for i=1,..., n, where d_i^+ and d_i^- represent non-negative deviations above and below the regression plane. We can write (3.1) as a linear programming problem.

$$\text{minimize } \Sigma_{i=1}^{n} (d_i^+ + d_i^-),$$

subject to

$$y_i - (\beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_m x_{im}) + d_i^+ - d_i^- = 0,$$

and

$$d_i^+ \geq 0,\ d_i^- \geq 0, \text{ and } i = 1, \ldots, n.$$

A straight-forward application of the simplex algorithm to this linear program is computationally cumbersome,mainly because of the size of the basis matrix (nxn). The dual problem requires only a working basis of m by m when solved using simple upper bounding techniques. The primal problem can also be solved with a working basis of this size, and Barrodale and Roberts (1966, 1974) report superior results with this approach. The algorithm proposed by Armstrong and Frome (1976) differs from that of Barrodale and Roberts (1974) mainly in that it is a revised simplex code, and only the basis inverse and certain indicators are updated at each iteration. In the present situation, it is possible to further reduce the solution time by selecting an initial basis with at least one point in each of the intervals that are defined by the knots. Further imporvements are possible when the dummy variables are included in the model--see Armstrong and Frome (1977).

4. APPLICATION

The cubic spline smoothing procedure has been applied to monthly Texas construction data. A numerical example is available from the authors as a supplement to this paper. In this example, Texas residential housing authorizations for 1967 to 1976 are analyzed. The cubic spline procedure with 9 knots located at 13-month intervals (i.e., $t_1 = 15$, $t_2 = 28$, etc.) was used to estimate the trend cycle component. The special structure of the x matrix makes it possible to obtain a good starting point by selecting at least one observation from each interval, so that the initial basis matrix is of full rank. For monthly data, we require that there be at least 12 observations per interval, so that the spline fit will not be affected by a seasonal component that may be present in the data. Examination of the residuals indicated that a seasonal component should be included in the model. This combined spline-plus-seasonal model ($k = 9$, $s = 12$, $n = 132$) was then fit to this same data using the LAV estimation procedure. The supplement to this paper contains the original data, the LAV estimates and residuals for both models, and an analysis of the quality of fit of the two models.

5. ACKNOWLEDGMENT

This research was supported in part by the Bureau of Business Research, University of Texas at Austin, and by the Energy Research and Development Administration through the Medical and Health Sciences Division of Oak Ridge Associated Universities, Oak Ridge, Tennessee.

6. REFERENCES

ARMSTRONG, R. D. and FROME, E. L. (1976). A Comparison of Two Algorithms for Absolute Deviation Curve Fitting. Journal of the American Statistical Association 71, 328-330.

ARMSTRONG, R. D. and FROME, E. L. (1977). A Special Purpose Primal Linear Programming Algorithm for Obtaining Least Absolute Value Estimators in a Linear Model with Dummy Variables. (Working Paper)

BARRODALE, I. and ROBERTS, A. (1966). A Note on Numerical Procedures for Approximation by Splines. Computer Journal 9, 318-320.

BARRODALE, I. and ROBERTS, F. D. K. (1974). Solution of an Overdetermined System of Equations in the L_1 Norm. Communications of the ACM 17, 319-320.

BUSE, A. and LIM, L. (1977). Cubic Splines as a Special Case of Restricted Least Squares. Journal of the American Statistical Association 72, 64-68.

CLEVELAND, W. P. and TIAO, G. C. (1976). Decomposition of Seasonal Time Series; A Model for the Census X-11 Program. Journal of the American Statistical Association 71, 581-587.

HARTER, H. L. (1974). The Method of Least Squares and Some Alternatives--Part I. International Statistical Review 42, 147-174.

SCHLOSSMACHER, E. J. (1973). An Iterative Technique for Absolute Deviations Curve Fitting. Journal of the American Statistical Association 68, 857-859.

SHISKIN, J., YOUNG, A. H., and MUSGRAVE, J. C. (1967). The X-11 Variant of Census Method II Seasonal Adjustment Program. Technical Paper No. 15, Bureau of the Census, U.S. Department of Commerce.

WOLD, S. (1974). Spline Functions in Data Analysis. Technometrics 16, 1-11.

BIOGRAPHIES

Edward L. Frome is a biostatistician with the Medical and Health Sciences Division of Oak Ridge Associated Universities, Oak Ridge, Tennessee. Frome received a Ph.D. in Statistics and Biometry from Emory University in 1973.

Ronald Armstrong is Assistant Professor of Operation's Research in the College of Business Administration, the University of Texas, Austin, Texas. Armstrong received his Ph.D. in Operation's Research from the University of Massachusetts in 1974.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

SOLVING THE GENERAL LINEAR MODEL WITH LINEAR PROGRAMMING

Steven R. Borbash, Jr.
West Virginia University, Morgantown, WV 26506

ABSTRACT

Solutions of the general linear model giving estimates of regression coefficients and residuals can be obtained by minimizing the sum of the absolute values of the residuals and by minimizing the residual largest in absolute value. These solutions can be easily obtained by solving associated linear programming problems. Problem formulations are reviewed and solutions are illustrated for a quadratic polynomial model.

Key words: computer method; general linear model; linear programming; regression; residuals; statistical computing.

1. INTRODUCTION

The general linear model is given as:

$$y = X\beta + \varepsilon$$

where y is an (n x 1) vector of observations on a dependent variable, X is an (n x p) matrix (with $n \geq p$) of independent variables, β is a (p x 1) vector of regression coefficients and ε is an (n x 1) residual vector which represents the difference between the observed and true values of the dependent variable. In practice y and X are given, and estimates β and ε are desired. These estimates are obtained by minimizing the length (or norm) of the residual vector.

A general definition of length is given below.

$$\ell_p = ||\varepsilon||_p = \Sigma|\varepsilon_j| \qquad ; p \geq 1$$

When p = 2, the Euclidean norm results, and (β, ε) are estimated by minimizing the sum of squares of the residuals. This definition of length is popular for statistical work because the ℓ_2 estimate of β is identical to the maximum liklihood estimate under the assumption of normality of the residual vector ε. The ℓ_2 solution for β is obtained by solving the well-known normal equations $X'X\beta = X'y$ Then ε is given by $\varepsilon = y - X\beta$. Two other definitions length are also of interest. When p = 1 and in the limit as p approaches infinity, we get

$$\ell_1 = |\varepsilon_j| = |\varepsilon_1| + |\varepsilon_2| + \ldots + |\varepsilon_n|$$

$$\ell_\infty = \max_j |\varepsilon_j| \qquad \text{(Chebyschev norm)}$$

These two special cases are important because they represent limiting values of the parameter p and also because the corresponding estimates (β, ε) are easily obtained by formulating the minimization problems as linear programs and solving them with one of the widely available software systems. The ℓ_1 and ℓ_∞ solutions carry none of the statistical richness of the ℓ_2 solution, but are of interest in their own right. The formulation of the ℓ_1 and ℓ_∞ problems as linear programs has been dealt with extensively in the literature, beginning with the article by Wagner (1959). See also, for example, Rabinowitz (1968) and Barrodale and Young (1966).

Some attention has been given to identifying situations where the ℓ_1 and ℓ_∞ solutions might be more appropriate than the ℓ_2 solution. See, for example, Rice and White (1964) and Barrodale (1968). As p increases, outlying residuals contribute an increasing amount to the length of the residual vector. For example, all residuals contribute equally to the ℓ_1 norm, while in the ℓ_2 norm the larger residuals contribute proportionally more. In the limiting case of the Chebyshev norm ($p = \infty$), the maximum residual is the length of the residual vector. This suggests that for linear models where the ϵ_j can be assumed to be drawn from distributions with more tail area than the normal (such as the Cauchy distribution), the ℓ_1 solution may be more appropriate than least squares because it is relatively insensitive to outliers. On the other hand, for ϵ_j from distributions with little or no tail area (such as the uniform distribution) the ℓ_∞ solution is perhaps more appropriate than least squares. This latter situation could arise when smoothing tabular data, such as thermocouple tables, etc.

2. EXAMPLE PROBLEM

Consider the problem of fitting y as a quadratic function of x with the following seven (x, y) data pairs: (-1.0, 1.0), (3.0, 3.0), (4.5, 14.0), (5.0, 8.0), (-3.5, 15.0), (1.0, 1.0), (-1.5, 8.0). The scalar model equation is

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \epsilon_i, \quad i = 1, 2, \ldots, 7$$

and the vector-matrix formulation is

$$y = X\beta + \epsilon$$

$$\begin{bmatrix} 1 \\ 3 \\ 14 \\ 8 \\ 15 \\ 1 \\ 8 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 \\ 1 & 3 & 9 \\ 1 & 4.5 & 20.25 \\ 1 & 5.0 & 25.0 \\ 1 & -3.5 & 12.25 \\ 1 & 1 & 1 \\ 1 & -1.5 & 2.25 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \\ \epsilon_7 \end{bmatrix}$$

Three solutions to this problem were obtained by minimizing the ℓ_1, ℓ_2 and ℓ_∞ norms of the residual vector. Plots, regression coefficients and residuals for the three solutions are shown on Figure 1. Note that the ℓ_1 and ℓ_2 parabolas are similar, but the ℓ_∞ fit attempts to maintain equal residuals at all data points, and the resulting parabola is somewhat different. The ℓ_1 parabola passes through 3 of the data points, giving zero residuals for these points. In general, at least $q \leq p$ of the residuals in the ℓ_1 solution will be zero where q is the rank of the matrix X. (See Barrodale and Roberts (1973)). Similarly, the ℓ_∞ solution will have $q + 1$ residuals equal at the maximum value. For the example problem note that four residuals are equal in absolute value to 3.88.

3. LINEAR PROGRAMMING

3.1. Standard form for linear programs

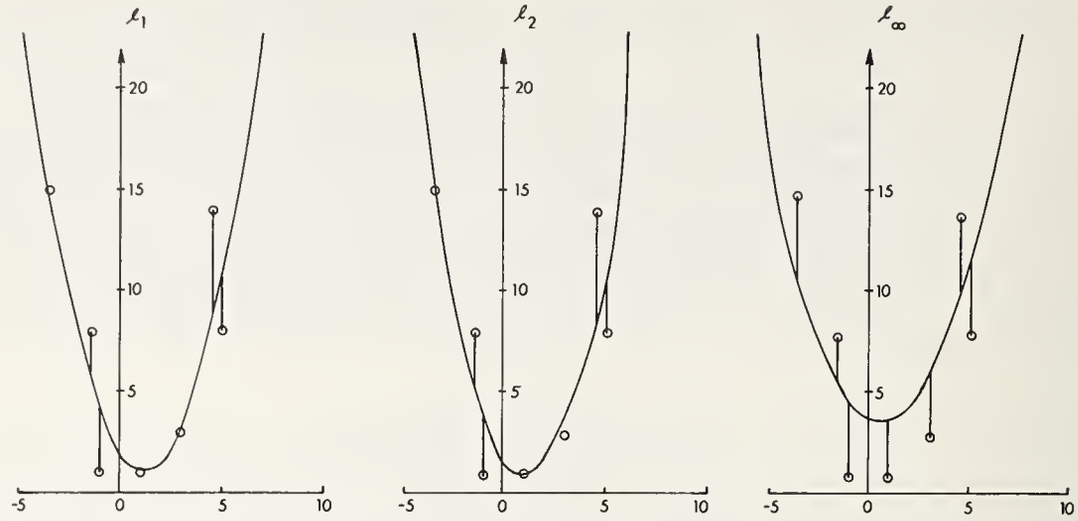
The linear programs for the ℓ_1 and ℓ_∞ problems will be stated in the standard form shown below, following Rabinowitz (1968).

$$\begin{aligned} &\text{Minimize } z = c^t x \\ &\text{Subject to } Ax = b \\ &\qquad\qquad x \geq 0 \end{aligned}$$

This form has m equalities in the constraint set and n non-negative solutions variables. Any linear program can be reduced to this form. See Rabinowitz (1968) or Wagner (1959)

FIGURE 1

FITTED CURVES AND RESIDUALS FOR THREE NORMS



REGRESSION COEFFICIENTS

	ℓ_1	ℓ_2	ℓ_∞
$\hat{\beta}_0$	1.90	1.779	3.90
$\hat{\beta}_1$	-1.53	-1.410	- .556
$\hat{\beta}_2$	.632	.680	.431

RESIDUALS AND PREDICTED VALUES

		ℓ_1		ℓ_2		ℓ_∞	
x_i	y_i	$\hat{y}_i$	$\hat{\varepsilon}_i$	$\hat{y}_i$	$\hat{\varepsilon}_i$	$\hat{y}_i$	$\hat{\varepsilon}_i$
-3.5	15	15.0	0	15.05	-.0450	11.12	3.88
-1.5	8	5.62	2.38	5.42	2.58	5.70	2.30
-1.0	1	4.06	-3.06	3.87	-2.87	4.88	-3.88
1.0	1	1.0	0	1.050	-.0496	3.77	-2.77
3.0	3	3.0	0	3.67	-.671	6.106	-3.11
4.5	14	7.82	6.18	9.21	4.79	10.12	3.88
5.0	8	10.06	-2.06	11.73	-3.73	11.88	-3.88
		$\|\|\hat{\varepsilon}\|\|_1 = \sum\|\varepsilon_i\|$ = 13.68		$\|\|\hat{\varepsilon}\|\|_2 = (\sum\hat{\varepsilon}_i^2)^{1/2}$ = 7.23		$\|\|\hat{\varepsilon}\|\|_\infty = \max_i\|\hat{\varepsilon}_i\|$ = 3.88	

for details. This choice for the standard form has the advantage that it is universally acceptable to software packages. The (m x n) matrix A (where $m \leq n$) is often called the structural or constraint matrix. The (m x 1) vector b is known as the right hand side vector, and the (m x 1) vector c is the cost vector. The (n x 1) optimal solution vector x has at most $m \leq n$ non-zero elements. See Hadley (1962) for further information on details of linear programming.

4. THE ℓ_1 NORM

4.1. The ℓ_1 norm-formulation I.

The appropriate linear programming problem in standard form is given below. A derivation of this and all other formulations presented below can be found in Wagner (1959), Rabinowitz (1968) or Borbash(1977).

$$\text{Minimize } z = \sum_1^n \varepsilon_j^+ + \sum_1^n \varepsilon_j^-$$

$$\text{Subject to } (X|-X|I_n|-I_n) \begin{bmatrix} \beta^+ \\ \beta^- \\ \varepsilon^+ \\ \varepsilon^- \end{bmatrix} = y$$

$$\beta^+, \beta^-, \varepsilon^+, \varepsilon^- \geq 0$$

Here I_n denotes the n-square identity matrix. The regression coefficients are recovered as $\beta_j = \beta^+_j - \beta^-_j$ and the residuals as $\varepsilon_j = \varepsilon^+_j - \varepsilon^-_j$. The constraint matrix above has

2·(n + p) columns and n rows. A slightly more compact form will be introduced below but it is instructive to look at the output for this form first because it is easily interpreted.

The sample problem was solved using the MPS/360 linear programming software system. (MPS/360 is an IBM-developed software system to solve linear programming problems. See IBM (1967)). Figure 2 summarizes the input data and the MPS/360 ℓ_1 solution together in tabular form for the sample problem. The body of the table is the (7 x 20) constraint matrix. The upper stub shows the column number j, the optimal value of the solution variable associated with the column (blanks are used for zero), the name of the column, and the associated cost coefficients. The right hand stub of the table shows the right hand side vector y, the row number i and the dual variable ω_i associated with each y_i. The lower stub shows the "reduced costs" r_j for each column in the structural matrix, (with blanks used for zero). Beneath the table are the calculations necessary to recover the regression coefficients and residuals. There are seven variables in the optimal basis; (β_0^+ , β_2^+ , β_1^- , ε_3^+ , ε_7^+ , ε_1^-). All other variables have optimal values of zero. The optimal value of the objective function is $z^* = \ell_1 = 13.68$.

FIGURE 2

EXAMPLE PROBLEM ℓ_1 NORM

j →	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	RIGHT HAND SIDE	i	DUAL VAR.	NAME
VALUE	1.897		.632		1.530				6.18				2.38	3.06			2.06							
NAME	β_0^+	β_1^+	β_2^+	β_0^-	β_1^-	β_2^-	ε_1^+	ε_2^+	ε_3^+	ε_4^+	ε_5^+	ε_6^+	ε_7^+	ε_1^-	ε_2^-	ε_3^-	ε_4^-	ε_5^-	ε_6^-	ε_7^-				
c_j →	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1				
1	-1.0	1.0	-1	1.0	- 1.0	1								-1							1.0	1	-1.000	y1
1	3.0	9.0	-1	-3.0	- 9.0		1								-1						3.0	2	.462	y2
1	4.5	20.25	-1	-4.5	-20.25				1							-1					14.0	3	1.000	y3
1	5.0	25.0	-1	-5.0	-25.0					1							-1				8.0	4	-1.000	y4
1	-3.5	12.25	-1	3.5	-12.25						1							-1			15.0	5	- .0171	y5
1	1.0	1.0	-1	-1.0	- 1.0							1							-1		1.0	6	- .444	y6
1	-1.5	2.25	-1	1.5	- 2.25								1							-1	8.0	7	1.000	y7
r_j →	0		0		0		2.00	.538		2.00	1.017	1.444			1.462	2.00		.983	.556	2.00				

Linear Programming Formulation

$$\text{minimize } Z = \sum_1^n \varepsilon_j^+ + \sum_1^n \varepsilon_j^-$$

$$\text{subject to: } (X|-X|I_n|-1_m) \begin{bmatrix} \underline{\beta}^+ \\ \underline{\beta}^- \\ \underline{\varepsilon}^+ \\ \underline{\varepsilon}^- \end{bmatrix} = \underline{y}$$

$$\underline{\beta}^+, \underline{\beta}^-, \underline{\varepsilon}^+, \underline{\varepsilon}^- \geq \underline{0}$$

Regression Coefficients

$\hat{\beta}_0 = \beta_0^+ - \beta_0^- = 1.897 - 0 = 1.897$

$\hat{\beta}_1 = \beta_1^+ - \beta_1^- = 0 - 1.530 = -1.530$

$\hat{\beta}_2 = \beta_2^+ - \beta_2^- = .632 - 0 = .632$

Objective Function

$z^* = ||\hat{\underline{\varepsilon}}|| = 3.06 + 6.18 + 2.06 + 2.38 = 13.68$

Residuals

$\hat{\varepsilon}_i = \varepsilon_i^+ - \varepsilon_i^-$

$\varepsilon_1 = 0 - 3.06 = -3.06$
$\varepsilon_2 = 0 - 0 = 0$
$\varepsilon_3 = 6.18 - 0 = 6.18$
$\varepsilon_4 = 0 - 2.06 = -2.06$
$\varepsilon_5 = 0 - 0 = 0$
$\varepsilon_6 = 0 - 0 = 0$
$\varepsilon_7 = 2.38 - 0 = 2.38$

+ Dual variables are the negative of the MPS/360 "DUAL ACTIVITY"

4.2. Dual variable interpretation

Every linear program has a dual variable associated with each element b_i of the right hand side vector. When the solution is optimal, these dual variables ω_i can be interpreted as sensitivity coefficients, or marginal rates of increase or decrease of the objective function with respect to each individual right hand side element.

$$\omega_i = \frac{\partial z}{\partial b_i}$$

A positive dual variable means that if the associated right hand side element b_i is increased by one unit, the objective function will increase by ω_i. In formulation I the right hand side elements are the observed dependent variables y_i, and the objective function is the ℓ_1 norm of the residual vector. Thus, the dual variables here reflect the sensitivity of ℓ_1 to the y_i. In the example problem, the solution is most sensitive to $y_3 = 14$ with a

dual variable $\omega_3 = 1.00$ and least sensitive to $y_5 = 15$ with a dual variable $\omega_5 = -.0171$. An increase in y_5 will cause ℓ_1 to increase, while an increase in y_5 will cause a decrease. Several dual variables other than ω_5 have absolute values of unity, indicating maximal sensitivites to variations in the corresponding y_i.

4.3. The ℓ_1 norm-formulation II

A slightly more compact form of this problem due to Barrodale and Young (1966) is given below.

$$\text{Minimize: } z = \sum_1^n \varepsilon_j^+ + \sum_1^n \varepsilon_j^-$$

$$\text{Subject to: } (X|\delta|I_n|-I_n) \begin{bmatrix} \alpha \\ d \\ \varepsilon^+ \\ \varepsilon^- \end{bmatrix} = y$$

$$\alpha, d, \varepsilon^+, \varepsilon^- \geq 0$$

This problem has $(p - 1)$ fewer columns in the constraint matrix than formulation I. It takes slightly less time to solve and requires less input data. The column vector δ is obtained by summing all the columns in X and then multiplying by (-1). The scalar solution variable d is associated with this column. Regression coefficients are given by $\beta_j = \alpha_j - d$ and residuals by $\varepsilon_j = \varepsilon^+_j - \varepsilon^-_j$.

4.4. The ℓ_1 norm-other formulations.

The ℓ_1 problem has a dual formulation given by Wagner (1959) with a (p x n) constraint matrix and 2n bounded variables. Barrodale and Roberts (1973) state that the dual formulation is not as efficient as formulation II above. A special algorithm for the ℓ_1 problem has been developed by Barrodale and Young (1966) and improved by Barrodale and Roberts (1973). Barrodale and Roberts (1974) claim this algorithm is more efficient than any other for the ℓ_1 problem and present a FORTRAN program for its implementation.

5. THE ℓ_∞ NORM

5.1. The ℓ_∞ norm-formulation I

This formulation is given below.

$$\text{Minimize: } z = u$$

$$\text{Subject to: } \begin{bmatrix} X|\delta|J|-I_n|0 \\ -X|\delta|-J|0|I_n \end{bmatrix} \begin{bmatrix} \alpha \\ d \\ u \\ S \\ s \end{bmatrix} = \begin{bmatrix} y \\ y \end{bmatrix}$$

$$\alpha, d, u, S, s \; 0$$

Here J is an (n x 1) vector of 1's, vectors S and s are surplus and slack variables, u is the ℓ_∞ norm, and d is the convenience variable used in ℓ_1 formulation II. The regression coefficients are given by $\beta_j = \alpha_j - d$ and the residuals by $\varepsilon_j = u - S_j$. This formulation has $2n + (p + 2)$ columns and 2n rows in the constraint matrix and is inefficient to solve relative to formulation II which follows.

5.2. The ℓ_∞ norm-formulation II

This formulation is the dual of formulation I.

$$\text{Maximize } g = y\,\omega_1 - y\,\omega_2$$

$$\text{Subject to: } \left[\begin{array}{c|c|c} X' & -X' & \\ \delta' & -\delta' & I_{p+2} \\ J' & J' & \end{array}\right] \left[\begin{array}{c} \omega_1 \\ \omega_2 \\ s \end{array}\right] = \left[\begin{array}{c} 0_{p+1} \\ 1 \end{array}\right]$$

$$\omega_1, \omega_2, s, \geq 0$$

This maximization problem has (p + 2) rows and 2n + (p + 2) columns. The dual variables associated with the rows of this problem correspond to the column variables α_j, d, u of the previous problem, and vice versa. The optimal value of the objective function is also equal to u at the maximum. The residuals are recovered from the reduced costs $r_j = (c_j - z_j)$ associated with all the columns which are not in the optimal basis. The r_j are given as part of the MPS/360 output. For the residuals we have $\epsilon_j = u + r_j$. The dual variables associated with the right hand side elements y_j of the original formulation appear now as the optimal values of the column variables in the new formulation. Figure 3 was constructed from the MPS/360 output to show the input data and solution variables in the same format as figure 2 for the example problem. Thus α_j, d and u are given by the optimal dual variables. The regression coefficients are computed as before, with $\beta_j = \alpha_j - d$.

FIGURE 3

EXAMPLE PROBLEM ℓ_∞ NORM

j →	1	2	* 3	4	* 5	6	7	* 8	9	10	* 11	12	13	14	15	* 16	17	18	19			
VALUE			.443		.0573			.1181			.382					0.00				RIGHT		DUAL+	
NAME	ω_1	ω_2	ω_3	ω_4	ω_5	ω_6	ω_7	ω_8	ω_9	ω_{10}	ω_{11}	ω_{12}	ω_{13}	ω_{14}	S_1	S_2	S_3	S_4	S_5	HAND		
Cj →	1	3	14	8	15	1	8	-1	-3	-14	-8	-15	-1	-8						SIDE	1	VAR.	NAME
	1.0	1.0	1.0	1.0	1.0	1.0	1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	1					0	1	4.45	α_0
	-1.0	3.0	4.5	5.0	-3.5	1.0	-1.5	1.0	-3.0	-4.5	-5.0	3.5	-1.0	1.5		1				0	2	0.00	α_1
	1.0	9.0	20.25	25.0	12.25	1.0	2.25	1.0	-9.0	-20.25	-25.0	-12.25	-1.0	-2.25			1			0	3	.986	α_2
	-1.0	-13.0	-25.75	-31.0	-9.75	-3.0	-1.75	1.0	13.0	25.75	31.0	9.75	3.0	1.75				1		0	4	.556	d
	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0					1	1	5	3.88	u
r_j →	-7.77	-6.99		-7.77		-6.66	-1.583		-.778	-7.77		-7.77	-1.111	-6.18	-4.45		-.986	-.556	-.388				

Linear Programming Formulation (Dual Problem)

maximize $g = \underline{\omega}_1' \underline{y} - \underline{\omega}_2' \underline{y}$

subject to: $\left[\begin{array}{c|c|c} X' & -X' & \\ \underline{\delta}' & -\underline{\delta}' & I_{p+2} \\ \underline{J}' & \underline{J}' & \end{array}\right] \left[\begin{array}{c} \underline{\omega}_1 \\ \underline{\omega}_2 \\ \underline{S} \end{array}\right] = \left[\begin{array}{c} 0_{p+1} \\ 1 \end{array}\right]$

$\underline{\omega}_1, \underline{\omega}_2, \underline{S} \geq \underline{0}$

Regression Coefficients

$\hat{\beta}_0 = \alpha_0 - d = 4.45 - .556 = 3.89$
$\hat{\beta}_1 = \alpha_1 - d = 0 - .556 = -.556$
$\hat{\beta}_2 = \alpha_2 - d = .986 - .556 = .430$

Residuals

$\hat{\epsilon}_1 = u + r_1 = 3.88 - 7.77 = -3.89$
$\hat{\epsilon}_2 = u + r_2 = 3.88 - 6.99 = -3.11$
$\hat{\epsilon}_3 = u + r_3 = 3.88 + 0 = 3.88$
$\hat{\epsilon}_4 = u + r_4 = 3.88 - 7.77 = -3.89$
$\hat{\epsilon}_5 = u + r_5 = 3.88 - 0 = 3.88$
$\hat{\epsilon}_6 = u + r_6 = 3.88 - 6.66 = -2.78$
$\hat{\epsilon}_7 = u + r_7 = 3.88 - 1.583 = 2.30$

$\delta^* = ||\hat{\epsilon}||_\infty = \max_i \{|\epsilon_i|\} = u = 3.88$

NOTE: Primal variables here (the ω_j) are the dual variables for the primal problem and vice versa.

$\frac{\partial z^*}{\partial y_j} = +\omega_j;\ j = 1,\ldots,n$

$\frac{\partial z^*}{\partial y_3} = \omega_3 = .443$

$\frac{\partial z^*}{\partial y_5} = \omega_5 = .0573$

$\frac{\partial z^*}{\partial y_{j-n}} = -\omega_j;\ j = n+1,\ldots,2n$

$\frac{\partial z^*}{\partial y_1} = -\omega_8 = -.1181$

$\frac{\partial z^*}{\partial y_4} = -\omega_{11} = -.382$

+ Dual variables are the negative of the MPS/360 "DUAL ACTIVITY"

Due to space limitations, MPS/360 programs and output were not included in this article. These items are included in a more extensive report available from the author. See Borbash (1977).

6. REFERENCES

BARRODALE, I. (1968). L_1 approximation and the analysis of data. Appl. Statist., 17, 51-56.

BARRODALE, I. and ROBERTS, F. D. K. (1973). An improved algorithm for discrete ℓ_1 linear approximation. SIAM J. Numer. Anal., 10, 839-848.

BARRODALE, I. and ROBERTS, F. D. K. (1974). Algorithm 478: solution of an overdetermined system of equations in the ℓ_1 norm. Commun. ACM, 17, 319-320.

BARRODALE, I. and YOUNG, A. (1966). Algorithms for best L_1 and L_∞ linear approximations on a discrete set. Numerische Mathematik, 8, 295-306.

BORBASH, S. (1977). Linear programming solutions to the general linear model. Internal working paper, Industrial Engr. Dept., W. Va. Univ., Morgantown, WV.

HADLEY, G. (1962). Linear Programming. Addison-Wesley.

IBM CORP. (1967). Mathematical Programming System/360: Control Language User's Manual, IBM H20-0292, IBM Corp., White Plains, NY.

RABINOWITZ, P. (1968). Applications of linear programming to numerical analysis. SIAM Review, 10, 121-159.

RICE, J. R. and WHITE, J. S. (1964). Norms for smoothing estimation. SIAM Review, 6, 243-256.

WAGNER, H. M. (1959). Linear programming techniques for regression analysis. J. Amer. Statist. Assoc., 54, 206-212.

BIOGRAPHY

Steven R. Borbash teaches and directs projects in the areas of data management, software utilization and engineering applications of statistics. He received a Ph.D. in Systems Management Engineering and Operations Research from the University of Pittsburgh in 1970.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

ANALYSIS OF VARIANCE INCORPORATING TREND ANALYSIS

Michael H. Kutner
Emory University; Atlanta, Georgia 30322

ABSTRACT

When a factor under investigation is quantitative, the analysis of the factor effects often includes a study of the nature of the response function (trend analysis). Without loss of generality, balanced and unbalanced data from single-factor experiments are considered.

Key words: Trend analysis; response curves; analysis of variance; regression analysis with repeated x's; polynomial regression.

1. INTRODUCTION

When a factor under investigation is quantitative, the analysis of the factor effects often includes a study of the nature of the response function (trend analysis). Without loss of generality, balanced and unbalanced data from single-factor experiments are considered.

2. REGRESSION MODEL APPROACH

Assume $Y = X\beta + e$ where Y is an Nxl vector of observed random variables, X is a full-rank $Nx(p+1)$ matrix of known fixed numbers, β is a $(p+1)xl$ vector of unknown parameters and e is an Nxl vector of random errors. For hypothesis testing purposes further assume $Y \sim N(X\beta, \sigma^2 I)$. The total sum of squares (SS) can be partitioned as follows:

$$Y'Y = Y'[jj'/N]\,Y + Y'[X(X'X)^{-1}X' - jj'/N]Y$$
$$+ Y'[I - X(X'X)^{-1}X']Y$$
$$= SS\ (Mean) + SS\ Reg\ (Adj\ for\ Mean) + SS(Residual)$$

where $j' = (1, \ldots, 1)$ is a 1xN row vector of ones. If repeated x's are available, i.e., some rows of the X matrix are identical, then the Residual SS can be further partitioned as follows:

$$SS(Residual) = Y'[I - X(X'X)^{-1}X']Y = Y'\,[WDD'W' - X(X'X)^{-1}X']Y$$
$$+ Y'[I - WDD'W']Y = SS\ (Lack\ of\ Fit) + SS\ (Pure\ Error)$$

where

$$W = \begin{bmatrix} j_{n_1} & 0 & \cdots & 0 \\ 0 & j_{n_2} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & j_{n_k} \end{bmatrix}_{Nxk}, \qquad D = \left[\text{Diag} \left(\frac{1}{\sqrt{n_i}}\right)\right], i = 1,\ldots,k,$$

j_{n_i} is an $n_i$xl column vector of ones, and n_i = # of repeat observations for the ith set of x's.

3. ANALYSIS OF VARIANCE MODEL APPROACH

The regression problem formulated above can be viewed as a one-way classification analysis of variance with k groups and n_i observations per group. Assuming evenly spaced levels of x then certain computational simplifications can be made by the use of orthogonal polynomials. The group sum of squares can be partitioned using orthogonal polynomials into linear, quadratic, and higher degree polynomial components in order to study the nature of the response function. Letting $\bar{x}$ be the average of the evenly spaced x levels and d be the spacing distance between consecutive x's then the orthogonal polynomials can be obtained recursively using

$$p_{r+1}(x) = p_r(x)\, p_1(x) - \frac{r^2(k^2-r^2)}{4(4r^2-1)}\, p_{r-1}(x) \quad , r=1, 2, \ldots$$

where $p_o(x)=1$ and $p_1(x) = \frac{x-\bar{x}}{d}$.

3.1 Balanced data

If $n_i = n$ for all i then both the regression model approach and the analysis of variance model approach yield identical results as demonstrated in Example 1.

Example 1. Data

Group 1	Group 2	Group 3
x = 1	x = 2	x = 3
1.3	2.4	3.0
1.7	2.0	3.2

$k = 3$, $n_i = n = 2$

Analysis of Variance Approach

AOV Table

S.V.	df	SS
Groups	2	2.5733
Error	3	0.1800
Total	5	2.7533

AOV Table (incorporating trend analysis)

S.V.	df	SS
Groups	2	2.5733
Linear	1	2.5600
Dev. from linear	1	0.0133
Error	3	0.1800
Total	5	2.7533

	x = 1	x = 2	x = 3	
$p_1(x)$ =	-1	0	1	(Linear)
$p_2(x)$ =	-1	2	-1	(Dev. from linear)

Regression Approach

S.V.	df	SS
Reg	1	2.5600
Residual	4	0.1933
Lack of Fit	1	0.0133
Error	3	0.1800
Total	5	2.7533

3.2 Unbalanced data.

If some n_i are different we have the unbalanced data case and the regression approach and the analysis of variance approach yield different results. This can be seen in Example 2.

Example 2: Data

Group 1	Group 2	Group 3
x = 1	x = 2	x = 3
1.3	2.4	3.0
1.7	2.0	3.2
		2.8
		3.1
		2.9

$n_1 = n_2 = 2$, $n_3 = 5$, $k = 3$

Analysis of Variance Approach

S.V.	df	SS
Groups	2	3.4289
Linear	1	3.2143
Dev. from linear	1	0.0037
Error	6	0.2600

SPSS ONEWAY Output

S.V.	df	SS
Groups	2	3.4289
Linear	1	3.2143
Dev. from linear	1	0.2146*
Within groups	6	0.2600
Total	8	3.6889

*Obtained by subtraction and not correct.

Regression Approach

S.V.	df	SS
Reg	1	3.4252
Residual	7	0.2637
Lack of fit	1	0.0037
Error	6	0.2600
Total	8	3.6889

Note that the regression approach yields SS Reg = 3.4252 and analysis of variance approach yields SS Linear = 3.2143. The two are not identical as in the balanced case. The SS Linear from the analysis of variance approach does not use the weights n_i in the hypotheses while the regression approach does. (See Speed (1976).) It seems reasonable that the meaning of linear, quadratic, and higher order polynomials should not be affected by the sample sizes in each group so that the preferred analysis comes from analysis of variance approach. Here a reasonable interpretation in terms of the population parameters can be made.

4. ACKNOWLEDGEMENT

This work was supported in part by USPHS grant # RR00039.

5. REFERENCES

NIE, N.H., HULL, C. H., JENKINS, J.G., STEINBRENNER, K., and BENT, D. H. (1975). Statistical Package for the Social Sciences. Second Edition. McGraw-Hill Book Company.

SPEED, F. M. (1976). Response curves in the one way classification with unequal numbers of observations per cell. Proc. Statistical Computing Section, American Statistical Association, 270-272.

BIOGRAPHY

Michael H. Kutner received a Ph.D. in statistics in 1971 from Texas A & M University. He is presently Associate Professor of Statistics and Biometry at Emory University School of Medicine.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

COMPUTERIZED ANALYSIS OF QUALITY CONTROL FOR RADIOIMMUNOASSAYS

Peter J. Munson and David Rodbard
NICHD, ERRB, National Institutes of Health, Bethesda, MD 20014

ABSTRACT

We have developed a new computer programs for analysis of Quality Control data arising in the Clinical Chemistry laboratory, applicable to RIAs. Detailed analyses including calculation of within- and between-assay variability, detection of non-random assay behavior, evaluation of indicators of assay instability and "lack of control" are performed automatically. The results allow the laboratory director to make informed decisions concerning the maintenance of assay control.

Key words: clinical chemistry; competitive protein binding assay; data processing; quality control; radioimmunoassay; RIA.

1. INTRODUCTION

Radioimmunoassay (RIA) methods now constitute one of the most popular and important class of methods in the Clinical Chemistry laboratory. RIAs need careful monitoring, or quality control (QC) -- perhaps more than most other procedures. RIAs are notoriously unstable, with problems of "blanks", large inter-assay and inter-laboratory variation, and fluctuating specificity.

In an attempt to satisfy this need, we have developed a computer program for quality control with the following features:

1. User oriented: Most laboratories do not employ a full time "on-line" statistician who can interpret the results of QC data. Therefore, a computer program is needed to perform routine calculations and print out readily interpreted results.

2. Ease of data entry: A minimum of data preparation is required by the user. Corrections for missing data, or unequal members of replicates are handled exactly.

4. Availability: The program is written in generally available PL/I for IBM/370. A prototype is available in BASIC.

5. Combination of results from several QC samples: Provides a compact summary of results, and improves reliability.

6. Criteria for rejecting assays: Some assays may be rejected on the basis of their QC results. In order to make such a decision, the laboratory director needs objective criteria which the computer program can provide.

2. METHODS

Large inter-assay variability may often go unnoticed and unappreciated by users of an RIA with possibly serious consequences, unless a competent quality control system is being

maintained. To measure such variability, samples from a QC pool, a relatively large amount of frozen serum, are assayed repetitively within each assay and in different assay runs. We assume that each aliquot of the pool has the same "true" value and does not suffer degredation over time. Thus, all observed variability is caused by experimental fluctuations. QC pools are maintained at several concentrations, usually low, normal, and high ranges, since the observed standard error varies with concentration and position on the standard curve.

Control chart techniques and analysis of variance (ANOVA) with components of variance estimation are the basic methods we use to analyze the QC data (DUNCAN, 1974). The statistical model assumes "random effects" due to assays and "fixed effects" for each QC pool and/or laboratory. Therefore, ANOVA allows us to estimate within- and between-assay components of variability for each QC pool. However, several problems arise which complicate the analysis. In a two-way ANOVA, unequal numbers of observations usually are balanced by deleting data or by duplicating the remaining observations. Unfortunately, for many RIA applications, dropping one of the measurements may have a disasterous effect on the estimator of residual variance. Therefore, the program makes use of ANOVA which explicitly takes account of unequal cell size. Since the variance is often non-uniform for different dose levels, and in for different laboratories, some transformation of the data is often needed. The program allows optional Ridit (percentile), square-root, logarithmic, or Studentizing transformations for this purpose.

The ratio of between assay variance to within assay variance is used as an index of assay stability and is compared with percentiles of the F distribution. The median value for this index was 3.5 with a range of 1.0 - 23.3 with data taken from a commercial RIA lab over a period of one month, using 16 different hormone assays. The ratio of current (or "local") to cumulative between assay variance provides a measure of assay control; a significantly elevated ratio indicates that the latest assay is probably out of control. The ratio of current to cumulative within assay variance gives an indication of the relative precision of the most recent assay. Significance for this test may indicate presence of outliers in the most recent results. Assay control can also be tested graphically by comparing the current results with the 95 or 99% control limits on the N most recent assays. The computer program makes these tests automatically and prints out warnings where appropriate.

A more powerful indicator of an assay "out of control" can be obtained by combining the results of several QC samples. All three samples falling outside their respective control limits strongly indicates that this assay should be rejected. If all three samples are above the previous average by an arbitrary percentage, one might consider applying a correction factor to the unknowns in the assay. This approach may be valid in cases when the errors for all QC pools are highly correlated. We may calculate the intra-class correlation coefficient for several QC pools, by "Studentizing" the assay means for each pool and re-applying a one-way ANOVA. The between-assay component may then be interpreted as the fraction of total variance arising between assays. This estimate is identical to the intra-class correlation coefficient (Snedecor, et al 1967).

Another approach to combining information from several QC pools is to plot today's result versus the mean of previous results on a log-log scale. Ideally, all the points should lie along the line of identity. Significant deviations from this line may be seen graphically and tested with regression analysis. Superimposition of the Studentized QC charts for all the samples, and plotting of the log-log graph is automatically performed by the program.

Trends, oscillations and other types of non-randomness can be detected in the QC charts by the Mean Square Successive Differences test. Experience has shown this test to be sensitive to types of non-randomness which may signal an imminent assay "crash". In one case a steroid assay showed oscillations of increasing magnitude before it crashed, the deterioration of performance was later determined to be a result of a bad solvent

extraction step. In other assays, significant drift of the QC samples has been due to reagent degradation. The program has been extensively field tested with data from NIH, commercial RIA laboratories and World Health Organization cooperative studies.

Details of the calculations used in this package are described in McDonagh (1977).

3. PROGRAM AVAILABILITY

The current version of our program available for distribution can be obtained from the National Technical Information Service, Springfield, Virginia 22151. Printed listings of the logit-log RIA program (for routine dose-interpolation), its documentation, sample input, sample output, and operating instructions, a guide to the interpretation of results, can be obtained as Report No. PB 246223, "RADIOIMMUNOASSAY DATA PROCESSING, third edition, Vol. 1". Similar materials for the Four Parameter Logistic RIA Program and the Quality Control program are designated Report No. PB246222. The contents of both booklets can be obtained on a magnetic tape for direct loading into a computer, by requesting Report NO. PB246222. The dose interpolation programs are in FORTRAN IV, level G, the Quality Control program is in PL/I. We have also developed programs for RIA dose interpolation and QC in extended BASIC.

Logit-log graph paper can be obtained from TEAM, box 25, Tamworth, New Hampshire, 03886, from Codex Book Co., Norwood, Mass., 02062 or from Heffer's Stationers, 26 King Street, Cambridge, England.

4. ACKNOWLEDGMENTS

Thanks are due to Bernard McDonagh for his helpful discussions and to S. Knisley for programming the original version.

5. REFERENCES

BENNETT, C.A. and FRANKLIN, N.L., (1954). Statistical Analysis in Chemistry and the Chemical Industry, John Wiley & Sons, Inc., New York.

RODBARD, D., (1974). Statistical quality control and routine data processing for radioimmunoassays (RIA) and immunoradiometric assays (IRMA). Clinical Chemistry 20, 1255-1270.

DUNCAN, A.J. (1974). Quality Control and Industrial Statistics, Richard D. Irwin, Inc., Homewood, Illinois.

McDONAGH, BERNARD F., MUNSON, P.J., AND RODBARD, D. (1977). A computerized approach to statistical quality control for radioimmunoassays in the clinical chemistry laboratory. To appear in Computer Programs in Biomedicine.

RODBARD, D., HUTT, D.M., (1972). Statistical analysis of radioimmunoassays and immunoradiometric (labeled antibody) assays: a generalized weighted, iterative, least-squares method for logistic curve fitting, in Symposium on RIA and Related Procedures in Medicine, Int. Atomic Energy Agency, Vienna, Austria, 165-192, (available from UNIPUB, box 433, Murray Hill Station, N.Y., N.Y. 10016).

SNEDECOR, G.W. COCHRAN, W.G. (1967). Statistical Methods, The Iowa State University Press, Ames Iowa.

YOUDEN, W.J. and STEINER, E.H. (1975). Statistical Manual of the Association of Official Analytical Chemists. Association of Official Analytical Chemists, Washington, D.C.

BIOGRAPHIES

Peter Munson received a MA. degree in mathematics in 1971 from the University of Wisconsin. He is now a statistician in the NICHD where he worked on the development of computer programs and statistical techniques for the analysis of data arising in Radioimmunoassays.

David Rodbard received an M.D. from Western Reserve University in 1964. He has been in the Endocrinology Branch of the NCI and the Endocrinology and Reproduction Research Branch of NICHD, since 1966, where he is now a Senior Investigator.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

AN INTERACTIVE GRAPHIC PROGRAM FOR SIMULATING THE DISTRIBUTION OF TRANSFORMATIONS OF SEVERAL INDEPENDENT RANDOM VARIABLES

C.F. Chung, S.R. Divi and A.G. Fabbri
Geological Survey of Canada, Ottawa, Canada, K1A 0E8

ABSTRACT

An interactive graphic computer program for simulating and displaying the distributions of transformations and extremes of several independent continuous random variables is presented. The parameters and distributions of the initial random variables can be interactively altered. Simulated distributions of transformations and extremes using the Monte Carlo technique are displayed with estimated means and standard deviations.

Key words: Interactive graphic program; simulation; distributions; transformations; extremes; Monte Carlo technique.

1. INTRODUCTION

During an early stage of statistical modelling, where modelling includes manipulations of several independent random variables, an approximate graphic form with mean and variance of the distribution of transformation usually provides sufficient information to characterize the distribution. The exact distributions of the transformations can be theoretically obtained as functions of the initial distributions. However, the analytic results are very complicated even in simple cases.

This program, SIMGRA1, was developed for simulating the density distribution and cumulative distribution of transformations of several independent continuous random variables using the Monte Carlo method and for displaying the results in graphic form. The program can also be utilized to simulate the distributions of extremes. Additionally, the program allows accumulation of several simulated distributions of a transformation based on different initial distributions and parameters. This feature can be used for studying the robustness of transformations.

At present, the following seven types of continuous distributions can be entered as distribution functions of initial variables; normal, lognormal, Cauchy, gamma, beta, uniform and triangular. The maximum number of initial variables which can be considered is eight.

The operational procedure of SIMGRA1 will be briefly described, followed by some computational details. A few simple examples of applications will be discussed.

2. OPERATIONAL PROCEDURE

The flow chart in figure 1 describes the operational procedure of the program. This can be outlined in steps as follows: (1) initialization of the terminal, (2) definition of distributions of the initial random variables and entry of their parameters, (3) definition of a transformation, in functional form, of the initial variables, (4) computation of simulation, plotting of the results and modification of the initial variables, (4.a) to (4.d), and (5) termination of the program.

As can be seen in figure 4, the screen consists of an input area on the left, a menu of options area in the centre, a simulation area on the upper right and a storage area on the lower right corner. The input area is used for displaying the distributions of all initial variables separately. The simulation area displays simulated distributions of transfomation with estimated means and standard deviations. Simulated distributions can be accumulated for comparisons with each other and then plotted in the storage area.

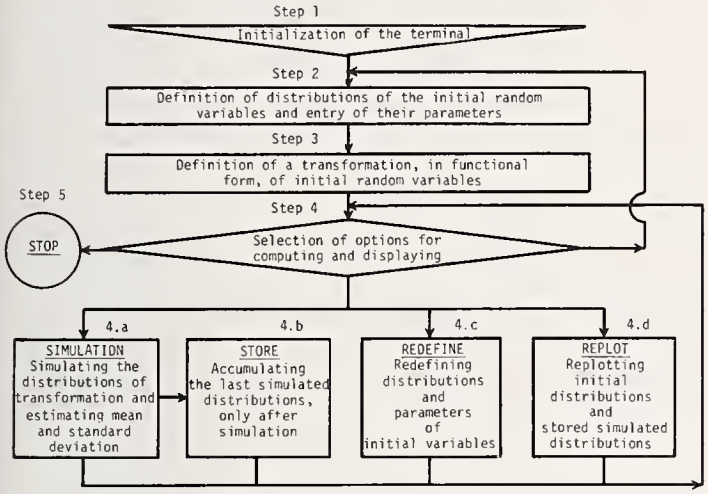


Figure 1. Flowchart of operational procedure for SIMGRA1.

The upper seven options of the menu of options are used for entering initial random variables. The bottom option FUNCTION is for specifying the function statement for transformation. The six options in between are used for computing simulations, plotting of results and manipulating the initial variables step(4) in figure 1. When the SIMULATION option is chosen, a simulated density distribution and cumulative distribution of the transformation defined by the user are separately plotted in the simulation area on the upper right of the screen as shown in figure 4. The STORE option can be selected for accumulating the last simulated distributions, only after simulation. The REDEFINE option is used for modification of the definitions of the distributions and the parameters of the initial variables. The REPLOT option is for clearing the simulation area before obtaining other simulated distributions and for plotting all accumulated simulated distributions.

3. SIMULATION ALGORITHM

Let $X_1, X_2, \cdots, X_k$ be independent random variables with the density distributions $f_{x_1}, f_{x_2}, \cdots, f_{x_k}$, respectively. Let $X = (X_1, X_2, \cdots, X_k)$, $f_x = f_{x_1} f_{x_2} \cdots f_{x_k}$, then X is the k-dimensional random vector with the density distribution f_x. Let $h : R^k \to R$ be a measurable transformation defined on R^k into R, so that $Y = h(X)$ is a random variable with the density distribution f_y. The Monte Carlo method is used for simulating f_y.

The pseudo random numbers $x_1, x_2, \cdots, x_k$ are first generated according to the given distributions $f_{x_1}, f_{x_2}, \cdots, f_{x_k}$ of the initial variables $X_1, X_2, \cdots, X_k$. Then $y = h(x)$ is computed where $x = (x_1, x_2, \cdots, x_k)$ and h is the transformation defined by the user. A transformation can also be an extreme or a combination of extremes and some other type of functional form as can be seen in example 2 in next section. These procedures are repeated at least 50000 times, or up to 500000 times depending upon the option selected by the user.

From all simulated y's, the first 4000 y's, $y_1, y_2, \cdots, y_{4000}$ are taken to obtain a range (y_{min}, y_{max}) for displaying the simulated distributions. The y_{min} and y_{max} are chosen such that $n_{min}/4000 = n_{max}/4000 = 0.005$, where n_{min} is the number of all

y_i's ($i \leq 4000$) with $y_i \leq y$ min , and n_{max} is the number of all y_i's ($i \leq 4000$) with $y_i \geq y_{max}$. Then the fixed range (y_{min} , y_{max}) is divided into 200 equal intervals with size $(y_{max} - y_{min})/198$. Finally, the numbers of the all simulated y's within each class are recorded and plotted as a simulated density distribution. A similar procedure is followed for the simulated cumulative distribution. Moment estimates of expected value $E(Y)$ and standard deviation $\sqrt{E(Y^2) - E^2(Y)}$, if it exists, of the transformation are also computed.

4. PRACTICAL EXAMPLES

Example 1. Let X_1, X_2 be independent random variables distributed as uniform (0.1, 1.1). Let us consider a transformation $Y = X_1 + X_2$. Then the density distribution of Y is distributed as triangular (0.2, 1.2, 2.2). This is shown in figure 2(a). A simulated density distribution by SIMGRA1 is shown in figure 2(b).

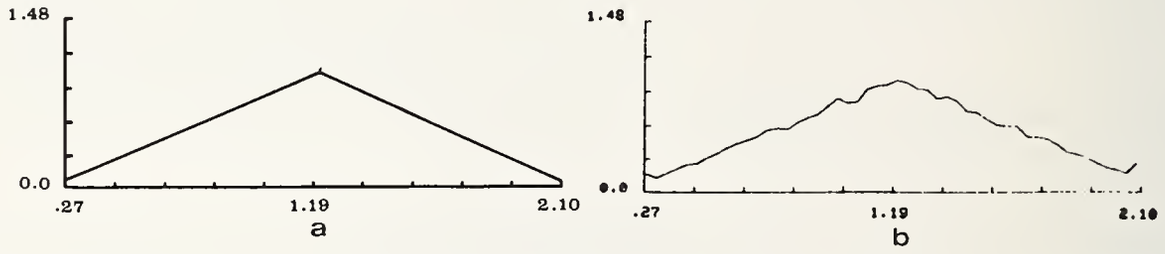


Figure 2. Density distribution of the sum of two identical independent uniform variables. (a) Density distribution of the sum, triangular (0.2, 1.2, 2.2). (b) Simulated density distribution. The scales along the axes in (a) were made identical to those in (b) for comparison.

Example 2. Let X_1, X_2, X_3, X_4 be independent random variables where X_1, X_2 are identically distributed as uniform (0, 1), and X_3, X_4 are also identically distributed as uniform(1, 2). Let $Y = Y_1 - Y_2$ where Y_1 is the minimum of X_3 and X_4 , and Y_2 is the maximum of X_1 and X_2 . Then, it is known, Matern(1960), that Y has the density distribution f_y where

$$f_y(y) = \begin{cases} 4y(1 - y) + \frac{2}{3} y^3 & 0 \leq y \leq 1 \\ \frac{2}{3}(2 - y)^3 & 1 < y \leq 2 \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

The distribution f_y is shown in figure 3(a). A simulated distribution of the transformation is also displayed in figure 3(b).

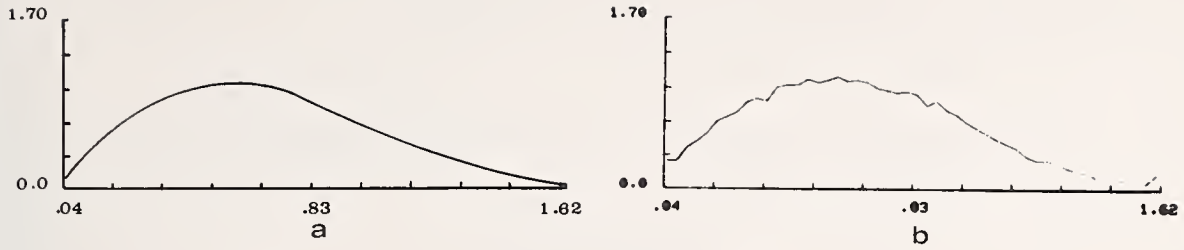


Figure 3. Density distribution of the transformation $Y = Y_1 - Y_2$ where $Y_1 = \text{minimum}(X_3, X_4)$, $Y_2 = \text{maximum}(X_1, X_2)$, and $X_3, X_4 \sim \text{uniform}(1, 2)$, $X_1, X_2 \sim \text{uniform}(0, 1)$. (a) Density distribution of Y by equation (1). (b) Simulated density distribution by SIMGRA1.

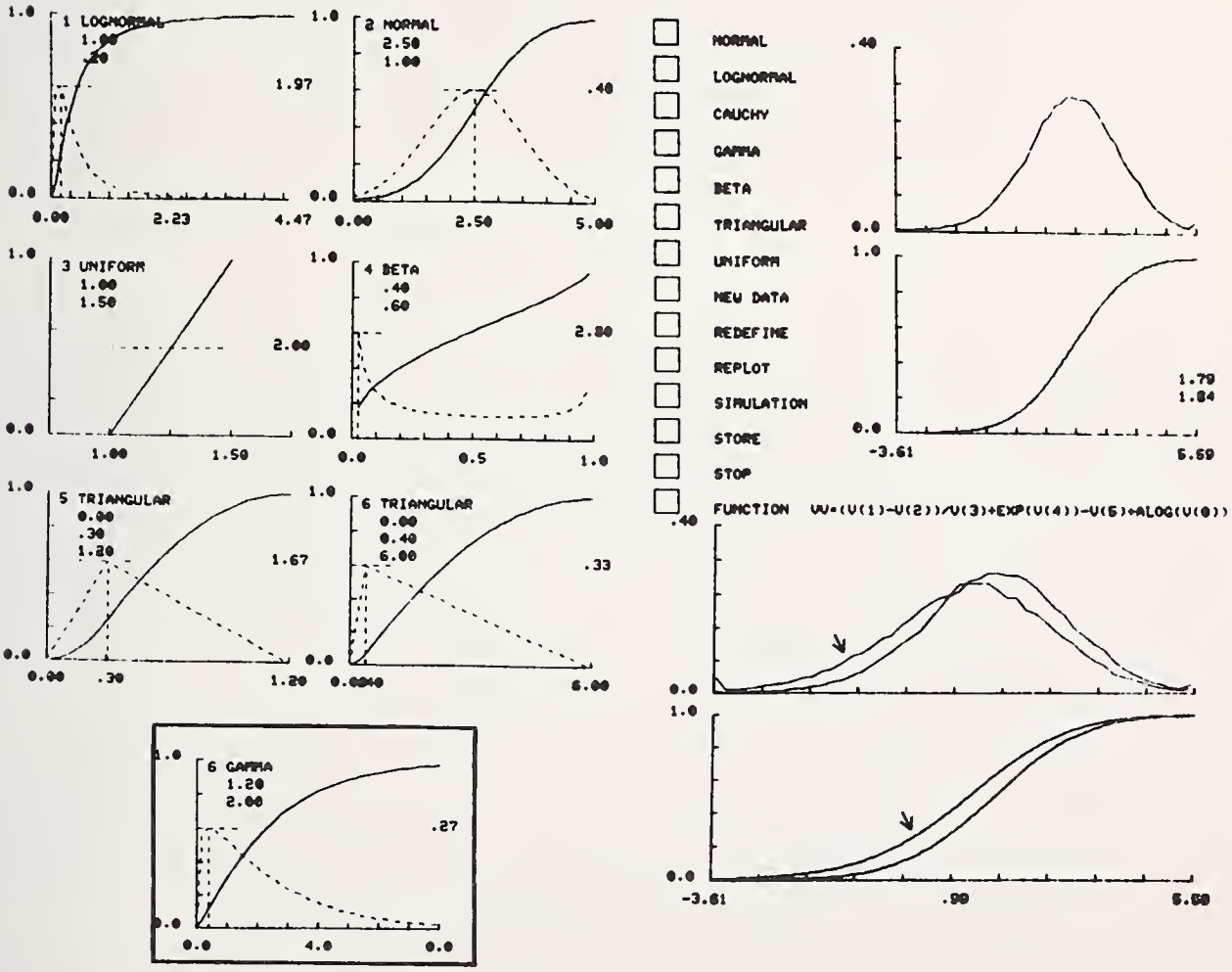


Figure 4. Display of two successive simulations by program SIMGRA1. The entire screen is shown. Detailed description of figure is in text. In the input area, scale values of density distributions of initial variables are printed to the right of each plot. The inset is part of the first simulation. Transformation Y in text is FUNCTION VV in the menu of options area. The storage area shows the distributions of both first and second simulations; the simulation area shows only the distributions of the second simulation.

Example 3. The artificial example shown in figure 4 demonstrates the accumulation feature of SIMGRA1. This feature is particularly useful for studying robustness of transformation. Let X_1, X_2, X_3, X_4, X_5, X_6 be independent random variables with

$X_1 \sim$ lognormal(1.0, 0.2), $X_2 \sim$ normal(2.5, 1.0), $X_3 \sim$ uniform(1.0, 1.5),

$X_4 \sim$ beta(0.4, 0.6), $X_5 \sim$ triangular(0.0, 0.3, 1.2) and $X_6 \sim$ gamma(1.2, 2.0). Let

$$Y = \frac{(X_1 - X_2)}{X_3} + e^{X_4} - X_5 + \log_e X_6 \qquad (2)$$

be a transformation. It would be difficult to obtain the density distribution of Y in analytic form in practice. Using SIMGRA1, a simulated density and a cumulative distribution of the transformation in equation (2) were first generated and stored. These first simulated distributions, indicated by arrows, are displayed in the storage area of the screen which is shown in figure 4. Figure 4 also shows the first five initial distributions in the input area. The sixth initial distribution, gamma, is shown in the inset. Estimated expected value and standard deviation of the transformation, not shown in the figure, were 1.26 and 1.71 respectively.

After the first simulation, the distribution of the sixth initial variable was interactively altered from gamma(1,2, 2.0) to triangular(0.0, 0.4, 6.0) as shown in the figure. The distribution of the same transformation in equation (2) of these six initial variables was then simulated and shown in the simulation area with estimated expected value 1.79 and standard deviation 1.84. For comparison with the first simulated distributions, the latter simulated distributions are also plotted in the storage area. In both simulation experiments, 50000 random numbers were generated for each initial variable.

5. CONCLUDING REMARKS

The program has been written as part of developments in statistical models for natural resource evaluation. Geological phenomena related to undiscovered mineral resources can be regarded as random variables as in Kaufman et al. (1975). In statistical modelling for resource evaluation, manipulations of random variables are required at an initial stage as in Miller et al. (1975). This program generates any type of transformation of a maximum of eight continuous random variables. It can be improved by adding routines for fitting known distributions (e.g. normal, lognormal etc.) to simulated distributions, and then performing statistical tests.

SIMGRA1 is in FORTRAN and, at present, is operational with a TEKTRONIX 4014/4015 on a CDC CYBER 74 computer. It requires 70000 octal words of core memory and uses the IMS subroutine library for generating pseudo random numbers. A user's guide for SIMGRA1, Chung et al.(1977), and the source program may be obtained upon request to the authors.

6. REFERENCES

CHUNG, C.F., DIVI, S.R. and FABBRI, A.G. (1977). User's guide for SIMGRA1: an interactive graphic program for simulating the distribution of transformations of several random variables. Geol. Surv. Can., Open File (in press).

KAUFMAN, G.M., BALCER, Y. and KRUYT, D. (1975). A probabilistic model of oil and gas discovery. Studies in geology no. 1 - Methods of estimating the volume of undiscovered oil and gas resources. The American Association of Petroleum Geologists, 113-142.

MATERN, B. (1960). Spatial variation. Medd. Skogforskningsinsitut, 49, no. 5, Stockholm.

MILLER, B.M., THOMSEN, H.L., DOLTON, G.L., COURY, A.B., HENDRICKS, T.A., LENNARTZ, F.E., POWERS, R.B., SABLE, E.G. and VARNES, K.L. (1975). Geological estimates of undiscovered recoverable oil and gas resources in the United States. U.S. Geological Survey Circular 725.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

MULTIPLE INCOMPLETE BETA INTEGRALS IN BAYES SUBSET SELECTION PROCEDURE FOR BINOMIAL PROBABILITY PARAMETERS

By

PREM NATH BHALLA
JACKSON STATE UNIVERSITY, JACKSON MS. 39217

ABSTRACT

In the Bayes subset selection procedure for the probability parameters of the binomial distribution we are faced with the problem of evaluation of incomplete beta integrals. By using the relationship between the beta distribution and the binomial distribution the incomplete beta integrals are reduced to a simple computational form.

Key words: Bayes; correct selection; expected size; incomplete integral; parameters; subset.

1. INTRODUCTION

There are several methods available for computing comulative distribution function of the beta distribution:

$$I_t(a,b) = \frac{1}{\beta(a,b)} \int_0^t \theta^{a-1} (1-\theta)^{b-1} d\theta$$

where β (a,b) is a beta function. These procedures for evaluating the incomplete beta integrals are approximate. In dealing with a Bayes subset selection problem a product of incomplete beta integrals occur in several inequalities. By using a relationship between beta distribution and binomial distribution these inequalities are reduced to a simpler computational form. In every case for which the total information was equal for each θ_i a check was made and the calculation found to be accurate to at least six decimal places. For example, if r = 3 and $n_1'' = n_2'' = n_3''$ and $X_1'' = X_2'' = X_3''$, then Pr $(\theta_i = \theta \max/\underline{x}) = .333333$.

2. SELECTION PROCEDURE FOR BINOMIAL PROBABILITY PARAMETERS

Set $\underline{\theta} = (\theta_1\ \theta_2,\ldots,\theta_r)$ be the vector of r independent binomial probability parameters where $0 < \theta_i < 1$, i = 1, 2,...,r. By using all available information a decision maker is to select a subset of binomial probability parameters which is asserted to contain the largest of such parameters. Bratcher and Bhalla [1] proved that the Bayes rule includes θ_i in the superior set if

$$\int_0^1 \prod_{k \neq i} G_k(\theta_i / x_k)\, dG(\theta_i / x) \geq \frac{1}{c+1}, \qquad (2.1)$$

where G_k represent the cumulative distribution function of θ_k and c is a constant.

Let $\underline{X} = (X_1, X_2,\ldots,X_r)$ be a vector of r independent binomial random variables. The probability distribution of X_i is binomial with parameters (n_i, θ_i),

$$f\ (\ x_i\ /\ \theta_i\ )\ =\ \binom{n_i}{x_i}\ \theta_i^{\,x_i}\ (\ 1\ -\ \theta_i)^{\ n_i\ -\ x_i},\ 0 \prec \theta_i \prec 1,\ x_i = 0,\ 1,\ldots n_i$$

It is assumed that the prior information available on each θ_i is approximately by a beta distribution,

$$g\ (\ \theta_i |\ x_i'\ ,\ n_i'\ )\ =\ [\beta\ (x_i',\ n_i'\ -\ x_i')]^{-1}\ \theta_i^{\ x_i'\ -\ 1}\ (1-\theta_i)^{\ n_i'-x_i'-1},\ 0 \prec \theta_i \prec 1. \qquad (2.2)$$

The unconditional distribution of X_i is beta binomial,

$$f\ (x_i)\ =\ (n_i'\ -1)\binom{n_i'}{x_i}\ \binom{n_i'-2}{x_i'-1}\ \Big/\ (n_i+n_i'-1)\ \binom{n_i+n_i'-2}{x_i+x_i'-1}\ ,\ x_i\ =\ 0,1,\ldots,n_i. \qquad (2.3)$$

The revised information on θ_i is given by the beta distribution with parameters $x_i' = x_i + x_i'$ and $n_i' = n_i + n_i'$. For diffuse information let the prior distribution be uniform,

$$g(\theta_i)\ =\ 1,\quad 0 \prec \theta_i \prec 1, \qquad (2.4)$$

where $x_i' = 1$ and $n_i' = 2$ in equation (2.2)

If the sample sizes are the same (i.e., $n_1 = n_2 = n_r = n$), then the unconditional distribution of x_i is

$$f\ (x_i)\ =\ \frac{1}{n\ +\ 1},\ x_i\ =\ 0,\ 1,\ldots,n. \qquad (2.5)$$

In the case of diffuse prior information and equal sample sizes, the revised information on θ_i is given by the beta distribution with parameters $x_i'' = x_i + 1$ and $n_i'' = n_i + 2$

From inequality (2.1) the Bayes procedure selects θ_i in the superior set S if $\Pr(\theta_i = \theta\ \max)$ is greater than or equal to the constant $c' = \frac{1}{c\ +\ 1}$. If a decision maker can specify c, each θ_i can be considered for inclusion in the superior set by evaluating inequality (2.1). The computational form of inequality (2.1) is obtained by substituting.

$$\prod_{K \neq i} G\ (\ \theta\ /\ \underline{x}\ )\ =\ \prod_{k \neq i}\ I_{\theta_i}\ (\ x_K''\ ,\ n_K''\ -\ X_K''), \qquad (2.6)$$

where I_t represent the incomplete beta function,

$$I_t\ (a,b) = [\beta(a,b)\ ]^{-1} \int_0^t \theta^{a-1}\ (\ 1 - \theta\ )^{b-1}\ d\theta.$$

Thus

$$\Pr\ (\theta_i = \theta_{max}\ \underline{x}) =$$

$$\int_0^1 \pi_{K\neq i}\ I_{\theta_i}\ (x_K'',\ n_K''-x_K'')\ [\beta(x_i'',n_i''-x_i'')\ ]^{-1}\ \theta_i^{x_i''-1}\ (1-\theta_i)^{n_i''-x_i''-1}\ d\theta_i. \tag{2.7}$$

Theorem 2.1. The probability in condition (2.1) for integer values of the prior parameter becomes

$$\Pr\ (\theta_i=\theta_{max}\ |\underline{x}) = \sum_{w_1=x_1''}^{n_1''-1} \sum_{w_2=x_2''}^{n_2''-1} \cdots \sum_{w_r=x_r''}^{n_r''-1} \binom{n_1''-1}{w_1} \binom{n_2''-1}{w_2} \cdots \binom{n_r''-1}{w_r} \cdot \tag{2.8}$$

$$\frac{\beta\left(\sum_{k\neq i} w_k + x_i'',\ \sum_{k\neq i} w_k + x_i'' - r - 1\right)}{\beta(x_i'',\ n_i'' - x_i'')},$$

where there is no summation over w_i.
Proof: By using the relationship between the beta distribution with integer parameters and the binomial distribution,

$$I_\theta\ (k,\ n-k+1) = \sum_{i=k}^{n} \binom{n}{i}\ \theta^i\ (1-\theta)^{n-i}, \tag{2.9}$$

and interchanging the order of summation and integration, we may reduce equation (2.7) to a simpler computational form

$$\Pr(\theta_i=\theta_{max}\ \underline{x}) = \sum_{w_1=x_1''}^{n_1''-1} \sum_{w_2=x_2''}^{n_2''-1} \cdots \sum_{w_r=x_r''}^{n_r''-1} \binom{n_1''-1}{w_1} \cdots \binom{n_r''-1}{w_r} \cdot$$

$$\frac{\beta\left(\sum_{k\neq i} w_k + x_i'',\ \sum_{k\neq i} w_k + x_i'' - r - 1\right)}{\beta\ (x_i'',\ n_i'' - x_i'')}$$

where thers is no summation over w_i.

Corollary 2.1 If the prior information is taken as the uniform distribution and $n_1 = n_2 = \ldots = n_r = n$, the computational form of equation (2.8) becomes

$$\Pr(\theta_i = \theta_{max} \mid \underline{x}) = \left(\frac{1}{r}\right)\binom{n}{x_i} \sum_{w_1=x_1+1}^{n+1} \cdots \sum_{w_{i-1}=x_{i-1}+1}^{n+1} \sum_{w_{i+1}=x_{i+1}+1}^{n+1} \cdots \tag{2.10}$$

$$\sum_{w_r=x_r+1}^{n+1} \binom{n+1}{w_1} \cdots \binom{n+1}{w_{i-1}} \binom{n+1}{w_{i+1}} \cdots \binom{n+1}{w_r} \Bigg/ \binom{rn+r-1}{x_i + \sum_{k=i} w_i}.$$

3. PROBABILITY OF CORRECT SELECTION AND EXPECTED SIZE OF THE SELECTED SUBSET

The binomial probability parameter θ_i is included in the superior set if $\underline{x} \in A_i$. The set A_i of $\underline{x}$'s are obtained from inequality (2.1) and equation (2.8). These sets A_i, (i=1,2,...,r) are employed to find probability of correct selection and the expected number of parameters in the superior set.

The probability of correct selection may be calculated by using equations (2.3) and (2.8). For a uniform prior distribution and equal sample sizes we make use of equations (2.5) and (2.10). The probability of correct selection,

$$\Pr(\theta_i = \theta_{max},\ \theta_i \in S) = \Pr(cs) = \frac{r}{(n+1)^r} \sum_{\underline{x} \in A_i} \Pr(\theta_i = \theta_{max}/\underline{x}) \tag{3.1}$$

In case of uniform prior and equal sample sizes from each binomial population

$$\Pr(\underline{x} \in A_i) = \frac{(\#\text{ of } x\text{'s in } A_i)}{(n+1)^r} \tag{3.2}$$

and this probability will have the same value for each i, (i=1 or 2,... or r). Thus

$$E(N) = \sum_{i=1}^{r} \Pr(\underline{x} \in A_i) = \frac{r(\#\text{ of } x\text{'s in } A)}{(n+1)^r}\ . \tag{3.3}$$

REFERENCES

BRATCHER, T. L. and BHALLA, P. N. (1974). On the properties of an Optimal Selection Procedure. Communications in Statistics, 3 (2), 191-196.

DUNCAN, D. D. (1961). Bayes rules for a common multiple comparison a related Student-t problems. ANN. MATH. STATIST. 32, 1013-33.

FERGUSON, T. S. (1967). Mathematical Statistics, A Decision Theoretic Approach. Acedemic Press, Inc...New York.

TRETTER, J. MARIETTA and WALSTER G. WILLIAMS, (1976). A Fast Algorithm For Significant Digit Computation Of The Incomplete Beta Function For Extreme Values. Proceeding of Computer Science and Statistics; 9th Annual Symposium on the Interface, Harvard University and MIT, Cambridge, MA.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

NUMERICAL SOLUTIONS OF THE INCOMPLETE GAMMA FUNCTION

Hubert Bouver, SUNY at Plattsburgh 12901
Rolf E. Bargmann, The University of Georgia, Athens 30601

ABSTRACT

The purpose of this paper is to present the derivation of standard and newly developed formulas, computational algorithms, and modules for comparison of numerical methods in the evaluation of the Incomplete Gamma function. Different series and continued fraction expansions were compared with the goal of finding the most efficient techniques for different domain of the shape parameter of the Gamma distribution. These methods, in addition to the standard serie solutions and continued fraction expansions, include the recent technique of the Hermite expansion around a local maximum which was investigated for large values of the shape parameter of the Gamma distribution along with the standard Wilson-Hilferty approximation. On the basis of time comparison, the most efficient and applicable modules were then combined into a distribution package where computer subprogram function were written in standard FORTRAN to evaluate 1) the cumulative density function, 2) the inverse of the cumulative density function and, 3) the probability density function of the Gamma distribution with guaranteed precision of at least 10 significant digits.

Key words: Asymptotic expansion; cumulative density function; fixed-length continued fraction and series; Gamma distribution; Hermite polynomials; probability density function; statistical computation; Taylor series; Wilson-Hilferty approximation.

1. INTRODUCTION

A comparison of modern computational algorithms, for mathematical functions (e.g. IBM library (1972), with those used twenty years ago, shows a trend toward higher efficiency with guaranteed precision. Even for elementary trigonometric, exponential and logarithmic functions, the classical series expansions have been replaced by optimized fixed-length continued fractions and Chebyshef minimax rational functions. The collection of mathematical functions by Abramowitz and Stegun (1968) have been used extensively, especially the formulas and mathematical properties of series expansions and rational fractions. Johnson and Kotz (1970), describe in detail properties of many statistical distribution functions and present formulas especially developed for approximations. They devote particular attention to formulas for small range of arguments and for modest precision. The techniques of numerical analysis are, for the most part, well known and are merely studied as they relate to statistical distribution functions. However, the Hermite expansion around a maximum, as described next, appears to be a novel approach. It seems to have superficial similarity with a method described by Daniel (1954) which Kendall and Stuart (1969) regarded as an entirely novel approach for the evaluation of distributions. The Hermite expansion proved very successful for large values of parameters in the Incomplete Gamma and was needed to fill a rather large gaps between continued fractions, series and Normal approximations, (see Figures 1 and 2). If one is satisfied with low

precision (e.g. 3 places) and a limited range of probabilities (e.g. 0.01 and 0.99 level) reference to the central limit theorem, variance stabilization transformations, and other approximations may be quite adequate. On the other hand, if high precision is required, these stand-by approximations have proven useless. For example, as will be noted later, the improved Normal approximation (Wilson-Hilferty) of the Incomplete Gamma Function cannot be used unless the shape parameter reaches an order of magnitude of 10 million (for 10-digit precision).

2. THE HERMITE EXPANSION

Let J_n be defined as an Incomplete Gamma function

$$J_n(b) = \frac{1}{\Gamma(n+1)} \int_0^b x^n e^{-x}\, dx \tag{1}$$

where $n > 0$ is a high power, not necessarily an integer, and $0 < x < \infty$.
In the previous integral eq. (1), write $x^n e^{-x} = e^{n[\log x - x/n]}$, and define $f(x) = \log x - x/n$ to obtain $x^n e^{-x} = e^{nf(x)}$. Thus, eq. (1) becomes

$$J_n(b) = \frac{1}{\Gamma(n+1)} \int_0^b e^{nf(x)}\, dx \tag{2}$$

First, we need to find the local maximum of $f(x)$ before we use the Taylor series expansion. The necessary derivatives are: $f(x) = \log x - x/n$, $f'(x) = 1/x - 1/n$, $f''(x) = -1/x^2$, ... $f^{(r)}(x) = (-1)^{(r-1)}(r-1)!/x^r$, $r = 2, 3, \ldots$.
Since $f'(\hat{x}) = 0$ implies $\hat{x} = n$, and $f''(\hat{x}) = -1/n^2 < 0$, it follows that $\hat{x} = n$ gives the local maximum. Hence the Taylor series expansion of $f(x)$ around its local maximum is

$$f(x) = \log n - 1 - \frac{(x-n)^2}{2n^2} + \frac{(x-n)^3}{3n^3} - \frac{(x-n)^4}{4n^4} + \ldots + \frac{(-1)^{r-1}(x-n)^r}{rn^r} + \ldots$$

and

$$e^{nf(x)} = n^n e^{-n} \cdot e^{-(x-n)^2/2n} \cdot e^{\hat{R}(x)}$$

where

$$\hat{R}(x) = (x-n)^3/3n^2 - (x-n)^4/4n^3 + (x-n)^5/5n^4 - \ldots$$

Hence eq. (2) may be written as

$$J_n(b) = \frac{n^n e^{-n}}{\Gamma(n+1)} \int_0^b e^{-\frac{(x-n)^2}{2n}} e^{\hat{R}(x)}\, dx. \tag{3}$$

Now let $z = x-n/\sqrt{n}$ in eq. (3), then

$$J_n(b) = \frac{n^n e^{-n}\sqrt{2\pi n}}{\Gamma(n+1)} \int_{-\sqrt{n}}^{(b-n)/\sqrt{n}} \phi(z)\, e^{R(z)}\, dz, \tag{4}$$

where $R(z) = \hat{R}\,(\sqrt{n}\, z + n)$ and $\phi(z)$ is the normal probability density function.
Now in the expansion of $e^{R(z)}$ we have

$$e^{R(z)} = 1 + z^3/\sqrt{n}[A(z)] + z^6/2n[A(z)]^2 + z^9/6n^{3/2}[A(z)]^3 + z^{12}/24n^2[A(z)]^4 + z^{15}/120n^{5/2}[A(z)]^5 + \ldots \quad , \tag{5}$$

where $$A(z) = 1/3 - z/4\sqrt{n} + z^2/5n - z^3/6n^{3/2} + z^4/7n^2 - z^5/8n^{5/2} + \ldots \quad .$$

We note from eq. (5) that all the terms of the third and higher power in the expansion of $e^{R(z)}$, have terms of order $1/\sqrt{n}$, $1/n$, $1/n^{3/2}$, ... in the denominator, thus forming the asymptotic series. Upon simplification of eq. (5), we obtain the following asymptotic expansion of $e^{R(z)}$ through terms up to $1/n^5$:

$$\begin{aligned} e^{R(z)} = 1 &+ 1/\sqrt{n}[c_{11}z^3] - 1/n[c_{21}z^4 - c_{22}z^6] + 1/n^{3/2}[c_{31}z^5 - c_{32}z^7 + c_{33}z^9] \\ &- 1/n^2[c_{41}z^6 - c_{42}z^8 + c_{43}z^{10} - c_{44}z^{12}] \\ &+ 1/n^{5/2}[c_{51}z^7 - c_{52}z^9 + c_{53}z^{11} - c_{54}z^{13} + c_{55}z^{15}] \quad \ldots\ldots \\ &+ 1/n^5[c_{10,1}z^{12} - c_{10,6}z^{22} + c_{10,7}z^{24} - c_{10,8}z^{26} + c_{10,9}z^{28} - c_{10,10}z^{30}] \quad , \end{aligned}$$

where the constants c_{ij} are presented in the i^{th} row and j^{th} column of lower triangular matrix C (see Bouver and Bargmann (1975) for unabbreviated results).

$$C = \begin{bmatrix} \frac{1}{3} & & & & & & & \\ \frac{1}{4} & \frac{1}{18} & & & & & & \\ \frac{1}{5} & \frac{1}{12} & \frac{1}{162} & & & & & \\ \frac{1}{6} & \frac{47}{480} & \frac{1}{72} & \frac{1}{1,944} & & & & \\ \frac{1}{7} & \frac{19}{180} & \frac{31}{1,440} & \frac{1}{648} & \frac{1}{29,260} & & & \\ \vdots & & & & \vdots & & & \\ & & & & & & \frac{1}{1,058,158,080} & \frac{1}{214,277,011,200} \\ \frac{1}{12} & \frac{42,131}{388,080} & \frac{59,197}{1,209,600} & \frac{1,722,811}{163,296,000} & \frac{143,921}{116,121,600} & \frac{44,021}{522,547,200} & \frac{1,699}{503,884,800} & \frac{137}{1,763,596,800} \end{bmatrix}$$

Taking the polynomial expansion of $e^{R(z)}$ in eq. (6), we first transform it into Hermite polynomials (A computer program (HERPOL) was developed to transform a power series into an Hermite polynomial and vice versa). This Hermite expansion for $e^{R(z)}$ is substituted into $J_n(b)$ in eq. (4). Integration is then performed and the resulting Hermite polynomials are re-transformed into power series expansions using HERPOL. Thus, the final asymptotic expression is

$$J_n(b) = G(b;n+1) = \frac{n^n e^{-n}\sqrt{2\pi n}}{\Gamma(n+1)} \left\{ 1/\sqrt{n}[a_{11}+a_{12}z^2]\,\phi(z) + 1/n[(a_{21}z+a_{22}z^3+a_{23}z^5)\,\phi(z) - b_2\phi(z)] \right.$$

$$+ 1/n^{3/2}\,[-a_{31}-a_{32}z^2-a_{33}z^4-a_{34}z^6+a_{35}z^8]\,\phi(z) + 1/n\,[(a_{41}z+a_{42}z^3+a_{43}z^5+a_{44}z^7$$

$$-a_{45}z^9+a_{46}z^{11})\,\phi(z)-b_4\Phi(z)] + \ldots\ldots + 1/n^5[(a_{10,1}z+a_{10,2}z^3+a_{10,3}z^5+a_{10,4}z^7+a_{10,5}z^9$$

$$+a_{10,6}z^{11}+a_{10,7}z^{13}-a_{10,8}z^{15}+a_{10,9}z^{17}-a_{10,10}z^{19}+a_{10,11}z^{21}-a_{10,12}z^{23}+a_{10,13}z^{25}$$

$$-a_{10,14}z^{27}+a_{10,15}z^{29})\,\phi(z)-b_{10}\,\Phi(z)]\Big|_{-\sqrt{n}}^{b-n/\sqrt{n}}\Bigg\}\quad,$$

where the constants a_{ij} and b_i of eq. (7) are, (for more complete details see Bouver and Bargmann (1975))

$a_{11} = 2/3,\ a_{12} = 1/3$;

$a_{21} = 1/12,\ a_{22} = 1/36,\ a_{23} = 1/18$; $\qquad b_2 = 1/12$

$a_{31} = 4/135,\ a_{32} = 2/135,\ a_{33} = 1/270,\ a_{34} = 11/324,\ a_{35} = 1/162$;

$a_{41} = 1/288,\ a_{42} = 1/864,\ a_{43} = 1/4{,}320,\ a_{44} = 103/4{,}320,\ a_{45} = 2/243,$

$a_{46} = 1/1{,}944$; $\qquad b_4 = 1/288$

$a_{51} = \ldots\ldots\ldots\ldots$

$a_{10,1} = 163{,}879/209{,}018{,}880,\ a_{10,2} = 163{,}879/627{,}056{,}640,\ a_{10,3} = 163{,}879/3{,}135{,}283{,}200,$

$a_{10,4} = 163{,}879/21{,}946{,}982{,}400,\ a_{10,5} = 163{,}879/197{,}522{,}841{,}600,$

$a_{10,6} = 53{,}260{,}675/706{,}144{,}158{,}720{,}000,$

$a_{10,7} = 13{,}927{,}905{,}283/2{,}172{,}751{,}257{,}600,$

$a_{10,8} = 2{,}882{,}490{,}481/423{,}263{,}232{,}000,$

$a_{10,9} = 1{,}048{,}924{,}927/423{,}263{,}232{,}000,$

$a_{10,10} = 35{,}964{,}223/84{,}652{,}646{,}400,\ a_{10,11} = 4{,}925{,}299/126{,}978{,}969{,}600,$

$a_{10,12} = 62{,}737/31{,}744{,}742{,}400,\ a_{10,13} = 443/7{,}936{,}185{,}600,$

$a_{10,14} = 347/428{,}554{,}024{,}000,\ a_{10,15} = 1/214{,}277{,}011{,}200,$

$b_{10} = 163{,}879/209{,}018{,}880.$

A computer program called COEF was written to calculate the coefficients c_i of the eq. (6). This main program uses the subroutine ERZ which calculates all the coefficients of eq. (5) of the series expansion $e^{R(z)}$. The first call to the subroutine HERPOL transforms the coefficients of the series expansion into Hermite polynomials. Next a reduction for the integration is performed. The second call to Herpol translates the resulting Hermite polynomial back into the coefficients a_{ij} and b_i of the final power series expansion, as stated in eq. (7). The computer program module GAMAX is the single precision version (CDC 48 bits mantissa) of the Hermite expansion about the maximum and includes the block data subroutine IGAMA, which contains all the coefficient values (8) of eq. (7). GAMAX uses only the terms up to $1/n^4$ in the final eq. (7). It may be of interest to the reader, at this point, to have a general idea of the degree of precision attainable. With the single precision version, (CDC-, approximately 14 significant digits, using a 48 bit mantissa) the Incomplete Gamma function, at $\alpha = 100$, has approximately 12 significamt digits of precision at the mean and 10 significant digits at the extreme tail ends ($\mu \pm 10\sigma$). Even when α is as low as 50 or 10, the precision is about 9 or 5 significant digits, respectively at the mean, and 7 or 4 at the extreme tail ends. Of course, in a program combining modules, GAMAX should be used when $\alpha \geq 100$ and its accuracy will be valid for 10 significant digits.

In the comparison of modules for the Incomplete Gamma function, a diagrammatic display

(Figures 1 and 2) will show the regions in which different modules are superior. It is well known that the Gamma distribution approaches normality as the shape parameter, α, approach infinity. As has been discussed in Abramowitz and Stegun, the cube root of a random variable which has the Gamma distribution, approaches normality much faster. Even so, to obtain 10 digits of relative precision, values of α as large as 10^8 are needed before the Wilson-Hilferty approximation can be used. The cut-off point $\alpha = 100$, which had been used in distribution programs before, could guarantee only 5 significant digits of precision. With the Hermite expansion the region from several hundreds to 10^8 can be covered very satisfactorily (fast as well as precise). With the appropriate choice of modules, in accordance with Figure (2), the average time of execution of GAMX 10 digits of precision is:

Shape Parameter	Time
for $\alpha < 100$	0.25 msec
for $100 < \alpha < 1000$	0.65 msec
for $1000 < \alpha < 10^8$	0.55 msec
for $\alpha > 10^8$	0.45 msec

The dotted line in Figure 2 represents the turning point between the series and the continued fraction.

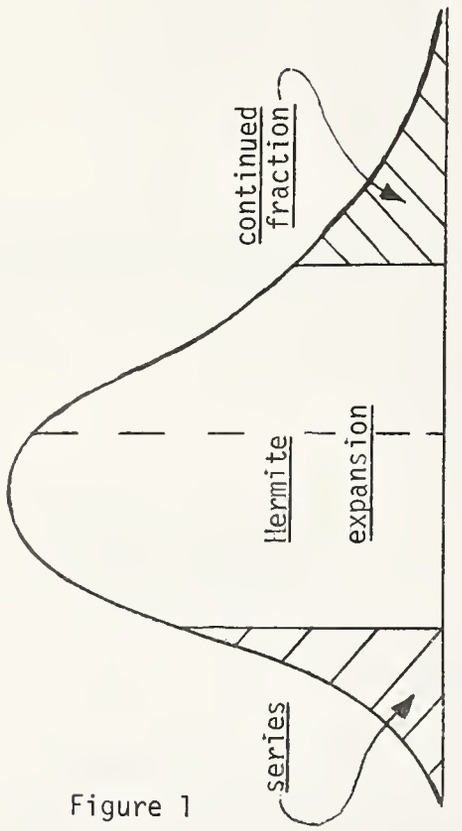


Figure 1

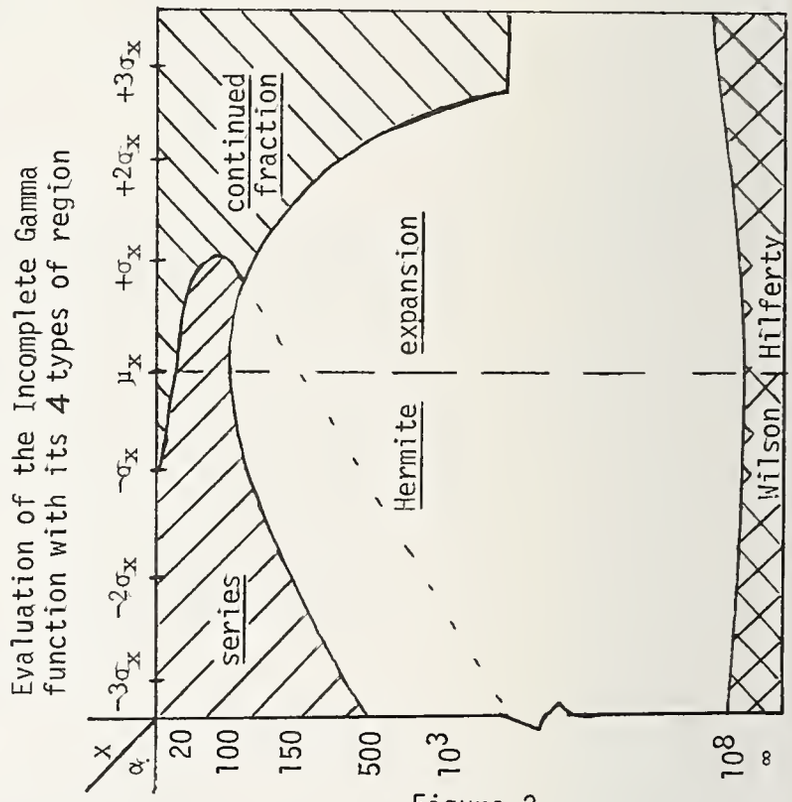


Figure 2

3. THE SERIES AND CONTINUED FRACTION SOLUTIONS

The Incomplete Gamma Function may be written as

$$G(x;n) = 1 - e^{-x} \sum_{i=1}^{n-1} \frac{x^i}{i!} = e^{-x} \sum_{i=n}^{\infty} \frac{x^i}{n!} .$$

Thus as an infinite series we simply have

$$G(x;\alpha) = e^{-x} \left[\frac{x^\alpha}{\Gamma(\alpha+1)} + \frac{x^{\alpha+1}}{\Gamma(\alpha+2)} + \frac{x^{\alpha+2}}{\Gamma(\alpha+3)} + \ldots . \right] .$$

For high precision evaluation, this series is very effective as long as x is much less than

α. As x approaches α, high precision will require many terms if α becomes large (> 100, say see Figures 1 and 2).
The continued fraction plays an important role for the evaluation of the Incomplete Gamma function usually for values of $x > \alpha$, and moderately large values of α (< 100). (i.e. for evaluation of the right side from mean and for small to moderately large degrees of freedom, see Figures 1 and 2). For the Mill's ratio we have the following rational fraction:

$$R(x) = \frac{1 - G(x;\alpha)}{g(x;\alpha)} = \frac{1 - e^{-x}\left[\frac{x^{\alpha}}{\Gamma(\alpha+1)} + \frac{x^{\alpha+1}}{\Gamma(\alpha+2)} + \frac{x^{\alpha+2}}{\Gamma(\alpha+3)} + \cdots\right]}{x^{\alpha-1} + x^{\alpha} + \frac{x^{\alpha+1}}{2!} + \frac{x^{\alpha+2}}{3!} + \frac{x^{\alpha+3}}{4!} + \cdots}$$

which, if converted into a continued fraction, becomes

$$G(x;\alpha) = 1 - g(x;\alpha)\, R(x) = 1 - g(x;\alpha)\left[\frac{1}{x+}\ \frac{1-\alpha}{1+}\ \frac{1}{x+}\ \frac{2-\alpha}{1+}\ \frac{2}{x+}\ \cdots\ \frac{n}{x+}\ \frac{n+1-\alpha}{1+}\ \cdots\right] .$$

4. REFERENCES

ABRAMOWITZ, M. and STEGUN, I. A. (1968). Handbook of Mathematical Functions. National Bureau of Standards, Washington, D. C.

BARGMANN, R. E. (1970). A statistical distribution computer package. Department of Statistics and Computer Science, the University of Georgia, Athens.

BOUVER, H. (1973). Curve fitting by method of moments. Themis Technical Report No. 29. The University of Georgia, Athens.

BOUVER, H. and BARGMANN, R. E. (1975). Computational algorithms and modules for the evaluation of statistical distribution functions. Themis Technical Report No. 36. The University of Georgia, Athens.

DANIELS, H. E. (1954). Saddlepoint approximations in statistics. Ann. Math. Statist., 25, 631.

IBM Systems Reference Library. (1972). Mathematical and service subprograms. IBM GC 28-6818.

JOHNSON, N. L. and KOTZ, S. (1970). Continuous Univariate Distributions-1, 2. Houghton-Mifflin Co..

KENDALL, M. G. and STUART, A. (1969). The Advanced Theory of Statistics. London, Griffin.

BIOGRAPHIES

Hubert Bouver received a Ph.D. in statistics from the University of Georgia in 1975 and is an assistant professor of Computer Science with SUNY at Plattsburgh. For the past 4 years, he has developed a Statistical Distribution Package (SDP10) which would guarantee the user 10 significant digits of precision.

Rolf E. Bargmann received a Ph.D. in statistics in 1957 from the University of North Carolina. He has been a professor in the department of Statistics and Computer Science at the University of Georgia since 1965. Bargmann is a fellow of ASA and AAS, his speciality is in multivariate analysis and in statistical computing.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

THE OPCS LONGITUDINAL STUDY

T J Orchard
Office of Population Censuses & Surveys
Titchfield, UK

ABSTRACT

The OPCS Longitudinal Study links data on birth, death and cancer registrations with that collected at the 1971 Census, for a 1% sample of the England and Wales population. The paper describes the current computer system and the investigations into the possible use of Data Base Management Systems.

Key words: Data Base Management Systems; Longitudinal Data.

1. INTRODUCTION

The Office of Population, Censuses and Surveys collects data in order to provide statistics on the population of England and Wales. In addition to size this includes social, economic and medical characteristics, such as mortality.

The basic data sources consist of the Population Census (16½ million households and 49 million persons collected each 10 or 5 years), Death Registrations (600,000 each year), Birth Registrations (600,000 each year), Cancer Registrations (140,000 each year) and Migrations (140,000 overseas and 3½ million internally).

It was recognised as long ago as 1839, by William Farr the Registrar General at that time, that cohort analysis would provide more information on mortality than could be obtained from a cross-sectional analysis. The costs and practical problems associated with linking data from the various sources have restricted the data analysis to a cross-sectional approach until fairly recently. The Longitudinal Study is designed to link the information collected at the Census with evens such as births. A major benefit of doing this is that the Census gives details on housing and education that are not recorded on the event files, this enables events to be related to housing and parental characteristics. Following individuals through time is of benefit in studies of occupational and area mortality, in which chronic diseases may develop over a period of years, since the characteristics on event files relate only to the time of the event and may therefore be misleading.

A full description of the expected benefits as well as the guidelines for a Longitudinal Study covering a 1% sample of the population are contained in a booklet (1) published in 1973.

A computer system for linking the data and producing tabulations has been developed over the last two years. At the time the system was designed it was recognised that a new system would be required once the project had stabilised. The object of this paper is to describe the current computer system, some of the problems with managing data of this type and the investigations into a new system which have just commenced.

2. THE DATA FILES

A sample member is defined to be "A person who, at the time of the 1971 Census enumeration, or subsequent addition to the sample, had a stated date of birth on a selected date and a usual residence within the area of the Longitudinal Study". By choosing four selected dates a 1% dample of the population was obtained giving a sample size of 500,000.

Each member of the sample is assigned a unique serial number which is used for the computer linking of records.

Clearly in order to identify sample members as events occur, an index of these serial numbers is required and this is facilitated by the fact that each resident of the UK has a family doctor and a unique National Health Service number. The National Health Service Central Register (NHSCR) is an index, maintained by the OPCS on behalf of the Department of Health and Social Security, giving names and addresses of individuals and their Family Practitioners.

The index records for the Longitudinal Study sample members are flagged so that they may be quickly identified.

Events such as Internal Migration, Enlistment in the Armed Forces, Immigration and Emmigration initiate actions by the staff maintaining the NHSCR and data files can be constructed clerically. Birth, Death and Cancer records are extracted from data files using date of birth. To get the LS serial number these records must be matched with the source documents, which contain name and address, and forwarded to the NHSCR for the LS number to be obtained from the index cards. The physical separation of the computer records, the source documents and the NHSCR is seen as being essential in order to ensure the privacy of data on individual sample members.

The 1971 Census Household File is hierarchical, persons within families within households, coded in binary. The file in the current system is recoded to character and contains some household and family information for each sample member. The Personal File is also a recoded extraction from 1971 Census data and contains some family and household information as well as personal details.

Since date of birth as recorded at an event identifies a potential sample member there are two possible types of error. An individual may state an LS date of birth at Census but not at an event, this we shall call a Type A error. Conversely an LS date of birth may be recorded at an event but was not at census, this is called a Type B error.

From the definition of a sample member the date of birth stated at census is taken as being correct. This implies that the records having aType A error should be included in the data and those with Type B should be excluded. Type A errors can only be detected however for those events, such as death, which result in action by the NHSCR. Currently Type B records are deleted and Type A records added whenever the reasons for not matching can be determined. The records concerned have also been saved separately and could be subjected to a statistical analysis.

3. THE CURRENT COMPUTER SYSTEM

The constraints imposed on the design were that COBOL should be used for all applications programs, that existing utility programs should be used where possible, that the data should be stored on magnetic tape and processed in a batch mode and that the data records must be fixed length and coded in character or numeric.

The system was designed to create temporary work-files of records for input to the standard tabulation system of the office. To create a work-file, records from one data file and are matched then merged with a second file in order to produce linked records. The matching programs require the data files to be sorted to LS serial number order and the tabulation programs require the linked records to be sorted on the primary key, or major axis of the table. The tabulation programs have the facility to use only those records having the specified attributes, hence extractions are not required.

As an example of the processes involved suppose that it was required to tabulate deaths in 1971 by a. year of birth, b. month of death, c. cause of death, d. social class and e. educational attainment. Items a. and e. are to be taken from the Census records and the remainder from the Death records.

The 500,000 Census Personal records and the 6,000 1971 Death records would be input to a record matching utility which outputs pairs of matched records, a census record followed by a death record. This file is passed to another utility which combines the two records and sorts them into year of birth order for passing to the tabulation system. If only a subset, perhaps all males, of the 1971 Deaths File was required or if the Deaths File contains all deaths to date in LS serial number order (as is planned), the tabulation system would process all records in order to tabulate perhaps only 10% of them.

Since most requests are census information all 500,000 Census records have to be processed and this has resulted in tabulation requests being batched together.

Another problem is that the Census records do not contain all items on the original, variable length binary, records and hence re-extractions have had to be made. The Census Household file also contains only a subset of the total household information.

The system design was dictated by the need for an economic approach to systems development during the experimental stages of the project. It was recognised however that the system would require enhancements to deal with future requests of the data and in order to add data from another Census.

4. INVESTIGATIONS INTO A NEW SYSTEM

The requirements of the system are for the production, on an ad-hoc basis with almost immediate response, of tabulations and extractions of items taken from many different data files. The only privacy requirement is that it must be impossible to identify individuals from the output. It therefore seems to be a good area for using a Data Base Management System (DBMS) and investigations into this have just begun.

The major constraint on the use of a DBMS is that only 360 million bytes of disc storage can be made available. This may be compared to the size of data file, 220 million bytes, which would result from putting all information (excluding Census household data) onto a single record in character form.

Another major constraint is that the system must be portable from an ICL 1900 (based on 24 bit words) to an ICL 2900 (based on 8 bit bytes). This means that assembler language cannot be used. The office has standardised on the use of COBOL in order to facilitate the transfer of programs but it appears that a language, such as Algol 68, which supports more data types would be better for this particular system, where data storage is a problem.

At present there seems to be three choices for software, the ICL version of Cullinane's IDMS, a self-written DBMS or just to improve the current system without any fundamental re-design. There are also several choices for the type of self-written DBMS with Codasyl or Relational approaches as the major alternatives. Since the fundamental requirement is for tables it would seem to be more efficient for the tabulation system to be part of the package, this again will require careful investigation. What is clear however is that any new system must be designed with a view to including data from the 1981 Census and also information from the Census Household files.

5. CONCLUDING COMMENTS

Any system design in central government requires careful examination of costs and benefits and quite often severe constraints are imposed on the designers. In the case of the Longitudinal Study the constraints imposed seem to have resulted in a system which falls short of what is now required. The programmers concerned with the current system have, in true programmer style, adapted the design to overcome some of the shortcomings but clearly there is still much to be gained from a complete re-design.

With unlimited resources the problem of developing a Data Base Management System for such a large and messy set of data could be guaranteed to keep any programmer happy for quite some time. Even the requirement of portability would be viewed as a challenge rather than a constraint.

Although the resources available are limited and the constraints are only slightly less severe than before we are confident of being able to design a system that satisfies us as well as meeting the less demanding requirements of the users.

BIOGRAPHY

Terry Orchard is a statistician with the Computer Division of OPCS and has responsibility for Statistical Computing, Data Base Studies and Tabulation Systems for very large data files

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

DERIVATIVE-FREE NONLINEAR REGRESSION

Mary L. Ralston and Robert I. Jennrich
Health Sciences Computing Facility and Dept. of Mathematics
University of California, Los Angeles, Ca. 90024

ABSTRACT

A new derivative-free algorithm for finding the minimum of a function of the form $Q(\theta) = \Sigma_{j=1}^{n}(y_j - f_j(\theta))^2$ has been developed. Like the Gauss-Newton algorithm, the new algorithm is based on a sequence of linear approximations to the $f_j(\theta)$. However, unlike the Gauss-Newton algorithm, the new algorithm doesn't use derivatives, and hence is called Dud. Dud uses secants to the $f_j(\theta)$ which pass through $(p + 1)$ previous estimates of the solution, where p is the dimension of θ. Since the $f_j(\theta)$ have already been computed for these values of θ, only one new function evaluation is required per iteration. Consequently, Dud is potentially economical in the use of function evaluations. The performance of a FORTRAN implementation of Dud was evaluated on a number of standard test problems from the literature. The results demonstrate that Dud can be used successfully on a variety of problems.

Keywords: Derivative-free; fitting differential equations; nonlinear least squares

1. INTRODUCTION

There is a growing recognition of the need for derivative-free methods of fitting, not only because it is inconvenient to provide the derivatives required by most nonlinear optimization algorithms, but because in a large class of important problems it is difficult and expensive to do so. We have in mind fitting problems in which the response function is defined by a system of not necessarily linear differential equations. In engineering such problems arise naturally in systems analysis. In biology they are found under the general heading of compartment analysis. In each case the response function is evaluated by numerical integration of the defining system. Parametric derivatives generally must be found by further integration of a derived system, one for each parameter. In situations such as these, derivative-free algorithms are particularly attractive - especially ones that make more efficient use of previously computed function values.

Numerical studies by Box (1966) and Bard (1970) have shown that when the function to be minimized takes the form of a sum of squares, algorithms that use the Gauss-Newton approach can be faster than those that do not. Moreover, in addition to being the classical and most extensively used algorithm for nonlinear least squares estimation, the Gauss-Newton algorithm has intimate connections with maximum likelihood estimation algorithms (Bradley, 1973; Charnes, et al., 1976; Jennrich and Moore, 1975 and Nelder and Wedderburn, 1972) and modern methods of robust estimation (Beaton and Tukey, 1974).

The need for derivative-free algorithms has inspired many practitioners, for example, Berman and Weiss (1967), to replace derivatives with difference approximations, and has inspired others, such as Powell (1965) and Peckham (1970), to develop special derivative-free least squares algorithms. The algorithm we consider here, called Dud (doesn't use derivatives), is basically a derivative-free Gauss-Newton algorithm that gives one iteration for each function evaluation.

2. DUD

We want to consider the least squares fitting problem wherein one seeks a parameter vector $\theta = (\theta_1,\ldots,\theta_p)'$ to minimize a sum of squares

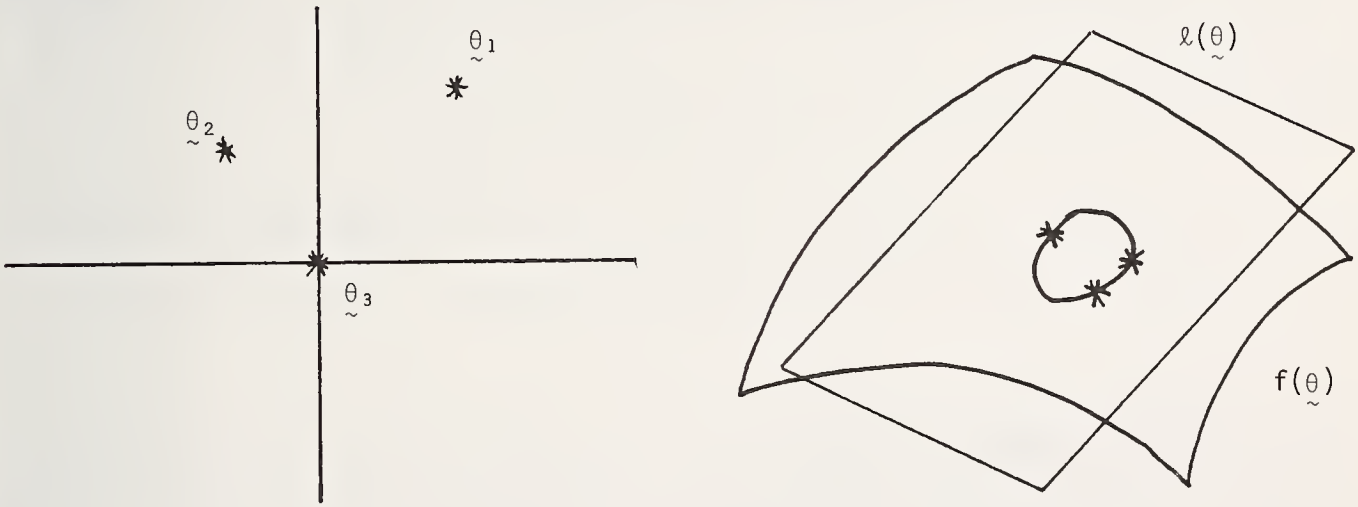
$$Q(\theta) = \sum_{i=1}^{n}(y_i - f_i(\theta))^2 \ . \tag{1}$$

The y_i are components of an observed data vector $y = (y_i)$ and the $f_j(\theta)$ are components of a vector valued response function $f(\theta) = (f_i(\theta))$. In vector notation, using the standard Euclidian norm,

$$Q(\theta) = ||y - f(\theta)||^2 \ . \tag{2}$$

In each iteration the Gauss-Newton algorithm approximates $f(\theta)$ by a first order Taylor expansion about the current value of the parameter vector θ and solves the resulting linear least squares problem to obtain a new value of θ. Dud, on the other hand, approximates $f(\theta)$ at each iteration by an affine function which agrees with $f(\theta)$ at p+1 previous values of the parameter vector. This also leads to a linear least squares problem which is solved to obtain a new value of θ. The new value replaces one of the currently used parameter vectors and the updated set is passed to the next iteration.

From a geometric point of view, the Gauss-Newton algorithm approximates the p-dimensional manifold spanned by the values of $f(\theta)$ by a tangent plane at a current value of $f(\theta)$. Dud approximates the manifold by the secant plane through p+1 previous values of $f(\theta)$ (see Figure 1).



Parameter Space Variable Space

Figure 1. A geometric picture of the affine approximation used by Dud.

3. DETAILS OF IMPLEMENTATION

Formulas are simplified if Dud's linear approximation is written as a function of the transformed parameters α, which are defined implicitly at each iteration by

$$\theta = \theta_{p+1} + \Delta\Theta\alpha \tag{3}$$

where $\theta_1,\ldots,\theta_{p+1}$ are estimates from previous iterations (numbered by age with θ_1 being the oldest), and the ith column of $\Delta\Theta$ is given by

$$\Delta\Theta_i = \theta_i - \theta_{p+1}; \quad i=1,\ldots,p .$$

The linear approximation is given by

$$\ell(\alpha) = f(\theta_{p+1}) + \Delta F\alpha \tag{4}$$

where the ith column of ΔF is given by

$$\Delta F_i = f(\theta_i) - f(\theta_{p+1})$$

for i=1,...,p. One iteration consists of minimizing

$$Q(\alpha) = (y - \ell(\alpha))'(y - \ell(\alpha)). \tag{5}$$

The solution is given by

$$\alpha = (\Delta F'\Delta F)^{-1}\Delta F'(y-f(\theta_{p+1})) \tag{6}$$

and a new value of θ, θ_{New}, is computed from eq. (3).

Gauss-Jordan pivoting (Jennrich and Sampson, 1968) is used for the required matrix inversion in eq. (6). Tolerance (Jennrich and Sampson, 1968) is used to prevent complete inversion of $\Delta F'\Delta F$ if it is essentially singular. The stepwise regression method described in Jennrich and Sampson (1968) is used to determine the order of pivoting. An updating procedure (such as that used by Powell, 1965) could be used to reduce the number of arithmetic operations, but we chose not to do this for two reasons. First, updating is incompatible with iterative reweighting, which is needed for many maximum likelihood and robust estimation procedures. Second, in our experience when the fitting of a function is expensive, most of the cost comes from evaluating $f(\theta)$. For such functions the reduction in cost from the use of an update procedure is minor.

Like the Gauss-Newton algorithm, Dud will not converge for some functions without the use of a step shortening procedure to decrease $Q(\theta)$. Since derivatives are not used, θ_{New} is not necessarily in a "downhill" direction from θ_{p+1}. The following procedure has a good chance of producing an estimate that decreases $Q(\theta)$. Select as the new parameter vector

$$\tilde{\theta}_{New} = d\theta_{New} + (1-d)\theta_{p+1} \tag{7}$$

where d is the first member of the sequence

$$d_i = \begin{cases} 1 & i=0 \\ -(-1/2)^i & i=1,\ldots,m \end{cases} \tag{8}$$

which makes $Q(\theta_{New}) < Q(\theta_{p+1})$ if there is such a d_i. Otherwise $d = d_m$. This procedure should be used sparingly because it can use several function evaluations per iteration and Dud performs satisfactorily without it on most problems. A convenient guideline is to set m = 5 when function evaluations are not too expensive and m = 0 for a first run if they are.

In order to insure that the search does not collapse into a subplane of the parameter space, the p parameter vector differences used in each linear approximation must span the parameter space. Theoretically, if the current set of parameter vector differences spans the parameter space, the new set will span it also, if and only if the component of α corresponding to the discarded parameter vector is nonzero. Normally the new estimate

will replace θ_1 (the oldest member of the set). However, if $|\alpha_1| < 10^{-5}$ two members of the set are replaced. First, θ_i is replaced by θ_{New} where α_i is the first component of α for which $|\alpha_i| \geq 10^{-5}$. Second, so old parameter values are not retained indefinitely, θ_1 is replaced by $(\theta_1 + \theta_{New})/2$.

The p+1 starting values required by Dud are generated from one user-supplied starting value θ_{p+1}. For i=1,...,p, θ_i is computed from θ_{p+1} by displacing its ith component by a nonzero number h_i. These vectors are renumbered so that $Q(\theta_1) \leq \ldots \leq Q(\theta_{p+1})$. In most examples we have looked at, 0.1 times the corresponding component of θ_{p+1} provides satisfactory values for the h_i's.

A specific convergence criterion is not an integral part of the algorithm. However, the one that is used (and found to be satisfactory) is to stop when

$$\frac{|Q(\theta_{New}) - Q(\theta_{p+1})}{Q(\theta_{p+1})} \leq T \tag{9}$$

for five successive iterations, where T is a small positive number such as 10^{-5}. The use of this particular criterion is not important. However, the use of a convergence criterion that requires something to be satisfied for several consecutive iterations is important.

An estimate of the asymptotic covariance matrix of the parameter estimates can be obtained by approximating the Gauss-Newton result given in Jennrich (1969),

$$\hat{\Sigma} = s^2\left(\frac{df'}{d\theta}\ \frac{df}{d\theta}\right)^{-1} \tag{10}$$

where $s^2 = Q(\theta)/(n-p)$. Here $\frac{df}{d\theta}$ can be approximated by $\Delta F \Delta\theta^{-1}$ where ΔF and $\Delta\theta$ are the values used in the last iteration. The resulting estimate is

$$\tilde{\Sigma} = s^2 \Delta\theta(\Delta F' \Delta F)^{-1} \Delta\theta' \ . \tag{11}$$

4. NUMERICAL TESTING

In this section we evaluate Dud's performance on some standard test problems found in the literature. Results for a variety of popular algorithms are included to provide measures of the difficulty of the problems. Box's (1966) "equivalent function evaluations" are used to compare algorithms. In this method each evaluation of the vector $f(\theta)$, or one of its partial derivatives is counted as a function evaluation. Computations with Dud were done on an IBM 360/91 using double precision arithmetic. Unless indicated otherwise results for other algorithms are taken from the originator's paper.

4.1 Rosenbrock's Valley

This problem was first proposed in Rosenbrock (1960). The function to be minimized is $Q(\theta) = 100(\theta_1^2 - \theta_2)^2 + (1 - \theta_1)^2$. The minimum occurs at $\theta = (1.0, 1.0)'$. Iterations begin at $\theta = (-1.2, 1.0)'$. Additional starting values for Dud were computed with $h = (-.012, .01)'$.

Table 1 contains the number of equivalent function evaluations and number of iterations required to reduce $Q(\theta)$ to the indicated values. When speed is measured by the

Algorithm	Iterations	Equivalent Fun Eval.	Final Log $Q(\theta)$
Derivative-Free Least Squares			
Dud	2	5	-13.7
Peckham (1970)	NA*	12	-17.4
Powell (1965)	20	70	-8.0
Marquardt (1963)‡	NA	92	-13.6
Spiral (Jones, 1970)	NA	17	$-\infty$
Polyalgorithm (Aird, 1973)	18	100	NA
Derivative-Requiring Least Squares			
BMDP3R (Dixon, 1975)	2	9	-7.1
Shanno (1970)	21	74	NA
Myer and Roth (1972)	17	NA	-13.6
Derivative-Free General			
Powell (1964)	13	151	-9.2
Brent (1973)	47	120	-17.2
Stewart (1967)	25	169	-11.5
Q.N. (Greenstadt, 1972)	20	199	-8.4
Rosenbrock (1960)	NA	200	-8.0
Nelder and Mead (1965)	NA	150	-9.5
Derivative-Requiring General			
Fletcher and Reeves (1964)	27	NA	-8.0
Davidon (Fletcher-Powell, 1963)	18	NA	-8.0
Fletcher (1970)	39	141	NA
Greenstadt I-S (1970)	24	221	-13.4
Greenstadt I-W (1970)	33	138	-12.1
Bass (1972)	66	231	-11.3
Oren - 1 (1973)	F+	F	-
Oren - 2 (1973)	35	104	NA
Oren - 3 (1973)	29	85	NA

* NA means not available
\+ F means the algorithm failed
‡ From Jones (1970)

Table 1. Rosenbrock's Valley

number of function evaluations BMDP3R is the only real competitor to Dud. This might be expected on this problem because most of the other algorithms attempt to minimize $Q(\theta)$ along the search direction of each iteration. Since $Q(\theta)$ has a curved valley short steps are taken at each iteration, and as a result these algorithms need several iterations to get to the solution.

4.2 Box's functions

These two problems were originally described in Box (1966). Box's first response function is

$$f_1(x,\theta) = e^{-\theta_1 x} - e^{-\theta_2 x} - (e^{-x} - e^{-10x}) \tag{12}$$

and the second is

$$f_2(x,\theta) = e^{-\theta_1 x} - e^{-\theta_2 x} - \theta_3(e^{-x} - e^{-10x}) \, . \tag{13}$$

The problems were run using several different starting values. Results are found in Tables 2 and 3. Again the performance of Dud is quite good relative to its competitors. Although it failed occasionally with the partial stepping option turned off, in this mode it rather consistently outperformed the Gauss-Newton algorithm, BMDP3R, and all of the others. With the partial stepping option on, Dud never failed and outperformed all of the competition, including BMDP3R, in 6 out of 14 cases. Overall no other algorithm did as well.

	Starting Values				
Algorithm	0 0	0 20	5 0	5 20	2.5 10
Dud, m=0	F*	11	41	F	10
Dud, m=5	32	12	46	19	8
Brown & Dennis (1972), FDGN	F	F	F	F	16
Brown & Dennis (1972), FDLM	22	25	25	31	16
BMDP3R (Dixon, 1975), m=0	27	F	F	F	15
BMDP3R (Dixon, 1975), m=5	24	18	48	24	35
Shanno (1970)	46	31	32	45	35
Powell (1965)+	38	22	46	29	12
Best of Box (1966)	27	15	39	27	6
Worst of Box (1966)	96	144	231	103	109

* F means the algorithm failed
\+ From Box (1966)

Table 2. Number of Equivalent Function Evaluations for Box's 2 Parameter Function.

	Starting Values								
Algorithm	10 20 1	2.5 10 10	0 0 10	0 10 1	0 10 10	0 10 20	0 20 0	0 20 10	0 20 20
Dud, m=0	10	F*	5	11	9	14	11	12	12
Dud, m=5	10	27	5	11	21	22	9	24	24
Brown & Dennis (1972), FDGN	21	21	13	17	17	17	21	21	21
Brown & Dennis (1972), FDLM	41	33	41	17	41	93	41	61	109
BMDP3R (Dixon, 1975)	20	12	8	16	16	16	20	20	20
Powell (1965) ‡									
Best of Box (1966)	34	68	104	15	28	28	19	46	33
Worst of Box (1966)	564	281	200	313	292	350	344	608	315

* F means the algorithm failed
\+ NA means not available
‡ From Box (1966)

Table 3. Number of Equivalent Function Evaluations for Box's 3 Parameter Problem.

4.3 Trigonometric functions

The response function (Peckham, 1970 and Powell, 1965) is

$$f_i(\theta) = \sum_{j=1}^{p} (a_{ij}\sin\theta_j + b_{ij}\cos\theta_j) \qquad (14)$$

and $y_i = f_i(\theta) + e_i$; i=1,...,n. Components of θ are random numbers on $[-\pi,\pi]$, the a_{ij} and b_{ij} are random numbers on [-100,100] and the e_i are random on $[-\delta,\delta]$. Tests were run with p=5, 10, 20, δ = .1, 1.0, and 10.0, and in all cases n = 2p. Starting values differ from θ by random numbers in the interval $[-\pi/10, \pi/10]$. This description of the problem was found in Peckham (1970). It differs slightly from Powell's (1965) version in that Powell describes the a_{ij} and b_{ij} as random integers. Dud generated its p additional starting values with $h = .001\theta_{p+1}$. The numbers in Table 4 are the number of function evaluations required to locate the least squares estimate of θ to an accuracy of .0001 for each component. Multiple entries in the table were obtained by generating the problems using a different random number of sequence.

Dud's performance on these problems looks encouraging. In all cases the number of function evaluations was a small multiple of the number of parameters. The number of function evaluations increases with the size of the residuals, but this trend is no worse with Dud than with the other algorithms.

Algorithm	Number of Parameters	δ = .1	δ = 1.0	δ = 10.
Dud	5	11	13	19
		14	15	22
	10	23	23	33
		19	21	29
	20	34	43	56
		39	43	57
Peckham (1970)	5	8	18	24
	10	15	27	34
	20	26	48	55
Powell (1965)	5	17	37	33
		20	29	34
	10	26	47	78
		29	47	86
	20	42	118	175
		36	88	73

Table 4. Number of Function Evaluations for the Trigonometric Functions Problems

The problems in Table 4 are unusually rich in factors that complicate the comparison of algorithms. Since the solutions of these problems are not known, it must be assumed that they are the best estimate that the algorithm produces. The use of a convergence criterion that causes an algorithm to stop prematurely makes the algorithm's performance look better than it is. For Dud, the iterations were continued until eq. (9) with T = .00001 was satisfied for five successive iterations. The figures for Dud in Table 4 are the number of function evaluations required to get each component of θ to within .0001 of the best estimate produced by the algorithm. Several more iterations were required to satisfy the stopping rule used in the program than to locate the solution to the accuracy required in the comparisons.

Totals for Dud include the p+1 function values required for starting. It is not always clear if other authors include these in their totals. For one of the five parameter problems in Table 4, Peckham claimed that his algorithm used only eight function evaluations. If the six function evaluations required by his algorithm for starting are counted in this total, the algorithm must have reached the solution on the first or second iteration. This is rather surprising since his first iteration should be the same as an iteration with Dud, and from Dud's output it seemed unlikely that any algorithm would converge in one more iteration.

4.4 Bard's Problem #3d1

The problem (Bard, 1970) is to minimize

$$Q(\theta) = \sum_{i=1}^{8} \sum_{r=1}^{5} w_r (z_r(t_i,\theta) - y_{ri})^2 \qquad (15)$$

where the $z_r(t,\theta)$ satisfy a system of nonlinear differential equations. This is the type of problem that motivated Dud's development. Initial values for the system of equations and the data are found in Bard (1970). The convergence criterion is that each component θ_i of θ differs from its previous estimate by less than 10^{-4} $(\theta_i + .001)$. Results are found in Table 5.

Algorithm	Iterations	Equivalent Fun. Eval.
Dud	26	33
Gauss-Newton*	9 - 10	74 - 101
Marquardt (1963)*	14	114
Davidon (Fletcher-Powell, 1963)*	30 - 91	392 - 1073
ROC, IROC (Bard, 1970)*	40 - 58	350 - 548

* From Bard (1970)

Table 5. Performance of Various Algorithms on Bard's Problem.

The algorithms used with Dud to numerically integrate the system and to constrain estimates of the parameter vector to lie between the upper and lower bounds differ from those used for the other algorithms. A fourth order Runge-Kutta routine was used to perform the integrations required by Dud. A quadratic programming technique described in Ralston (1975) was used for the constraints. The algorithms used in Bard's paper require partial derivatives of the components of z with respect to components of θ. Bard used sensitivity equations for these derivatives. These equations plus the original system were integrated with a third-order variable step predictor-corrector routine. He used penalty functions for the constraints.

When speed is measured by the number of equivalent function evaluations, Dud's performance looks impressive. For this problem the actual cost of obtaining the solution depends heavily on the accuracy to which the integral of system of equations must be computed. The problem was run using various stepsizes in the integration. Stepsizes as large as 2.5 produced satisfactory results. When a stepsize of 2.5 was used, computed values of the z_r were accurate to five to eight significant digits. With this stepsize, less than two seconds of cpu time were required to obtain the solution.

The comparisons in this section can be criticized from several points of view. The examples, although they have all been canonized by the literature, seem for the most part

to be artificial. The basic criterion for comparison, equivalent function evaluations, is somewhat arbitrary and not always a relevant index. All the problems had small (in two cases zero) fitted residuals, a situation which is known to make Gauss-Newton type algorithms perform well and probably makes all the algorithms considered look artificially good. In some cases the performances recorded may depend more on details of implementation than on the basic algorithms considered.

Nevertheless such comparisons are valuable if we don't take them too seriously. They suggest that Dud is at least a competitive algorithm. This, together with its simplicity and potential for application to problem of fitting functions defined by differential equations, where function evaluation is expensive, is enough to make the algorithm attractive.

5. ACKNOWLEDGMENT

This work was partially supported by National Institutes of Health Grant RR-3 of the Health Sciences Computing Facility at the University of California at Los Angeles.

6. REFERENCES

AIRD, T.J. (1973). A computational solution of global nonlinear least squares problems. Ph.D. Dissertation. Purdue University.

BARD, Y. (1970). Comparison of gradient methods for the solution of nonlinear parameter estimation. SIAM J. Numer. Anal. 7, 157-186.

BASS, R. (1972). A rank 2 algorithm for unconstrained minimization. Math. Comp. 26, 129-143.

BEATON, A.E. and J.W. TUKEY (1974). The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data. Technometrics 16, 147-185.

BERMAN, M. and M.F. WEISS (1967). Users Manual for SAAM, U.S. Public Health Service 1703, U.S. Department of Health, Education and Welfare. U.S. Government Printing Office, Washington, D. C.

BOX, M.J. (1966). A comparison of several current optimization methods, and the use of transformations in constrained problems. Computer J. 9, 67-77.

BRADLEY, E.L. (1973). The equivalence of maximum likelihood and weighted least squares estimates in the exponential family. J. Amer. Statist. Assoc. 68, 199-200.

BRENT, R.P. (1973). Algorithms for Minimization Without Derivatives. Englewood Cliffs, N.J., Prentice-Hall, Inc.

BROWN, K.M. and J.E. DENNIS (1972). Derivative-free analogues of the Levenberg-Marquardt and Gauss algorithms for nonlinear least squares approximation. Numer. Math. 18, 289-297.

CHARNES, A., E.L. FROME and P.L. YU (1976). The equivalence of generalized least squares and maximum likelihood estimation in the exponential family. J. Amer. Statist. Assoc. 71, 169-171.

DIXON, W.J. ed. (1975). BMDP Biomedical Computer Programs. Los Angeles, University of California Press.

FLETCHER, R. (1970). A new approach to variable metric algorithms. Computer J. 13, 317-322.

FLETCHER, R. and M.J.D. POWELL (1963). A rapidly convergent descent method for minimization. Computer J. 6, 163-168.

FLETCHER, R. and C.M. REEVES (1964). Function minimization by conjugate gradients. Computer J. 7, 149-152.

GREENSTADT, J.L. (1970). Variations on variable metric methods. Math. Comp. 24, 1-22.

GREENSTADT, J. (1972). A quasi-Newton method with no derivatives. Math. Comp. 26, 145-166.

JENNRICH, R.I. (1969). Asymptotic properties of nonlinear least squares estimators. Ann. Math. Stat. 40, 633-643.

JENNRICH, R.I. and R.H. MOORE (1975). Maximum likelihood estimation by means of nonlinear least squares. Statistical Computing Section Proceedings of the American Statistical Association, 57-65.

JENNRICH, R.I. and P.F. SAMPSON (1968). Application of stepwise regression to nonlinear estimation. Technometrics 10, 63-72.

JONES, A.P. (1970). Spiral -- a new algorithm for nonlinear parameter estimation using least squares. Computer J. 13, 301-308.

MARQUARDT, D.W. (1963). An algorithm for least squares estimation of nonlinear parameters. J. SIAM 11, 431-441.

MYER, R.R. and P.M. ROTH (1972). Modified damped least squares -- an algorithm for nonlinear estimation. J. Inst. Math. Applics. 9, 218-253.

NELDER, J.A. and R. MEAD (1965). A simplex method for function minimization. Computer J. 7, 308-313.

NELDER, J.A. and R.W.M. WEDDERBURN (1972). Generalized linear models. J. Roy. Stat. Soc. A 135, 370-384.

OREN, S.S. (1973). Self scaling variable metric algorithm without line search for unconstrained minimization. Math. Comp. 27, 873-885.

PECKHAM, G. (1970). A new method for minimizing a sum of squares without calculating gradients. Computer J. 13, 418-420.

POWELL, M.J.D. (1964). An efficient method for finding the minimum of a function of several variables without calculating derivatives. Computer J. 7, 155-162.

POWELL, M.J.D. (1965). A method for minimizing a sum of squares of nonlinear functions without calculating derivatives. Computer J. 7, 303-307.

RALSTON, M.L. (1975). Dud, a Derivative-Free Algorithm for Nonlinear Regression. Ph.D. Dissertation. University of California, Los Angeles.

ROSENBROCK, H.H. (1960). An automatic method for finding the greatest or least value of a function. Computer J. 3, 175-184.

SHANNO, D.F. (1970). Parameter selection for modified Newton methods for function minimization. SIAM J. Numer. Anal. 7, 366-372.

STEWART, G.W. (1967). A modification of Davidon's minimization to accept difference approximations of derivatives. J. ACM 14, 72-83.

BIOGRAPHIES

Mary L. Ralston received a Ph.D. in biostatistics from UCLA in 1975 and is a research statistician at Health Sciences Computing Facility.

Robert I. Jennrich received a Ph.D. in mathematics from UCLA in 1960. He was a member of the Mathematics Department at the University of Wisconsin for two years. Then he held a joint appointment in the Mathematics and Biomathematics Departments at UCLA until 1975, during which time he worked on the development of the BMD programs. Presently he is a Professor of Mathematics at UCLA, where his primary interest is in statistical computing.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

IMPROVING THE APPARENT RANDOMNESS OF PSEUDORANDOM NUMBERS GENERATED BY THE MIXED CONGRUENTIAL METHOD

Peter Peskun
Department of Mathematics, York University, Downsview, Ontario, Canada M3J 1P3

ABSTRACT

This paper deals with sequences of pseudorandom numbers generated by the mixed congruential method. That is, a sequence of integers is started with a value X_0 and continued as $X_{i+1} \equiv \lambda X + \mu \pmod P$ where λ, μ, P, and X_0 are integers. The fractions $U_i = X_i/P$ or $U_i = X_i/(P-1)$ $(i=0,1,2,\ldots)$ are the derived pseudorandom numbers in the intervals $[0,1)$ and $[0,1]$, respectively. If X_0 is a "true" random number, then choices of λ and μ have been based, for example, on making small the serial correlations $\rho_s = \mathrm{Cov}(U_i, U_{i+s})/\mathrm{Var}(U_i)$ $s = 1,2,\ldots$. That is, choices of λ and μ have been made to make the sequence $U_0, U_1, U_2,\ldots$ appear random. Exact determinations of the serial correlations ρ_s have been made (Ahrens and Dieter (1971); Jansson (1964/66); Knuth (1969)) except that the evaluation of generalized Dedekind sums is involved making the necessary computations cumbersome. It is the purpose of this paper to indicate how certain subsets of the sequence $U_0, U_1, U_2,\ldots$ can each be made to consist of mutually statistically independent random variables, how simple exact expressions for the serial correlations ρ_s, $s = 1,2,\ldots$ can be obtained, and how the ρ_s can be minimized if X_0 (and in general $X_0, X_1,\ldots, X_r$ for some r) and μ are chosen randomly, and if λ is chosen appropriately.

Key words: Pseudorandom numbers; mixed congruential method; serial correlation; mutually statistically independent random variables.

1. INTRODUCTION

In all practical applications of the Monte Carlo method, we need samples of random numbers but, because of practical considerations, we are usually forced to use samples of pseudorandom numbers instead. One possible way of generating a sample of pseudorandom numbers is by the mixed congruential method.

Let us consider the stochastic process $X_0, X_1, X_2,\ldots$ with values in the set of integers $A = \{0,1,2,\ldots,P-1\}$ defined by

$$X_{i+1} \equiv \lambda X_i + \mu \pmod P, \quad i = 0,1,2,\ldots \tag{1}$$

where the two process parameters $\lambda, \mu \in A$ and $X_0 \sim U[A]$ (i.e. X_0 has the discrete uniform distribution on A). In practice, a physical method (e.g. specially constructed dice, tables of random numbers, specially constructed machines) is used to randomly generate a value of

X_0. The fractions $U_i = X_i/P$ or $U_i = X_i/(P-1)$ $(i = 0,1,2,\ldots)$ are the derived pseudorandom numbers in the intervals $[0,1)$ and $[0,1]$, respectively. With respect to the computation of the sequence $X_0, X_1, X_2,\ldots$, we obtain fast and short calculating routines by choosing $P = 2^b$ or 10^b for a binary or a decimal computer, respectively. We shall initially consider $P = 2^b$ and then deal with $P = 10^b$ later on in the paper.

2. A MIXTURE OF MIXED CONGRUENTIAL GENERATORS

Since $X_0 \sim U[A]$, one way in which the statistical sample $X_0, X_1, X_2,\ldots$ defined by eq. (1) can be made to appear random is to also have the discrete uniform distribution on A as the marginal distribution of each of the random variables $X_1, X_2,\ldots$. In order to achieve this, we see from eq. (1) that the transformation $y \equiv \lambda x + \mu \pmod{2^b}$ must be one-to-one from A onto A.

> Theorem 1. The transformation $y \equiv \lambda x + \mu \pmod{2^b}$ is one-to-one from A onto A iff $\lambda \equiv 1 \pmod 2$.

A second way in which the statistical sample $X_0, X_1, X_2, \ldots$ can be made to appear random is to have as much statistical independence among the random variables $X_0, X_1, X_2,\ldots$ as is practically possible. As a first step in achieving such, let us now consider a simple mixture of mixed congruential generators each defined by eq. (1); in particular, let us consider the new stochastic process $X_0, X_1, X_2,\ldots$ with values in A defined by

$$X_{i+1} \equiv \lambda X_i + M \pmod{2^b}, \quad i = 0,1,2,\ldots \tag{2}$$

where the process parameter $\lambda \equiv 1 \pmod 2$ and X_0, M is a random sample of size 2 on $X \sim U[A]$. In practice, a physical method will be used to randomly generate values of X_0 and M. We note that the process parameter μ in eq. (1) has been chosen randomly, i.e., $M \sim U[A]$.

> Theorem 2. The stochastic process $X_0, X_1, X_2,\ldots$ defined by eq. (2) is strictly stationary with $X_i \sim U[A]$ $(i = 0,1,2,\ldots)$. Also, the pairs of random variables $X_i, X_{i+(2m+1)2^k}$ $(i,m = 0,1,2,\ldots)$ for fixed k $(k = 0,1,2,\ldots)$ have identical joint distributions; in particular, for k = 0 and for each i,m = 0,1,2,..., the random variables X_i, X_{i+2m+1} are statistically independent.

3. SERIAL CORRELATIONS

A necessary condition for the stochastic process $X_0, X_1, X_2, \ldots$ defined by eq. (2) to appear as a random sample is that, for $i = 0,1,2,\ldots$ and $s = 1,2,\ldots$, the serial correlations

$$\rho_s = \rho_{X_i, X_{i+s}} = \mathrm{Cov}(X_i, X_{i+s})/\mathrm{Var}(X_i) = \mathrm{Cov}(U_i, U_{i+s})/\mathrm{Var}(U_i) \tag{3}$$

be small.

Theorem 3. For the strictly stationary stochastic process $X_0, X_1, X_2, \ldots$ defined by eq. (2), the serial correlations ρ_s $(s = 1,2,\ldots)$ defined by eq. (3) are minimized for $\lambda \equiv 1 \pmod 4$, and for $m = 0,1,2,\ldots$, are given by

$$\rho_{(2m+1)2^k} = \rho_{2^k} = \begin{cases} 0 & , \quad k = 0 , \\ (2^{2k} - 1)/(2^{2b} - 1), & k = 1,2,\ldots,b-1, \\ 1 & , \quad k = b, b+1, \ldots . \end{cases}$$

With respect to Theorem 3, there are a number of notes we would like to make. Firstly, the choice $\lambda \equiv 1 \pmod 4$ has previously been suggested in the literature with respect to the mixed congruential generator defined by eq. (1) but on the seemingly non-statistical basis that the generator's period (i.e., the maximum number of numbers which can be generated without repetition) be a maximum. Secondly, the desired joint discrete uniform distribution of X_0 and X_{2^k} is given by

$$\Pr(X_0 = x_0, X_{2^k} = x_{2^k}) = \begin{cases} 1/2^{2b} & , \quad (x_0, x_{2^k}) \in A \times A , \\ 0 & , \quad \text{otherwise} , \end{cases}$$

where $A \times A = \{(x_0, x_{2k}) | x_0, x_{2k} \in A\}$. For fixed b and fixed $k \leq b$, the best approximation to it is obtained when $\lambda \equiv 1 \pmod 4$ and is given by

$$\Pr(X_0 = x_0, X_{2^k} = x_{2^k}) = \begin{cases} 1/2^{2b-k} & , \quad (x_0, x_{2^k}) \in (A \times A)_k , \\ 0 & , \quad \text{otherwise} , \end{cases}$$

where $(A \times A)_k = \{(x_0, x_{2k}) | x_0, x_{2k} \in A , \; x_{2k} \equiv x_0 + 2^k x \pmod{2^b}$ for some $x \in A\}$. Finally, on the basis of minimizing the serial correlations ρ_s $(s = 1,2,\ldots)$, the choice of the process parameter $\lambda = 1$ is as good as any other choice $\lambda \equiv 1 \pmod 4$. For the choice $\lambda = 1$, the stochastic process $X_0, X_1, X_2, \ldots$ defined by eq. (2) becomes strictly "additive" and from a practical point of view can be computed quickly given the values of X_0 and M.

4. COMPUTATIONS ON A DECIMAL COMPUTER ($P = 10^b$)

For $P = 10^b$, we could go through a discussion similar to the discussion which we carried out in the previous two sections for $P = 2^b$; however, for the sake of brevity, we shall just list Theorems 4, 5 and 6 which would replace Theorems 1, 2 and 3, respectively.

Theorem 4. The transformation $y \equiv \lambda x + \mu \pmod{10^b}$ is one-to-one from A onto A iff $\lambda \not\equiv 0 \pmod{q}$ for $q = 2,5$.

The stochastic process defined by eq. (2) would be replaced by the stochastic process $X_0, X_1, X_2, \ldots$ with values in A defined by

$$X_{i+1} \equiv \lambda X_i + M \pmod{10^b}, \quad i = 0,1,2,... \qquad (4$$

where the process parameter $\lambda \not\equiv 0 \pmod{q}$ for $q = 2,5$ and X_0,M is a random sample of size on $X \sim U[A]$ generated by a physical method.

Theorem 5. The stochastic process $X_0, X_1, X_2,...$ defined by eq. (4) is strictly stationary with $X_i \sim U[A]$ $(i = 0,1,2,...)$. Also, for a given λ, the pairs of random variables X_i, $X_{i+(10m+n)2^k5^\ell}$ $(i,m = 0,1,2,...;$ $n = 1,3,7,9)$ for fixed k and ℓ $(k,\ell = 0,1,2,...)$ have identical joint distributions; in particular, for $k,\ell = 0$ and $\lambda \equiv 1 \pmod{10}$, the random variables X_i, $X_{i+10m+n}$ for each $i,m = 0,1,2,...$ and $n = 1,3,7,9$ are statistically independent. Moreover, for $\lambda \equiv 3,7,9 \pmod{10}$, the random variables X_i, X_{i+2m+1} for each $i,m = 0,1,2,...$ are statistically independent.

Theorem 6. For the strictly stationary stochastic process $X_0, X_1, X_2,...$ defined by eq. (4), the serial correlations ρ_s $(s = 1,2,...)$ defined by eq. (3) are minimized for $\lambda \equiv 1 \pmod{20}$ and $\lambda \equiv 9,13,29,33,37,49,53,57,69,$ $73,77,97 \pmod{100}$; and for $m = 0,1,2,...,$ and $n = 1,3,7,9$ are given as follows:

(i) for $\lambda \equiv 1 \pmod{20}$

$$\rho_{(10m+n)2^k5^\ell} = \rho_{2^k5^\ell} = 0, \qquad k = \ell = 0,$$

$$= (5^{2\ell} - 1)/(10^{2b} - 1), \qquad k = 0;\ \ell = 1,2,..., b-1,$$

$$= (5^{2b} - 1)/(10^{2b} - 1), \qquad k = 0;\ \ell = b, b+1,...,$$

$$= (2^{2k}\, 5^{2\ell} - 1)/(10^{2b} - 1), \qquad k=1,2,...,b-1;\ \ell=0,1,...,b-1$$

$$= (2^{2k}\, 5^{2b} - 1)/(10^{2b} - 1), \qquad k=1,2,...,b-1;\ \ell=b,b+1,...,$$

$$= (2^{2b}\, 5^{2\ell} - 1)/(10^{2b} - 1), \qquad k=b,b+1,...;\ \ell=0,1,...,b-1,$$

$$= 1, \qquad k,\ell = b,b+1,...;$$

(ii) for $\lambda \equiv 13,33,37,53,57,73,77,97 \pmod{100}$,

$$\rho_{(10m+n)2^k5^\ell} = \rho_{2^k5^\ell} = 0 \quad , \quad k = 0;\ \ell = 0,1,2,\ldots,$$

$$= 3/(10^{2b} - 1) \quad , \quad k = 1;\ \ell = 0,1,2,\ldots,$$

$$= (2^{2k}\, 5^{2\ell+2} - 1)/(10^{2b} - 1), \quad k = 2,3,\ldots,b-1;\ \ell=0,1,\ldots,b-2,$$

$$= (2^{2k}\, 5^{2b}-1)/(10^{2b} - 1) \quad , \quad k = 2,3,\ldots,b-1;\ \ell=b-1,b,\ldots,$$

$$= (2^{2b}\, 5^{2\ell+2} - 1)/(10^{2b} - 1), \quad k = b,b+1,\ldots;\ \ell=0,1,\ldots,b-2,$$

$$= 1 \quad , \quad k = b,b+1,\ldots;\ \ell=b-1,b,\ldots;$$

(iii) for $\lambda \equiv 9,29,49,69 \pmod{100}$,

$$\rho_{(10m+n)2^k5^\ell} = \rho_{2^k5^\ell} = 0 \quad , \quad k = 0;\ \ell = 0,1,2,\ldots,$$

$$= (2^{2k}\, 5^{2\ell+2} - 1)/(10^{2b} - 1), \quad k = 1,2,\ldots,b-1;\ \ell=0,1,\ldots,b-2,$$

$$= (2^{2k}\, 5^{2b} - 1)/(10^{2b} - 1) \quad , \quad k = 1,2,\ldots,b-1;\ \ell=b-1,b,\ldots,$$

$$= (2^{2b}\, 5^{2\ell+2} - 1)/(10^{2b} - 1), \quad k = b,b+1,\ldots;\ \ell =0,1,\ldots,b-2,$$

$$= 1 \quad , \quad k = b,b+1,\ldots;\ \ell=b-1,b,\ldots .$$

On the basis of both statistical independence and the magnitudes of serial correlations, we would prefer the values $\lambda \equiv 13,33,37,53,57,73,77$ or $97 \pmod{100}$ for the process parameter λ.

5. A GENERALIZATION OF THE GENERATION PROCESS

As a generalization of the pseudorandom number generation process, we shall consider the strictly stationary stochastic process $X_0, X_1, X_2,\ldots$ with values A defined by

$$X_{i+r} \equiv \lambda X_i + M \pmod{P}, \qquad i = 0,1,2,\ldots \tag{5}$$

where the process parameter $\lambda \equiv 1 \pmod 4$ if $P = 2^b$ and $\lambda \equiv 13,33,37,53,57,73,77$ or $97 \pmod{100}$ if $P = 10^b$; and $X_0, X_1, X_2,\ldots, X_{r-1}$, M is a random sample of size r+1 on $X \sim U[A]$. In practice, a physical method will be used to randomly generate values of $X_0, X_1, X_2,\ldots, X_{r-1}$, M. We have constructed a new stochastic process $X_0, X_1, X_2,\ldots$ by selecting in turn the random variables from r statistically independent stochastic processes each defined by eq.(2) for $P = 2^b$ and by eq. (4) for $P = 10^b$.

From the definition of the new stochastic process $X_0, X_1, X_2,\ldots$, we see that each (r+1)-tuplet $X_i, X_{i+rn_1+1}, X_{i+rn_2+2}, \ldots, X_{i+rn_{r-1}+r-1}, X_{i+r2n_r+r}$ $(i,n_1,n_2,\ldots,n_r=0,1,2,\ldots)$ consists of mutually statistically independent random variables and all the possible pairs, triplets, ..., and r-tuplets, each consisting of mutually statistically independent random

variables, are obtained as subsets of them. As a consequence, we see that the serial correlations

$$\rho_{rm+1} = \rho_{rm+2} = \ldots = \rho_{rm+r-1} = 0, \qquad m = 0,1,2,\ldots \tag{}$$

while for m = 0,1,2,...

$$\rho_{r(2m+1)2^k} = \rho_{r2^k} = \begin{cases} 0 & , \quad k = 0 \ , \\ (2^{2k}-1)/(2^{2b}-1), & k = 1,2,\ldots,b-1, \\ 1 & , \quad k = b, b+1,\ldots, \end{cases}$$

if $P = 2^b$; and for m = 0,1,2,..., and n = 1,3,7,9

$$\begin{aligned} \rho_{r(10m+n)2^k 5^\ell} = \rho_{r2^k 5^\ell} &= 0 && , \quad k = 0;\ \ell = 0,1,2,\ldots, \\ &= 3/(10^{2b}-1) && , \quad k = 1;\ \ell = 0,1,2,\ldots, \\ &= (2^{2k}\,5^{2\ell+2}-1)/(10^{2b}-1), && k = 2,3,\ldots,b-1;\ \ell=0,1,\ldots,b \\ &= (2^{2k}\,5^{2b}-1)/(10^{2b}-1) && , \quad k = 2,3,\ldots,b-1;\ \ell=b-1,b,\ldots \\ &= (2^{2b}\,5^{2\ell+2}-1)/(10^{2b}-1), && k = b,b+1,\ldots;\ \ell=0,1,\ldots,b-2 \\ &= 1 && , \quad k = b,b+1,\ldots;\ \ell=b-1,b,\ldots\ . \end{aligned}$$

6. REFERENCES

AHRENS, J., DIETER, U. (1971). An exact determination of serial correlations on pseudo-random numbers. Numer. Math., 17, 101-123.

JANSSON, B. (1964). Autocorrelations between pseudo-random numbers. BIT, 4, 6-27.

JANSSON, B. (1966). Random Number Generators. Stockholm: Almqvist and Wiksell.

KNUTH, D.E. (1969). The Art of Computer Programming. Vol. 2./Semi-numerical Algorithms. Reading, Mass.: Addison-Wesley Co.

BIOGRAPHY

Peter Peskun received a Ph.D. in statistics from the University of Toronto in 1970 and is an Associate Professor in the Department of Mathematics, York University.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

ADVANCED SPSS CROSSTABS: FITTING MODELS TO CATEGORICAL DATA

Ervin H. Young
Institute for Research in Social Science
University of North Carolina, Chapel Hill, NC 27514

ABSTRACT

This is a progress report of an effort to integrate into the SPSS CROSSTABS procedure both weighted least squares and maximum likelihood techniques for fitting models to categorical data. It includes a partial draft of a users' manual.

Key words: Categorical data; linear models; loglinear models; maximum likelihood; statistical package; SPSS; weighted least squares.

1. INTRODUCTION

The art of analyzing categorical data has advanced rapidly in recent years. Two lines of research have proved especially fruitful. One approach has been to fit hierarchical linear models to the logarithms of the joint probabilities using maximum likelihood estimation (MLE). These models are subsets of the fully crossed design; the term hierarchical denotes the fact that if any particular interaction effect is included in the model, then the model also includes the interactions of all subsets of the set of variables involved in the first interaction. Researchers prominently associated with this approach include Goodman, Bishop, Fienberg, and Holland.

A second approach has been to fit linear models to conditional probabilities (or to their logarithms, or to logits) using weighted least squares (WLS). This approach has been pursued principally by Koch and his associates.

A variety of self-contained computer programs is available for one approach or the other. In addtion, one of the major statistical packages already has a procedure for MLE of hierarchical loglinear models (BMDP3F), and a WLS procedure is anticipated in SAS in the near future.

No single currently available program offers both approaches, however. This presentation is a progress report of the effort to integrate both approaches into the integer-mode CROSSTABS procedure of SPSS. A partial draft of a users manual for the new features is attached. Comments and suggestions will be appreciated.

Progress to date consists of implementing the ECTA program computational and output features in CROSSTABS. ECTA is a self-contained program for the log-linear MLE approach,

written by Leo Goodman and Robert Fay. The second stage of the project will consist of implementing NONMET, a program for weighted least-squares analysis which was adapted by Herbert M. Kritzer from programs originally written by Gary Koch and his associates. The third step will be to integrate their output features with a single output format.

The users manual will of course have to be expanded. An introductory section or sections will have to be written. Descriptions of the output, and the options, and limitation will have to be documented.

Suggestions from readers of this paper will be appreciated. Work on the implementatio of NONMET is already well along. Completion of all remaining work is scheduled for March, 1978.

2. ADVANCED CROSSTABS: 2-WAY TO 8-WAY CROSSTABULATION OF INTEGER-MODE CATEGORICAL DATA, AND FITTING OF A MODEL TO THEM

Subprogram CROSSTABS enables the user to compute two-way to eight-way joint frequency distributions, and to fit models to the tables thus obtained, using either maximum-likelihood or weighted least squares techniques. The advanced features of CROSSTABS, however, operate only in the integer mode. Furthermore, each variable must actually take on every valu within the range specified for it in the VARIABLES=list (see below).

2.1 Required components of the Advanced CROSSTABS procedure card. The specification field of the CROSSTABS procedure card has a fairly large number of portions or segments, bu of these, only three are required. Of those three, the first two are almost identical to the two segments of the CROSSTABS procedure card for integer mode. The first part, the VARIABLES=list, specifies the variables to be used in building the tables, and the range of their values. The second part, the TABLES=list, specifies the tables to be generated. These two parameters have the general form:

1	16
CROSSTABS	VARIABLES = variable list / TABLES = tables list /

Both of these parameters are discussed in ample detail in sections 16.2.1 through 16.2.3 of the SPSS manual (second edition). The advanced features depart from that discussion in only one respect: where formerly the TABLES= parameter could be specified only once, now it may be specified up to 20 times. (The limitation of 20 on the total number of primary table requests still obtains. The primary table requests may now be distributed among TABLES= parameters at the user's discretion.) The third required parameter, the ESTIMATE parameter, requests the type of parameter estimation technique, either maximum likelihood or weighted least squares, used to fit a model to the data. The appearance of this parameter invokes the advanced features of CROSSTABS. Without the ESTIMATE parameter, CROSSTABS works precisely as documented in chapter 16 of the SPSS manual. In particular, if the ESTIMATE parameter is omitted, but some of the other parameters discussed below are included, then those other parameters will be unrecognizable symbols, causing premature termination of the run. Also, if the advanced features of CROSSTABS are invoked, the first ESTIMATE parameter must immediately follow the first TABLES= list.

Since some of the optional procedure specification segments are relevant only to maximum likelihood estimation, while others refer only to weighted least squares estimation, and

still others have different meanings, depending on which type of analysis is requested, the remaining procedure card segments will be treated separately in the following discussion.

2.2 Requesting a maximum likelihood analysis. To request a maximum likelihood analysis, one must first specify ESTIMATE = MLE, as follows:

```
1          16
           ESTIMATE = MLE /
```

If the user specifies only the three parameters just discussed (a VARIABLES = list, a TABLES = list, and ESTIMATE = MLE), CROSSTABS will estimate only the saturated log-linear model, with standard effects for all variables, and the problem will be interpreted as a "no-factor, multi-response" problem in the sense described in section 1.x. To specify other models or other types of effects, or to specify that the variable named first in each table request of the TABLES = list is to be treated as a dependent variable, the user must use some of the optional control card segments described below.

2.2.1 Adjusting cell frequencies with ADDCELL. For a variety of reasons, some authorities recommend that a fixed constant, usually 0.5, be added to the observed frequency in each cell when estimating the saturated loglinear model. This option is available to the user through the ADDCELL parameter. To add 0.5 to each observed cell frequency, specify ADDCELL = 0.5/. To cause some other value to be added to each cell, the user should specify

```
           16
           ADDCELL = value /
```

ADDCELL = 0. is the default for each TABLES = list. Once the ADDCELL parameter has been specified for a particular TABLES = list, the specified value will remain in force for succeeding TABLES = lists, until it is suppressed by specifying ADDCELL = 0., or until it is changed by ADDCELL = some other value.

2.2.2 Using the COMPARISONS parameter to specify the comparisons to be made among the categories of each variable. One way of drawing comparisons among the categories of a variable is discussed in this manual in section 21.2.1, "Dummy Variables: Coding and Interpretation." In this section, we will review dummy variables briefly, and then go on to discuss other types of comparisons that can be made.

In the example of section 21.2.1, an unnamed dependent variable Y is being regressed on religion, which has been coded into four categories: Protestant, Catholic, Jewish, and Other. From these four categories, three dummy variables have been created: variable D1 has value 1 for Protestants, and so on. The category for Others is called the reference category, since it has no dummy variable of its own, and since Others have the value 0 on all three dummy variables. Note that the reference category here is the highest-numbered category. To specify the creation of dummy variables for religion, with the highest-numbered category as the reference category, in a CROSSTABS analysis, the user would specify a COMPARISON parameter as follows:

```
      16
      COMPARISON = RELIGION (DUMMY, HI) /
```

Another type of comparison is traditional in the analysis of variance for balanced designs, where each class of the independent variable has the same number of cases. This type of comparison uses variables that are similar to dummy variables except that the reference category is coded -1 in each of the variables. We have called this the BALANOVA comparison for want of a better unambiguous term; its values are tabulated for the religion example in Table 1. To analyze the data from the religion example as though the comparison variables

Table 1: Scores for BALANOVA Comparison Variables for Religion

Types of Cases	B1	B2	B
Protestant	1	0	
Catholic	0	1	
Jewish	0	0	
Other	-1	-1	-

shown in Table 1 had been used, one would specify COMPARISON = RELIGION (BALANOVA, HI).

This type of comparison is useful when each category of the independent variable has the same number of cases, for in that situation the regression coefficient for each comparison variable is equal to the difference between the mean of the dependent variable in the corresponding category of the independent variable and the overall mean of the dependent variable. The coefficients of BALANOVA comparison variables lose this neat interpretation when the marginal distribution of the independent variable is not uniform.

When the marginal distribution of the independent variable is not uniform, there is no generally applicable set of comparison variables that can be used to give the deviations of each category's effect from the mean effect. The comparison variables that will work for one marginal distribution will not work for another. Nonetheless, it is possible to specify that the effects for each category should be printed out as the deviation of that category's effect from the mean effect. To do this for the religion example, one would code COMPARISON = RELIGION (DEVIATION). When deviation comparisons are specified, an effect coefficient is printed for each category of the independent variable. (Any one of them is redundant, since they sum to zero.) Thus there is no need to specify a reference category with DEVIATION comparisons, and the reference category will be ignored if it is specified.

Since deviation comparisons are the usual choice of authors of published social science research (as of this writing) using this technique, they are the default when maximum likelihood estimation is specified. If no comparisons are specified for a particular variable, DEVIATION comparisons will be computed.

In some cases, even though a variable is categorical, its categories represent real numbers. When the categories represent equally-spaced real numbers, polynomial comparisons can be generated by specifying COMPARISON = variable name (POLYNOMIAL). For instance, if the variable NCHILDRN has five categories representing 0, 1, 2, 3, and 4 children respectively, the specification COMPARISON = NCHILDRN (POLYNOMIAL) will generate polynomial comparisons for linear, cubic, quadratic, and quartic effects.

Still another type of comparison is called the Helmert comparison, after the statistician who first suggested it. With Helmert comparisons, the first category is compared with

all the rest, taken together; then the second category is compared with the third through the last, all taken together; the third is compared with the fourth through the last, and so on, until the next-to-last category is compared with the last. Alternatively, one can compute reverse Helmert comparisons by starting with the last category and working backward to the first. Both types of comparisons are especially useful when the categories are measured on an ordinal scale. Suppose now that NCHILDRN has four categories, representing no children, 1 child, 2 children, and 3 or more children respectively. COMPARISON = NCHILDRN (HELMERT, LO) specifies Helmert comparisons for NCHILDRN, while COMPARISON = NCHILDRN (HELMERT, HI) specifies reverse Helmert comparisons.

Finally, if the user desires a type of comparison not available through the keywords, it is possible to specify the keyword SPECIAL and enter the comparison matrix directly. To see how this works, consider the religion example used above. One could enter the dummy variable matrix directly as follows:

COMPARISON = RELIGION (SPECIAL, 1, 0, 0, 0 / 0, 1, 0, 0 / 0, 0, 1, 0)

Here we have entered the matrix of values of variables D1, D2, and D3 from Table 21.2. Note that this matrix has as many rows as the underlying variable has categories, but one column fewer. It has been entered by columns, not by rows. The size of the matrix is fixed, therefore, by the number of categories of the variables; the order of its entry is fixed by the program. The alignment of the columns one above the other in the example, and the use of slashes to separate them, is purely optional. The program ignores slashes in the matrix specification. Their use is highly recommended, however, to improve control card readability (and therefore to make the user's eyeball check more effective).

2.2.3 Specifying the model to be fit to the data with the MODEL = parameter. A single model is specified by a list of marginals that are to be fit to the data. This specification has the form

MODEL = (marginals list) (marginals list) ...(marginals list)

where a "marginals list" is either the name of a single variable, or several variable names separated by asterisks. For example, consider a five-way table created by TABLES = A BY B BY C BY D BY E. To test the hypothesis that all of the relationships in this table can be adequately summarized in the table of A by B by C, the table of B by D and the marginal distribution of E, we would write MODEL = (A*B*C) (B*D) (E).

In the early stages of the analysis of an unfamiliar table, one will often not want to test hypotheses as specific as the example just given. One may want to estimate the coefficients in the saturated model (the model that fits all effects up to the highest possible order of interaction). For the saturated model, it is only necessary to specify MODEL = SATURATED. One might also want to fit first the model consisting of all (K-1)-way subtables (where the table being analyzed has K dimensions), then the model consisting of all (K-2)-way subtables, and so on down to the model consisting of all 2-way subtables, then the model consisting of all the 1-way marginals, and the model that hypothesizes equal frequencies in every cell. To fit all of these models, beginning with the saturated model, specify MODEL = BYLEVEL.

2.2.4 Controlling the iterative proportional fitting with MAXITER and DELTA. The iterative process that is used to estimate the cell frequencies, based on the specified model, is set to stop after 25 steps, or after the estimated frequencies at any one step are all within .01 of the corresponding frequencies at the previous step, whichever comes first. These limits should be adequate for the vast majority of data situations. In exceptional cases, however, the user can modify the maximum number of iterative steps by specifying

MAXITER = number, and can modify the maximum discrepancy by specifying DELTA = value.

2.2.5 Summary of the CROSSTABS control card for maximum likelihood analysis. The general form of the CROSSTABS control card for maximum likelihood analysis is as follows:

```
1                16
CROSSTABS        VARIABLES = variable list/ TABLES = tables list/

                 ESTIMATE = MLE/ ADDCELL = value/

                   { variable list (DUMMY, LO )                  }
                   { variable list (DUMMY, HI )                  }
                   { variable list (BALANOVA, LO )               }
                   { variable list (BALANOVA, HI )               }
COMPARISON =       { variable list (DEVIATION)                   }  /   MODEL = { model specification }  /
                   { variable list (POLYNOMIAL)                  }              { or                    }
                   { variable list (HELMERT, LO )                }              { SATURATED             }
                   { variable list (HELMERT, HI )                }              { or                    }
                   { variable list (SPECIAL, matrix)             }              { BYLEVEL               }

                 DELTA = value/ MAXITER = number
```

2.3 Requesting a weighted least squares analysis. To request a weighted least square analysis, one must first specify ESTIMATE = WLS, as follows:

```
      16
      ESTIMATE = WLS
```

If the user specifies only the VARIABLES = and TABLES = parameters, in addition to ESTIMATE = WLS, CROSSTABS will estimate the saturated linear model, with the variable named first in the table specification interpreted as the dependent, or response, variable. The highest-numbered category of the response variable will be taken to be the reference category. Effects of the independent variables will be computed using dummy comparisons with the highest-numbered category of each independent variable omitted, as described in section 2.2.3, above. Separate contrasts will be computed for each of the effect parameters, but not for any combination of parameters. To specify another response function, other models, other ways of computing effects, and so on, it is necessary for the user to code additional parameters on the CROSSTABS procedure card, described below.

2.3.1 Adjusting cell frequencies with the ZEROCELL parameter. Grizzle, Starmer, and Koch (1969:491) recommend that in cells in which the observed frequency is zero, the zeroes be replaced by the quantity 1/r, where r is the number of categories of the dependent variable. This substitution will be made automatically by subprogram CROSSTABS when weighted least-squares analysis is requested. To specify the substitution of some other number for observed zeroes, the user should include the ZEROCELL parameter, as follows:

```
      16
      ZEROCELL = n
```

Where n is the number to be substituted. In particular, to suppress this substitution, code ZEROCELL = 0.

2.3.2 Specifying one of the standard response functions for weighted least-squares analysis. Subprogram CROSSTABS allows the user somewhat greater flexibility in specifying the response function for weighted least-squares estimation than for maximum-likelihood estimation. Here, the response function is not uniquely determined as was the case with maximum-likelihood estimation. With ESTIMATE = WLS, the user may specify either linear, log-linear, or logistic response using the RESPONSE subparameter. The RESPONSE parameter has the general form

```
16                 { LINEAR    }
                   {   or      }
RESPONSE =         { LOGLINEAR }
                   {   or      }
                   { LOGISTIC  }
```

2.3.3 Identifying the dependent variable. In the current version of CROSSTABS, the first dimension of the table is taken to define the dependent variable for weighted least squares analysis. For example, the table specification

```
TABLES = X Y Z BY A BY B BY C BY D
```

specifies that X, Y, and Z are to be taken in turn as the dependent variables.

2.3.4 Specifying COMPARISONs and MODELs. The comparisons of the categories of the independent variables using weighted least squares are specified precisely like comparisons for maximum likelihood analysis, except that DEVIATION comparisons are not available. Models are also specifiable in the same way, where fully crossed hierarchical designs are being employed. WLS estimation, however, offers the opportunity of fitting non-hierarchical models, and in particular of including nested and contingent effects in those models.

In a non-hierarchical model, including in a model the interaction of a set of variables does not automatically cause the inclusion of the main effects of those variables and the interaction of all subsets of them. Consider this example: suppose we have opinion on some issue cross-tabulated by several variables, including education and income. Then a hierarchical model including the effects of education, of income, and of their interaction may be specified as follows:

```
CROSSTABS    VARIABLES = OPINION EDUCATION INCOME (1, 2). . . /
             TABLES = OPINION BY EDUCATION BY INCOME BY . . . /
             ESTIMATE = WLS /
             MODEL = (EDUCATION * INCOME) . . . /
```

To specify the same model in a non-hierarchical fashion, it would be necessary instead to specify

```
MODEL (NH) = (EDUC) (INCOME) (EDUC * INCOME) . . . /
```

with all other parameters remaining unchanged.

Now suppose further that the other variables in the table of the preceding example ar race (1 = white, 2 = black) and sex (1 = male, 2 = female), and suppose that an initial ru with a saturated model had shown a significant interaction at the highest level; in other words, the education by income by race by sex interaction effect was statistically signifi cant. One might then decide to examine the effects of income, education, and their intera tion separately within the cells of the race by sex subtable. In this case, one would spe ify

```
MODEL (NH) = (EDUC WITHIN (RACE * SEX)) (INCOME WITHIN (RACE * SEX)) . . .
```

to get the effects of income and education nested within the sub- table of race by sex. Finally, suppose that examination of the output from the nested moc suggested that the education effect really was about the same for black males and black an white females, but was different for white males; and that the income effect was significa only for males, but was much stronger for white males. To test that hypothesis, one could fit a model of contingent effects, as follows:

```
MODEL (NH) = (EDUC WHEN (RACE EQ 1 AND SEX EQ 1))
             (EDUC WHEN (RACE EQ 2 OR SEX EQ 2))
             (INCOME WHEN (RACE EQ 1 AND SEX EQ 1))
             (INCOME WHEN (RACE EQ 2 AND SEX EQ 1)) / . . .
```

Thus the user has much more latitude in specifying a model for weighted least squares estimation than for maximum likelihood estimation.[1] First of all, a model may be specifie hierarchically, just as with MLE estimation, where each effect specified automatically gen erates a whole family of main effects and lower-order interaction effects. (Of course, if the response is not LOGLINEAR, then the name of the dependent variable will not appear in the model description.) Models may also be specified non-hierarchically, so that the spec ification of an effect generates only the effect specified. When models are specified in this way, the effects may optionally be nested within subtables of the full table, or may be contingent on the truth of some logical proposition about the cells of the table. Thus a model specification may have the form

```
(marginals list) (marginals list) . . .
```

for hierarchical models, or the form

```
effect [WITHIN (subtable)] [WHEN (logical expression)]
effect [WITHIN (subtable)] [WHEN (logical expression)]
. . .
```

for non-hierarchical models.

The logical expressions that are permitted here have the same form as those that are permitted in the IF statement of SPSS, but their use is much more restricted here. In particular, a relation in a logical expression here must have the form

```
variable name    relational operator    value
```

[1]This difference in versatility is not a result of inherent differences between the two statistical estimation methods, but rather of differences between the computational algorithms used in advanced CROSSTABS. Direct maximum likelihood estimation algorithms with the same latitude as the weighted least squares algorithm implemented here do exist, but they are prohibitively slow.

In other words, each relational operator must be preceded by a single variable name, and must be followed by a single value of that variable. As with the IF statement, if a variable name is omitted, the last one mentioned will be inferred; if both variable name and relational operator are omitted, the last-mentioned of each of them will be inferred. Also be aware that a variable name may be used in only one part of each effect description within a model specification; it may be used in the "effect" part, or in naming the "subtable," or in the "logical expression", but not in any two of them.

2.3.5 Identifying variables using dimension numbers in MODEL = parameters. Model specifications may, at the user's option, use the dimension numbers of variables rather than their names. Consider the earlier example, where we had

```
TABLES = OPINION BY INCOME BY EDUC BY RACE BY SEX/
MODEL (NH) = (EDUC WITHIN (RACE * SEX)) (INCOME WITHIN (RACE * SEX))
```

We could just as well have written

```
TABLES = OPINION BY INCOME BY EDUC BY RACE BY SEX/
MODEL (NH) = (3 WITHIN (4 * 5)) (2 WITHIN (4 * 5))
```

2.3.6 Specifying contrasts. Using the CONTRASTS = parameter, it is possible to test the statistical significance of any one of the effect parameters taken separately, or of any set of them taken together. It is also possible to answer questions of the form, "Is this effect equal to that one?" or "Is this effect equal to twice that one?" and so on. This subject was covered in greater detail in Section 1.x, above. The CONTRASTS = parameter may be used to test the significance of each of the model parameters, taken separately in sequence, by specifying CONTRASTS = EACH. To test the significance of each parameter separately, and in addition to test the significance, taken together, of all of the parameters whose separate chi-squared value fall below a certain critical value, specify CONTRASTS = ALLBELOW (value), where the critical value, in parenthesis, immediately follows the word ALLBELOW.

For any other set of contrasts, it is necessary first to code CONTRASTS = MATRIX(c), where c is the number of effects, or linear combinations of effects, being set to zero simultaneously, and second to supply the contrast matrix following the READ MATRIX card. See Section 2.5, below, for details.

2.4 Summary of CROSSTABS control card for weighted least-squares analysis.

```
1        16
CATFIT   VARIABLES = variables list/ TABLES = tables list/
         ESTIMATE = WLS/ ZEROCELL = value/

RESPONSE = { LINEAR | or | LOGLINEAR | or | LOGISTIC } /

COMPARISON = { variable list (DUMMY, LO|HI)
             | variable list (BALANOVA, LO|HI)
             | variable list (POLYNOMIAL)
             | variable list (HELMERT, LO|HI)
             | variable list (SPECIAL, matrix) } /
```

```
MODEL = { model specification }
        { or                  } /
        { BYLEVEL             }
        { or                  }
        { SATURATED           }

CONTRASTS = { EACH             }
            { or               }
            { ALLBELOW (value) } /
            { or               }
            { MATRIX (c)       }
```

2.5 Special conventions for table and matrix input to subprogram CATFIT. The user ma optionally read in the table itself and as well as contrast matrices following the READ MA-- TRIX card. The table and matrices appear in the order and in the formats indicated in Tabl 2.1.

Table 2.1: Table and Matrix Input to Subprogram CATFIT

Item	format	Shape	read by	how requeste
Tables	8F10.0	Vector		Option 1
C-matrix	16F5.0	matrix	row	CONTRASTS = MATRIX(c)

2.6 Options for subprogram CATFIT.

1. Read the table

2. Write the table on FT09F001.

3. ACKNOWLEDGMENTS

The source code of Leo A. Goodman's ECTA program, which he generously placed in the public domain, has been borrowed freely. We are grateful also to Herbert M. Knitzer for the donation of the source code of his program NONMET, parts of which were adapted from still earlier programs written by Gary G. Koch and his students. The Institute for Research in Social Science has been more than generous in committing computer time and the author's work time to this project. Bonita Samuels and Vonda Hogan typed the manuscript swiftly and accurately.

4. REFERENCES

GRIZZLE, J. E., STARMER, C. F., and KOCH, G. G. (1969). The analysis of categorical data by linear models. Biometrics, 25, 489-504.

BIOGRAPHY

Ervin H. Young received an M.S. in mathematics from Rensselaer Polytechnic Institute in 1966. In 1973, he received the M.A. in sociology from UNC-CH, where he is currently a candidate for the Ph.D. He has been employed as a systems analyst at the Research Triangle Institute and at the University of North Carolina. He is currently employed as a statistician at the Institute for Research in Social Science, University of North Carolina at Chapel Hill.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

GENERAL CRITERIA AND CONSIDERATIONS FOR THE EVALUATION OF TIME SERIES PROGRAM PACKAGES AND LIBRARIES

Herbert T. Davis
Sandia Laboratories, Albuquerque, New Mexico 87115

ABSTRACT

The general criteria for statistical software packages is discussed in application to time series software.

1. INTRODUCTION

Since the Section on Statistical Computing of the American Statistical Association formed the Committee on Evaluation of Statistical Program Packages in 1973, there has been considerable interest and activity centered around establishing the desirable features of a statistical software package. A recent article documenting general criteria for packages is that of Francis, Heiberger and Velleman (1). In this document we propose some criteria and considerations for the more specific needs of evaluating computing software for time series analysis.

The computing problems of time series algorithms stand quite distinct from the mainstream of statistical computing for several reasons. One immediately apparent reason is the variation in sample sizes encountered in time series problems. Where as a sample size of 50 to 100 may be very adequate for most statistical analyses, here it is insufficient; but a sample size of 500,000 would not shock anyone active in the field. The wide range and magnitude of sample sizes encountered make issues of speed and accuracy important considerations, as well as generally preventing any one algorithm from being even nearly optimal in every situation. Another problem specific to time series is the distance between alternative approaches to the analysis of a given set of data. Whereas everyone would recognize a factor analysis problem and treat it as such, with some dissention perhaps on the method of rotation, time series analysts are immediately split between "Frequency Domain" and "Time Domain" approaches. Even within these broad categories there is apt to be differences on such issues as window shape or differencing. More than the individual preferences though, certain fields of application by their nature dictate different approaches.

The large number of alternative methods of analysis in time series gives rise to a third difference. Since no package can be sufficiently versatile to encompass all algorithms used without being a storage nightmare, packages must either be specialized or the routines in a library (a collection of subroutines which require user-written main routines and can be retrieved from mass storage when needed). The criteria discussed by Francis, Heiberger and Velleman tend to apply more closely to "packages" than to libraries, so some additional detail for libraries is included in this report.

In this document, we will use the outline provided by Francis, Heiberger and Velleman since the criteria established in that report applies to statistical computing in general and hence to time series computing in particular. For completeness, all of the criteria in that report will be repeated; but to prevent excessive redundancy, we will elaborate on those aspects more specific to time series analysis. The brevity given to any topic is therefore not to indicate any lack of importance.

2. USER INTERFACE

2.1 User's Documentation. User documentation for time series packages, as in general, needs to be on two levels: a novice document and an advanced document. However, perhaps

more in time series than elsewhere, several types of novice must be identified and reached: the time series, statistics and computer novice; the time series and computer novice knowledgeable of statistics; and finally the computing novice knowledgeable of time series analysis. Recently several text books have been published with associated software, in which case the text serves also as user documentation.

The advanced manual has some unusual needs for time series software. The large sample sizes encountered make punch cards sometimes an inefficient means of data storage or input. Hence the capabilities for reading tape input, together with options for storage format, must be treated clearly in the text, and not briefly in an appendix. Also, the variety of algorithms for accomplishing the same end result need discussion with some clear guidelines for their use (not criteria such as "with large sample sizes," but with input to what is "large").

2.2 Control Language and Output. Criteria for control language are discussed in (1). An important addition to their considerations is criteria for libraries. The calling sequence for a subroutine can be very clearly documented with comment cards, which is quite handy since the programmer usually has a source listing. However documented, though, several items must be included. In addition to the actual calling sequence, a clear explanation must be given to the values for control options, to the nature of the variables in the calling sequence (real, double precision, complex, etc.) and to the sizes needed for arrays.

2.3 Data Structures. Much of the data available for analysis is part of a larger data structure maintained by a Data Base Management System. The large sample sizes dealt with in time series analysis make the interface of the packages with data structures more important. This is not always a simple manner since most DBMS's are written in languages such as COBOL while the scientific subroutines used to build a time series package are typically written in languages such as ALGOL, BASIC or FORTRAN. Any trend towards interactive or semi-interactive packages certainly complicates this problem.

The problem of missing values mentioned in (1) is also very important in time series analysis. There is in general even less agreement here on how to handle missing values than in the rest of statistics.

2.4 Graphics. The use of graphics is of even greater importance in time series analysis. Whereas line printer plots are adequate to spot trends in residuals or patterns in factor loadings, high resolution plots are needed to differentiate such subtle differences as between spectral peaks and side lobe effects. Hence in addition to the criteria given for graphics in (1), the issue of "graphics portability" emerges as a very difficult and important problem.

True graphics portability is an extremely difficult achievement as graphics software differs from device to device. A higher level graphics language must either be used or contained as a part of the package.

2.5 Cost. The problems of accuracy verses running time have been discussed before. There are, however, additional cost considerations that should be discussed at least in the appendix of the advanced manual. First would be the speed considerations for input and output. For example, very large series are typically analyzed by segmenting the series. While this method is memory efficient, it is I/O inefficient. Information should be available to ascertain the trade off's as well as information such as buffer size to help select the most optimal segment size. Another cost consideration for large jobs is the overlay structure of the package and how to use it in sequencing commands to minimize swapping.

2.6 Audience and Pedagogy. The specialized nature of many time series packages makes these considerations even more noteworthy. For example, a filtering routine may be seriously out of place in a strictly time domain oriented package, and hence only wasted storage space.

3. STATISTICAL EFFECTIVENESS

3.1 Versatility and Accuracy. These considerations are discussed in (1), and as previously noted they are of increased importance for time series analysis. Even when information on speed and numerical accuracy is not available, careful statements of what algorithms are used is absolutely mandatory.

4. IMPLEMENTATION

4.1 Programmer's Documentation. Many of the things normally relegated to the programmer's document (things useful to the "keeper of the package" at an installation) have been moved to the advanced user manual in previous paragraphs. It is still important that adequate information exist to allow changes necessitated by local pecularities in an operating system in order to make the package operate.

4.2 Extensibility. Time series analysis continues to be a rapidly developing and growing body of knowledge. The last two decades have seen several major revolutions in approach to analyzing a time series. Consequently, if a package is not easily extended, it suffers early obselescence.

4.3 Portability and Source Language. These considerations are discussed in (1).

5. DISCUSSION

The criteria listed are obviously idealistic in the sense that the "perfect time series package," (TUTTIPACK) optimal for all environments and situations is not possible. These criteria therefore are not meant to measure or rank packages, but rather to help delineate the differences between packages and to help designers of future packages. One may argue that a "pedagogical" or teaching package needs to be concerned only with easy control, small (teaching) examples and simple I/O, making many of the above criteria irrelevant. However, experience has shown that students are fond of taking their familiar packages with them after graduation. Converse arguments can also be made about the desirability of using the same type package in the classroom that will be used in "real applications." Hence these criteria and considerations are important and should be considered in the study of any time series software.

6. REFERENCES

(1) I. FRANCIS, R. M. HEIBERGER, P. F. VELLEMAN, Criteria and Considerations in the Evaluation of Statistical Program Packages. The American Statistician, 29 (1975) pp. 52-55.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

COMPARISON OF STATISTICAL PACKAGES: A FEATURES MATRIX APPROACH

Kenneth A. Hardy and William C. Reynolds
Social Science Statistical Laboratory, Institute for Research in Social Science
University of North Carolina at Chapel Hill, Chapel Hill, NC 27514

David R. Kniefel
North Carolina State University, Raleigh, NC 27607

ABSTRACT

A features matrix with accompanying glossary of terms is proposed for the comparison of features of statistical packages available for IBM 360-370 environments. Improved definitions and further enumeration of features are seen as necessary, continuing tasks in the further refinement of such a matrix. The proliferation of statistical packages and their various versions makes such a task quite difficult.

Key words: Statistical package; features matrix; BMDP; DATA-TEXT; OMITAB; OSIRIS; SAS; SOUPAC; SPSS; TSAR.

1. INTRODUCTION

There was once a time when a person wishing to use a computer to analyze data had a very simple task before him/her. After becoming expert enough in a programming language the user simply wrote a program to perform the necessary calculations feeling fortunate indeed if a library of useful subroutines was already available for use. The process was terribly time consuming and often made accomplished programmers out of analysts with little desire to become so. With the advent of the statistical package and problem oriented languages all this has changed for the better--or has it? When there were only one or two such packages life was simple. Either one or both of the packages could perform the required data input, transformations and statistical calculations or it was back to roll your own. However, now package proliferation is poignantly problematic for both the novice and sophisticate alike. Not only are there many more packages available but each has grown in complexity as well as flexibility. The problem has become which package best solves a particular class of problems rather than whether or not there is a package that will solve them.

In order to help users select among packages several surveys of packages have appeared which have offered feature by package matrices to compare package capabilities (Allerbeck, 1971; Schucany, et. al., 1972; Slysz, 1974; CUNY, 1976). While such matrices generally offer no evaluation of the degree of accuracy or ease of features, they do serve the important function of defining bases for comparisons which might later be more thoroughly investigated and quantified (E. G. Rollwagen, 1974; Francis, 1973).

This paper attempts to make a contribution to the construction of such matrices by constructing a matrix of features which does not concentrate completely on available statistical procedures but also on data management and transformation capabilities. It also attempts to update past efforts by including more recent versions of packages reviewed in the past.

2. PACKAGES

The packages chosen for this effort were those most commonly available on IBM 360/370 machines running under OS or VS. They were BMD-P, DATA-TEXT, OMNITAB, OSIRIS III, PSTAT, SAS-76, SOUPAC, SPSS and TSAR. Unfortunately, manuals for GENSTAT and OMNITAB II were not secured in time for this effort but, hopefully both will appear in a later version. Every attempt was made to secure the most recent documentation for each package. (See references.) However, since not only packages proliferate but versions of packages also have multiplied, it is very possible that any omissions might be the result of not having up to date documentation.

3. FEATURES

An attempt has been made to provide some organization to the list of features which corresponds to the steps involved in using most packages--data definition, data transformation, data summarization and estimation. Other modes of organization are certainly possible. In fact the best mode of organization should be a matter of further investigation. However, it is clear that some form of logical organization must be brought to such features in order to make such lists useful for both evaluation and reference purposes.

The determination of whether a package has or does not have a particular capability is sometimes not as clear as it might seem. Many packages can be made to do almost any form of data manipulation by applying enough programming effort to the task. However, other packages can accomplish the same manipulation in one or two statements. For example, SPSS's RECODE statement can easily accomplish the same remapping of variable values that requires many SAS76 IF statements. Thus, for this matrix an indication that a package has a certain capability is based on whether or not an operation can be accomplished but not necessarily how easily that operation might be done. Admittedly some consideration was given to ease of programming in making some determinations about a package's capability. This has undoubtedly interjected a subjective element into the construction of the matrix which, hopefully, can be improved upon by better definition of features and constructive comments.

Unlike some other papers presented in this area, the originators of the packages have not been given the opportunity to examine the entries for their package in advance. It is hoped that they will be able to do so in the near future and provide feedback on any errors and omissions as well as suggest improvements in the features list itself. Comments from others interested in this effort are also appreciated. The matrix has been automated using SAS76 so that adding or changing features or correcting entries is not quite as problematic as retyping the matrix anew.

4. GLOSSARY

A glossary of terms for input data capabilities, data management facilities, package file capabilities and output capabilities is appended to the features by packages matrix to help clarify the meaning of terms used. No attempt has been made to provide definitions of statistical terms. The need for such definitions or at least references to appropriate literature is recognized. However, such a task would require more effort than is possible to devote to this project at the present time. It is hoped that this glossary can be expanded and improved upon through comments from interested readers.

COMPARISON OF STATISTICAL PACKAGE FEATURES
APRIL, 1977

Input Data Capabilities	BMDP	DTXT	OMNI	OSIR	PSTA	SAS	SOUP	SPSS	TSAR
Types of Input Data									
Case by variable	X	X	X	X	X	X	X	X	X
Hierarchial records		X		X	X	X			
Variable length records						X			
Correlation matrices	X	X		X		X	X	X	X
Program Data Definition									
Mnemonic variable names	X	X			X			X	X
Column defined		X		X	X	X		X	X
FORTRAN format	X		X				X	X	
Freefield						X		X	
Data Types									
Character < 5	X	X		X	X		X	X	X
Character > 4						X			
Multipunched (col. binary)		X		X		X			
Multivalued observations		X		X					
Real binary	X	X	?		X	X	X	X	
Integer binary	X	X	?			X	X	X	
Packed decimal						X			
Zoned decimal						X			
Data Management									
Data Editing									
Input sequence check			X					X	
Wild code check		X		X					
Range check	X			X					
Missing Values									
Automatic deletions	X	X		X	X	X	X	X	X
Pair-wise deletions	X	X		X	X		X	X	
List-wise deletions	X	X		X	X	X		X	X
Checked in transformations	X	?	?	?	?	X		X	X
Data Transformation									
Recode statement	X	X		X			X	X	X
Character to numeric transform	X	X		X	X	X	X	X	X
Arithmetic computes	X	X	X	X	X	X	X	X	X
List functions		X		X	X	X	X		X
Crosscase transformations	X	X	X	X	X		X		X
Ranking	X		X	X	X	X	X		
Standardization (Z scores)	X	X	X	X	X	X	X		
Data aggregation	X	X		X	X		X	X	
Transpose data (e.g., case to variable)						X	X		
Contingent transformation	X	X	X	X	X	X	X	X	X
Case weighting	X	X	X	X	X	X	X	X	X
Sort functions			X	X	X	X	X	X	
Case Selection									
Random samples	X		X	X	X	X	X	X	X
Selective samples	X	X	X	X	X	X	X	X	X
Automatic storage of data subsets					X	X			

Package System File	BMDP	DTXT	OMNI	OSIR	PSTA	SAS	SOUP	SPSS	TSAR
File Manipulation									
Save and process system files	X	X		X	X	X	X	X	X
Update system files	X	X		X	X	X	X	X	X
Add variables to file		X		X	X	X	X	X	X
Add cases to file		X		X	X	X	X	X	X
Merge files		X		X	X	X		X	
File Interfaces									
Read other system files						X		X	
Write other system files		X							
Output									
Labeling									
Variables		X						X	
Values		X						X	
Other Output									
Data listing statement		X				X		X	X
Data to other (tape, etc.)		X	X	X	X		X	X	
Matrix (correlation, etc.)		X		X			X	X	
Statistical Procedures									
Univariable Descriptive Measures									
Mean	X	X	X	X	X	X	X	X	X
Median	X	X	X	X			X	X	
Mode	X	X	X	X	X		X	X	X
Variance	X	X	X	X	X	X	X	X	X
Standard deviation	X	X	X	X	X	X	X	X	X
Range	X	X	X	X	X	X	X	X	X
Frequency distribution	X	X	X	X	X	X	X	X	X
Histogram	X			X			X	X	
Contingency Table Analysis									
Row percent	X	X		X	X	X	X	X	X
Column percent	X	X		X	X	X	X	X	X
Cell percent of total	X	X		X	X	X	X	X	
Expected values	X				X	X			
Chi-square	X	X		X	X	X	X	X	X
Fisher's exact test	X	X		X	X		X	X	X
Yate's correction	X						X		
Non-Parametric Measures of Association									
Cramer's V	X	X		X			X	X	
Tau A	X	X		X	X				•
Tau B	X	X		X	X			X	X
Tau C	X							X	
Gamma	X	X					X	X	
Somer's D									
Symetric	X	X						X	
Asymetric	X	X						X	
Lambda									
Symetric	X	X					X	X	
Asymetric	X	X					X	X	
Phi	X							X	

	BMDP	DTXT	OMNI	OSIR	PSTA	SAS	SOUP	SPSS	TSAR
Simple Correlation (Parametric)									
Scaiter plots	X	X	X		X	X	X	X	X
Pearson's product moment	X	X	X	X	X	X	X	X	X
Eta2				X				X	
Simple regression coefficients	X	X	X	X	X	X	X	X	X
Sample Tests									
Parametric									
Student's T test	X	X		X		X	X	X	X
Related T-test	X	X							X
Non-Parametric									
Chi-square	X				X	X	X	X	X
Kolmogorov - Smirnov					X		X	X	X
Wilcoxon	X								
Runs test			X						
Tau A	X			X	X			X	
Concordance	X			X					
Mann Whitney U	X						X		
Kruskal-Wallis Anova	X								
Friedman 2-way Anova	X								
Analysis of Variance									
Dimensions									
One-way	X	X		X	X	X	X	X	X
Two-way	X	X	X	X	X	X	X	X	X
Three-way	X	X	X	X	X		X		X
Three-way	X	X	X	X	X	X	X	X	X
N-way	X	X	X	X	X	X	X	X	
Estimation									
Unweighted means	X	X					X		X
Exact anova	X		X	X	X	X		X	
A priori contrasts								X	
Posterior comparisons									
Duncan multiple range test	X				X	X		X	
Dunnet's T	X				X				
Student-Newman-Keuls						X		X	
Tukey							X	X	
Tukey's alternative								X	
Modified least significant difference								X	
Scheffe							X	X	
Repeated measures	X	X					X		
Multiple Regression									
Variable entry									
Stepwise	X	X		X	X	X	X	X	X
Automatic polynomial	X		X				X		X
Parameter estimation technique									
Ols	X	X	X	X	X	X	X	X	X
Weighted least squares									
Least squares estimates of nonlinear bet	X					X			
Assessment of resultant equation									
Residual printing	X	X	X	X	X		X	X	
Residual plotting	X	X	X	X	X		X	X	
Durbin-Watson		X		X	X	X	X	X	
Multiple R	X	X	X	X	X	X	X	X	

	BMDP	DTXT	OMNI	OSIR	PSTA	SAS	SOUP	SPSS	TSAR
F-test for multiple R	X	X		X	X	X	X	X	X
Parameters estimated									
Unstandardized beta	X	X	X	X	X	X	X	X	X
Standardized beta	X	X		X	X		X	X	X
Normalized beta					X				X
Regression through origin	X	X		X	X	X			X
T or F test for coefficients	X	X	X	X	X	X	X	X	X
Analysis of covariance									
One-way	X	X		X	X	X		X	
N-way		X		X	X	X		X	
With multiple covariates	X			X	X	X		X	
Factor analysis									
Factor structure estimation									
Principal components	X	X		X	X	X	X	X	
Principal axis	X			X	X	X	X	X	
More advanced techniques	X			X		X	X	X	
Rotational methods									
Orthogonal									
Varimax	X	X		X	X	X	X	X	
Other	X	X		X		X	X	X	
Celique				X			X	X	
User supplied communalities	X					X	X	X	
Miscellaneous multivariate techniques									
Discriminant function	X			X	X	X	X	X	
Canonical correlation	X			X	X	X	X	X	
Probit	X				X	X	X		
Logit									
Cluster analysis	X			X		X	X		
AID (automatic interaction detector)				X					
MCA (multiple classification analysis)				X					
Spectral analysis	X					X	X		
Time series analysis	X					X	X		
Miscellaneous mathematical techniques									
Linear programming							X		
Matrix algebra operations			X			X	X		
Miscellaneous scaling									
Nonmetric multidimensional scaling							X		
Guttman scaling				X	X		X	X	
Roll call analysis				X					

5. GLOSSARY OF TERMS

I. Input data capabilities--input data may have various forms of organization and coding. Some statistical packages have great flexibility with respect to the forms of data organization and coding which are acceptable. Others are much more restricted in what they are capable of accepting.

A. Types of input data

Case by variable - the standard mode of data input corresponding to the statistical notation of a case by variable matrix of data.

Hierarchical file - a data set with an aggregate level record preceding records for each unit composing the aggregate unit. (e.g.: a record with family characteristics followed by records with individual level data about each member of the family.)

Variable length records - records with different physical lengths. A package must be able to determine the length of the record as well as read the information on it.

Correlation matrices - matrices are often a more compact form of feeding data to multivariate analyses. Some packages have the capacity to input them directly to these statistical routines.

B. Program data definition - every statistical program must have a method for defining the characteristics of the data to the program. Statistical packages vary in the ease with which this may be done.

Mnemonic variable names - the capacity to refer to variables in the data using user created mnemonic names.

Column defined data formatting - a convenient method for detailing the position of variables on data records in terms of record location without having to write a pseudo-Fortran format statement.

Fortran format - use of FORTRAN-like format statements to describe positions of variables on data records.

Freefield - data for variables is simply placed on record in order separated by one or more blanks.

C. Data types - data are not always represented as simple numeric codes. Packages differ in their capabilities for reading nonstandard data types.

Character < 5 - character strings of length four or less.

Character > 4 - character strings of length greater than four are generally less easily handled, if at all, by many packages.

Multipunched - non-EBCDIC or BCD codes used to represent data, usually found in old Harris and Roper opinion surveys.

Multivalued observations - possible with multipunched codes where several numeric codes in a single card field comprise a legitimate code combination.

II. Data management facilities - data is not often in the form an investigator wishes immediately after being read in from a deck of cards, tape, etc. The data management facilities of statistical packages permit the investigator to manipulate the values of variables, construct indices of concepts, select subsamples for analysis as well as edit out "bad" data from analyses. Various packages provide these facilities to different degrees.

A. Data editing - operations performed to insure that data has been correctly recorded and in proper order for processing.

Automatic data sequence checking - the program requires the user to specify to it the case identification field and the case sequencing field in the data. Using this information, the program then checks the user's input data to confirm that it is sorted properly.

Wild code check - ability to specify permissable codes for variables and have package check for illegitimate values.

Range check - ability to define permissable upper and lower numeric bounds for variables and have package check for out of range values.

B. Missing data handling - data are seldom complete for all cases across all variables. Nevertheless, many researchers feel that they have adequate data to perform an analysis on the data in hand. Statistical packages often provide a means for identifying values for variables that indicate "bad" data and provide a means for editing out cases containing these values.

Automatic elimination - elimination of cases with identified missing values from the analyses without user intervention.

Pair-wise deletion - the elimination of cases from each of the bivariate relationships entering into a multivariate analysis such that only cases missing data from a particular bivariate relationship are eliminated from the calculation of that relationship's coefficient.

List-wise deletion - the elimination of cases from a bivariate or multivariate analysis such that if any variable of a case has a missing value, all variables for that case are eliminated from the analysis.

Checked in transformations - missing values encountered in data transformations cause the result of the transformation to be set to a missing value.

C. Transformation and selection features

Recode statement - the ability to easily change the values of a variable to other values. Usually most useful in collapsing many categories of responses into fewer categories for analysis.

Arithmetic computations - the ability to easily perform arithmetic transformations of a variable or variables. (E.g., sum several attitude variables to construct a Likert index.)

List functions - functions useful in arithmetic computations which perform summary operations on a case-wise basis or several variables at a time. (E.g., Y=mean (A, B, C, D, E) where Y is the mean of the variable values of A, B, C, D and E.)

Crosscase transformations - ability to perform computations that involve values aggregated across cases. [E.g., Y = (X - mean (X))]

Transposition of data matrix - the ability to transpose the datamatrix from a cases by variables matrix to a variables by cases matrix. This is useful for performing Q factor analysis and/or cluster analysis.

Contingent transformations - the ability to transform the value of a variable, assign a value to a new variable or perform a computation if and only if some logical expression based on values of the data is true.

Ranking function - the ability to assign rank order numbers to cases based on the values of a variable.

Standardization - a useful crosscase transformation which will automatically transform the values of a variable into Z score form.

Aggregate data - the ability to calculate and assign to cases values for variables which are the sums of all members of a class of units in the data of which that case is a member. The units used as the basis for aggregation are user definable.

Character variable conversion - the ability to assign numeric values to the alphabetic codes for a variable or variables.

Case weighting - the ability to weight cases in terms of marginal frequencies in a manner prescribed by the user.

Sort functions - the ability to sort the data based on the values of some variable(s). Usually the case identification number and the record sequencing field are used as the variables.

D. Case selection

Random sampling - the ability to select a random subset of cases in the data.

Selective samples - the ability to select out cases for analysis on the basis of some selection criterion. (E.g., perform the analysis only on middle class males.)

III. Package file capabilities - many systems are capable of storing input data as well as variable mnemonics and other descriptive information in a form that makes them more easily accessible and quickly read by that particular statistical package. These system files also save the user the machine time necessary to define the location and description of the data each time the package is run on that particular data set.

File manipulation - the ability to modify a package file.

Save and process system files - has the ability to store and retrieve a system file.

Update files - the capacity to correct values for given variables for given cases on an existing file.

Add variables to system file - the ability to add additional data in the form of variables to a statistical package system file.

Add cases to system file - the ability to add data in the form of cases to a statistical package system file.

Merge files - a procedure for merging two or more existing files.

File interfaces

Read other system files - procedure for reading another package's system file.

Write other system files - a procedure for writing another package's system file.

IV. Output capabilities

Variable labelling - the ability to append short descriptive phrases to each variable and have those phrases printed out when the variable is displayed on the printout for an analysis.

Value labelling - the ability to assign a short descriptor to any or all values of a

variable and have these descriptors printed out adjacent to the values whenever that variable is displayed on the printout.

Data output - the ability to output the data from the package to some other storage medium such as tape or cards.

Matrix output - the ability to output correlation and/or factor score matrices from the program for storage on some medium such as tape or cards.

6. REFERENCES

ALLERBECK, K. S. (1971). "Data Analysis Systems: A User's Point of View." Social Science Information 10(3), 23-35.

ARMOR, D. J. and COUCH, A. S. (1972). Data-Text Primer. New York: Free Press.

EARR, A. J., et al. (1976). A User's Guide to SAS76. Raleigh: SAS Institute Inc.

BUHLER, R. (1975). P-Stat: A Computing System for File Manipulation and Statistical Analysis of Social Science Data. Princeton, N. J.: Princeton University Computation Center.

CHAMBERLAIN, R. L. and JOWETT, D. (1970). The OMNITAB System: A Guide for Users. Ames, Iowa: University Bookstore, Iowa State University.

COMPUTING SERVICES OFFICE. (1975). SOUPAC Program Descriptions, Vols. I and II. Urbana-Champaign, Computing Services Office, University of Illinois at Urbana-Champaign.

DIXON, W. J. (ed.) (1975). BMDP Biomedical Computer Programs. Berkeley: University of California Press.

DUKE UNIVERSITY COMPUTATION CENTER. (1970). Tele-Storage and Retrieval System User's manual. Durham: Mimeographed.

FRANCIS, I. (1973). "A Comparison of Several Analysis of Variance Programs." Journal of the American Statistical Association, 68(344), pp. 860-865.

INSTITUTE FOR SOCIAL RESEARCH. (1973). OSIRIS III, Vol. 1. Ann Arbor: Institute for Social Research, University of Michigan.

NIE, N., et al. (1970). SPSS Statistical Package for the Social Sciences: Second Edition. New York: McGraw-Hill.

RATTENBURG, J. and PELLETIER, P. (1974). Data Processing in the Social Sciences with OSIRIS. Ann Arbor: Survey Research Center, Institute for Social Research.

ROLLWAGEN, R. I. (1974). "Statistical Computer Systems -- An Evaluation." Paper presented at Meeting of the ASA, St. Louis, Missouri.

SCHUCANY, W. R., MINTON, P. D. and SHANNON, B. S., Jr. (1972). "A Survey of Statistical Packages." Computing Surveys, 4(2), pp. 65-79.

SLYZ, W. D. (1974). "An Evaluation of Statistical Software in the Social Sciences." Communications of the ACM 17(6), 326-332.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

INTERACTIVE PLOTTING WITH THE ST PACKAGE

Robert M. Dunn and Jane F. Gentleman
University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

Abstract

The low "overhead" learning philosophy of the ST package is discussed. Examples of the interactive dialogue used to produce plots, and of the resulting plots are presented. A word on the current state of development concludes the report.

Key words: Graphics; interactive computing; plots.

1. Some Philosophy

The ST interactive statistical plotting package was developed for use by statisticians, as opposed to computer programmers. Computer programmers have invested in a body of knowledge which allows them to enter data and instructions into the computer, thus acheiving some desired result. In other words, they make use of a certain syntax to communicate with the computer. On the other hand, a statistician may not be familiar with any other syntax than that of the English language. The ST package allows the statistician to enter specifications for a particular type of plot, usually using only the English language. Thus the need for programming experience or reference to thick manuals is minimized.

The obvious way of getting the plot specifications into the computer is for the program to ask appropriate questions, which when answered in English (usually "yes" or "no") obtain the necessary information to produce the desired plot. This is the approach used by the ST package. (Hence the name "interactive plotting.") In this manner, the "overhead" of computer related knowledge required of the user is kept minimal. More complex plots will require more questions, and answering many questions can be tedious. This is the "operating cost," and is measured in terms of human patience. In writing an ST program, we try to minimize this operating cost in several ways--among which are keeping questions terse yet unambiguous, and allowing experienced users to "answer ahead" (supply answers before the question appears)--but not, if possible, at the expense of increased overhead.

2. An Example

At the poster session described by this paper, 15 recorded examples of on-line interaction with the ST package were played back on a graphics terminal, and copies of the ST user's guide were made available. A condensed version of one of the examples follows:

Example: Analysis of U.S. draft data using enhanced scatter plots.

Data:

Y Data: Birthdates, represented as the integers from 1 to 366. These were drawn from a box, supposedly randomly, to determine an order for drafting people in the U.S. in 1969.

X Data: The number of the draw on which the corresponding birthdate was selected.

Background to the Analysis:

An analysis by Fienberg (1973) using regression and goodness-of-fit tests showed that earlier birthdates tended to be selected later, probably because they were nearer the bottom of the box and the box was inadequately mixed--although this cannot be perceived in the scatter plot.

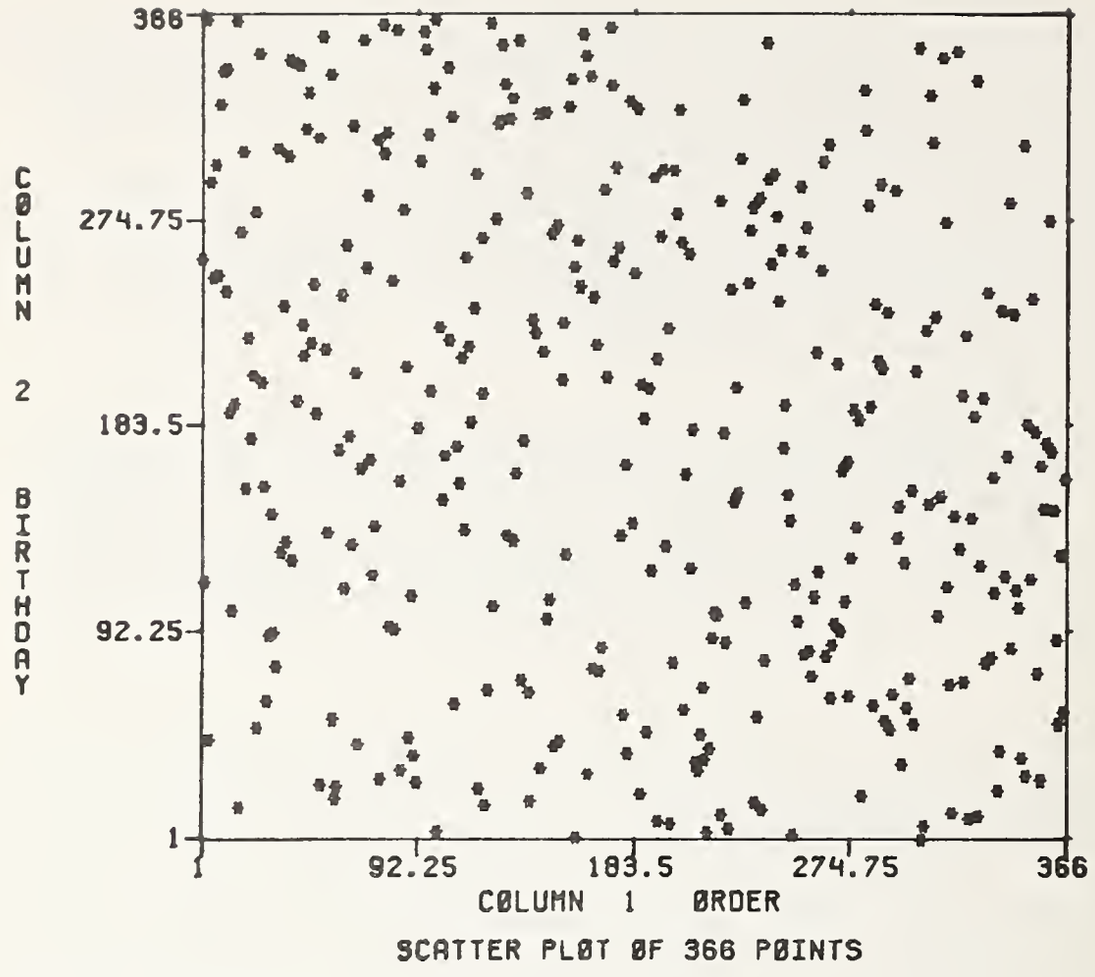
The Cleveland/Kleiner technique is to compute smoothed moving statistics to summarize the behaviour of the original scatter plot of Y versus X. Four vectors are computed given a group size R for the moving statistics: the vector S0 is smoothed X's; S1 is smoothed values of the lower values of Y; S2 is smoothed values of the middle values of Y; S3 is smoothed values of the upper values of Y. The points (S0,S1), (S0,S2), and (S0,S3) are then plotted.

Description of the Terminal Session:

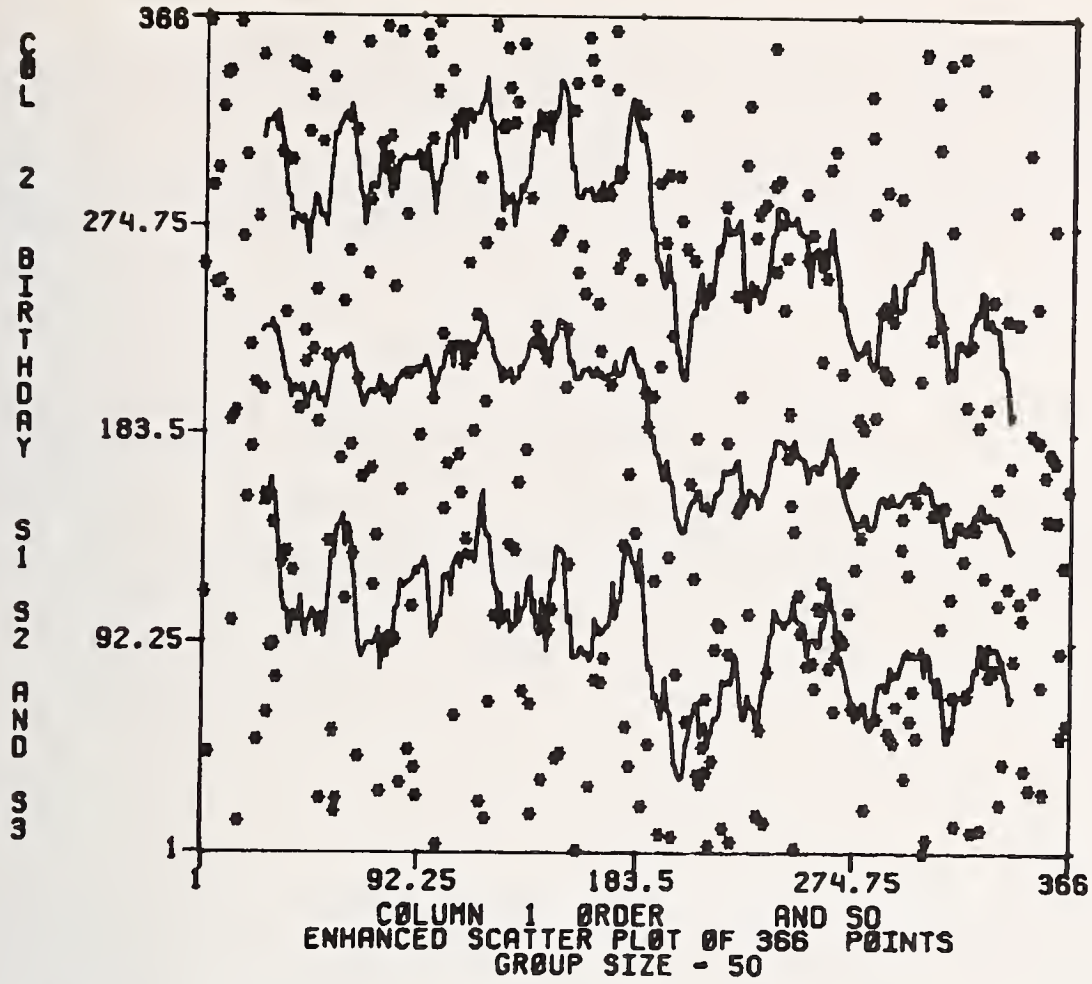
The user runs the program ST/scat: the X and Y data are typed and saved on a file, and a scatter plot is produced. Not satisfied, the user then runs ST/SCATPLUS, which uses the Cleveland/Kleiner technique to enhance scatter plots. The saved data are read from the file, a group size of R=50 is specified, and the resulting enhanced scatter plot is displayed. (Notice that the results agree with Fienberg's analysis.) The user saves the moving statistics on a separate file for future reference.

Terminal Session:

```
SYSTEM?st/scat
IF TEKTRONIX TERMINAL, TYPE BAUD RATE: 9600
VARIAN HARD COPY CAPABILITY REQUIRED? no
(screen clears)

   INTERACTIVE SCATTER PLOTS

USER CONTROL OF AXIS LIMITS? yes
  OF X-AXIS LIMITS? y
  OF Y-AXIS LIMITS? y
IF DATA ON A FILE, TYPE FILE NAME: (carriage return)
IF DATA CONTAINS MISSING VALUES,
  TYPE A NUMBER THAT WILL REPRESENT THEM:
TYPE THE DATA: AN X VALUE, A Y VALUE, ANOTHER X, ETC.
  AN EMPTY LINE SIGNIFIES END OF DATA.
DATA: 305 1  159 2  251 3  215 4  101 5  224 6  306 7  199 8  194 9
DATA: 325 10  329 11  221 12  318 13  238 14  17 15  121 16  235 17
DATA: (etc.)
DATA: 95 359  84 360  173 361  78 362  123 363
DATA: 16 364  3 365  100 366
DATA:
TITLES FOR X DATA AND Y DATA (8 CHARS EACH): order birthday
SAVE DATA ON A FILE? y
TO SAVE THE X DATA, TYPE FILE NAME: usdraft
TO SAVE THE Y DATA, TYPE FILE NAME: usdraft
CONNECT POINTS WITH STRAIGHT LINES? n
PLOTTING CHARACTER:
X-AXIS LIMITS: 1 366
Y-AXIS LIMITS: 1 366
(the screen clears, and the plot appears:)
```




```
(user types carriage return to continue)

ANOTHER SCATTER PLOT? no

SYSTEM?st/scatplus
IF TEKTRONIX TERMINAL, TYPE BAUD RATE: 9600;no
(screen clears)

   INTERACTIVE ENHANCED SCATTER PLOTS

USER CONTROL OF AXIS LIMITS? y;y;y
IF DATA ON A FILE, TYPE FILE NAME: usdraft
THERE ARE   2 COLUMNS OF DATA.
  WHICH COLUMN FOR X-COORDINATES? 1
  WHICH COLUMN FOR Y-COORDINATES? 2
GROUP SIZE FOR MOVING STATISTIC: 50
PLOTTING CHARACTER FOR (X,Y), IF ANY: *
X-AXIS LIMITS: 1 366
Y-AXIS LIMITS: 366
TYPE  1 MORE NUMBER: 1
(the screen clears, and the plot appears:)
```



(user types carriage return to continue)

```
SAVE RESULTS ON A FILE? yes
TO SAVE THE ORDERED X-COORDINATES, TYPE FILE NAME: results
TO SAVE THE CORRESPONDING Y-COORDINATES, TYPE FILE NAME: results
TO SAVE THE MOVING STATISTICS SO, TYPE FILE NAME: results
TO SAVE THE MOVING STATISTICS S1, TYPE FILE NAME: results
TO SAVE THE MOVING STATISTICS S2, TYPE FILE NAME: results
TO SAVE THE MOVING STATISTICS S3, TYPE FILE NAME: results
ANOTHER ENHANCED SCATTER PLOT? no
```

3. Current State of Development

There are presently 19 different ST routines. They will perform scatter, Q-Q, ECDF, PDF, PF, and CDF plots, polynomial regressions, user supplied function plots in rectangular or polar coordinates, histograms, a Central Limit Theorem demonstration involving histograms of means of random samples, summary sample descriptive statistics plots, multi-dimensional data plots, bar graphs, enhanced scatter plots, multiple regressions, and contour plots. They will also generate random numbers to be stored in a file for further analysis, and evaluate PF's, PDF's, CDF's, and inverse CDF's for various discrete and continuous distributions. Certain programs are used for research, while others are designed for primarily for teaching purposes (e.g. the Central Limit Theorem demonstration). These programs will run on a Tektronix terminal (for high quality plots) or an arbitrary terminal (for crude character plots). High quality hard copy is available from any terminal. The ST programs are available to and are widely used by both faculty and students throughout the Faculty of Mathematics at the University of Waterloo.

The ST package is being developed at the University as a research project. As such, it is in a state of constant flux. (We are currently involved in making the package device independent.) It is written in standard Fortran, with exceptions and system dependent features documented and separated from the main body of code.

4. References

CLEVELAND, W.S. and KLEINER, B. (1975). A Graphical Technique for Enhancing Scatterplots with Moving Statistics. Technometrics, 17, 447-454.

FIENBERG, S.E. (1973). Randomization for the Selective Service Draft Lotteries. Statistics by Example. Finding Models. Edited by Mosteller, Pieters, Kruskal, Rising, and Link, 1-13.

Biography

Robert M. Dunn is a graduate student in the Department of Computer Science at the University of Waterloo. He received a Bachelor of Mathematics in Computer Science and Statistics from Waterloo in 1976.

Jane F. Gentleman is an Associate Professor in the Department of Statistics at the University of Waterloo, and has a cross appointment with the Department of Computer Science. She received a Ph.D. in Statistics from Waterloo in 1973.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

GENERALIZING THE FUNCTION CALL TO STATISTICAL ROUTINES: AN APPLICATION FROM THE DATATRAN LANGUAGE

John Brode
M.I.T., Cambridge, MA. 02139

ABSTRACT

A standard inconvenience of most statistical computer programs is the awkwardness of passing the results from one routine to another.

Key words: Algorithm, attributes; computer; Consistent System; data; DATATRAN; linguistic expression; mnemonics; random number generation; statistical routines; time-sharing.

1. INTRODUCTION

There are several stages to be passed before this data transmission can be done automatically at the cost of minimum effort from the user. Obviously, the data must be retrieved and stored in a way that other routines can get at it. On some time-sharing installations, such data handling is available. However, there is a further stage that has been neglected. The linguistic expression of this data transmission must be concise and clear. The expression should resemble the way in which the problem is normally stated, say to a colleague.

2. GENERAL

As part of the Consistent System developed at M.I.T., I proposed a language, DATATRAN, that would accomplish this linguistic expression. This language has been implemented within an interactive version of a large statistical package (TSP which was written primarily for econometricians). Two notable features of this application are the table driven syntax and the context dependent semantics.

A basic element of what was proposed is to treat all statistical routines as multi-valued functions. This parallels our natural usage. We regress "x" on the "log(y)" rather than on some other attribute that was created as the log of y.

Several examples are attached. To better understand these examples, the reader is urged to first read the note on "An algorithm to derive mnemonics for computer usage."

The first example compares two time series; the original dependent attribute and the predicted values of that attribute regressed on several independent attributes.

The second and third examples introduce random number generators as dyadic operators. A random number generation can be thought of as relating location (some measure of centrality) to scale (some measure of dispersion).

All of the examples show how the computer creates names for the attributes that are returned by the indicated functions. These created names are just what the user called them in writing out the instructions to the computer. "log(y)" is called just that. "x-y" becomes "x-y". "5 rnnd 2" (random numbers normally distributed) becomes "5.rnnd2.".

An algorithm to derive mnemonics for computer usage

As the number of statistical procedures available on a computer grows, the problem of what to call the procedures arises. Some computers will not accept more than a small number of characters in a name (six in many cases). This makes it impossible to use the full name of many statistical techniques. Thus, we have seen a great number of abbreviations. ANOVA for analysis of variance is a well-known one.

These abbreviations, however, are arbitrary. They have to be memorized for each usage. Besides, which analysis of variance should be called ANOVA since there are a variety of techniques available?

To get around these encumbrances, an abbreviation algorithm was developed at MIT.* The advantage of an algorithmic approach is that the user is absolved from memorizing abbreviations. Instead, each abbreviation can be recreated from the normal name for a statistical technique by the application of the algorithm.

The algorithm for abbreviating names is as follows:

```
! The first letter and the next following consonant (if any)
! of the first word to which are added the first letter of
! each subsequent word in the name.  (N.B., prepositions,
! conjunctions, definite and indefinite articles are passed
! over in scanning the subsequent words.)
```

As an example, "analysis of variance for complete layouts" becomes "anvcl". "an" comes from the first word. "of" is skipped as is "for" so that the first letters of the remaining words make up "vcl". "under the name" would become "unn" even though "under" is a preposition. It is not skipped over since it is the first word in the name to be abbreviated.

Inevitably, we have had to make exceptions but they are well-defined and limited in scope.

1. In order to avoid confusion and redundancy, short names are not abbreviated. Short is defined as any name composed of only one word which has four or fewer letters. E.g., "for", "with", "plot" are not abbreviated.

2. The few commands coming from FORTRAN, such as "format", have been left unabbreviated. It was felt that most people would already know them in their long form.

3. Function names, such as "log", "tan", etc. that are already widely used in an abbreviated form have been left untouched. For the most part, these abbreviations have become names in themselves and, as such, would not be abbreviated under the above algorithm, exception 1. The few, like "conjg" that would be abbreviated by this algorithm have been left untouched so as to avoid undue confusion.

* This algorithm was developed largely by Jeffery Stamen and Robert Wallace as part of their work at the Cambridge Project, MIT.

An expanded version of TSP/DATATRAN will be available for general use on Multics beginning in September 1977. Some parts of the expanded version are currently available by special arrangement. For more information, please contact: John Brode, 23 Berkeley St., Cambridge, MA. 02138, (617) 864-8319.

```
cmts t prv of rso(t on los(t) cnt) with plot;end;
t
los(t)

equation  12

entities vector
   1 to   12
t

independent                    estimated          standard        t-
 attribute                    coefficient          error        statistic
  los(t)                        4.51319            .4885          9.24
  constant_term                -1.01717            .8872         -1.15
r-sq =  0.8951
f-stat (  1, 10) =     85.34
d-w (  0 vars) =    0.4519
number entities =   12
st. error =   1.22469

r =  0.9461
r-sq = 0.8951
rms =   1.250
mean abs. error =    .9567
mean error =  0.5782e-18
res. coeff. =  0.8951
U = 0.1537
 U-m = 0.0000
 U-s = 0.0277
 U-c = 0.9723
 U-r = 0.1049
 U-d = 0.8951

entities vector
   1 to   12

* = t
+ = prv_of_rso

 -1.02                                                          12.0
|+        *
|              *+
|                    *   +
|                        *     +
|                              *    +
|                                   *    +
|                                        *  +
|                                             * +
|                                                 +*
|                                                    + *
|                                                      +    *
|                                                       +       *

T  4:45   1.133 $0.13
```

EXAMPLE 1

Generalizing the function call — John Brode

```
⍝lts location rnnd scale location rnud scale⍝end⍝
location
scale
location
location
scale
location

entities vector
   1 to   12

* = locationrnndscale
+ = locationrnudscale

 -22.3                                                          13.9
¦                        *        +
¦                                          +                    *
¦                        *     +
¦                                 *      +
¦*                              +
¦                       *       +
¦                             +       *
¦                    *                  +
¦                       *   +
¦                           +    *
¦                                + *
¦                                   .          +                *

T  2: 2   1.025 $0.11
```

EXAMPLE 2

```
rso t on location rnnd scale cnt;end;
location
scale
location

equation  12

entities vector
   1 to   12
t

independent                    estimated          standard        t-
 attribute                    coefficient          error        statistic
  locationrnndscale             -.176445           .1229          -1.44
  constant_term                  4.50642           1.708           2.64
r-sq =  0.1708
f-stat (  1, 10) =      2.06
d-w (  0 seps) =    0.3045
number entities =    12
st. error =    3.44347

T  1:59   0.381 $0.08

plts t - prv of rso(t on location rnnd scale cnt with step none);end;
location
scale
location
t
prv_of_rso
t-prv_of

entities vector
   1 to   12

* = t-prv_of_rso

 -5.54                                                          5.51
|*
|      *
|          *
|              *
|                  *
|                      *
|                          *
|                              *
|                                *
|                                    *
|                                        *
|                                            *
|                                                *
|                                                    *
|                                                        *
|                                                            *
|                                                                *

T  2: 1   1.109 $0.14
```

EXAMPLE 3

Generalizing the function call — John Brode

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

INTEGER PROGRAMMING WITH A COMPUTER: A STATISTICAL APPROACH

William Conley and Derrick S. Tracy
University of Windsor, Windsor, Ontario, Canada

ABSTRACT

Using Monte Carlo techniques it is possible to solve any and all integer programming problems in a very simple and direct fashion. Starting with problems that have a million or less feasible solutions, the authors write Fortran IV programs to search all possible solutions to obtain and record the optimum one.

When dealing with an integer programming problem with more than a million feasible solutions, which is usually the case in applications, the authors take a random sample of approximately one million feasible points and find the optimum solution of this sample.

It is the authors' contention that the sampling distributions of feasible solutions of practical integer programming problems have thick enough tails, no isolated extreme points, to make this approach useful in obtaining a solution that is very close to the true theoretical optimum. This contention is investigated by finding and graphing the sampling distributions of the feasible solutions of hundreds of integer programming problems. Copies of the graphs are available from the authors.

Key words: Computer; integer programming; Monte Carlo techniques; optimum solution; random sample of feasible solutions; statistical approach and justification.

1. INTRODUCTION

Integer programming is the study, and hopefully solution, of functions of several variables that are to be maximized or minimized. These variables are subject to certain constraints, usually inequalities. They further have the property that each variable can take only integer values.

Therefore in most practical problems there are only a finite number of possible (feasible) solutions. So theoritically it is possible to examine all possible solutions and take the one that produces the true optimum.

Until recently there was no real point in pursuing this approach, because even the simplest of integer programming problems would have thousands of feasible solutions. These problems were just too large to solve in this manner without some computational aid. But a modern high speed computer is quite capable of looking at thousands or millions of points and recording and printing the optimum solution in a matter of minutes or seconds.

2. AN EXAMPLE

Let's look at an example of an integer programming problem. Maximize

$$P = 7x_1 + 2x_2^3 + x_3^2 + 8x_4$$

ubject to $0 \leq x_i \leq 9 \quad i = 1,4, \quad x_1 + x_2 + x_3 + x_4 \leq 30$ and $x_1^2 + 2x_2 + x_3 \leq 70$.

Without considering the constraints one can see that there are 10 choices for each variable and therfore at most $10^4 = 10000$ possible solutions. Using Fortran (or any comparable language with loops and IF statements) a short program can be written to run through the 0000 possibilities, throw out the ones that don't meet the constraints and find the optium solution.

However, most practical problems, although they have a finite number of solutions, involve at least billions or trillions of possible solutions. This makes the above approach mpractical at best. For example, if we had a function of twenty variables and each could ake the values from 0 to 99 then we would have 100^{20} possible solutions.

3. THE TECHNIQUE

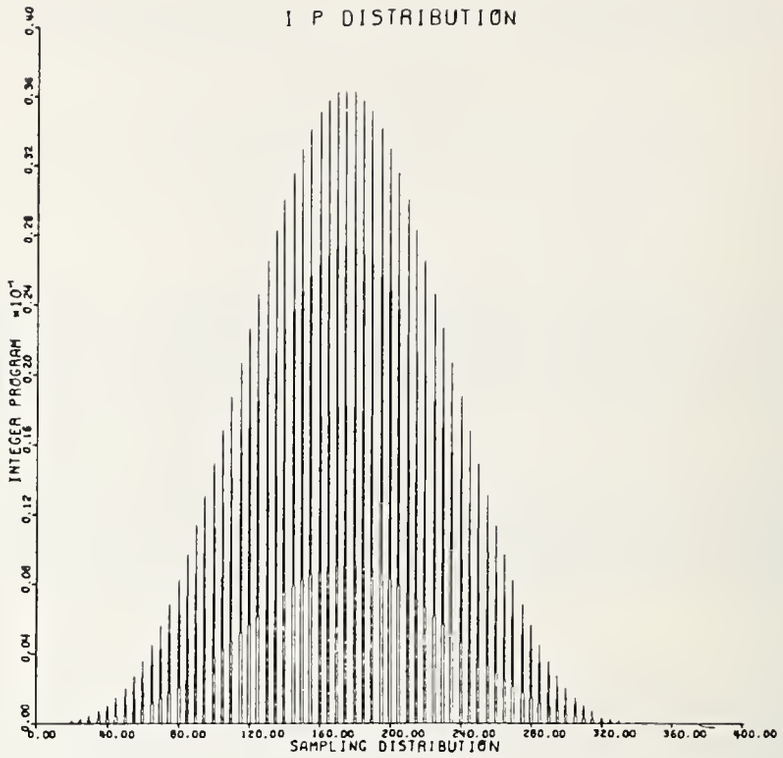
The authors propose to solve integer programming problems of this size by taking a random sample of say one million possible solutions and finding the maximum or minimum of this ample of solutions as desired.

The approach is quite straightforward. Just read in a random number for each variable nd check it to see if it meets the constraints. If they all meet the constraints, have he program evaluate the function and check to see if it is the optimum so far. If it is, hen store this solution. The program continues like this through the loop, say one million imes, and then prints the optimum solution. The programming details are available from the uthors.

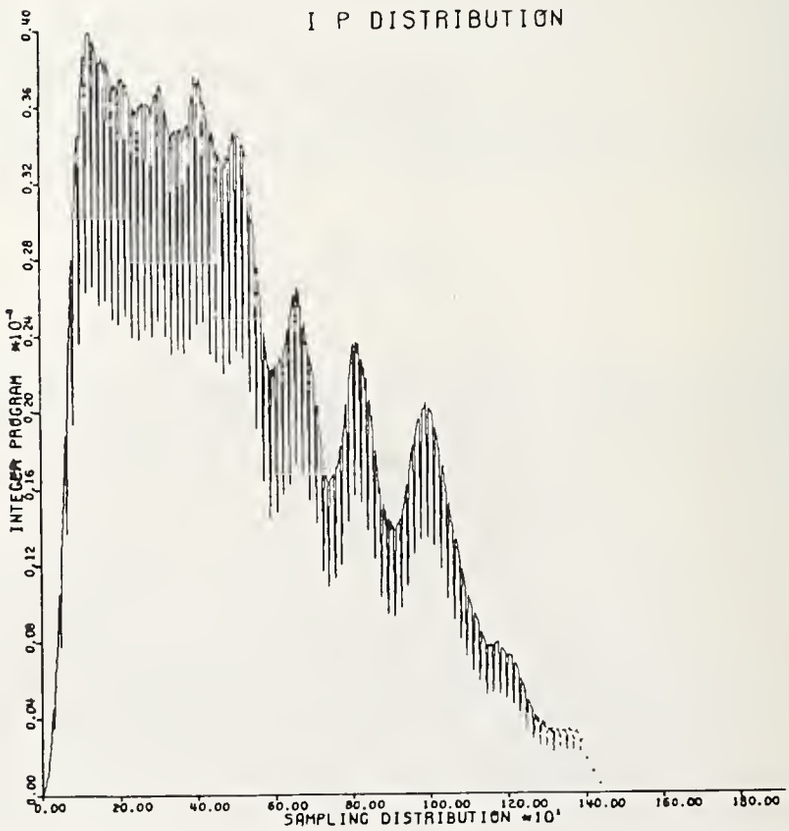
4. JUSTIFICATION

The only remaining question is how good is the answer obtained through random sampling? irst, it is very easy to obtain an answer quickly this way. This reduces costs. Also, the echnique works on virtually any integer programming problem whether linear or nonlinear, nd regardless of the type or number of constraints. Therefore very little time has to be pent figuring out how to approach the problem.

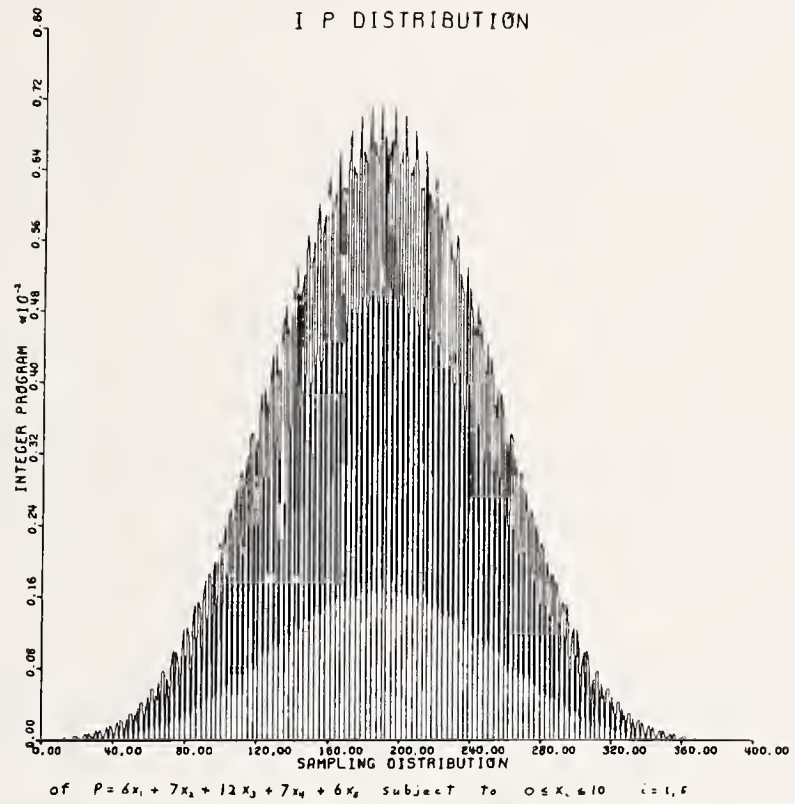
Each integer programming problem has a sampling distribution of all feasible solutions. y taking a random sample of about one million possible solutions, the odds are overwhelming hat the maximum or minimum solution from the sample will be in the upper or lower .001 percent region of the distribution. Assuming that the tails of the distributions of the integer programming problems are reasonably thick, no isolated extreme points, our random sample olution should be near the optimum. It is the authors' contention that this is true in ractical integer programming problems. This contention was investigated by finding and graphing the sampling distributions of hundreds of integer programming problems using the technique in Conley and Tracy (1976). Four of these graphs are presented here. Copies of thers are available from the authors.

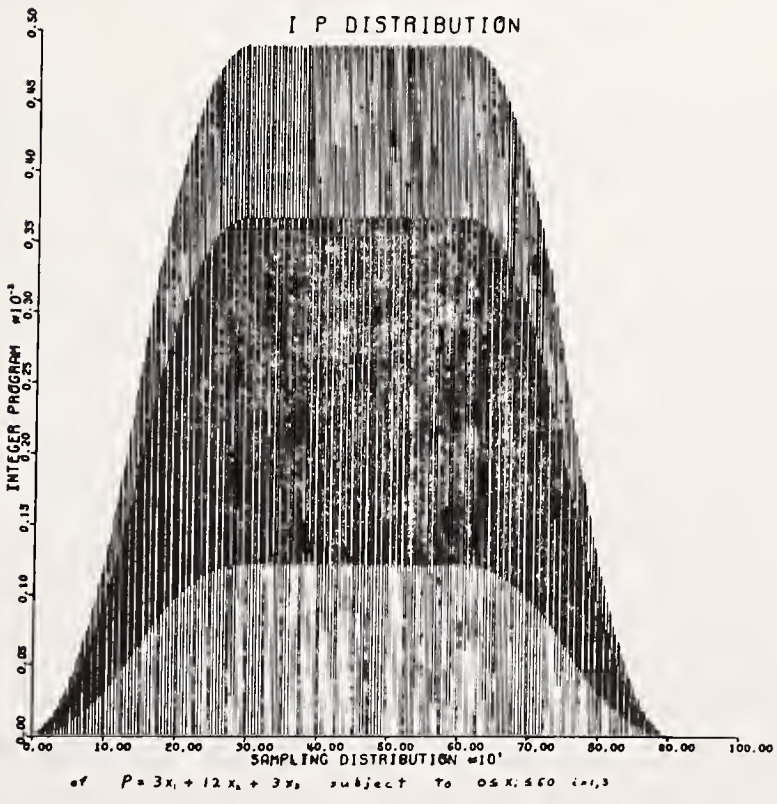


of $P = 5x_1 + 10x_2 + 5x_3 + 10x_4 + 5x_5$ subject to $0 \leq x_i \leq 10 \quad i = 1,5$



of $P = 6x_1 + 9x_2^2 + 7x_3 + 4x_4^2 + 2x_5$ subject to $0 \leq x_i \leq 10 \quad i = 1,5$


I P DISTRIBUTION
INTEGER PROGRAM *10⁻³
0.00
0.08
0.16
0.24
0.32
0.40
0.48
0.56
0.64
0.72
0.80
0.00
40.00
80.00
120.00
160.00
200.00
240.00
280.00
320.00
360.00
400.00
SAMPLING DISTRIBUTION
of P = 6x₁ + 7x₂ + 12x₃ + 7x₄ + 6x₅ subject to 0 ≤ xᵢ ≤ 10 i = 1, 5


I P DISTRIBUTION
INTEGER PROGRAM *10⁻³
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45
0.50
0.00
10.00
20.00
30.00
40.00
50.00
60.00
70.00
80.00
90.00
100.00
SAMPLING DISTRIBUTION *10¹
of P = 3x₁ + 12x₂ + 3x₃ subject to 0 ≤ xᵢ ≤ 60 i = 1, 3

This does not by any means prove our contention. However, the well behaved nature of the distributions does tend to reassure one that similar results would follow for large integer programming problems if those results were obtainable. Even if the contention isn't justified in every case, one still will have the best solution of a million or ten million possible answers, depending on how long the program is run.

Also sensitivity analysis can be done by checking the points around the random sample optimum to discover if a better answer is close by. Other variations can be used to improve the answer. In addition, if the constraints and/or objective functions are subject to slight variations, the random sampling approach is more likely to produce a solution that will be valid with these variations than a theoretical solution that is frequently near a "corner" of the constraint region.

5. CONCLUSION

With the recent and future advances in capacity, speed and miniaturization of computers we believe this technique will be a promising alternative when the theory approach to integer programming problems becomes complex.

6. REFERENCE

Conley, W. C. and Tracy, D. S. (1976). Small sample sampling distributions. Proceedings in Computational Statistics: COMPSTAT. Physica-Verlag, Wurzburg-Vienna.

BIOGRAPHIES

William Conley received a Ph.D. in mathematics from the University of Windsor in 1976. For the past four years, he has been teaching in the Faculty of Business at Windsor. Conley, who specializes in computer mathematics research, has just been appointed Director of Quantitative Methods in the Managerial Systems Concentration at the University of Wisconsin - Green Bay.

Derrick S. Tracy received a science doctorate in statistics from the University of Michigan in 1963. Tracy, a Fulbright Fellow, has been a distinguished researcher, lecturer and teacher of theoretical statistics for many years. He is Professor of Mathematics at the University of Windsor.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

A SYSTEM FOR DICTIONARY-DRIVEN DATA ENTRY USING AN INTELLIGENT TERMINAL

Brent A. Blumenstein and Robert K. O'Day
Emory University, Department of Biometry, Atlanta, Georgia 30322

ABSTRACT

This paper discusses a generalized dictionary-driven data entry system which runs on an intelligent terminal in conjunction with a large-scale data base computer. An Input Control Program (ICP) controls the sequencing, branching and edit checking. The ICP is interpreted by a data-independent BASIC program which runs on the intelligent terminal. Data are entered, verified, and recorded on diskettes for later transmittal to the data base computer. The ICP is prepared for the intelligent terminal on the data base computer by a program which runs under control of a data base dictionary. This system of data entry has been found to have significant advantages over more traditional approaches to data entry. The main advantages include flexibility and an orientation to the goal of data analysis. This paper presents an analysis of these advantages and a favorable cost comparison over card punching for one large-scale data entry problem.

Key words: Data base dictionary; data editing; data entry; data independence; data management; distributed processing; intelligent terminal; statistical data systems.

1. INTRODUCTION

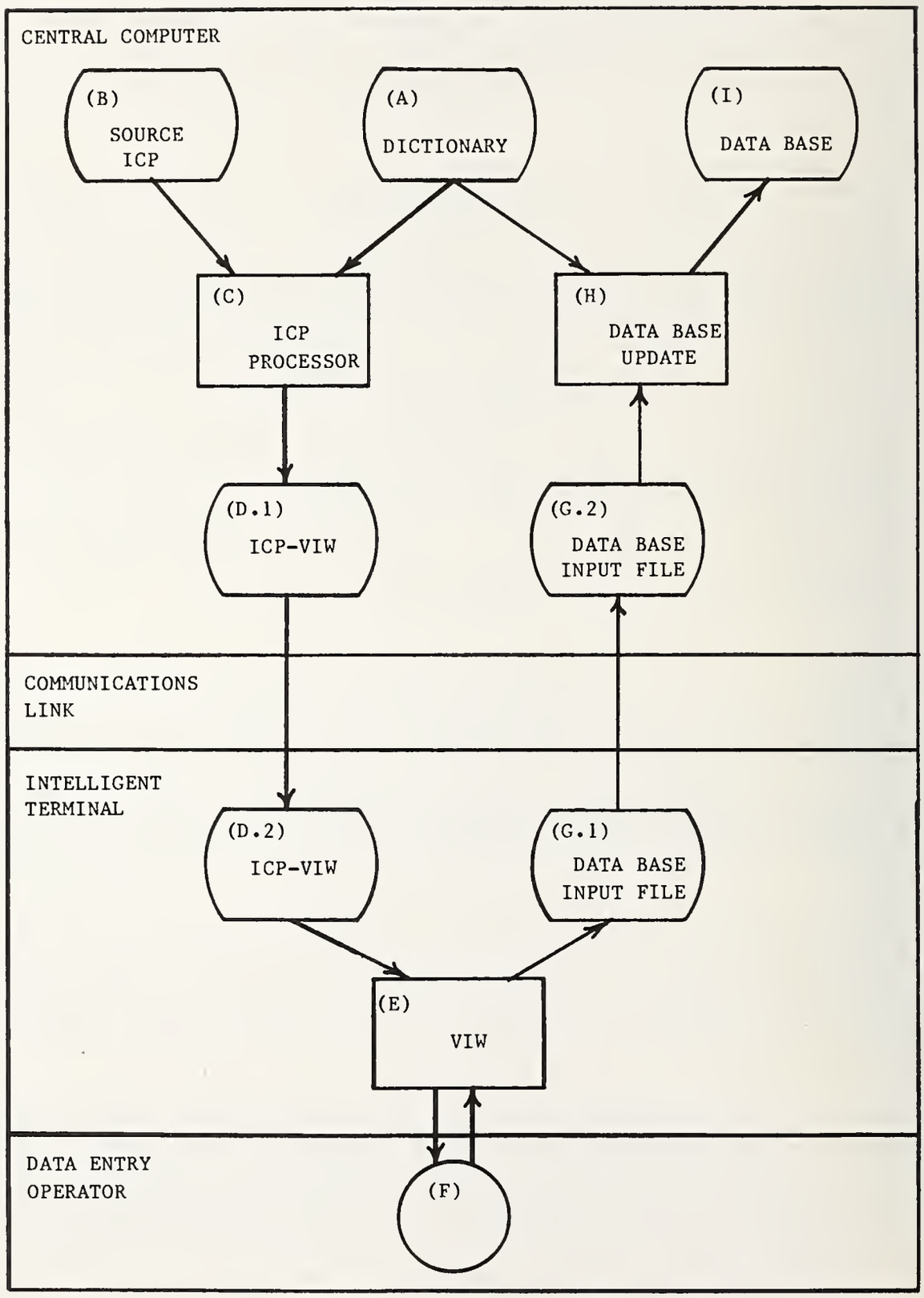
This paper describes an intelligent terminal data entry extension to the Dictionary-Driven Datasystem (DDD) described in Blumenstein (1976). DDD is a general purpose data system providing within entity content flexibility through the use of a data base dictionary. The primary goals of DDD are to facilitate the entry and maintenance of data in a flexible and easily extendible data base and to allow for the extraction of selected data in a form compatible with statistical analysis software and report programs.

The DDD within-entity content flexibility allows the set of attributes on which values exist to vary from entity to entity. However, control of a minimum set of attribute values which must exist is possible. The introduction of a new attribute into the data base is accomplished simply by adding the definition of the attributes to the dictionary and modifying the input program(s) to request values of the new attribute. This data system design has already proven itself to be successful on several large-scale longitudinal data bases.

One of the most useful components of DDD is VIC, a conversational value input program. VIC is controlled by an ordered list of attribute names called an Input Control Program (ICP). The ICP is interpreted by VIC and causes requests for attribute values to be made to the terminal. The values are edited under control of the dictionary as they are entered and a second entry for verification is manditory. The value input procedure is modified (for example, an order change or the addition of a new attribute) simply by changing the ICP.

There are two problems in using VIC: (1) it is expensive to run and (2) its operational efficiency is affected by response time degradation of the multi-user central computer on

FIGURE 1: SYSTEM SCHEMATIC



which it runs. The motivation to develop the intelligent terminal data entry system came from a desire to overcome these two problems.

The intelligent terminal used consists of a central processing unit with 16K bytes of memory, a very fast video display, an upper/lower case keyboard, an audible signal and two diskette handlers. A diskette holds 262,262 bytes. In addition, a printer is necessary to facilitate program development. The software available is a very advanced and augmented BASIC interpreter. A detailed description of the intelligent terminal is found in the manufacturers reference manual (1975).

2. OPERATION OF THE INTELLIGENT TERMINAL DATA ENTRY SYSTEM

Figure 1 is a schematic of the relationship between the central computer and the intelligent terminal relative to the operation of the intelligent terminal data entry system. The intelligent terminal version of VIC is named VIW. The dictionary file (A) is the DDD dictionary for the data base being processed. The dictionary is maintained using the DDD definition facility. The ICP source file (B) is created and modified using the general purpose file editor available in the central computer. The ICP processor (C) interprets the source ICP, fetches the required value editing specifications from the DDD dictionary and creates the file ICPVIW (D.1), the VIW version of the ICP on the central computer. Hence, the ICPVIW contains both control logic and value editing specifications. The ICPVIW is transmitted to the intelligent terminal using a telephone linkage. The ICPVIW may not be modified except by modifying the ICP or the dictionary and re-running the ICP processor. Hence, the dictionary is the only source of attribute value editing specifications and the source ICP will always represent the control logic of the input procedure as it will run on the intelligent terminal. The transmission of the ICPVIW to the intelligent terminal is done one time.

The execution of VIW (E) causes the intelligent terminal version of the ICPVIW (D.2) to be interpreted, attribute value requests to be displayed on the video display, values accepted from the keyboard (F) and a data base input file (G.1) to be written. Periodically (for example, at the end of each day) the intelligent terminal version of the data base input file is transmitted to the central computer. The central computer copy of the data base input file (G.2) is input to the data base update program (H) and causes the data base (I) to be updated.

The design of this intelligent terminal data entry system allows for multiple input procedures for a single data base (a different ICP for each). Furthermore, multiple data bases may also be processed on a single intelligent terminal.

3. DESCRIPTION OF VIW

VIW is independent of the data base being processed. It is an interpretive program written in the BASIC programming language. Like its VIC counterpart, VIW accepts six value types: literal, metric, categorical, indicator, date and time. Each value type is vigorously edited according to its specifications in the data dictionary. These edits include range checks, category code verification, and date structure validation. A single display line is used for each value requested, and each value is prompted by displaying its respective attribute name from the data dictionary. If an error condition is detected by VIW, an appropriate diagnostic is displayed on the screen and the operator is provided with the opportunity for immediate error correction. A programmable audio tone (BEEP) is used to capture the operators attention. Missing values are entered and validated in accordance with the missing value permissions specified in the data dictionary. A second entry or verification is mandatory on all values to test equality with the previously entered value. However, the operator is provided with the capability of batching input, and verifying an entire batch of input at once. If there is a discrepancy during verification, an algorithm is executed which is designed to evoke the intended value from the operator. VIW has many additional features which include: the ability to suspend and subsequently restart data entry, a very tight file integrity mechansim and the ability to cancel the input of a single

entity or batch of entities if necessary. Figure 2 provides an example of part of a VIW run.

Figure 2: VIW run example

```
DCMICMET? hist
DEFAULT ENTERED FOR DCOTHCON.
POINV? y
POSIDE? left
SEQNUM? 1
PSCODE? 742
EDPEOD? &-000-010000
DEFAULT ENTERED FOR EOD1967.
TSREC01? y
TSPROC01? L mod rad mastectomy, quadrant bx Rt breast
VERIFICATION DISCREPENCY. RE-ENTER.
TSPROC01? Lt mod rad mastectomy, quadrant by Rt breast
TSSTAT01? done
```

4. THE INPUT CONTROL PROGRAM

The ICP implements the ordering and control logic for an input procedure. At present, the only control logic provided is the conditional input of values depending on the value of condition codes. The condition codes are set as a result of the execution of ICP commands which can assess the value set inclusion relationships. For example, an ICP can be designed so that the outcome of a given treatment will be requested only if the treatment was actuallly performed. The substitute value for a conditional input is indicated in the conditional input request command. There are also ICP commands which perform logical operations on two or more conditional codes. Figure 3 presents an ICP source code example.

Figure 3: ICP example

```
REQO FORMNUM
REQO SOURCE
RETC 0,1,A
REQC HOSP,0,!NA
REQO LASTNAME
REQO FIRSTNM
REQO SOCREC
REQO DATEADM
REQO DATBIRTH
REQO BIRTHPL
REQO CASAME
RETC 0,1,N
REQC CASTRNUM,0,!NC
REQC CASTREET,0,!NC
REQC CACITY,0,!NC
REQC CASTATE,0,!NC
REQC CAZIP,0,!NC
```

5. EXPERIENCES IN DEVELOPMENT AND USE

The development of this system has not been smooth. The first type of intelligent terminal selected was rejected because of inadequate processing capacity, ineffective language processors and inefficient use of memory by the language processors. The current manufacturer was then selected. However, the first type of processor tried was too slow. The current target intelligent terminal is quite satisfactory.

It is natural to wonder how the cost of using this intelligent terminal data entry system compares with punched card preparation. The project providing the stimulus for the development of this system is concerned with the collection of data on a large number of cancer cases, both incidence and follow-up. The number of attributes on which data are collected exceeds 300. In this data base it is highly desirable to collect a large amount of open ended literal values such as "other diagnostic procedures performed." The monthly cost of the two intelligent terminals and two data entry operators (salary + fringe + overhead) is $3880. The goal is to enter 48 forms per day and this requires 75% utilization of the data entry operators and intelligent terminals. Therefore, the cost per form for data entry is

$$\$3880/(23 \text{ days per month})/(48 \text{ forms per day}) = \$3.51.$$

The estimate for card punching using a card punching service is based on 40 cards per form and a cost of $0.14 per card (local Atlanta prices for an alphanumeric punch job). Hence the cost per form for card punching would be:

$$(40 \text{ cards/form}) \times (\$0.14/\text{card}) = \$5.60$$

Please note the following:

* The total cost of the intelligent terminals and the data entry operators are used in the estimate.

* The estimate of the card preparation cost does not include the cost of editing. The intelligent terminal cost does include a significant amount of editing.

* The card preparation cost estimate could have been computed for an "in-house" operation. However, because of card handling problems this option was rejected.

* Verification discrepancies are corrected more quickly on the intelligent terminal and this is probably the main reason the use of the intelligent terminal costs less.

6. PLANS FOR THE FUTURE

A planned major enhancement to this data system is the implementation of the capability to perform value cross checks within the intelligent terminal. For example, it would be desirable to be able to assure that a sex specific cancer diagnostic procedure indicated as having been performed is valid for the sex of the patient. The implication of this feature will require a major restructuring of VIW so that values may be addressed in a random fashion rather than sequentially as is now the case. Because of the limited capabilities of the intelligent terminal it may be very difficult to accomplish this rewrite. However, the potential saving in processing cost on the central computer is significant.

SUMMARY

An intelligent terminal data entry system has been developed. It is an extension to a dictionary-oriented data management system and therefore benefits from the high degree of content flexibility and data base extendibility inherent in the dictionary-oriented design.

The system implements many desirable data entry features not available using conventional methods and is cost effective for one very complex data entry problem.

8. ACKNOWLEDGEMENTS

This work was sponsored by National Cancer Institute contract NO1-CP-6-1027, to Metropolitan Atlanta Surveillance, Epidemiology and End Results Program.

9. REFERENCES

BLUMENSTEIN, B. A. (1976). American Statistical Association 1976 Proceedings of the Statistical Computing Section. American Statistical Association.

____ (1975). Wang BASIC Language Reference Manual. Wang Laboratories, Inc, Teuksbury, Mass

BIOGRAPHIES

Brent A. Blumenstein received a Ph.D. in statistics from Emory University in 1974 and is an assistant professor of Biometry and Medicine at Emory University. His interests and responsibilities are in the direction of the management and analysis of data from large epidemiological studies. He is currently secretary of the Atlanta Chapter of the American Statistical Association.

Robert O'Day received his M.S. in statistics and computer science from the University of Georgia in 1976. He is the programmer/analyst and statistical technician for the Atlanta Cancer Surveillance Center located at Emory University.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

COMPARISONS OF ALGORITHMS FOR MINIMUM L_p NORM LINEAR REGRESSION

W. J. Kennedy and J. E. Gentle
Iowa State University

ABSTRACT

Minimization of the p-th power of the residuals as a criterion for fitting regression models has been suggested by a number of authors recently. Various algorithms have been proposed for computing these L_p estimators. Some of the more promising algorithms are considered, and computational experience relating to their speed is reported.

Key words: Computer timings; curve fitting; estimation; gradient; Newton-Raphson; perturbation methods; quasi-Newton; simplex method; variable metric method.

1. INTRODUCTION

The usual solution to the common problem of estimation of the parameters in the linear odel has traditionally been to use an estimator that minimizes the sum of squares of the eviations of the observations from their estimated mean values. While these least-squares stimators enjoy optimal properties among certain classes of estimators and/or under some airly weak assumptions, when the class of permissible estimators is extended or when the ssumptions are not met, the least squares estimators may loose some of their appeal.

We consider the linear model,

$$\underline{y} = X\underline{\beta} + \underline{\epsilon}, \tag{1}$$

here $\underline{y}$ is an n-vector of observations, X is an n x m ($n > m$) matrix of constants, $\underline{\beta}$ is an -vector of parameters to be estimated, and $\underline{\epsilon}$ is an n-vector of disturbances. The L_p stimator of $\underline{\beta}$ is a vector $\tilde{\underline{\beta}}$ which is a solution to the problem

$$\min_{\underline{\beta}} \Sigma \left| y_i - \underline{x}_i'\underline{\beta} \right|^p \tag{2}$$

where $\underline{x}_i'$ is the i-th row of X. For p = 2 this is the least squares criterion. In addition to their possible statistically attractive properties, the least squares estimators are particularly simple to compute, being a solution to the consistent linear equations

$$(X'X)\underline{\beta} = X'\underline{y} \tag{3}$$

(although this formulation is not necessarily the best way to obtain these estimators). Other members of the class of L_p estimators, however, may be more desirable from a statistical standpoint than the least squares estimators under various conditions on the model (see, e.g., Forsythe, 1972, or Rice and White, 1964). For $p \neq 2$ the estimators are more difficult to compute.

In recent years a number of algorithms for computing L_p estimates have been proposed. However, little is known about their relative computational efficiency. The authors undertook a study to compare some of the more popular algorithms, for $p > 1$, with regard to computational efficiency.

2. THE ALGORITHMS

The L_p estimation problem (2) is essentially one of unconstrained minimization; hence, there are a number of algorithms available for the computation of the L_p estimates. As an initial classification, we may categorize such algorithms based on the degree of regularity of the objective function that they require, such as convexity, existence of derivatives, etc. In general, the extent to which the optimization method takes advantage of special properties of the objective function is indicative of the time-efficiency of the procedure.

A widely-used algorithm requiring very few conditions on the objective function is the Nelder-Mead method (Nelder and Mead, 1965). In this procedure the objective function is evaluated at the vertices of a simplex and, based on the function values, a new point is chosen to replace one of the simplex vertices in such a way that the sequence of points leads toward the function minimum. This and other direct search procedures would not be expected to perform as well as some other algorithms that utilize more properties of the objective function (2) for the problem at hand.

Members of the class of gradient procedures, when applicable, should perform more efficiently. Letting $r_i = y_i - \underline{x}_i'\underline{\beta}$ in (2), we have the i-th component of the gradient,

$$-p \sum_{i=1}^{n} |r_i|^{p-2}(y_i - \underline{x}_i'\underline{\beta})x_{ij}, \tag{4}$$

which, when equated to zero, gives a weighted least squares problem, aside from the presence of $\underline{\beta}$ in the r_i. The following iterative procedure is immediately suggested.

Solve

$$X'R^{(k)}X\underline{\beta}^{(k+1)} = X'R^{(k)}\underline{y} \tag{5}$$

where

$$R^{(k)} = \text{diag}\,(|y_i - \underline{x}_i'\underline{\beta}^{(k)}|^{p-2}) \qquad \text{for } K = 1, 2, \ldots$$

and

$$R^{(o)} = I.$$

This procedure was investigated by Fletcher, Grant, and Hebden (1971) and found to diverge for $p > 2$. If $p < 2$, the weights in (5) may become infinite. Merle and Spath (1974) investigated this algorithm and found it to converge for $1 < p < 2$ on all problems they considered, when they assigned to any quantity $|y_i - \underline{x}_i'\underline{\beta}^{(k)}|$ less than E, the value E, for some small $E > 0$. Following Merle and Spath, we refer to this method as Algorithm 1.

A straightforward application of the Newton-Raphson method using second derivatives of (2) gives the iterative procedure, solve

$$X'R^{(k)}X\underline{\delta}^{(k+1)} = X'R^{(k)}\underline{y} \tag{6}$$

and take $\underline{\beta}^{(k+1)} = [(p-2)\underline{\beta}^{(k)} + \underline{\delta}^{(k+1)}]/(p-1)$, where $R^{(k)}$ is as in Algorithm 1. Again following Merle and Spath, we refer to this procedure as Algorithm 2. This algorithm was studied by Gentleman (1965), Fletcher, Grant, and Hebden (1971), Kahng (1972), and Rey (1975), among others. As long as the Hessian matrix remains positive definite, the algorithm is known to converge. In the L_p estimation problem for $p > 2$, this requirement is satisfied if no more than $n - m$ residuals are equal to 0 at any stage.

To overcome the possible problems of a singular Hessian matrix, Ekblom (1973) introduced a perturbation in problem (2), yielding the objective function

$$\min \Sigma\,[(y_i - \underline{x}_i'\underline{\beta})^2 + e^2]^{p/2}, \tag{7}$$

and suggested using a Newton-Raphson method on a sequence of problems in which e^2 is decreased to zero. In addition, Ekblom recommended a Goldstein-Armijo steplength in (6) instead of the constant $(p-2)/(p-1)$. Ekblom's modification allows the procedure to perform effectively for all $p > 1$.

Algorithms 1 and 2 and Ekblom's algorithm all make use of the normal equations (5). An essential difference is in the method of updating the solution, Let

$$\underline{\beta}^{(k+1)} = \underline{\beta}^{(k)} + \gamma^{(k+1)}\underline{\sigma}^{(k+1)}, \tag{8}$$

where

$$\underline{\sigma}^{(k+1)} = (X'R^{(k)}X)^{-1}X'\underline{w}^{(k)}$$

with $R^{(k)}$ as before and

$$w_i^{(k)} = |y_i - \underline{x}_i'\underline{\beta}^{(k)}|^{p-1} \text{ sign } (y_i - \underline{x}_i'\underline{\beta}^{(k)}).$$

Then $\gamma^{(k+1)} = 1$ gives Algorithm 1, $\gamma^{(k+1)} = \frac{1}{p-1}$ gives Algorithm 2, and $\gamma^{(k+1)}$ set to the Goldstein-Armijo steplength divided by (p-1) gives the interior loop of Ekblom's algorithm.

Another gradient procedure applicable for any value of $p > 1$ is the Davidon-Fletcher-Powell method (Fletcher and Powell, 1963). This widely-used algorithm is one of the most efficient of the class of gradient procedures known as variable metric or quasi-Newton methods.

3. TIMING COMPARISONS

The five algorithms described in Section 2 were implemented in FORTRAN using double precision and run on an IBM 360/65 for various artificial data sets. Available codings known to be generally efficient were used when available. The Nelder-Mead implementation by O'Neill (1971) (with the modification and corrections given in subsequent issues of Applied Statistics) was used in some preliminary timing trials, but was found to be very time consuming relative to the other algorithms. For example, with p = 3.5, n = 20, and m = 5, the Nelder-Mead procedure required approximately six times as much CPU time as the modified Newton method (Algorithm 2) and, with p = 1.5, n = 20, and m = 5, required over three times as long as Davidon-Fletcher-Powell.

The IBM (1968) SSP implementation, DFMFP, of the Davidon-Fletcher-Powell method was used. For the other three algorithms the authors wrote a subroutine, LPFIT, incorporating a least squares procedure HFTI and associated routines given by Lawson and Hanson (1974). A key given to LPFIT determined whether Algorithm 1, i.e., a weight of 1 in (8), Algorithm 2, i.e., a weight of 1/(p-1) in (8), or Ekblom's method, i.e., a weight of $\delta^{(k+1)}/(p-1)$ in (8), with $\delta^{(k+1)}$ being the Goldstein-Armijo steplength, and a sequence of values of e, was to be used in the computations. Residuals less in absolute value than a small tolerance were set to a small positive number. In the Ekblom method, e was set to 100 initially and was decreased by a factor of 1/100 for three iterations.

The various tolerances and convergence criteria of the algorithms were tuned so as generally to give seven place accuracy on well-conditioned data. Table 1 gives the CPU times, in hundredths of seconds, for the four algorithms for various well-conditioned data sets with n observations and m independent variables (including a constant) and for various values of p.

TABLE 1

CPU Times in Hundredths of Seconds for Four L_p Algorithms

N	M	P	DFMFP	Algorithm 1	Algorithm 2	Ekblom
20	5	1.25	86	145	*	115
40	5	1.25	253	380	*	377
40	10	1.25	310	573	*	365
20	5	1.50	100	73	35	114
40	5	1.50	157	216	78	180
40	10	1.50	271	271	191	318
20	5	3.50	44	*	24	84
40	5	3.50	144	*	46	154
40	10	3.50	227	*	100	360
20	5	7.50	117	*	33	91
40	5	7.50	348	*	89	198
40	10	7.50	501	*	157	397

* -- process did not converge

4. DISCUSSION

Investigation of CPU time for the various algorithms, as shown in Table 1, points to the need for consideration of two cases defined by the user's situation.

First, if a general purpose algorithm, applicable for all $p > 1$, is desired, then the only two candidates are the Fletcher-Powell and Ekblom algorithms. Programs based on these algorithms required roughly the same amount of CPU time in execution of the test datasets. Since Ekblom's algorithm allows for more user control over the iteration, it seems preferable to the authors, particularly for larger p values. Also, we suspect that for very large p the Fletcher-Powell algorithm will not compare favorably with Ekblom's algorithm.

Secondly, if the user is only interested in p values near p = 2, say $1.5 \leq p \leq 7.5$, then algorithm 2 (key = 2) seems to be a logical choice since it is significantly faster. However, the user must be aware of the fact that proof of convergence has not been found for the range $1.5 \leq p < 2$. Also, it must be expected that as p increases above 7.5, this algorithm will begin to perform less well in comparison with the Ekblom algorithm.

5. REFERENCES

EKBLOM, HAKAN. (1973). Calculation of linear best L_p-Approximations. BIT 13, 292-300.

FLETCHER, R., GRANT, J. A., and HEBDEN, M. D. (1971). The calculation of linear best L_p approximations. Computer Journal 14, 276-279.

FLETCHER, R. and POWELL, M. D. J. (1963). A rapidly convergent descent method for minimization. Computer Journal 6, 163-168.

FORSYTHE, ALAN B. (1972). Robust estimation of straight line regression coefficients by minimizing p-th power deviations. Technometrics 14, 159-166.

GENTLEMAN, W. M. (1965). Robust estimation of multivariate location by minimizing p-th power deviations. Ph.D. Dissertation, Princeton University.

IBM Corporation. (1968). IBM System/360 Scientific Subroutine Package. IBM Systems Reference Library, White Plains, NY.

KAHNG, S. W. (1972). Best L_p approximation. Mathematics of Computation 26, 505-508.

LAWSON, C. L. and HANSON, R. D. (1974). Solving Least Squares Problems. Prentice-Hall, Englewood Cliffs, NJ.

MERLE, G. and SPATH, H. (1974). Computational experiences with discrete L_p-approximation. Computing 12, 315-321.

NELDER, J. A. and MEAD, R. (1965). A simplex method for function minimization. Computer Journal 7, 308-313.

O'NEILL, R. (1971). Algorithm AS47--function minimization using a simples procedure. Applied Statistics 20, 338-345.

REY, W. (1975). On least p-th power methods in multiple regressions and location estimations. BIT 15, 174-185.

RICE, JOHN R. and WHITE, JOHN S. (1964). Norms for smoothing and estimation. SIAM Review 6, 243-256.

BIOGRAPHIES

William Kennedy received a Ph.D. in statistics from Iowa State University in 1970. For the past seven years he has been Professor-in-charge of the Statistical Numerical Analysis and Data Processing Section of the Statistical Laboratory at Iowa State

James E. Gentle received the Ph.D. in statistics from Texas A & M University in 1974. Since then he has been assistant professor in the Department of Statistics at Iowa State.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

THE METHOD OF MIDPOINTS

Frances Yu Lu
Biola College, La Mirada, California 90639

ABSTRACT

This Mathematical Model is used for finding an estimated regression line by successive midpoints. It can be applied to computer science, statistics and operations research.

Suppose the set of n points (x_1,y_1), $\ldots(x_n,y_n)$ is given, then the estimated regression line can be determined by the kth set of two midpoints, $M_k = [(x_{k_1},y_{k_1}), (x_{k_2},y_{k_2})]$, $k = n-2$,

and
$$\begin{cases} x_{k_1} = (1/2^k)\left[\sum_{i=0}^{K}\binom{K}{i} x_{i+1}\right] \\ y_{k_1} = (1/2^k)\left[\sum_{i=0}^{K}\binom{K}{i} y_{i+1}\right] \end{cases} \quad \begin{cases} x_{k_2} = (1/2^k)\left[\sum_{i=0}^{K}\binom{K}{i} x_{i+2}\right] \\ y_{k_2} = (1/2^k)\left[\sum_{i=0}^{K}\binom{K}{i} y_{i+2}\right] \end{cases}$$

This method would benefit both statistics and computer science in the following ways: (1) The Mathematical Model can be derived easily without using any calculus; (2) The model may be used as an example for teaching "Model Building"; (3) It is easy to show the estimated regression line by graphing and making the calculations by a simple table; (4) It is a simple example for learning computer programming by using the FACT(N) and COMBINATION subroutines.

Keywords: Binomial coefficients; COMBINATION subroutine; FACT(N) subroutine; estimated regression equation; estimated regression line; least-squares prediction equation; Mathematical Model; midpoints; midpoints prediction equation.

1. INTRODUCTION

How to build a Mathematical Model is one of the interesting topics in applied mathematics. A real world problem is given for showing the process of deriving the model. Some examples are illustrated for presentation of the techniques and applications. Then some of the results are checked by the method of least-squares.

2. A REAL WORLD PROBLEM

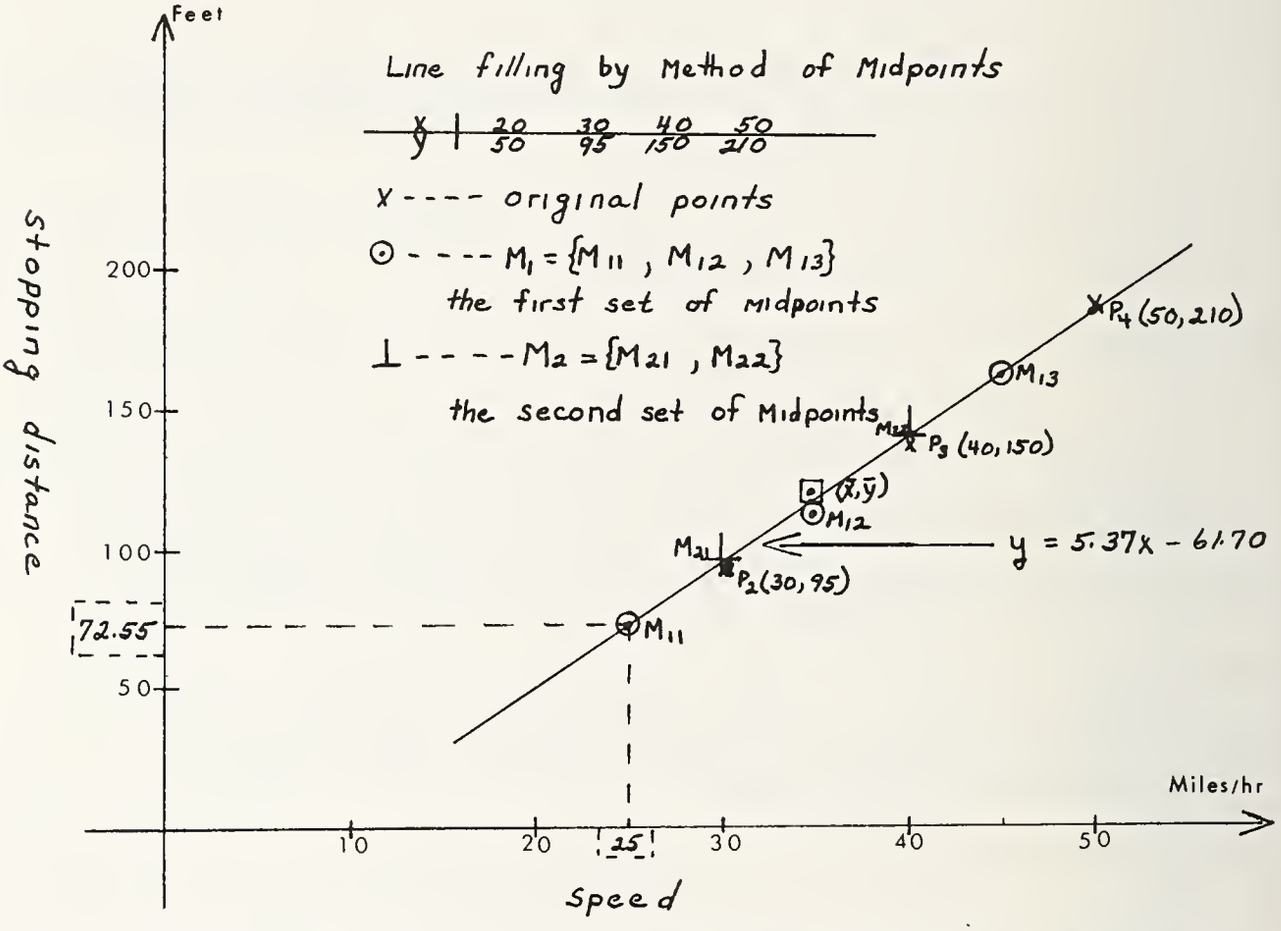
2.1 Problem. Estimate the stopping distance of the car traveling at 25 miles per hour from the following data:

Speed (miles/hour) of the car	20	30	40	50
Stopping distance (feet)	50	95	150	210

2.2 Solution (by graphing). In this figure: let x=speed and y=stopping distance.

$P_i(X_i, Y_i)$ are the points on the xy-plane where i=1,2,3,4. Draw $\overline{P_1P_2}$, $\overline{P_2P_3}$, $\overline{P_3P_4}$ and find the midpoints from each segment. We get three points: $M_{11} = (25,72.5)$, $M_{12} = (35,122.5)$ and $M_{13} = (45,180)$. They belong to the set M_1. Then we find the two midpoints of $\overline{M_{11}M_{12}}$ and $\overline{M_{12}M_{13}}$ respectively: $M_{21} = (30,97.5)$ and $M_{22} = (40,151.25)$, which belong to the set M_2. Finally, use these two points to determine a line. Since the line does not pass through the mean $(\bar{x},\bar{y})$ (where $\bar{x}$=35 and $\bar{y}$=126.25) we can make another line pass through the point (35,126.25) by using M_{21} and M_{22} as the slope. Then we obtain the predicted regression line which equation is:

$$y = 5.37x - 61.70 \qquad (1)$$



From this predicted midpoints line we estimate the stopping distance of the car traveling at 25 miles/hour is 72.55 feet.

2.3 Comparison. The least-squares prediction equation is

$$y = 5.35x - 61.00 \qquad (2)$$

The stopping distance of the car is 72.75 feet when derived from equation (2). Thus the results from these two methods are approximately equal.

2.4 Solution by using vectors. We may solve the problem by using vectors as follows:

Let P_i be the set of vectors, M_1 = the set of 1st midpoints, M_2 = the set of 2nd midpoints.

$$P_i: \quad \underset{(20,50)}{P_1} \qquad \underset{(30,95)}{P_2} \qquad \underset{(40,150)}{P_3} \qquad \underset{(50,210)}{P_4}$$

$$M_1: \quad \left(\frac{20+30}{2}, \frac{50+95}{2}\right), \left(\frac{30+40}{2}, \frac{95+150}{2}\right), \left(\frac{40+50}{2}, \frac{150+210}{2}\right)$$

$$M_2: \quad \underbrace{\left(\frac{20+2(30)+40}{2^2}, \frac{50+2(95)+150}{2^2}\right)}_{M_{21}*}, \underbrace{\left(\frac{30+2(40+50}{2^2}, \frac{95+2(150)+210}{2^2}\right)}_{M_{22}}$$

*We can get M_{21} directly from the data i.e. in M_{21}:

$$x = \frac{(1)\ (20) + 2(30) + (1)\ (40)}{2^2}, \qquad y = \frac{(1)\ (50) + 2(95) + (1)\ (150)}{2^2}$$

or $x = 1/2^2\ [(1,2,1) \bullet (20,30,40)]$, $\qquad y = 1/2^2\ [(1,2,1) \bullet (50,95,150)]$

$\bullet$ means the dot product of two vectors. Similarly, we can calculate x, y in M_{22}.

3. GENERALIZATION

3.1 Generalization for finding the last two midpoints.

If $P_i = \{P_1, P_2, P_3, \text{--} P_n\}$ with $P_1 = (x_1,y_1)$, $P_2 = (x_2,y_2)$, $\text{--}P_n = (x_n,y_n)$ we can make a table as follows:

$$P_i: (x_1,y_1), (x_2,y_2), (x_3,y_3), (x_4,y_4), (x_5,y_5), \text{---} (x_{n-1},y_{n-1}), (x_n,y_n)$$

$$M_1: \left(\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2}\right), \left(\frac{x_2+x_3}{2}, \frac{y_2+y_3}{2}\right), \left(\frac{x_3+x_4}{2}, \frac{y_3+y_4}{2}\right), \text{---}\left(\frac{x \quad +x}{2}, \frac{y \quad +y}{2}\right)$$

$$M_2: \left(\frac{x_1+2x_2+x_3}{2^2}, \frac{y_1+2y_2+y_3}{2^2}\right), \left(\frac{x_2+2x_3+x_4}{2^2}, \frac{y_2+2y_3+y_4}{2^2}\right), \left(\frac{x_3+2x_4+x_5}{2^2}, \frac{y_3+2y_4+y_5}{2^2}\right), \text{----}$$

$$M_3: \underbrace{\left(\frac{x_1+3x_2+3x_3+x_4}{2^3}, \frac{y_1+3y_2+3y_3+y_4}{2^3}\right)}_{M_{31}}, \underbrace{\left(\frac{x_2+3x_3+3x_4+x_5}{2^3}, \frac{y_2+3y_3+3y_4+y_5}{2^3}\right)}_{M_{32}}, \text{----}$$

Now we have a pattern for calculating the last two points:

If $n = 4$ $M_2 = M_{n-2}$ is the end $\qquad n = 5$ $M_3 = M_{n-2}$ is the end.

In M_2 : $x_{21} = \dfrac{(1,2,1) \cdot (x_1,x_2,x_3)}{2^2}$ $\qquad$ Where 1, 2, 1 are the values $\binom{2}{0}, \binom{2}{1}, \binom{2}{2}$ Binomial coefficients in combinations.

When $n = 5$, $M_3 = M_{5-2}$, $x_{31} = \dfrac{x_1+3x_2+3x_3+x_4}{2^3} = \dfrac{(1,3,3,1) \cdot (x_1,x_2,x_3,x_4)}{2^3}$

Where 1,3,3,1 are the values of $\binom{3}{0}, \binom{3}{1}, \binom{3}{2}$ & $\binom{3}{3}$.

- -

- -

When n = n, let k = n - 2

In M_k : $x_{k,1} = ½k[(1,k,½k(k-1),\text{---}1) \bullet (x_1, x_2, \text{----}, x_{n-1})]$

$x_{k,2} = ½k\ [(1,k,½k\ (k-1),\text{---}1) \bullet (x_2, x_3, \text{---}, x_n)]$

$y_{k,1} = \frac{1}{2}k\,[(1,k,\frac{1}{2}k\,(k-1),\text{---}1) \cdot (y_1, y_2, \text{---}, y_{n-1})]$

$y_{k,2} = \frac{1}{2}k\,[(1,k,\text{---},1) \cdot (y_2, y_3, \text{---}, y_n)]$

3.2 The final formulas. Therefore, suppose the set of n points (x_1,y_1), (x_2,y_2), -- (x_n,y_n), is given, then the estimated regression line can be determined by the kth set of two midpoints,

$$M_k = (x_{k1},y_{k1}),\ (x_{k2},y_{k2}),\ k = n-2,\ \binom{K}{i} = \frac{k!}{i!\,(k-i)!},$$

and

$$\begin{cases} x_{k1} = (1/2^k)\left[\sum_{i=0}^{K} \binom{K}{i} x_{i+1}\right] \\ y_{k1} = (1/2^k)\left[\sum_{i=0}^{K} \binom{K}{i} y_{i+1}\right] \end{cases} \qquad \begin{cases} x_{k2} = (1/2^k)\left[\sum_{i=0}^{K} \binom{K}{i} x_{i+2}\right] \\ y_{k2} + (1/2^k)\left[\sum_{i=0}^{K} \binom{K}{i} y_{i+2}\right] \end{cases}$$

4. APPLICATIONS

4.1 Example 1.

Given: x	1	3	4	6	8	9	11	14
y	1	2	3	4	5	7	8	9

The work of this example may be arranged as in the following table

	$\binom{6}{i}x_{i+1}$	$\binom{6}{i}y_{i+1}$	x	y	i	$\binom{6}{i}$	$\binom{6}{i}x_{i+2}$	$\binom{6}{i}y_{i+2}$
	1	1	1	1	0	1		
	18	12	3	2	1	6	3	2
	60	60	4	4	2	15	24	24
M = 8	120	80	6	4	3	20	90	60
k = 6	120	75	8	5	4	15	160	100
	54	42	9	7	5	6	135	105
	11	8	11	8	6	1	66	48
			14	9			14	9
Sum	384	278					492	348

$$M_{61}: \begin{cases} x_{61} = \frac{384}{64} = 6.00 \\ y_{61} = \frac{278}{64} = 4.34 \end{cases} \qquad M_{62}: \begin{cases} x_{62} = \frac{492}{64} = 7.68 \\ y_{62} \quad \frac{348}{64} = 5.44 \end{cases}$$

The points: (6.00, 4.34) and (7.68, 5.44) approximately satisfy the least-square line:

$$y = .545 + .636x$$

4.2 Example 2. To show "The Method of Midpoint" by using computer programming.

Given: x	43	44	36	38	47	40	41	54	37	46
y	74	76	60	68	79	70	71	94	65	78

From the general program (by using the COMBINATION subroutine), we get the results which are shown as follows: where n = 10 and k = 8

x = 0.4171094521E 02 x = 0.4288282021E 02
y = 0.7197267168E 02 y = 0.7417970293E 02

The final midpoints approximately satisfy the least-squares line: $y = 73.5 + 1.68\,(x-42.6)$

4.3 Final remarks. 1) Sometimes a midpoint line does not pass through the point (centroid), $(\bar{x},\bar{y})$. We better make the line passing through $(\bar{x},\bar{y})$ by using the following equation: $y-\bar{y} = [(y_{K2}-y_{K1})/(x_{K2}-x_{K1})](x-\bar{x})$. 2) The "Midpoint Method" is useful for curve smoothing. We can also fit a midpoint parabola. But some of the results are not very good.

5. ACKNOWLEDGMENT

I thank Janet McDougall and Greg Enas for much assistance in computer programming.

6. REFERENCES

Kreyszig, Erwin (1970). Introductory Mathematical Statistics. John Wiley & Sons, Inc.

Newmark, Joseph (1975). Statistics and Probability in Modern Life. Rinehart Press.

Spiegel, M.R. (1961). Theory and Problems of Statistics. McGraw-Hill.

BIOGRAPHY

Frances Yu Lu received a Ph.D. in Mathematics from Ohio State University in 1967 and is a Professor of Mathematical Sciences at Biola College. From 1970 to 1976 she was also chairperson of the department of Mathematical Sciences at Biola College.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

CRITERIA FOR EVALUATION OF INTERACTIVE STATISTICAL PROGRAMS AND PACKAGES

Richard A. Plattsmier
Computation Center, University of Texas at Austin 78712

ABSTRACT

With the growth of interactive computing in general, attention needs to be paid to the quality and style of statistical software written or adapted to on-line computing. Criteria are presented for evaluation of interactive statistical software which may be of use to both designers and purchasers of such software.

Key words: Conversational computing; interactive computing; online computing; statistical programs.

Statistical processing, like many other computing genres, is showing no signs of foregoing the advantages of interactive computing. The ability of a student or researcher to sit down at a remote terminal, key in a few, simple, natural language commands, and have available, at the terminal, summary, descriptive and inferential statistical analyses is not to be denied. Interactive statistical processing (ISP) is particularly useful in instructional applications, both in the teaching of statistics itself and in "Statistical Analysis in" type courses in many academic fields. The student is relieved of the burden of translating specific assignments into program instructions into punched cards into card decks into interpretation by collapsing the intervening steps into a single key-in process.

ISP is inherently different from batch processing in more ways than the difference between a keypunch and an interactive terminal. Interactivity implies two-way communication: the user keying in instructions and receiving responses and the program receiving instructions, translating them, providing diagnoses of errors and responding with pathologies and requested analyses. The interactive terminal is not, then, just a convenient substitute for the keypunch and batch card reader. Of course, it is generally possible to execute batch-type statistical programs in an interactive environment, but this is not what is meant or implied by interactive computing. The interactive processor is one which prompts, asks questions, lexically scans for syntax errors, recovers from errors (usually with a meaningful diagnostic), and provides results at the terminal in a form designed for the interactive terminal.

Some proposed criteria for an ISP are as follows:

1. Generality
2. Conversability
3. Error-recoverability
4. Linguistic style of the program
5. Data restrictions
   --size
   --form
   --on/off line entry

6. Transportability

These criteria are, obviously, interrelated to a great extent but we shall examine each separately (with appropriate references to the interrelatedness).

1. GENERALITY

Generality in any statistical processor is the capability of that processor to perform multiple statistical tasks in the same "job." That is, one should be able to request an analysis of variance and multiple regression on the same data without having to execute multiple separate programs. The BMD and BMDP programs are examples of uniprocessors; SPSS and STATPAK are examples of multiprocessors. Not only should multiprocessing be available, it should be available for distinct subsets of the data.

In addition to multiprocessing, the "ideal" ISP should be able to produce a wide variety of "popularly available" analytic types, specifically summary descriptive statistics, frequency distributions, bi- and multivariate frequency distributions and associational statistics, correlational analyses, regression, factor analysis, analysis of variance, scaling, and perhaps other multivariate treatments such as cluster analysis, discriminant analysis, factor comparison, canonical analysis, and the like.

2. CONVERSABILITY

There is, as previously stated, a difference between running an essentially batch-type processor in an interactive environment and executing a true conversational program. One thinks of a conversation as a two way street: the receiver being more than passive and the sender being not the only active participant. Rather, conversability is a characteristic of an ISP such that the user can provide the program with instructions, the program provide the user with diagnosis of errors in the instructions or (ideally) the results the user desires, and both provide each other general information concerning needs and requirements. This conversability is not just post-hoc error diagnosis with a (usually) cryptic message. Rather, it is approximately real-time syntactic analysis with maximum opportunity for the user to request, at any stage in analysis, additional information from the program concerning the user's options.

3. ERROR-RECOVERABILITY

The ability of a user to converse with an ISP and the ability of the program to recover from syntactic or logical errors on the part of the user are obviously related; the latter is of little value unless the former is also available. The ideal is, of course, for the user to commit no errors of any kind. It is unfortunate that such a number of available statistical programs seem to assume that this will be the case or, if errors do exist in the user's instructions, to give an error diagnostic in the form of a memory "dump."

The process of error-recovery consists of four major steps:
(a) detection
(b) diagnosis
(c) prognosis
(d) prescription

The first step, detection, is the rather straightforward "catching" of (usually syntax) errors. This step is the most crucial to the entire process since a faulty syntax scanner is not likely to produce very meaningful subsequent steps. After an error is detected, the program should promptly notify the user that an error has been encountered, providing both the general "geographic" area of the suspected error and probable cause of the error. The third and fourth steps are branches from the second: if the error is too severe to remedy by internal corrections ("patching"), the program should be able to recognize this and, coupled with the previous diagnostic message, inform the user to completely re-enter the command. If, on the other hand, the error is not so serious, the program should be able to merely request an on-the-spot correction for the faulty part without requiring complete re-entry. In summary, if the program is capable of detecting a "small" error, it should be likewise capable of asking the user to correct this small error without requiring that the user completely re-enter the erroneous command.

4. LINGUISTIC STYLE

The style of command entry for analyses and data retrieval is what comprises the linguistic style of a program. In brief, an ideal ISP should permit the user to enter commands in a natural-language manner, including subject(s), verb(s), object(s) and modifiers. And, for the advanced user, abbreviated syntax should be available. The following example from OMNITAB may serve to illustrate this particular quality:

EXTENDED:

FIT INCOME IN COLUMN 1, USING WEIGHTS OF 1.0, 3 INDEPENDENT VARIABLES IN COLUMNS 2, 3, AND 4, PUT COEFFICIENTS IN COLUMN 5, RESIDUALS IN COLUMN 6, AND STANDARD DEVIATIONS OF PREDICTED VARIABLES IN COLUMN 7

CONCISE:

FIT 1, 1.0, 1, 3, 2, 3, 4, 5, 6, 7

Even this style is rather inflexible, however, since regardless of the "verbiage" inserted, parameters must still follow a particular order. What might be better still would be a scanner which required and recognized a keyword (or appropriate abbreviation) prior to a given parameter instead of requiring a fixed order, viz:

EXTENDED:

FIT 3 DEP VARS IN COLS 2, 3 AND 4 AGAINST INDEP VAR IN COL 1 AND WEIGHTS OF 1.0, PUTTING RESIDS IN COL 6, COEFS IN COL 5 AND SDS IN COL 7.

CONCISE:

FIT 3 DV 2, 3, 4 IV 1 WT 1.0 RESID 6 COEF 5 SDS 7.

This approach is the one generally used by SPSS and the BMDP programs. Certainly it is more helpful for the novice who is likely to be intimidated by the entire concept of timesharing analysis anyway and does not interfere with the flexibility for the more sophisticated user.

5. DATA RESTRICTIONS

5.1 Size. The amount of data which can be analyzed is chiefly a function of two components: machine size and the philosophy of the program used to analyze it. The second component is one which merits the greater attention.

Two competing philosophies exist here and both have merits and faults. The first is that, to insure speed of execution, data should be held in-core in a (typically) elastic segment of high-core. The second is that, to decrease execution-time field length, data should be held out of core in a rapidly accessible (random-access?) disk file. The decision on the part of the program designer of which of these two philosophies to adopt is based upon a number of criteria:

(A) Maximum field length available at execution time;
(B) Queueing algorithm used;
(C) Ease/difficulty of opening, accessing, and closing external data files at execution time; and
(D) Ease/difficulty of expanding/contracting execution field length at execution time.

In general, it may be said that out-of-core data storage is preferable to in-core storage since most timesharing systems utilize field-length size criteria in job queueing. The by-product of this philosophy is that much more data can be held/generated out of core than in core. Given the rapid technological advances in disk storage density and retrieval speed, the rationale for in-core storage promoting more rapid turnaround has been largely obviated.

5.2 Form. Data comes in many forms, ranging from the traditional "rectangular" set usually associated with batch-type card image to structured trees to matrices (correlation, covariance, etc.). The minimum capabilities of the ideal ISP should be the capability to enter any of these several different forms of data without the necessity of either "off-line" preparation (such as sorting, collating, and so forth) or "pre-program" preparation (such as writing a data justifying, "rectangularizing" or "data cleaning" program) prior to statistical analysis by the statistical processor itself.

For non-rectangular data sets, the user should be able to specify a fixed number of records ("cards") per entity and a unique case number for each entity so that the statistical program could re-justify the data to be examined. Matrix input should be permitted for those types of analyses which can make use of this type of input (regression, factor analysis, analysis of variance, and the like) and should be flexible enough to allow different formats for matrices (full matrix, serial string, triangular, etc.).

Admittedly, the greatest amount of statistical analysis is performed on rectangular, fixed-variable, fixed-record, fixed-observation data. Yet many times, this requirement is entirely inappropriate, the raw data resembling a tree much more than a rectangle (such as PUS data). An ideal ISP should, then, be capable of selectively accepting traditional rectangular data as well as tree-type data, if necessary "rectangularizing" the tree (by padding or aggregating) for subsequent analyses.

In addition to the usual numeric input, the ideal ISP should permit alphanumeric input for those circumstances in which it may be appropriate or necessary and should allow the maximum use (of an admittedly limited range) of this kind of data in statistical analyses.

Last, the ideal ISP should permit (and encourage by faster execution time!) alternate types of input to raw data where appropriate such as correlation or variance/covariance matrices into multivariate processors. A user should be able to specify the style of this input (typically generated by the program itself or by other statistical processors but occasionally not) such as full matrix, serial string, triangular, and so forth, and the format of the matrix or alternate input. The key here is maximum flexibility.

5.3 On/off line entry. Quite often, researchers and students are not the originators of the data being analyzed. They may not, then, have much or any control over the recording medium used. The data may reside on magnetic tape or disk from which it must be retrieved by any computer program, statistical or not. The ideal ISP should, therefore, be able to access data from sources external to the user's command terminal.

6. TRANSPORTABILITY

Insofar as statistical programs are concerned, transportability includes both adaptability of the computer code to multiple hardware manufacturer equipment and compatibility of analytic methodologies and report-generation to established (albeit somewhat fuzzy) criteria. The first of these can best be achieved through use of ANSI FORTRAN or COBOL (or other easily adaptable compiler) with as few machine-dependencies as possible (and those unavoidable dependencies clearly and accurately noted). Particular attention should be paid here to inconsistency of word-size and its possible implication for accuracy and report appearance. The second is simply a point that should be made early-on in the design of the program to use accepted (and well documented) techniques in the calculation of particular statistical analyses and to use accepted terminology in report generation and program documentation.

CONCLUSION

What has been examined here are some rather general concepts concerning the design of an ISP. But the user (or purchasing agent or other concerned individual) might keep these criteria in mind when considering purchase of interactive statistical software. And, indeed, these criteria should not be limited to statistical software; any program which purports to be "interactive" should at minimum meet the requirements of conversability and error-recoverability. As use of timesharing systems becomes more prevalent, these criteria will enable users - be they students, researchers or the "general public" - to make easy transitions between the verbal world of statistical analysis and the machine world.

BIOGRAPHY

Richard A. Plattsmier is a Computer Programmer at the Computation Center, The University of Texas at Austin, where he manages the Social Sciences Computing Laboratory. In this capacity, he is responsible for installation, maintenance, and user consultation for statistical software on both the CDC 6600-6400 system and the DEC-10 system. This software consists of SPSS, Statpak, SPSS-10, SPSS/ONLINE, OMNITAB, IMP, BMD, BMDP, OSIRIS and the statistical portions of the IMSL library. He holds B.A. and M.A. degress in Political Science.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

TWO CONCEPTUALIZATIONS OF DISCRIMINANT ANALYSIS AND THEIR IMPLEMENTATION IN COMPUTER PROGRAMS

John Hohwald and Richard M. Heiberger
University of Pennsylvania

ABSTRACT

Examination of Discriminant Analysis computer programs in several widely distributed packages (BMDP, GENSTAT, SAS76, SPSS) reveals that two analyses, termed classification and canonical variate analysis, are subsumed under the one phrase. The paper defines the two techniques, shows their relation, and considers how each of the packages handles the two methods.

Keywords: Canonical variate analysis; classification analysis; discriminant analysis; evaluation of statistical software.

1. INTRODUCTION

Examination of the Discriminant Analysis programs in several widely used statistical packages shows that two distinct but related analyses, which will be called classification analyses and canonical variate analysis, are encompassed by the term discriminant analysis. This paper defines the two techniques and shows their relation. It then reviews the capabilities of four programs (SAS76 DISCRIM, SPSS6 DISCRIMINANT, BMDP7M, GENSTAT CVA) to perform the analyses. It concludes with the observation that the programs emphasize one or the other of the two techniques. The comparison of the programs shows occasional holes in the packages' abilities, some of which can be filled by using other features of the packages.

2. CLASSIFICATION AND CANONICAL VARIATE ANALYSIS

Classification analysis involves techniques designed primarily for classifying observations into groups. Here one supposes that there exists a vector of observations $\underset{\sim}{x}_{(p\times 1)}$, on each sampling unit, and on the basis of such measurements, each sampling unit is to be classified as belonging to one of k distinct and mutually exclusive populations. In canonical variate analysis one seeks to find those dimensions along which the k groups show maximal separation. Geometrically, one seeks a set of axes along which the differences in group centroids are maximum relative to within-groups scatter. The canonical variates that are derived may be thought of as these axes. An outline for the mathematical procedures for achieving these goals is available in Hohwald and Heiberger(1977), the relevant formulas are included in TABLE 2.

3. PACKAGES

The computer programs examined are each part of a widely distributed statistical package as available to the University of Pennsylvania on the UNI-COLL Corporation's IBM 370/168 during summer 1976. They are not necessarily the most recent versions available from the package distributor. The packages were selected for availability and illustrativeness. They are not the only programs which compute discriminant analyses and their selection should not be interpreted as endorsement.

The four programs were compared by studying their manuals and noting the features included. A test data set (Fisher's Iris data) was run on each program with optional output requested. Table 1 lists possible control options and output content expected from programs for discriminant analysis, classification analysis, and canonical variate analysis. The list is a union of features culled from a review of the equations in this paper and those observed in the programs. Additional comments on the packages based on the general characteristics discussed in Francis, Heiberger, and Velleman (1975) and Heiberger(1976) are made in the discussion.

The table indicates that both SAS76 and GENSTAT have sufficiently flexible command languages that complex procedures can be written in a macro form and later used as if they were part of the language. Using a system-provided feature does not require the user to pay attention to either the arithmetic or the formatting details. Writing a macro requires both. Potential macros are included in the table because writing them is a significantly simpler task then writing the same procedure directly in either host language (e.g. Fortran for GENSTAT,PL/I for SAS76).

3.1 SAS76. The SAS76 DISCRIM procedure classifies observations using a generalized square distance measure, $D_i^2(x) = 2\tilde{S}_i$. Here $\tilde{S}_i$ is given either by equation (3) or (4) depending on whether the group covariance matrices or the pooled matrix is used, and with sample estimates of $\underset{\sim}{\Sigma}$ and $\underset{\sim}{\mu}_1$ used instead of the population parameters. The observation x is then classified as belonging to the group which minimizes $D_i^2(\underset{\sim}{x})$.

Table 1 shows that SAS76 provides most of the features needed for classification analyis and very few of those associated with canonical variate analysis. The SAS76 macro facility together with its MATRIX procedure provices sufficient flexibility such that it is possible for a sophisticated user to write a canonical variate analysis macro.

3.2 SPSS6. The SPSS6 SUBPROGRAM DISCRIMINANT is designed for two research objectives: (1) "analysis" to determine whether several populations are statistically distinquishable, and (2) "classification" to determine to which population an observation belongs.

The analysis can be based on a user-determined set of discriminating variables or by a subset of these selected by a stepwise procedure using one of five possible criteria. Either equation (3) or (4) can be used for classification. Classification of observations into groups can be based on all s canonical variates or on only the statistically significant ones - statistical significance being determined by a partitioning and sequential testing of Wilk's lambda, $\hat{\Lambda}$, equations 15-18. Sample misclassification probabilities (equation 5) are computed, but as noted in section 2, should be viewed with some caution. New observations cannot be classified.

SPSS6 DISCRIMINANT computes both the standardized and the unstandardized canonical variate coefficients (equations 9 and 6). The standardized coefficients correspond to the standardized discriminating variables:

$$z_i = \frac{x_i - \mu_i}{\sigma_i} \qquad i=1,2,\ldots,p.$$

Thus, $z_i \sim N(0,1)$ $i=1,2,\ldots,p$, and applying equation (9) to the first equation in (8), one sees that:

$$\underset{\sim}{I} = \underset{\sim}{A}'\underset{\sim\sim}{\Sigma}\underset{\sim}{A} = (\underset{\sim}{A}'\underset{\sim}{D}(\sigma_i))(\underset{\sim}{D}(\sigma_i)^{-1}\underset{\sim\sim}{\Sigma}\underset{\sim}{D}(\sigma_i)^{-1})(\underset{\sim}{D}(\sigma_i)\underset{\sim}{A}) = \underset{\sim}{A}^{(s)'}\underset{\sim\sim}{P}\underset{\sim}{A}^{(s)}$$

where $\underset{\sim}{P}$ is the correlation matrix corresponding to $\underset{\sim}{\Sigma}$. The discussion of the z_i in the manual incorrectly indicates that the z_i are independent rather than correlated.

In sum, therefore, SPSS DISCRIMINANT attempts a combination of both conceptualizations.

3.3 BMDP7M. The BMDP7M program emphasizes a stepwise approach, selecting variables that contribute most to the separation of the groups at each stage; the procedure is outlined in the last paragraph of section 5. The program makes the implicit assumption that all within-group covariance matrices are equal and uses equation (4) for classification. The user has the option of obtaining detailed output for the classification results and significance tests at each stage.

The primary emphasis of the BMDP7M program is on classification, although as Table 1 shows it does compute several pieces of information needed for a canonical variate analysis.

3.4 GENSTAT. The Genstat CVA directive is designed for a canonical variate analysis; and, as can be seen from Table 1, provides most of the needed features. Genstat does not have a classification procedure, although its computational language and macro facility provide sufficient flexibility that one could be constructed by a sophisticated user.

4. COMPARISION

All four programs examined are part of widely distributed statistical packages which include many data handling features and statistical capabilities not mentioned here.

Three of the programs (SAS76 DISCRIM, SPSS6 DISCRIMINANT, BMDP7M) have classification capabilities. SAS76 is the most complete in this respect since it is the only one that can save the classification information and use it to classify additional observations. Both SPSS and BMDP7M can select subsets of the discriminating variates by a stepwise procedure, and SPSS can further select a subset of the canonical variates for use in classification. SAS76 and GENSTAT can do neither. SAS76 and SPSS can accomodate unequal with-group covariance matrices; BMDP7M cannot. Three of the programs (SPSS, BMDP7M, GENSTAT CVA) have canonical variate capabilities. Genstat is the most complete among these. SAS 76 and Genstat both have macro facilities that provide the user with the opportunity to write additional components of analysis.

5. ACKNOWLEDGEMENTS

Statements on specific packages have been confirmed by the package authors.

The research was supported by the National Secience Foundation through grant DCR-75-13994 to the University of Pennsylvania.

6. REFERENCES

AFIFI,A. A., and AZEN, S. P. (1972). Statistical Analysis: A Computer Oriented Approach. Academic Press, New York.

ANDERSON, T.W.(1958). An Introduction to Multivariate Statistical Analysis. John Wiley and Sons, Inc., New York.

BARTLETT, M.S. (1941). "The statistical significance of canonical correlations." Biometrika 32, 29-37.

BARTLETT, M.S. (1947). "Multivariate Analysis." Journal of the Royal Statistical Society, Series B, 9, No. 2, 176-189.

FINN, JEREMY D. (1974). A General Model for Multivariate Analysis. Holt, Rinehart, and Winston, Inc. New York.

FRANCIS, IVOR, HEIBERGER, RICHARD M. AND VELLEMAN, PAUL (1975). "Criteria and considerations in the evaluation of statistical program packages." Amer. Stat., Vol. 29, No. 1, pp. 52-56.

HEIBERGER, RICHARD M. (1976). "Criteria and considerations for computer programs for the analysis of designed experiments." Proceedings of the Ninth Symposium on the Interface of Computer Science and Statistics, Boston.

HOHWALD, JOHN and HEIBERGER, RICHARD M, (1977) "Two Conceptualizations of Discriminant Analysis and their Implementation in Computer Programs," Technical Report 25, Department of Statistics, University of Pennsylvania.

MORRISON, D. F. (1976). Multivariate Statistical Methods (2nd edition), McGraw-Hill, New York.

RAO, C. R. (1973). Linear Statistical Inference and its Applications (2nd edition), John Wiley and Sons, Inc., New York.

Package Manuals

BMDP: Biomedical Computer Programs, W. J. DIXON (editor). University of California Press, 1975.

Genstat: Reference Manual User's Guide, J. A. NELDER, et al. Rothamstead Experimental Station, 1973-1975.

SAS76: A User's Guide,ANTHONY J. BARR, JAMES H. GOODNIGHT, JOHN P. SALL, JANE T. HELWIG, Sparks Press, Raleigh, North Carolina, 1976.

SPSS: Statistical Package for the Social Sciences (2nd edition), NORMAN H. NIE, C. HADLAI HULL, JEAN G. JENKINS, KAREN STEINBRENNER, and DALE H. BENT (editors), McGraw-Hill, New York, 1975.

TABLE 1. Control options and output content of discriminant analysis programs

	SAS76 DISCRIM 76.2	SPSS6 DISCRIMINANT 6.02	BMDP7M July '75	CENSTAT CVA 3.06
General				
1. data transformation capabilities	0	0	0	0
2. means and standard deviations of the discriminating variates, $[\bar{x}_i, s_i]$	0	0	0	0
3. minimum and maximum values of the discriminating variates, $[\min x_{ij}, \max x_{ij}]$	0	A	A	0
4. covariance matrix calculated from all available data	-	0	-	-
5. pooled covariance matrix estimate, $[S]$, or the error sums of squares $[E]$	0	0	0	0
6. covariance matrix for group i, $[S_i]$ i=1,2,...,k	0	0	A	C
7. test for the equality of the group covariance matrices, H_0: $\Sigma_1=\ldots=\Sigma_k$	0	0	-	C
8. ability to detect singularities in the covariance matrix and continue by automatically deleting a variate	0	0	0	0
9. pairwise Mahalanobis distance (D_{ij})	0	-	A	0
Classification Analysis				
1. ability to classify observations using the group covariance matrices $[S_i]$ e.g. (4)	0	0	-	M
2. prior probabilities for membership in group i, q_i, can be specified by the user	0	0	0	MC
3. prints the posterior probability of group membership	0	0	0	MC
4. classification coefficient and constant terms of linear classification function	0	0	0	MC
5. summary table of classification results	0	0	0	MC
6. labels misclassified cases for easy identification	0	0	0	MC
7. classify additional observations in the data set that lacks a group code	0	0	0	MC
8. save classification information for later use	0	-	-	MC
Stepwise options, (#9-14)				
9. Wilk's lambda based on a subset of $\ell<p$ variates, $[\hat{\Lambda}_{(\ell)}]$	M	0	0	MC
10. Rao's F-approximation to $\hat{\Lambda}_{(\ell)}$ with $\ell<p$ variates, (equation 14)	MC	0	0	MC
11. maximize the minimum Mahalanobis distance between two groups	M	0	-	M
12. maximize the minimum F ratio between pairs of groups	M	0	-	M
13. minimize the variance in the set of dummy variables indicating group membership not explained by the predictor variates	M	0	-	M
14. maximize the change in Rao's V, (equation 11)	M	0	-	M
15. user ability to specify a number w<s, of canonical variates to be used in the classification process	MC	0	A	MC
16. summary table of classification results using the jackknife procedure	M	-	0	M

TABLE 1 (continued)

	SAS76 DISCRIM 76.2	SPSS6 DISCRIMINANT 6.02	BMDP7M July '75	CENSTAT CVA 3.06
Canonical Variate Analysis				
1. unstandardized canonical variate coefficients, $[A]$	M	0	0	0
2. standardized coefficients, $[A^{(s)}]$	MC	0	A	0
3. means of canonical variates $[\bar{d}_i]$	MC	0	0	0
4. canonical variate scores, $[d_{ji}]$	MC	0	0	0
5. residuals between the group centroids and the largest canonical variate	MC	-	-	0
6. Wilk's lambda using all available discriminating variates, $[\hat{\Lambda}_{(p)}]$	M	0	0	0
7. partitioning of Wilk's lambda and significance testing using a chi-square approximation, $[\hat{\Lambda}_j]$	M	0	A	0
8. save canonical variables for future use	MC	A	0	0
9. one-dimensional plot of the first canonical variate, $[d_{j1}]$	MA	0	-	A
10. two-dimensional plot of cases on the first two canonical axes $[d_{j1}$ vs. $d_{j2}]$	MA	0	0	A
11. correlation between the canonical variates, d_i, and the set of dummy variables indicating group membership	MA	0	0	M

'0' : the program performs the operation

'-' : the program does not perform the operation

'C' : the operation requires minor calculation command statements

'M' : the operation requires the user to write a macro

'A' : the operation if available in the package (via another program)

'MC' : the operation requires minor calculations once an appropriate macro has been written

'MA' : the operation is available in the package once an appropriate macro has been written

Table 2. Formulas related to classification and canonical variate analysis.

Formula #	Formula	Explanation of Symbols
1.	$S_i = -\{q_1 f_1(x) C(i\|1) + \ldots + q_k f_k(x) C(i\|k)\}$ $i = 1,2,\ldots,k$	$S_i = i^{th}$ discriminant score; $q_i = i^{th}$ prior probability; $f_i(x)$ = probabili density under population i; $C(i\|j)$= cost of missclassification
2.	$S_i = q_i f_i(x)$ $i=1,2,\ldots,k$	i^{th} discriminant score when all costs are equal
3.	$S_i = -\frac{1}{2}\ell n\|\Sigma_i\| - \frac{1}{2}(x-\mu_i)'\Sigma_i^{-1}(x-\mu_i) + \ell n q_i$ $i = 1,2,\ldots,k$	quadratic discriminant function; $\Sigma_i = i^{th}$ covariance matrix; $\mu_i = i^{th}$ mea vector
4.	$S_i = (\mu_i'\Sigma^{-1})x - \frac{1}{2}\mu_i\Sigma^{-1}\mu_i + nq_i$ $i = 1,2,\ldots,k$	linear discriminant function; all Σ_i assumed equal
5.	$\hat{P}(i\|j) = \frac{n_{ij}}{n_j}$ $i \neq j$	estimated missclassification probabilities
6.	$d_{s\times 1} = A'_{s\times p} x_{p\times 1}$	The canonical variates; A defined by formula #8
7.1	$\hat{H} = \{h_{rs}\} = \sum_{i=1}^{k} n_j(\bar{x}_{.jr} - \bar{\bar{x}}_{..r})(\bar{x}_{.js} - \bar{\bar{x}}_{..s})$	hypothesis SSCP matrix with $n_h = k-1$ degrees of freedom
7.2	$\hat{E} = \{e_{rs}\} = \sum_{j=1}^{k}\sum_{i=1}^{n_j}(x_{ijr} - \bar{x}_{.jr})(x_{ijs} - \bar{x}_{.js})$	error SSCP matrix with $n_e = \sum_{j=1}^{k} n_j - k$ degrees of freedom
8.	$A'\Sigma A = I$ $A'HA = \Lambda$ $\Lambda = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_s \end{bmatrix}$	general eigenvalue problem where Σ is estimated by $\frac{1}{n_e}\hat{E}$.
9.	$A^{(s)} = D(\sigma_i) A = \{a_{ji}^{(s)}\}$	Standardized discriminant function coefficients; $D(\sigma_i)$ = diagonal matrix of standard deviations
10.	$V = \sum_{j=1}^{k} n_j(\bar{x}_{j.} - \bar{x}_{..})'S^{-1}(\bar{x}_{j.} - \bar{x}_{..})$	Hotelling generalized T_o^2 statistic; used to test for no overall differences among the k groups along all s dimensions simultaneously.
11.	$\frac{\|\hat{\Sigma}_\Omega\|}{\|\hat{\Sigma}_{\Omega_0}\|} = \frac{\|\hat{E}\|}{\|H+E\|} = \hat{\Lambda} = U = \prod_{i=1}^{s}\left(\frac{1}{1+\hat{\lambda}_i}\right)$	GLR test statistic for testing of no overall differences among groups; $\lambda_i = i^{th}$ largest sample eigenvalue of H in the metric of E
12.	$-(n_e + n_h - \frac{p+k+1}{2}))\ell n\hat{\Lambda} = \chi^2$	approximate central chi-square with pn_h degrees of freedom, used for testing the significance of $\hat{\lambda}$
13.	$R = \left(\frac{1-\hat{\Lambda}^{1/s}}{\hat{\Lambda}^{1/s}}\right)\left(\frac{k_1 k_2 - 2\lambda}{pn_h}\right)$ $k_1 = (n_h + n_e) - \frac{n_h + p + 1}{2}$; $k_2 = \sqrt{\frac{p^2 n_h^2 - 4}{p^2 + n_h^2 - 5}}$ $\lambda = \frac{pn_h - 2}{4}$ $s = \min(p, k-1)$	approximate F statistic with pn_h and $(k_1 k_2 - 2\lambda)$
14.	$\hat{\Lambda}_j = \prod_{i=j}^{s}\left(\frac{1}{1+\hat{\lambda}_i}\right)$ $j<s$	statistic for the remaining j through s eigenvalues assuming 1 through j-1 are significant
15.	$\chi^2 = -(n_e + n_h - \frac{n_h + p + 1}{2}))\ell n\hat{\Lambda}_j$	approximate central chi-square with $(p-j+1)(n_h-j+1)$ degrees of freedom; used for testing the significance of the remaining j through s eigenvalues.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

SIGNIFICANCE ARITHMETIC--A FORTRAN APPROACH

Marietta J. Tretter and G. W. Walster
Penn. State University and University of Wisconsin--Madison

ABSTRACT

The first part of this paper presents a brief history of automatic computer error analysis for uninitiated computer users. A chronological bibliography is also presented for further reference. The second part of the paper briefly describes an improved significance arithmetic system which is implemented by Fortran callable arithmetic routines. The system is never liberal and has special routines to overcome the problem of ultraconservatism. Technical details of this system will appear in a future paper.

Key words: Automatic error monitoring; computer calculations; Fortran error analysis; rounding error; significance arithmetic; significant digit algorithms.

1. INTRODUCTION

This paper consists of two parts: 1) a brief history of automatic error analysis; 2) an improved system of significance arithmetic. Before outlining an improved system of significance arithmetic, it would no doubt be useful for many computer users if a brief history of automatic error analysis, including significance arithmetic, is presented. The numerous articles on this subject span twenty years. To ease the effort required to obtain a quick overview of the literature, a chronologically ordered bibliography is included at the end of this paper.

2. A BRIEF HISTORY OF AUTOMATIC ERROR ANALYSIS

The errors associated with computer computations are traditionally classified as:

> discrepancies due to uncertainties in input data, discrepancies due to the use of approximation formulas, and discrepancies due to the necessity of rounding or otherwise truncating symbolic representations of numbers obtained as computed results, Ashenhurst, 1971.

These errors are referred to respectively as inherent, analytic, and generated errors. The perceived need for some sort of "automatic" analysis of these errors resulted from the concern over the neglect numerical error analysis received from the introduction of floating point arithmetic. Floating point arithmetic eliminates the numerical analysis previously needed to determine the location of the decimal point when fixed point arithmetic was used.

Floating point arithmetic results give absolutely no indication of the number of good, significant, digits. Only the most naive user assumes all digits carried by a computer are good digits. To get an indication of the accuracy of results, some form of error analysis

must be performed. Floating point arithmetic complicates this error analysis with the result that it is rarely performed by the majority of computer users. To those concerned over the neglect of error analysis, it seemed clear that the only way to induce the majority of users to perform this vital analysis was to make it as automatic as the placement of the decimal point.

Basically all attempts at automatic error analysis are classified as significance arithmetic, SA, or interval arithmetic, IA. Cheydleur (1949) originated the concept of significance arithmetic which received the major emphasis in various researcher's attempts to automatically determine error in floating point computations. Sterbenz (1974), refers to significance arithmetic as an automatic method of analysis of rounding error. Significance arithmetic generally incorporates variations on the unnormalized representation of computer numbers--leading zeroes replace non-significant digits in the decimal coefficient. If the coefficient is not normalized, the digits trailing the leading zeroes are significant digits, thus automatically indicating accuracy. Arithmetic operations have to be implemented to produce the correct number of leading zeroes in arithmetic results using unnormalized representations--see Sterbenz (1974) for more details. Note that there is a distinction between unnormalized representations and unnormalized arithmetic operations. In the past, disastrous versions of SA were produced by simply assuming that only unnormalized arithmetic needed to be implemented to produce automatic error analysis. A most important other consideration is the effect of rounding on unnormalized results. Normalization produces a cushion (of bad digits) on the end of a computer word which minimizes the effects of rounding on good digits. With straight unnormalized results, the good digits are on the end of the word and subject to severe effects from rounding.

Unnormalized representation is not needed when SA is implemented by carrying a separate index of significance with each variable in the result, Gray and Harrison (1959). Interval arithmetic carries an interval of significance for each value in a computation and produces an interval result. The wider the interval representing a number, the less accurate the number is. Interval arithmetic is attributed to Moore (1959, 1966). The majority of work on significance arithmetic was accomplished by Ashenhurst and Metropolis and is reported in the numerous articles appearing in the references. Unfortunately, despite the heroic efforts of many individuals, automatic error analysis remains unknown to many floating point computer users.

The cause of the demise of automatic error analysis, especially significance arithmetic, is forensic. None of the proposed systems were completely automatic, and certainly not as mindless to use as floating point arithmetic. Sterbenz (1974), p. 204, indicates several disadvantages of automatic error analysis, the most serious being the often severe loss of digits due to the unnormalized representation's increased sensitivity to round off error. Even more serious, though, for floating point users concerned with precise error analysis, is the fact that error estimates may indicate more good digits than there really are; it can be liberal, Miller (1964). Serious users in lieu of hand error analysis or no analysis often prefer interval arithmetic because it is never liberal. IA, however, is not without serious disadvantages including the fact that it can be very conservative, and that the interval requires two storage locations to represent a number rather than one. Neither SA nor IA can easily or efficiently handle correlated errors which most often affect matrix computations. Thus, for matrix computations, an alternative to SA or IA is gaining favor. The alternative is to select algorithms and techniques that are numerically stable for the computation of interest and then not worry about final error which is usually much less than SA or IA would predict, G. W. Stewart (1973).

The concept of automatic error analysis has not completely vanished from lack of success. Attempts are still being made including those by Metropolis (1976), Stoutmeyer (1977) and the authors. At this point one thing seems clear; it is unreasonable to expect totally automatic error analysis--the problems involved are more complex than the placement of a decimal point. It also seems apparent that large mainframe-builders will never change the architecture to favor automatic error analysis. However, with the advances in microcomputers this possibility should not be completely forgotten. At present, it is reasonable and possible to obtain a software system which is relatively easy to use,

gives the best possible non-liberal results, and is not ultraconservative--it does, however, require some insight on the part of the user.

3. PITHY BITS AND SIGNIFICANCE ARITHMETIC

In light of the bad connotations that SA has acquired for some users, the term pithy bit arithmetic (PA) will be used for the system briefly discussed here. Technical and analytic details of the PA system will appear in another paper. The goal of PA is to retain as many meaningful or pithy digits (bits, when referring to computer representations) as possible while eliminating the possibility of giving liberal accuracy estimates. As it currently exists, PA is implemented by calling Fortran functions that produce modified computer arithmetic. The usual arithmetic symbols are replaced by functions. This system requires little adaptation by programmers when initially coding a routine. It does require recoding of existing programs. In the future it is hoped to modify some Fortran compilers to include a 'type other' declaration which will allow PA variables. A precompiler is another possibility that would be useful for converting existing coding. However, the precompiler might be less desirable as it would allow conversion of sloppy coding and algorithms, which PA should eliminate initially. PA increases running time by 1/3 to 1/2 more than required by straight coding. The programs these estimates were based on contained much "brute force" error analysis so the estimates are probably low for programs not doing any error analysis. This loss of speed seems a small price to pay for knowing the accuracy of results.

Keeping the three sources of error in mind, the basic strategy adopted by PA computing any function is similar to that suggested by Ashenhurst (1965): First, compute the theoretical minimum (inherent) error based on the accuracy of all function arguments and the linear term of the Taylor series expansion of the function; second, normalize all function arguments; third, compute the function using the normalized arguments and pithy bit routines, taking a sufficient number of terms in any approximation to insure that the analytic error is less than the inherent or generated error, whichever is greater; and fourth, if the generated error is less than the inherent error, unnormalize the result to display only pithy bits. This procedure has the advantage that when sufficient word length exists, the returned accuracy of any function is neither liberal nor conservative. When word length is not sufficient, then other adjustments must be made to the PA system. These adjustments are made by special routines available in the PA system.

The PA system uses unnormalized number representation--double or single precision--with appropriate rounding. The interpretation of zeroes--there can be an infinite number of interpretations--is analogous to Carr's "shifting zero" (1959). Algorithms are provided for optimally converting Input/Output from decimal to binary and vice versa.

Estimates are given for the theoretical minimum inherent and generated errors for single and multiple argument function calculations. Strict use of PA and these estimates of error can lead to a severe loss of accuracy due to the sensitivity of unnormalized representations to accumulated rounding error. An example of where this can occur is in summing the series $S = \Sigma\ x^n$, Miller (1964). Previous versions of SA gave liberal estimates of accuracy for this geometric series, which contributed to mistrust of SA. PA is designed to never be liberal in such cases. Obviously, never being liberal can lead to ultraconservatism, i.e., losing all accuracy. PA solves this problem by establishing special routines for handling these recursive calculations. Metropolis (1965), Ashenhurst and others at various times devised similar special routines but they never seemed to be incorporated into a single system. Also, they were unwilling to entirely eliminate liberalism which would have had the effect of forcing the use of special routines (if all digits were not to be lost). The disadvantage of using extra routines is that the user must intervene and decide when a routine is appropriate.

A criticism of the pithy bit system is that it does not specifically take correlated error into account--estimate it. PA does provide some control or knowledge of correlated

errors, in that, using PA in cases where correlated error exists may yield an unusually conservative number of pithy digits. The implication of this result is that the algorithm should be changed or chosen to yield more digits and eliminate correlated error. This is consistent with matrix calculations proposed by G. W. Stewart (1973), with the addition that PA should be used in conjunction with appropriate computational algorithms. Metropolis (1976) proposes variance calculations to estimate correlated error, however the system becomes cumbersome for large matrix calculations. Thus it is believed the PA system offers as practical an approach to correlated error as any existing system.

The major differences between PA and other versions of SA is that it is never liberal; it has routines incorporated into the system which eliminate the ultraconservatism which can result from not being liberal; it allows full use of the computer word without requiring extra storage for variables or accuracy information. The system cannot be used blindly. The user must be aware of what he is doing but does not need to be concerned about liberalism. Further information on the PA system will be available from the authors.

4. REFERENCES

CHEYDLEUR, B. F. (1949). Binary notations in automatic computer algorithms and operation codes. 3rd ACM Nat'l. Conference, 1949 (unpublished).

METROPOLIS, N., ASHENHURST, R. L. (1958). Significant digit computer arithmetic. IRE Trans. on Elect. Comp., 7, 265-267.

CARR, J. W. (1959). Error analysis in floating point arithmetic. Comm. ACM, May, 10-15.

GRAY,H. L., HARRISON, C. (1959). Normalized floating-point arithmetic with an index of significance. Proc. JCC, 16, 244-248.

METROPOLIS, N., ASHENHURST, R. L. (1959). Unnormalized floating point arithmetic. Journ. ACM, 6, 415-428.

MOORE, R. E. (1959). Automatic error analysis in digital computation. Lockheed Tech. Rept. LMSD-48421.

WADEY, W. G. (1960). Floating-point arithmetics. Journ. ACM, 7, 129-139.

ASHENHURST, R. L. (1962). The Maniac III arithmetic system. Proc. JCC, 21, 195-202.

NORDSIECK, A. (1962). Automatic numerical integration of ordinary differential equations. Proc. Symposium on Experimental Arithmetic: Amer. Math. Soc. 241-250.

GOLDSTEIN, M. (1963). Significance arithmetic on a digital computer. Comm. ACM, 6, 111-117.

ASHENHURST, R. L. (1964). Function evaluation in unnormalized arithmetic. Journ. ACM, 11, 168-187.

MILLER, R. H. (1964). An example in "Significant-digit" arithmetic. Comm. ACM, 7, 21.

ASHENHURST, R. L. (1965a). Techniques for automatic error monitoring and control. Error in Digital Computation Vol. I, Rall, ed., Wiley, 43-59.

ASHENHURST, R. L. (1965b). Experimental investigation of unnormalized arithmetic. Error in Digital Computation Vol. II, Rall, ed., Wiley, 3-37.

ASHENHURST, R. L., METROPOLIS, N. (1965). Error estimation in computer calculation. Am. Math. Monthly, 72, 47-58.

KANNER, H. (1965). Number base conversion in a significant digit arithmetic. Journ. ACM, 12, 242-246.

METROPOLIS, N. (1965). Algorithms in unnormalized arithmetic I: recurrence relations. Num. Math., 7, 104-112.

METROPOLIS, N., ASHENHURST, R. L. (1965). Radix conversion in an unnormalized arithmetic system. Math. of Comp., 19, 435-441.

MOORE, R. E. (1966). Interval Analysis, Prentice-Hall.

MENZEL, M., METROPOLIS, N. (1967). Algorithms in unnormalized arithmetic II: unrestricted polynomial evaluation. Num. Math., 10, 451-462.

BLANDFORD, R. C., METROPOLIS, N. (1968). The simulation of two arithmetic structures. Los Alamos Labs. Tech. Rept., LA-3979.

BRIGHT, H. S. (1968). A proposed numerical accuracy control system. Proc. ACM Symposium on Experimental Applied Mathematics, Academic Press, 314-334.

FRASER, M., METROPOLIS, N. (1968). Algorithms in unnormalized arithmetic III: matrix inversion. Num. Math., 12, 416-428.

HANSEN, E. R. (1968). On solving systems of equations using interval arithmetic. Math. Comp., 22, 374-384.

FORSYTHE, G. E. (1970). Pitfalls in computation, or why a math book isn't enough. Stanford Univ. Tech. Rept. CS-147.

GARDINER, V., METROPOLIS, N. (1970a). Significant digit arithmetic on a CDC 6600. Los Alamos Labs. Rept. LA-4470.

GARDINER, V., METROPOLIS, N. (1970b). A comprehensive approach to computer arithmetic. Los Alamos Labs. Rept. LA-4531.

ASHENHURST, R. L. (1971). Number Representation and significance monitoring. Mathematical Software, Rice, ed. Academic Press, 68-92.

BRIGHT, H. S., COLHOUN, B. A., MALLORY, F. B. (1971). A software system for tracing numerical significance during computer program execution. AFIPS: Proc. SJCC, 387-391.

METROPOLIS, N. (1973). Analyzed binary computing. IEEE Trans. on Computers, 22, 573-576.

STEWART, G. W. (1973). Introduction to Matrix Computation, Academic Press.

STERBENZ, P. H. (1974). Floating-Point Computation, Prentice-Hall.

BIVINS, R. L., METROPOLIS, N. (1975). Significance arithmetic: application to a partial differential equation. Los Alamos Labs. Rept. LA-UR-75-1763.

FALTIN,R., METROPOLIS, N., ROSS, B., ROTA, G. C. (1975). The real numbers as a wreath product. Advances in Math., 16, 278-304.

MILLER,W. (1975) Computer search for numerical instability. Journ. ACM, 22, 512-521.

METROPOLIS, N. (1976). Methods of significance arithmetic. Los Alamos Labs. Rept. LA-UR-76-661.

STOUTEMEYER,D. R. (1977). Automatic error analysis using computer algebraic manipulation. ACM Trans. on Math. Software, 3, 26-43.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

DEVELOPMENT OF A COMPUTER TERMINAL BASED INTERACTIVE STATISTICAL ANALYSIS PACKAGE

Richard E. Lund
Montana State University, Bozeman, Montana 59715

ABSTRACT

An interactive package containing about twenty statistical analysis programs has been developed to the user testing stage. Example programs are multiple regression, analysis of variance, chi-square contingency tables, plotting and one and two sample multivariate statistics. Major concepts in program construction and examples of program input-output are provided. The program is written largely in XDS Sigma 7 Extended Fortran but utilizes some assembly language instruction subroutines for input and systems control. It is now being operated on the Montana State University XDS Sigma 7.

Key words: Interactive; program; statistical analysis.

1. GENERAL

A package of interactive programs for statistical analysis is being developed at Montana State University. The current version contains the twenty-five programs listed in table 1 and was recently made available to campus users. The package is written largely in XDS Sigma 7 Extended Fortran but utilizes some assembly language instruction subroutines for input-output and systems control. It is being operated on the Montana State University XDS Sigma 7.

The programs are designed for use by the novice in statistical analysis by computer. Default settings for most parameters enable the beginner to obtain results on simplified data sets without previous instruction. Assistance information is printed on request. But while one goal is to serve the novice, another is to provide sufficient options and flexibility to meet the day-to-day needs on moderate sized data sets of the more sophisticated user.

A major construction goal is to simplify addition of new programs to the package and modernizing old ones as new statistical procedures are developed. Appendage of notes to program output is possible without recompilation of programs. This facilitates references to new tables or recent journal articles (and even tells of bugs in a newly developed program).

A DRIVER program is utilized to select the specified statistical analysis program as well as to provide assistance instructions from a DICTIONARY file upon request. Each statistical analysis program obtains needed values for parameters through an ARGUMENTS subprogram. Data input is also always handled by a separate INPUT subprogram.

The subprogram ARGUMENTS and its associated file DICTIONARY is central to the interactive control of programs by the user as well as to simplications in statistical analysis program construction. A call of the following form for example

```
CALL ARGUMENTS(71,11,12,14,21,16,17,19)
```

types out records 71,11,, ,19 in the DICTIONARY and stores the user's response when appropriate for use by all programs. When the response pertains to setting parameters for input-output, these are automatically set. Table 2 is an example of input-output.

Table 1

Programs in Current Version

DESCRIPTIVE:	
BIPLOT	BIVARIATE PLOTS & SUMMARY STATISTICS
BICOUNT	TWO-WAY FREQUENCY TABLES
HISTOGRAM	HISTOGRAMS & SUMMARY STATISTICS
NPLOT	NORMAL PLOTS
SUMSTAT	SUMMARY STATISTICS
NON-PARAMETRIC:	
NPCORR	RANK CORRELATION TESTS (UNDER CONSTRUCTION)
NPGROUPED	MANN-WHITNEY 2-SAMPLE TEST
NPPAIRED	SIGN, SIGNED-RANK & FRIEDMAN TESTS
PROBABILITY (&INVERSES):	
BINPROB	BINOMIAL PROBABILITY
BPROB	BETA PROBABILITY
CHIPROB	CHISQUARE PROBABILITY
FPROB	F PROBABILITY
NCFPROB	NON-CENTRAL F PROBABILITY
TPROB	STUDENT T PROBABILITY
ZPROB	NORMAL PROBABILITY
ZINVERSE	Z FOR GIVEN NORMAL PROBABILITY
ATTRIBUTE DATA:	
CHISQR1	ONE-WAY CHI-SQUARE ANALYSIS
CHISQR2	TWO-WAY CHI-SQUARE ANALYSIS
ANALYSIS OF GROUP MEANS:	
ANOV1	ONE-WAY ANALYSIS OF VARIANCE
ANOV2	ONE FACTOR ANOV FOR RANDOMIZED BLOCK DESIGNS
TSINGLE	SUMMARY STATISTICS & TESTS FOR SINGLE SAMPLES
TGROUPED	SUMMARY STATISTICS & TESTS COMPARING TWO SAMPLE
TPAIRED	SUMMARY STATISTICS & TESTS FOR PAIRED RESPONSES
COMPARE	MULTIPLE COMPARISONS & CONTRASTS
REGRESSION & CORRELATION:	
MREGRESS	MULTIPLE LINEAR REGRESSION

The current version is written as 2735 fortran records including comments, 465 assembly language records, and 220 dictionary file records. It requires 22K in core exclusive of blank common as an operating module.

Immediate future plans are to develop an output subprogram suitable for use by most statistical analysis programs. It will eliminate much of the duplication in the current version due to each program handling its own output by fortran instructions. Programs to be added include random data generation, more general analysis of variance and covariance programs, and considerable extension of options in multiple regression. As may be expected, progress is dependent upon adequate funding.

Table 2

Example of Input-Output for Histograms

```
ENTER DESIRED PROGRAM OR HELP FOR ASSISTANCE.
>HISTOGRAM

PRODUCES SUMMARY STATISTICS & HISTOGRAMS OF FREQUENCY.

ENTER CONTROL PARAMETERS (* INDICATES DEFAULT).
DATA SOURCE (TTY* OR 8 CHARACTER FILE NAME)      =DATA
OUTPUT DESTINATION (TTY*, LP OR 8 CHAR FILE NAME)=
DATA FORMAT (*=(20G), MAX 40 CHAR. & PARENTHESES)=(T3,3F3.0)
NO. OF INPUT VARIABLES (1* TO 20)                =2
NO. OF VARIABLES USED (N*=NO.INPUT, 1 TO 20)     =
DATA TRANSFORMATIONS DESIRED (YES OR NO*)        =
CORRELATIONS DESIRED (YES OR NO*)                =NO
ENTER CUTPOINTS (UPPER BOUNDS) FOR CLASSES FOR HISTOGRAMS.
    USES 1 TO 10 CUTPOINTS IN ASCENDING ORDER PER VARIABLE
    DEFAULT* USES 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

VAR# :  CUTPOINTS
   1 :  10 20 30 40
   2 :  10 20 30 40 50
NO. OF CASES (N>2 OR N=0* EOF)                   =19
                  25.00      18.00

 NO. OF CASES READ =     19

 VARIABLE             1          2
 MEAN  (N=   19)=  25.84      34.05
 STD DEVIATION  =  9.535      14.21
 SKEWNESS       = -.8041E-01  .6208E-02
 KURTOSIS       = -2.889     -2.915
 MAXIMUM        =  40.00      55.00
 MINIMUM        =  10.00      15.00

 FOR VARIABLE  1:
 UPPER     %    FREQ 0        10        20        30        40        50
 BOUND               +---------+---------+---------+---------+---------+
 10.00       .1    2 +XX
 20.00       .1    2 +XX
 30.00       .4    8 +XXXXXXXX
 40.00       .4    7 +XXXXXXX
                     +---------+---------+---------+---------+---------+
                     SYMBOL X =  1

 FOR VARIABLE  2:
 UPPER     %    FREQ 0        10        20        30        40        50
 BOUND               +---------+---------+---------+---------+---------+
 10.00       .0    0 +
 20.00       .3    5 +XXXXX
 30.00       .2    3 +XXX
 40.00       .3    5 +XXXXX
 50.00       .1    2 +XX
 LAST        .2    4 +XXXX
                     +---------+---------+---------+---------+---------+
                     SYMBOL X =  1

RESTART WITH SAME CONTROL PARAMETERS (YES OR NO*)?
```

BIOGRAPHY

Richard E. Lund received a Ph.D. in Statistics from Iowa State University in 1957. He has fulfilled teaching and consultation duties for the past 8 years at Montana State University.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

MINITAB II, 1977

T. A. Ryan, Jr., B. F. Ryan
The Pennsylvania State University, University Park, Pa. 16802

and B. L. Joiner
University of Wisconsin, Madison, Wisconsin 53706

ABSTRACT

Minitab II is a general purpose statistical computing system, written in machine compatible FORTRAN IV. It is designed especially for students and researchers who have no previous experience with computers. It is very easy to use, flexible, and fairly powerful, and has been found especially useful for exploring data, for plotting, and for regression analysis. In this note we review three aspects of the development of Minitab in the past year.

1. REGRESSION OUTPUT

A new regression output was designed. One crucial aspect of this design is data dependent formats. In this note, we use the word format to mean the number of places printed after the decimal point. Controlling the format based on the data values has many advantages, including:

(1) It allows compact output (no need to allow for 5 decimal places and for numbers of the order of a million at the same time).

(2) The user never sees digits which are statistically (or numerically) meaningless.

(3) The user always sees all the relevant digits, no matter how small or large the data are.

(4) Output printing is faster; this is especially important on typewriter terminals.

The selection of formats is best shown by an example (see Figure 1). Note that the formats are chosen to print the same number of digits for an entire vector or table; this makes comparing of numbers easier. (The number of digits printed is partly a matter of taste - some people might prefer one more digit printed in parts of the output.)

1.1 Notes on the regression output.

(a) The coefficients of the regression equation are usually printed with 4 significant digits (sig. d.). (If the coefficients are too large, exponential format is used.)

(b) In the table of coefficients, a format is chosen (separately for each coefficient) so that the standard deviation of the coefficient is printed out to 3 sig. d. The coefficient is printed with the same format. This insures that just the statistically meaningful digits are printed.

(c) Entries in the AOV and Further AOV tables are all printed with the same format. It is normally chosen so that SS(RESIDUAL) is printed to 3 sig. d. (which allows computing F to about 2 places). (Fewer sig. d. are printed if necessary to avoid printing digits which are numerically meaningless.)

(d) X1 is printed out mainly for identification, so it is only printed to 3 sig. d. (in the largest values).

(e) The format for ST. DEV. PRED. Y is chosen so that ST. DEV. PRED. Y corresponding to an observation located at the centroid of the X values would be printed to 2 sig. d. Then all are printed to $\geq$ 2 sig. d. This same format is used in the printout of Y, Y-hat, and raw residuals. (The standardized residuals are printed with 2 decimal digits.)

(f) The table of X1, Y, Y-hat, etc. has been shortened to 4 rows in this note to save space. The normal output would include all 39 rows.

(g) The amount of output can be controlled with the BRIEF and NOBRIEF commands. The output is arranged so that the most important (and short) parts of the output are first, so if you are using Minitab interactively, you can terminate the output by using the break or attention key when you have the output you need.

(h) A second example is shown in Figure 2. The X1 in this example is the X1 of Example 1 divided by 100; the Y here is the Y of Example 1 multiplied by 10,000.

2. HODGES-LEHMANN ESTIMATES

A MANN-WHITNEY command was added to Minitab. This command does the 2-sample rank test and the corresponding point and confidence interval estimates (Hodges-Lehmann estimates). The algorithm for finding the estimates was developed by J. W. McKean and T. A. Ryan, Jr. (Transactions on Mathematical Software, June 1977, pp. 183-185).

These estimators have very desirable efficiency properties when compared to X-bar and the t-confidence interval. (The efficiency is always at least 86.4%, and can be infinite. Typical values are 95.5% for normal data, 100% for uniform data, and 200% on moderately long-tailed data.)

The traditional way of computing the point and interval estimates is to calculate and order the values (x-y) for every pair with x from the first sample and y from the second sample. Since there are mn such pairs (if there are m x values and n y values) the storage required is large (1,000,000 if m = n = 1,000), and the time required to order the differences is of order mnlog(mn). This method, then, is obviously unsuitable for moderate size data sets.

The McKean-Ryan algorithm finds the estimates by iteration. Let $U(\theta) = \#(y-x \leq \theta)$ be the Mann-Whitney statistic for testing the hypothesis that the difference of the population medians is θ. The Hodges-Lehmann point estimate of θ is defined by the solution of the equation $U(\theta) = mn/2$. (If the hypothesis is true, $E(U) = mn/2$.) Since $U(\theta)$ is monotone and asymptotically linear, the point estimate can be found by a modification of linear interpolation (regula falsi). Modifications of linear interpolation are needed to prevent a large number of interactions in bad cases. The confidence interval is found in a similar manner.

The time required on a typical example, involving two samples of 1,000 observations each, was only 0.2 second (about $0.02) on Penn State's IBM 370/168.

Donald B. Johnson and others have studied an entirely different algorithm for finding the Hodges-Lehmann estimate. Their method has the advantage that the solution can be found

in the order (m+n) log(m+n) time even in the worst case. The principal disadvantage is that it requires 2-3 times as much array storage.

3. PORTABILITY
(especially minicomputers)

A major part of the programming effort in the last year has been to increase the portability of Minitab. The most important advances have been in making Minitab suitable for large minicomputers.

Minitab has been written in standard FORTRAN IV, has been checked by the PFORT verifier, has been installed on many different large computers, and the program, especially the main root, is relatively small, so the implementation on minicomputers is not as difficult as it would be for most statistical programs.

Most minicomputers use 16-bit words for integer variables. Three important implications of this are:

(1) Integer variables must not store large values. (4-digit integers is a safe limit.)

(2) Integer and real word sizes are different, so it is necessary to be careful about word alignment, if real and integer arrays are equivalenced, or common blocks are used for both integer and real variables.

(3) Only two characters can be stored in each integer word.

The first and second problems are relatively simple to solve. The third problem required considerably more effort, particularly with object time formats.

3.1 Packing formats. Minitab makes extensive use of computed (object time) formats for printing. For example, instead of printing the median using a fixed format:

```
      WRITE (IPRINT,10) XMED
10    FORMAT (4X,8HMEDIAN = ,F12.4)
```

we initialize an array KFMT with (4X,8HMEDIAN = ,F12.n). We then replace the n with a character which is computed to print XMED to, say, 5 significant digits, and print the median using

```
WRITE (IPRINT,KFMT) XMED
```

There are difficulties with this approach, however. Suppose we initialize KFMT in a way which is appropriate for an IBM 370, (which stores 4 characters per word). We would then store KFMT in an array of length 7 as follows:

(4X,	8HME	DIAN	=,F	12.	n	)

and KFMT(6) is to be calculated.

Note first that this would not work on a 16-bit computer, which can only store 2 characters per word. The problem is deeper than this however. This format will not work on a computer which stores more than 4 characters per word. For example, on a DECsystem 10, which stores 5 characters per word, each word of the array would have an extra blank added, with the result being

```
(4X, 8HME DIAN  =,F 12.   n   )
```

The extra blanks do not matter, except in the Hollerith field, where they are disasterous.

Thus for a DECsystem 10, we want the format stored

4X,8	HMEDI	AN =,	F12.	n	)

with KFMT(6) to be computed, while on a 2-character per word machine, the format should be stored

(4	X,	8H	ME	DI	AN	=	F1	2.	n	)

which is of dimension 11, and KFMT(10) is to be calculated.

The solution, for Minitab, is to write a "master source" in a pseudo-FORTRAN, which contains a description of the format, with the items to be computed marked with a $ sign. A program, called "the packer", then processes this master source, and produces formats packed for word sizes from 1 to 10. It also creates variables which point to the locations of the items to be computed. The result is a FORTRAN deck with format items suitable for 2-characters per word computers marked with F2 in the first two card columns, etc. The appropriate cards are then selected for the target computer by a small preprocessor, similar in concept to that described by Roald Buhler (7th, 8th Interface Proceedings).

Minicomputers present many other problems, and we are gradually beginning to find out how to solve them. Minitab has recently been installed on several PDP-11 computers with some difficulty (primarily finding a suitable overlay structure) and it has been routinely installed on several HP 3000's.

```
THE REGRESSION EQUATION IS
Y =  0.06090 -0.00268 X1 +0.00122 X2

                                          ST. DEV.     T-RATIO =
       COLUMN       COEFFICIENT          OF COEF.     COEF/S.D.
        --             0.0609             0.0143         4.26
X1      C1           -0.002677          0.000722        -3.71
X2      C2            0.001217          0.000234         5.21

THE ST. DEV. OF Y ABOUT REGRESSION LINE IS
S =     0.00978
WITH (  39- 3) =  36 DEGREES OF FREEDOM

R-SQUARED = 47.3 PERCENT
R-SQUARED = 44.4 PERCENT, ADJUSTED FOR D.F.

ANALYSIS OF VARIANCE

 DUE TO      DF              SS             MS=SS/DF
REGRESSION    2         0.0030901         0.0015450
RESIDUAL     36         0.0034414         0.0000956
TOTAL        38         0.0065314
```

Figure 1. Minitab Regression Output

FURTHER ANALYSIS OF VARIANCE
SS EXPLAINED BY EACH VARIABLE WHEN ENTERED IN THE ORDER GIVEN

DUE TO	DF	SS
REGRESSION	2	0.0030901
C1	1	0.0004981
C2	1	0.0025920

ROW	X1 C1	Y C4	PRED. Y VALUE	ST.DEV. PRED. Y	RESIDUAL	ST.RES.
1	0.48	0.1700	0.1460	0.0038	0.0240	2.66
2	2.73	0.1200	0.1223	0.0022	-0.0023	-0.25
3	2.08	0.1250	0.1235	0.0024	0.0015	0.16
4	0.42	0.1480	0.1340	0.0029	0.0140	1.50

Figure 1. (Continued)

THE REGRESSION EQUATION IS
Y = 609.0 - 2677. X1 + 12.17 X2

	COLUMN	COEFFICIENT	ST. DEV. OF COEF.	T-RATIO = COEF/S.D.
	--	609.	143.	4.26
X1	C3	-2677.	722.	-3.71
X2	C2	12.17	2.34	5.21

THE ST. DEV. OF Y AEOUT REGRESSION LINE IS
S = 97.8
WITH (39- 3) = 36 DEGREES OF FREEDOM

R-SQUARED = 47.3 PERCENT
R-SQUARED = 44.4 PERCENT, ADJUSTED FOR D.F.

ANALYSIS OF VARIANCE

DUE TO	DF	SS	MS=SS/DF
REGRESSION	2	309007.	154504.
RESIDUAL	36	344136.	9559.
TOTAL	38	653143.	

FURTHER ANALYSIS OF VARIANCE
SS EXPLAINED BY EACH VARIABLE WHEN ENTERED IN THE ORDER GIVEN

DUE TO	DF	SS
REGRESSION	2	309007.
C3	1	49807.
C2	1	259201.

ROW	X1 C3	Y C5	PRED. Y VALUE	ST.DEV. PRED. Y	RESIDUAL	ST.RES.
1	0.0048	1700.	1460.	38.	240.	2.66
2	0.0273	1200.	1223.	22.	-23.	-0.25
3	0.0208	1250.	1235.	24.	15.	0.16
4	0.0042	1480.	1340.	29.	140.	1.50

Figure 2. Second Example of Regression Output

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

GR-Z: A System of Graphical Subroutines for Data Analysis

Richard A. Becker

John M. Chambers

Bell Laboratories
Murray Hill, New Jersey 07974

The GR-Z system is a set of FORTRAN subprograms, designed to provide a basis for the graphical operations useful in data analysis and related areas. They provide a wide range of general and specialized statistical graphical operations, and are designed to facilitate both simple graphical computations and the design of new graphical methods.

Features of the system include the following.

- The user has access to a large number of graphical operations, designed to provide powerful, attractive graphical facilities in data analysis, including easy-to-use high-level operations and a wide range of flexible intermediate- and lower-level routines. High-level routines are provided for scatter plots, time-series plots, histograms, probability plots and other graphical operations. The system allows user programs to extend, alter or replace these operations, in a flexible manner.
- The display or page is organized into pictorial components (figure and plot) and related co-ordinate systems which allow graphical operations to be expressed simply and logically.
- A set of *graphical parameters* controls the pictorial results, with sensible default values.
- Extensive documentation exists, including tutorial and reference manuals, and detailed descriptions of individual routines.

Simple GR-Z programs usually consist of a sequence of calls to high-level routines, each of which produces a complete plot; for example,

```
REAL X(50),Y(50)
READ(01)X,Y

CALL BEGINZ
CALL SPLOTZ(X,Y,50)
CALL FINISZ

STOP
END
```

The call to the scatter-plot routine, SPLOTZ, could be replaced with any of the other high-level routines. For example,

```
CALL EEPLTZ(X,50,Y,50)
```

produces an empirical-empirical probability plot of the two sets of data, and

```
CALL HPLOTZ(X,50)
```

produces a histogram of one of the sets. Other simple subroutine calls allow titles, additional lines or points, or other information to be added to the plot.

In a simple program such as the above, the GR-Z system chooses default values for graphical parameters which determine the details of the appearance of the plot. When greater control over the appearance of output is desired, or when users wish to create their own graphical operations, a wide range of additional routines may be used. For example, users can control the layout of figures on a page, the style in which plots are produced, plotting characters and many other characteristics. The parameters which control such features all have system default values, so that the user need not be concerned with them unless special effects are desired.

GR-Z is designed to be highly portable. The source code conforms to a portable subset of standard FORTRAN. Device-dependent code is kept to a minimum and is isolated into a small number of routines. On the other hand, it is possible to improve the efficiency of the system by writing device-dependent routines to replace system routines at a higher level. Operating system dependencies are also isolated and kept to a minimum.

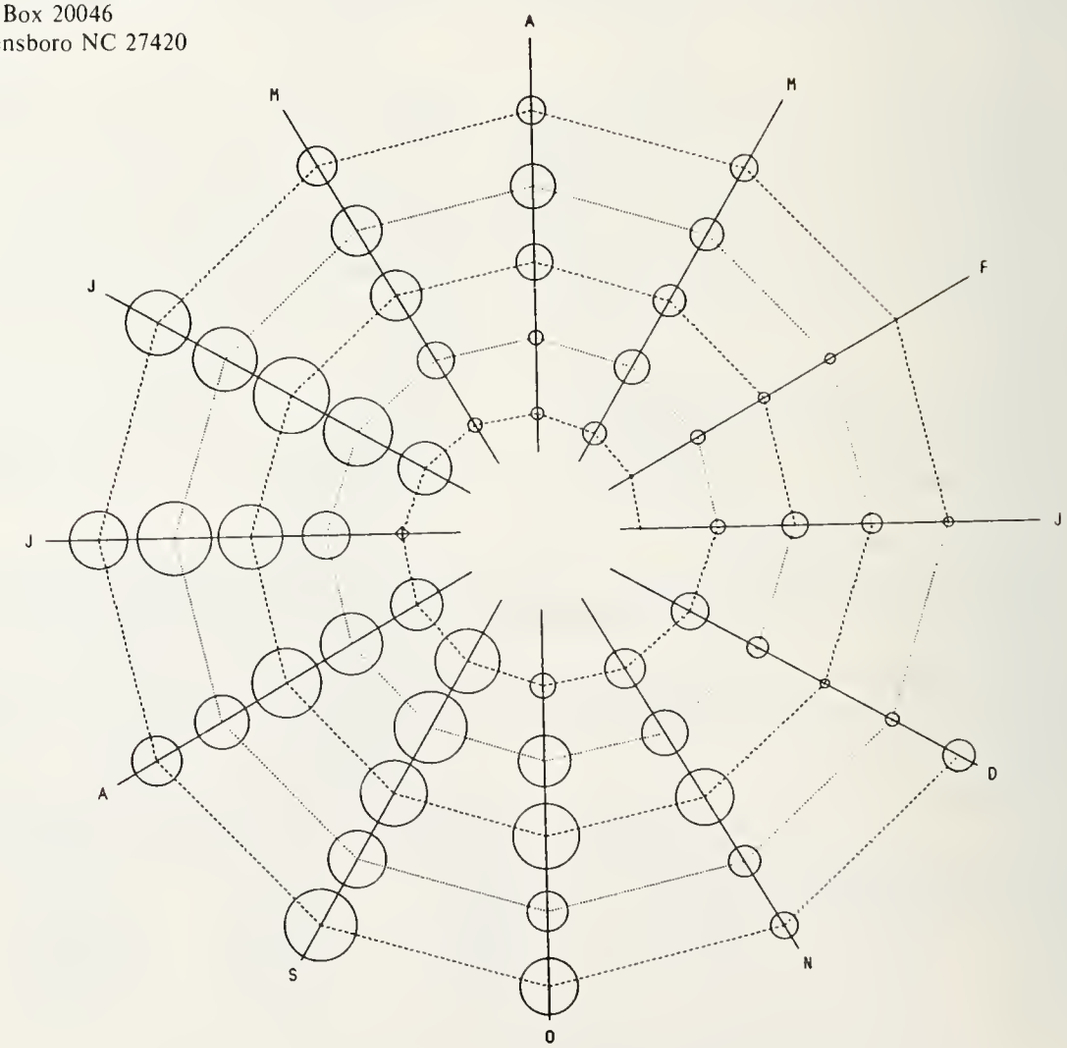
Attached are a set of examples of GR-Z graphical output from users' programs. (We make no attempt here to explain the varied applications involved in the plots.)

The GR-Z programs are available from Bell Laboratories on a license basis. A non-profit educational institution may obtain a royalty-free license to use GR-Z for educational and academic purposes. There is a small service charge to help defray the cost of distribution. Inquiries for an educational license should be directed to:

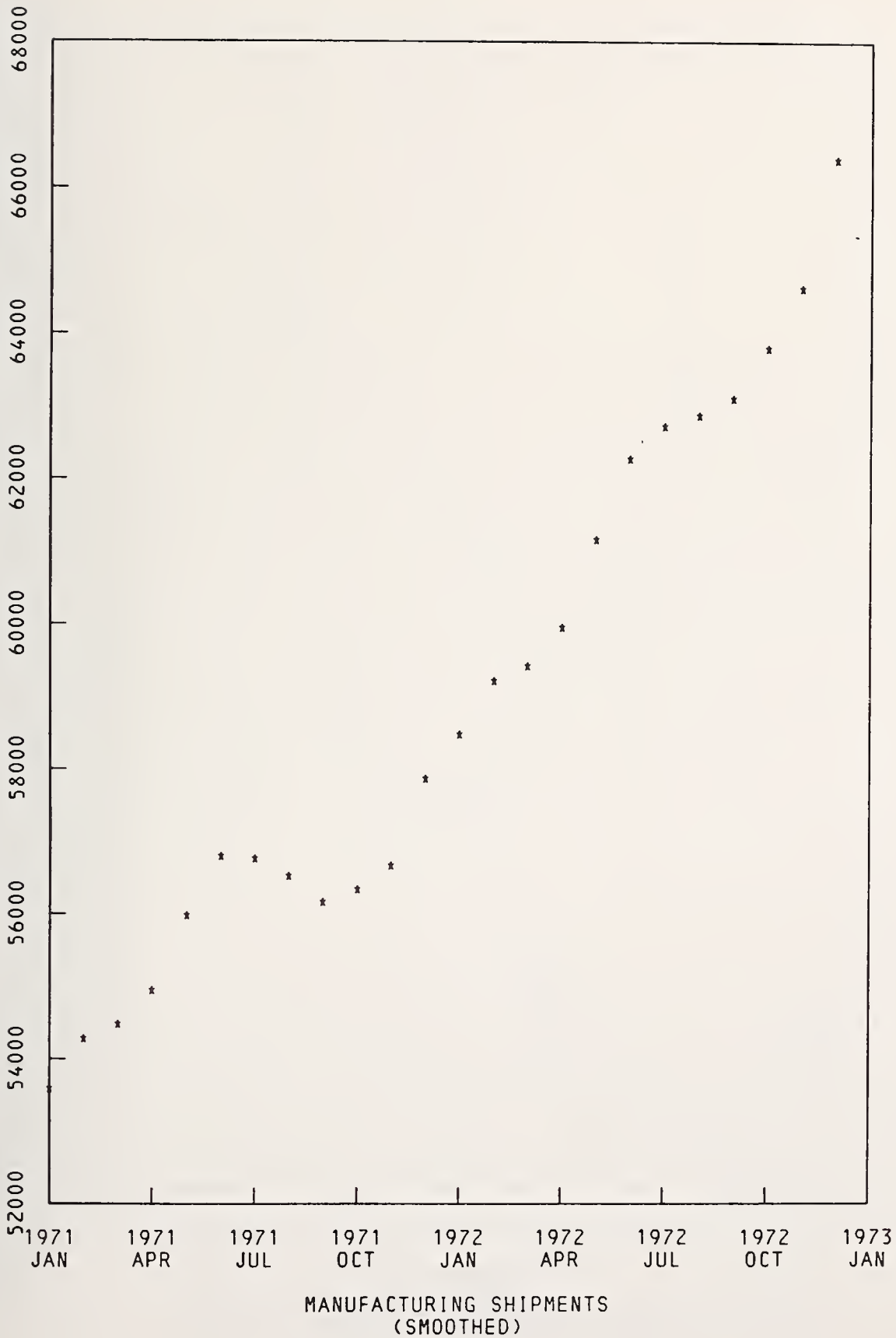
Bell Laboratories
Computing Information Service
600 Mountain Avenue
Murray Hill, New Jersey 07974
USA

For commercial and governmental organizations, and for educational institutions desiring to use the system for commercial purposes, a royalty is charged. Inquiries should be directed to:

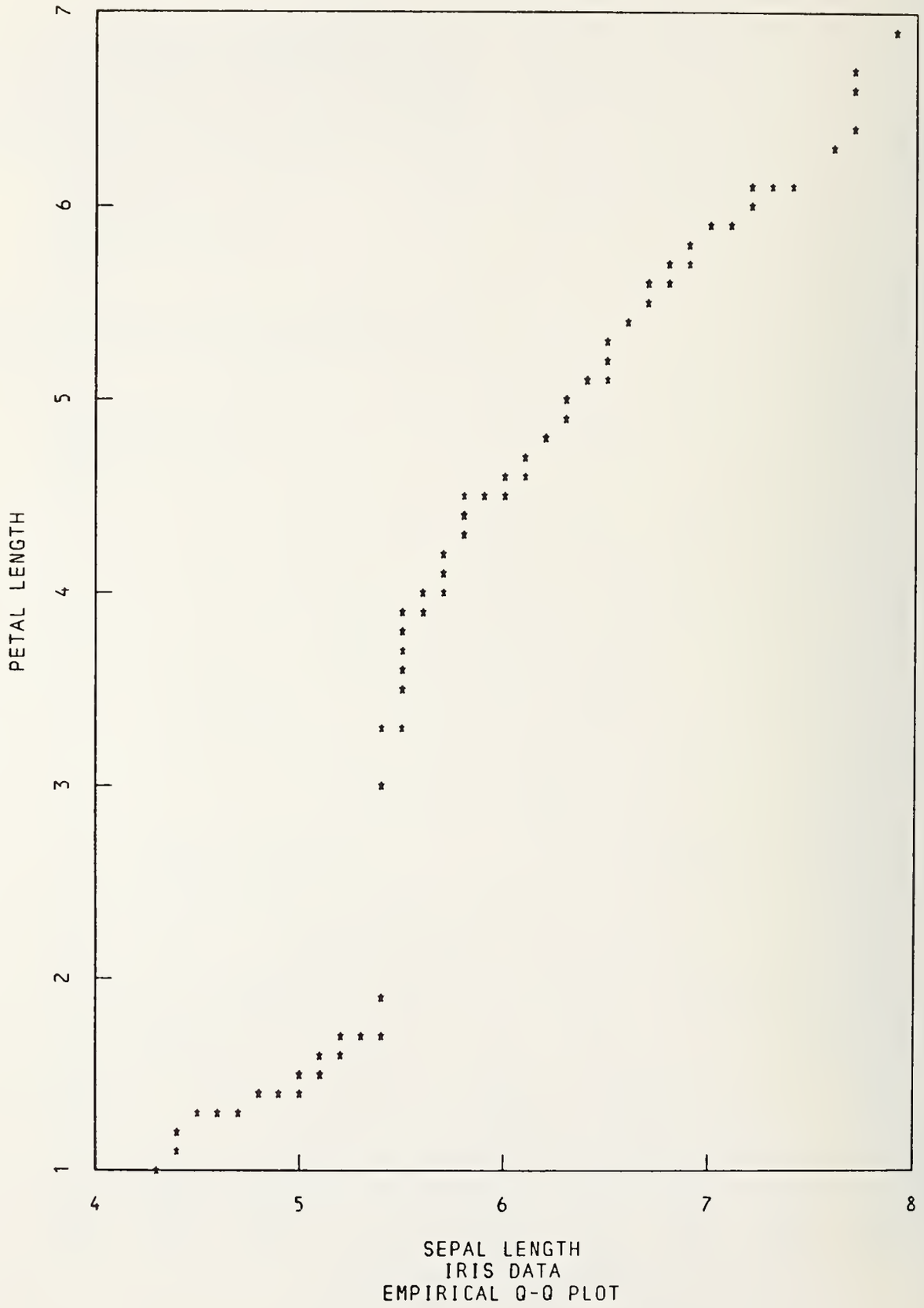
Western Electric Co.
Patent Licensing Manager
P.O. Box 20046
Greensboro NC 27420
USA

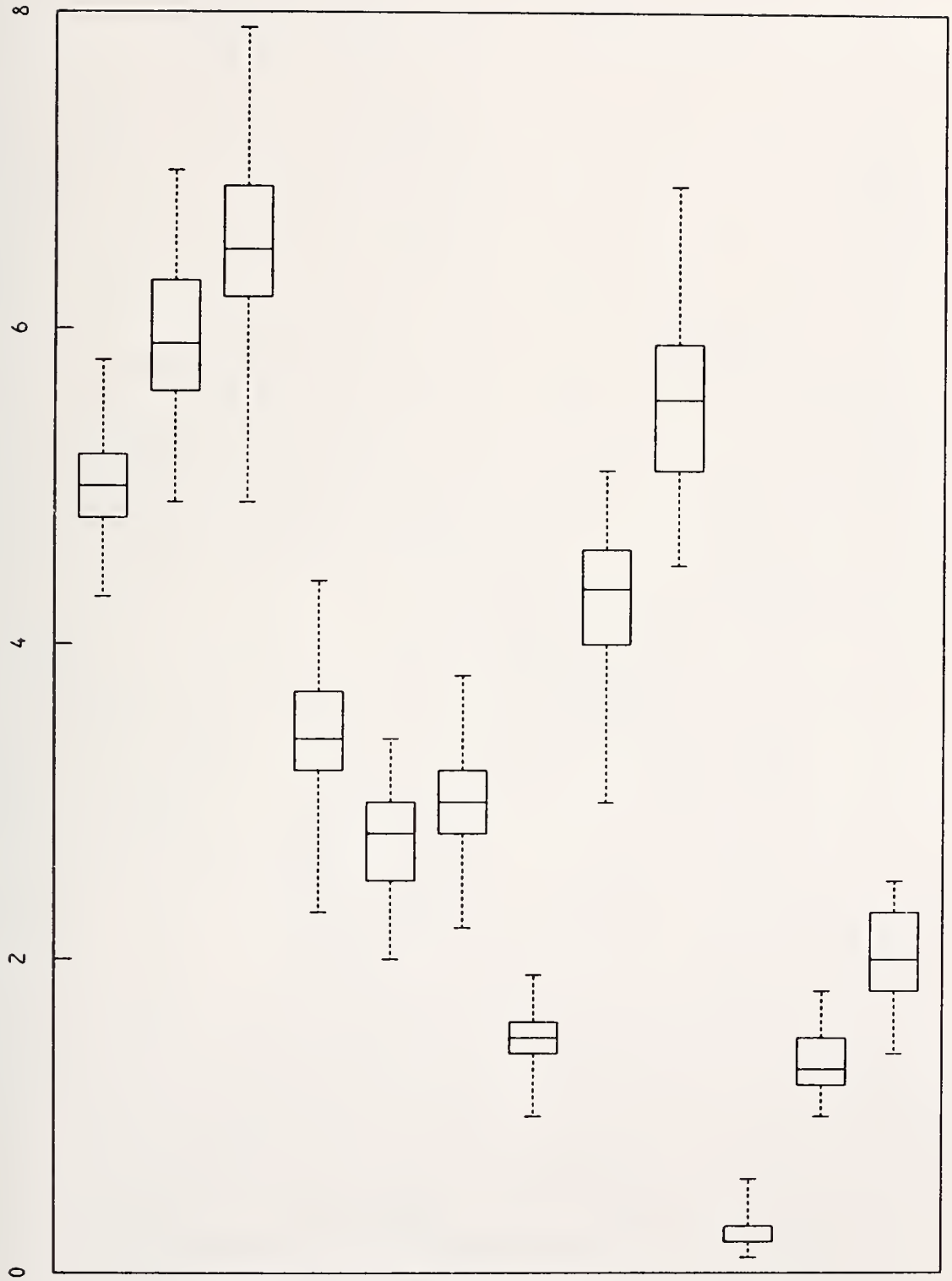


SPIRAL SEASONALITY PLOT FOR MONTHLY DATA

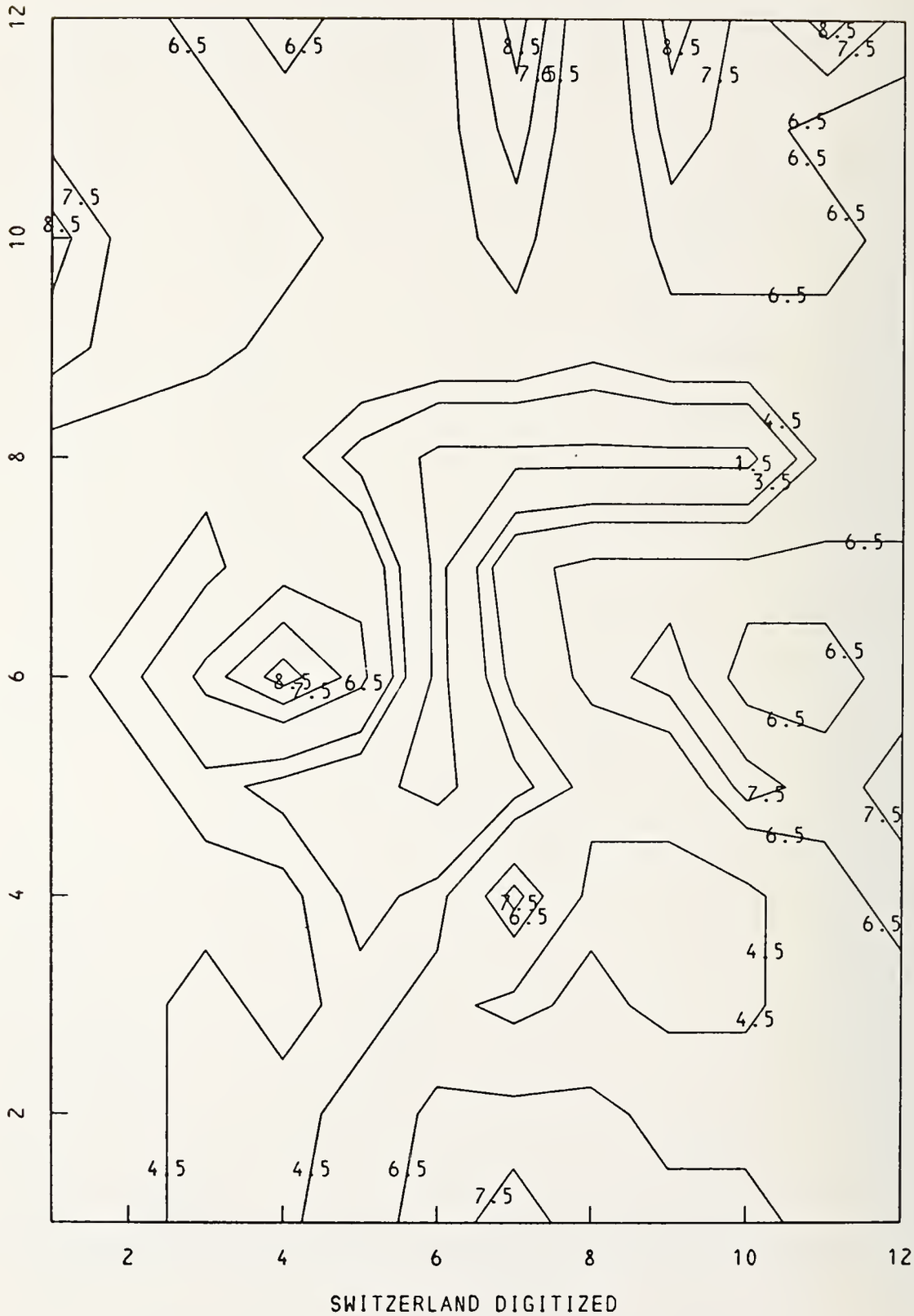


MANUFACTURING SHIPMENTS
(SMOOTHED)

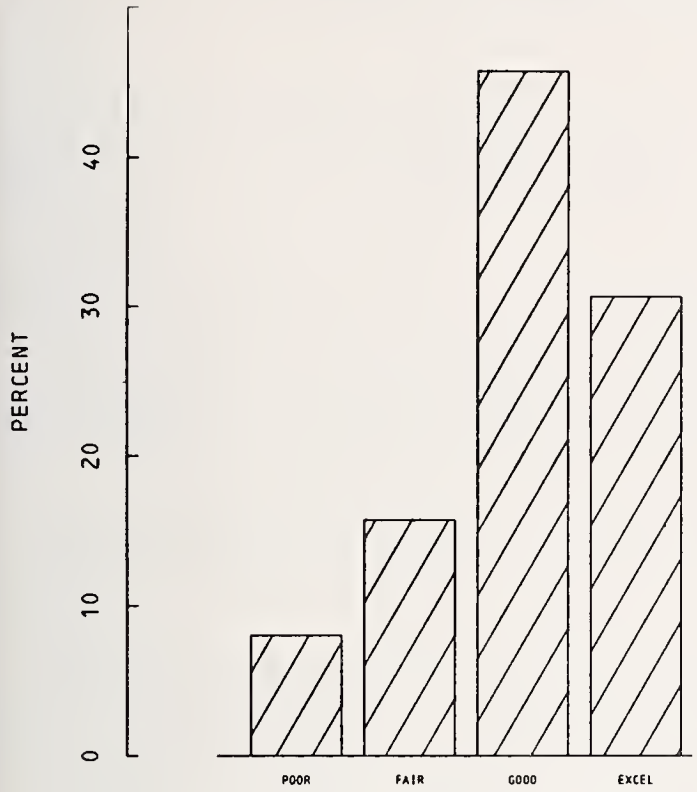

PETAL LENGTH
1
2
3
4
5
6
7
4
5
6
7
8
SEPAL LENGTH
IRIS DATA
EMPIRICAL Q-Q PLOT



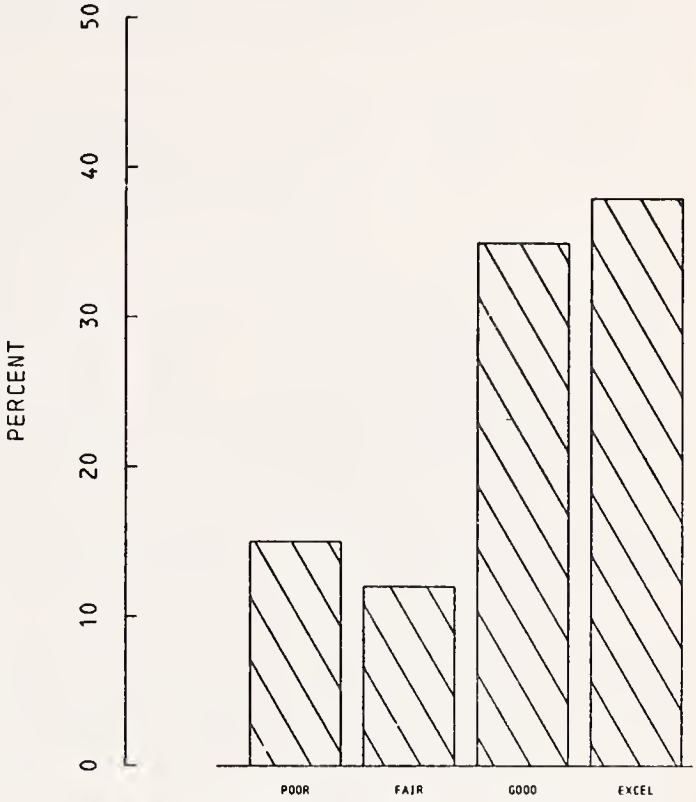
IRIS DATA
SEPAL LENGTH AND WIDTH, PETAL LENGTH AND WIDTH
3 GROUPS WITHIN EACH VARIABLE


12
10
8
6
4
2
2
4
6
8
10
12
6.5
7.5
8.5
4.5
1.5
3.5
SWITZERLAND DIGITIZED

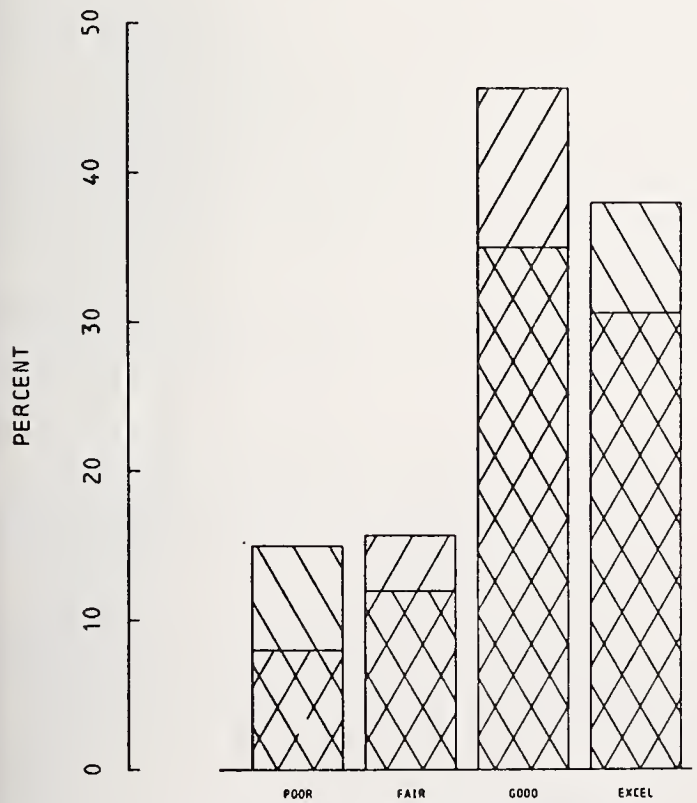
RESIDUAL BAR GRAPHS



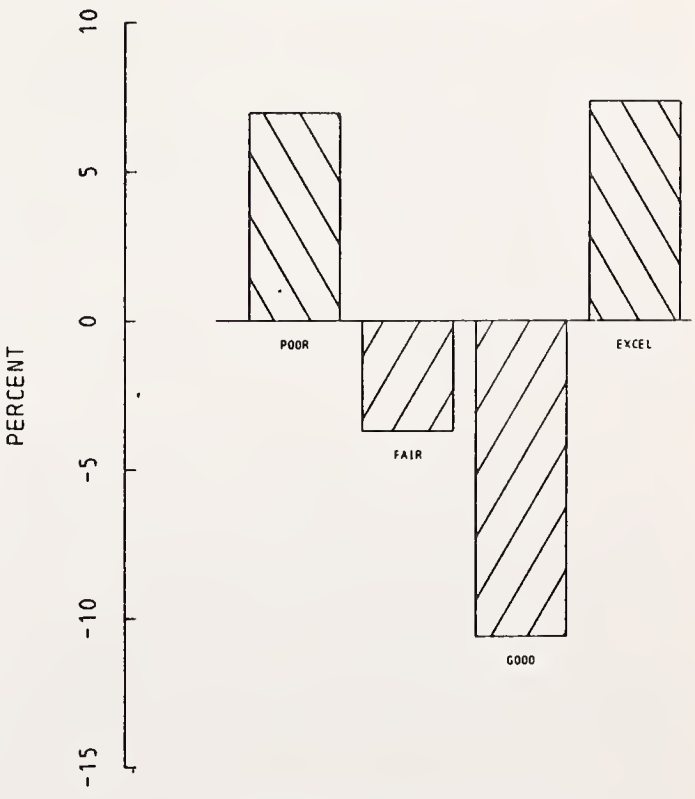
(1) OVERALL RESPONSE TO Q1



(2) RESPONSE BY SUBPOPULATION
TO BE COMPARED WITH PLOT 1



(3) PLOTS 1 AND 2 SUPERIMPOSED



(4) SINGLY SHADED AREA OF PLOT 3
REARRANGED TO FORM
RESIDUAL BAR GRAPH

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

AN APPLICATION OF A RECORD LINKAGE THEORY IN CONSTRUCTING A LIST SAMPLING FRAME

Richard W. Coulter and James W. Mergerson
U.S. Department of Agriculture

ABSTRACT

The Statistical Reporting Service (SRS), USDA is presently involved in the task of building a master list sampling frame of farms in each of its field offices. Lists from various sources with various formats and data content are combined to form a composite list in each state. An automated record linkage system is being developed to format and standardize the lists and to detect, display, and remove the duplication from this composite list. An overview of the system is presented with a brief explanation of the functions of the subsystems involved. This is followed by a discussion of the mathematical model employed to detect duplication and the computer processing used to implement this model.

Keywords: Address match; blocking; data manipulation; group resolution; identical match; linkage group; linkage model; list sampling frame; record linkage; reformat.

1. OVERVIEW

The Statistical Reporting Service, USDA, has developed an automated system to combine many list sources to form a master list sampling frame in each of its field offices. The present system consists of three subsystems. These are referred to as the Source List Editor Subsystem, the Record Linkage Subsystem, and the Group Resolution Subsystem.

1.1 Source list editor subsystem. The Source List Editor Subsystem consists of three major operations. These operations are Reformat, Identical Match, and Data Manipulation. While together these form a large and complex set of logic and perform a vital role in the total system they can be mentioned only briefly here.

Lists are obtained from various sources and do not conform to a standard format. The primary function of Reformat is to convert all source lists into a common format. Also place, state, and zip code are validated against each other and their spellings standardized.

In Identical Match, the first attempt is made to identify and remove duplication. The input file is sorted on all variables that will be used for record linkage. Any two or more records which have identical character by character linkage information will be considered to be the same record. These records will be compressed into one record, and one identifying number will be assigned.

In Data Manipulation, the information that is necessary to perform record linkage is obtained. The purpose of Data Manipulation is to identify all words in the primary name, secondary name, and address fields of each input record; to determine the use of these words; to manipulate them into a common structure; to code all given names and surnames; and to assign each record to one of three classes: individual, partnership, or corporate.

1.2 Record linkage subsystem. A separate linkage procedure has been designed for each class of records.

Partnership and corporate class records tend to be unique in their forms and for these a simple set of decision rules is used to match records. Depending upon the results of this testing each pair is classified as a link, possible link, or non-link. Links and possible

links are arranged in linkage groups, the premise being that records in the same linkage group will generally represent one farming operation.

The individual class of records comprise by far the largest portion of records on the file. A substantial amount of effort has been undertaken in developing a linkage system for these records in which the amount of identifying data is often meager. The probability model used and the necessary computer processing are described in Section 2.

1.3 Group resolution subsystem. The Group Resolution Subsystem consists of four major operations. They are as follows: Automated Resolution I, Address Match, Manual Review, and Automated Resolution II.

The main functions of Automated Resolution I are to generate microfiche output for all linkage groups, to select a sampling unit from each linkage group, and to identify linkage groups that contain a record from the drop file (a predetermined file of non-farms).

In Address Match, all records with sufficient address, regardless of class, that have identical addresses are identified. This program will help to identify between class duplication.

Manual review is a manual process in which certain linkage decisions made by the automated system are examined. The reviewer will decide to either accept or reject the decision made by the automated system.

In Automated Resolution II the final List Sampling Frame is created. All overrides made by the manual reviewer are processed.

2. INDIVIDUAL CLASS LINKAGE

2.1 Linkage model.

2.1.1 General technique. The mathematical model employed to identify the duplication between the individual type names on the composite list incorporates some of the concepts developed by Ivan Fellegi and Alan Sunter. The model is based on estimating two probabilities for the results of each comparison pair and converting these into a weight for the pair. Pertinent portions of the theory are described below.

Let L_A be the list to be unduplicated which covers the population A with members $a_i \epsilon A$. Members of L_A will be denoted by $\alpha(a_i)$.

Define: $M = \{(a_i, a_j);\ a_i = a_j,\ i < j\}$

$U = \{(a_i, a_j);\ a_i \neq a_j,\ i < j\}$

as the matched and unmatched sets respectively.

Denote by $\gamma = (\gamma^k)$ the vector of coded results of the comparison of the components in the comparison pair $[\alpha(a_i), \alpha(a_j)]$, where the result of the comparison on the k^{th} component is denoted by γ^k.

1. $m(\gamma^k) = P\{\gamma^k\ [\alpha(a_i), \alpha(a_j)];\ (a_i, a_j) \in M\}$

2. $u(\gamma^k) = P\{\gamma^k\ [\alpha(a_i), \alpha(a_j)];\ (a_i, a_j) \in U\}$

A component weight for each γ^k is defined by:

$$w(\gamma^k) = \log_{10} [m(\gamma^k)/u(\gamma^k)]$$

Once weights have been assigned to the outcome of each of those components being compared, a total weight for the comparison pair is computed by summing together all of the component weights.

Two threshold values are calculated prior to making the comparisons and are used in classifying each. If the total comparison pair weight is larger than the upper threshold, then the pair is classified as a definite link. If the total weight is less than the lower threshold, then the pair is classified as a definite non-link. Pairs with total weight between the two are classified as possible links.

2.1.2 Weight calculation. Rather than describe in detail the computations for each component, some of which are rather lengthy, only the computations for the simplest condition are given here. This is the weight calculation procedure for those components (prefix, suffix, route, street name) which use only a simple agreement or disagreement weight.

Define e = P(the component is misreported on a record given the pair is associated with M)

e_t = P(the component is different, though correctly reported in a pair of records from M)

e and e_t will be referred to in the following as error terms.

e_o = P(the component is missing on a record)

f_j = the frequency of the j^{th} value of the component on the file (e.g. frequency of route number '1')

N = the total number of records with the component present on the file.

Then, $m(\text{component agrees and is the } j^{th} \text{ value}) = (f_j/N)(1-e)^2(1-e_t)(1-e_o)^2$

$u(\text{component agrees and is the } j^{th} \text{ value}) = (f_j/N)^2(1-e_o)^2$

$m(\text{component disagrees}) = [1-(1-e)^2(1-e_t)](1-e_o)^2$

$u(\text{component disagrees}) = [1-\Sigma_j(f_j/N)^2](1-e_o)^2$

$m(\text{component missing in one or both records}) = 1-(1-e_o)^2$

$u(\text{component missing in one or both records}) = 1-(1-e_o)^2$

The weight for each condition is $\log_{10}(m/u)$. Note that one agreement weight is calculated for each different value of each component. Many modifications have been made to this basic theory to allow more sophistication. These include the use of given name and surname codes and partitioned disagreement weights in the model.

2.1.3 Estimating error rates and thresholds. Prior to processing the entire file through linkage, a sample of blocks is selected. Weights are calculated for the entire file but only the sample is processed through linkage using initial estimates of error terms and thresholds.

The sample results are then manually reviewed and verified. Counts for each component are kept for those pairs classified as links. These are used to update the error terms. This update then changes the various weights already calculated.

Also, the thresholds are revised as necessary based on this manual review. Upon completion of this step, which may require processing the sample through several iterations, the entire file is then ready to be processed through linkage.

2.2 Linkage Software. In the Record Linkage Subsystem there are eight major programs involved in the process of detecting and grouping together those individual class of records which have a high probability of representing the same person or farm operation. These programs are: Data Selection, Blocking, Sample Selection, Frequency Count, Weight Calculation, Weight Insertion, Linkage Match, and the Master File Update Program.

Since the master file is very large, the entire file is not passed through each program in the subsystem. Instead, in data selection only those data fields that are used in the Record Linkage Subsystem are extracted. Also frequencies for Identification numbers are calculated.

Blocking consists of putting all records with the same surname code in one block. A maximum allowable block size is parameter input since in some cases the internal tables could get too large. For this reason the surname code blocks that exceed the maximum block size are further broken down by other variables. These variables are currently a first name initial group code and a location code.

Since the entire file is not passed to the blocking program a special technique is used to define blocks. The input file contains only the value of the blocking variables for a given record and the record number for that record. The program outputs one record for each block which contains all the record numbers of the records in that block.

The sample select program selects a subset of the blocks to be used in the iterative process to calculate error probabilities. Blocks are selected by strata where each strata is a range of block sizes. To extract blocks, a particular number of blocks from each strata and the starting block for each strata are parameter specified and a systematic sample selected.

The frequency count program calculates frequencies of all linkage variables on the input file except for identification numbers. These frequencies are used by the weight calculation program in calculating agreement constants which are the portion of the agreement weights which do not include the error terms.

In the weight calculation program partial agreement weights, agreement constants, and disagreement weights which are used by the linkage model are calculated. This program can operate in two different modes, called Mode A and Mode B. When running in Mode A, partial agreement weights, agreement constants and disagreement weights are calculated. In Mode B, only partial agreement weights and disagreement weights are calculated.

Since an iterative procedure is used to calculate error terms, it is necessary to recalculate weights after each iteration. In order to greatly reduce costs, a special technique is used to eliminate the need for reinserting weights after each subsequent calculation. This is accomplished by initially operating in Mode A, while each subsequent sample iteration is done in Mode B. To obtain the agreement weights the appropriate agreement constant is added to each partial agreement weight in the linkage match program. The initial threshold values are also calculated by this program.

The weight insertion program takes the agreement constants and inserts them into the internal master records. The output of this process are records which contain the original linkage variables and their corresponding partial agreement weights concatenated on the end.

The linkage match program performs the actual comparisons of components and classification of records. This program runs in two modes which are referred to as the test mode and the production mode. The test mode is used during the iterative process to calculate error probabilities. In this mode, the program automatically terminates when enough comparison pairs that match have been obtained for calculating error probabilities. The production mode runs to completion and does not do the processing for the error probability revision.

The processing starts by reading a block. Every possible combination within a block is generated and passed one pair at a time to the model. A total weight for each pair is calculated and the pair is classified according to the relationship of this weight to the threshold values, and is placed in a linkage group.

The master file update program reads the master file serially and outputs an updated serial version of the master to be passed to the Group Resolution Subsystem.

3. REMARKS

While results are encouraging, analysis is continuing on all subsystems for both improved results and improved efficiency.

Persons interested in more information should contact the List Sampling Frame Section, Statistical Reporting Service, U.S. Dept of Agriculture, Washington, D.C.

4. ACKNOWLEDGMENT

The development of this List Frame System has involved the efforts of many people in SRS. The authors wish to acknowledge their effort which has made this paper possible.

5. REFERENCES

Fellegi, Ivan P. and Sunter, Alan B. (1969). A Theory for Record Linkage. JASA.

BIOGRAPHIES

Richard W. Coulter received a M.S. in Mathematics from Montana State University in 1972.

James W. Mergerson received a M.S. in Mathematics from Stephen F. Austin University in 1974.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

LONG RANGE PLANNING MODELS
LRPM2, LRPM3, and LRPM4/PDM

Joseph Quinn, Roger Bove, Ta-Lin Liau
I.S.P.C., U.S. Bureau of the Census, 20233

ABSTRACT

The Bureau of the Census has built three LRPM (Long-Range Planning Models) packages for use by planners in the developing countries: LRPM2, LRPM3, and LRPM4/PDM (which was originally developed under contract by the Agricultural Economics Department of the University of Purdue). The packages differ in their level of sophistication and data needs -- starting with simple population projections and limited and flexible data needs in LRPM2 and going to linear programming optimization and extensive data requirements in LRPM4/PDM.

The subjects treated in submodels include: demographic projections; family planning; projections of urban and rural populations; labor force, health, food, and economic consumers; health services; education projections; housing; social security; electricity, gas, and water; families; mortality by cause; food consumption and production by crop; energy; social mobility; construction; government budget; regional projections; employment by industry and profession. There are also adaptions of programs developed by others for graphing and data management.

Because the models were designed for use in the developing countries, they were designed to be:
Easy for social scientists who were not computer technicians to use;
Small enough to be run on most computers;
Flexible in their data needs and in allowing many alternative paths to be followed when building a country model;
Segmented so that submodels such as education projections could be run independently;
Reasonably accurate in any projection mechanisms used;
Useful for historical and structural analysis as well as for projections.

Key words: Demographic projections; developing countries; economic projections; long range; planning models, social services.

1. TEXT

The LRPM packages were built to show planners how to use census and survey data to see how demographic, economic, and social factors interrelate. Until about ten years ago, most planning models ignored the effects of population growth and structure and assumed that this subject should be treated outside of planning models, particularly Neo-Keynesian and Neo-Classical models. With the passage of time, many analysts have decided that a country's population should be the center of development plans, programs, and projects.

If a planner wants to account for population in an explicit and an analytic way, he must be able to: first, identify specific subgroups by their characteristics and needs -- age, sex, educational level, training needs, requirements for food, medical and social services. Second, he must be able to test whether changes in the structure of these groups will be consistant with his plans and vice-a-versa. Last, he should be able to see what interactions will occur as changes in some characteristics of the population affect other characteristics and these in turn affect plans.

Based on this need the LRPM2 modules or submodules were designed to have the following features:

(1) They treat a fairly wide range of relevant problems;
(2) They are easy for persons, who are not computer technicians, to use. Data formating and computer instructions are clear and simple to follow;
(3) The data needs are flexible since statistical systems differ in the kinds and quality of data produced;
(4) Any projection or accounting mechanisms used are reasonably accurate;
(5) All of the submodels can be run separately;
(6) The models are designed so that they can be used for checking historical data and structural analysis as well as for making forecasts and projections;
(7) All of the programs are small enough to be run on the medium size computers generally found in less developed countries;
(8) The LRPM2 flexibility allows any researchers to follow as he sees fit a large array of alternative paths when building his choice of a structural model;
(9) The basic test for a model as set forth by Hannes Hyrenius is considered: "(a) that all factors judged necessary, according to the criteria laid down, must be included; (b) that these are indicated and measured in correct (unbiased) forms and measurements; (c) that relations and feedback loops are included correctly and to the extent necessary; (d) that all constants, parameters, relations and feedbacks are quantified in a satisfactory way."

The twenty-two submodels of LRPM2 have the following functions. Projections of:

(1) Demographic variables;
(2) Urban and rural population;
(3) Special population groups such as labor force, economic consumers, health service consumers, food consumers, and school-age groups;
(4) Health Services;
(5) Education;
(6) Housing (also electricity, sewerage, and water);
(7) Economic simulations;
(8) Family planning;
(9) The number of families;
(10) Mortality by cause;
(11) Food consumption and agricultural demand;
(12) Energy;
(13) Construction;
(14) Government budget;
(15) Regional projections;
(16) Employment by industry and profession;
(17) Social Security;
(18) Graphing;
(19) Table formating and a management information system;
(20) Patterns of development submodel;
(21) Income Distribution;
(22) Transition matrices;

All of the submodels have several short manuals which include:

(1) Program listing;
(2) Methodology;
(3) Input instructions and data needs;
(4) Example runs using various options;
(5) Useful statistical routines for preparing or analyzing data;
(6) Special uses of this submodel, e.g., housing to forecast water, sewerage, and electricity demands;
(7) Actual case studies;
(8) Flow charts;
(9) Data needs (required and optional);

The analyst can use the combination of submodels he desires. LRPM3 and LRPM4/PDM, also developed by the SEA Staff, deal with the data in a more integrated fashion with LRPM3 focusing on keeping track of the educational attainments of the population and income distribution and LRPM4/PDM concentrating on the relationships between agriculture and the rest of the nation.

An interactive version of LRPM2 was built by the Demographic Projections Analysis Group at the American University in Cairo.

TABLE I-A -- SAMPLE DEMOGRAPHIC OUTPUT - LRPM2

1975 Base Population (Thousands)

	NUMBER		PROPORTION OF TOTAL POPULATION	
AGE	MALES	FEMALES	MALES	FEMALES
0 to 4	427.94	423.22	0.0817	0.0808
5 to 9	349.89	346.22	0.0668	0.0681
10 to 14	308.42	304.85	0.0585	0.0582
15 to 19	258.23	257.66	0.0493	0.0511
20 to 24	200.61	233.69	0.0383	0.0445
25 to 29	152.42	202.71	0.0291	0.0367
30 to 34	123.09	175.99	0.0235	0.0336
35 to 39	121.62	166.56	0.0232	0.0318
40 to 44	112.61	136.71	0.0215	0.0261
45 to 49	104.76	119.42	0.0200	0.0228
50 to 54	95.35	97.42	0.0182	0.0186
55 to 59	81.19	85.90	0.0155	0.0164
60 to 64	67.57	68.62	0.0129	0.0131
65 to 69	46.09	50.81	0.0088	0.0097
Over 69	48.71	61.81	0.0093	0.0118
TOTAL	2498.38	2740.99	0.4766	0.5234

TABLE I-B SAMPLE DEMOGRAPHIC OUTPUT - LRPM3

Rural Males - 1971

Age	Illiterates	Literates	Primary	Secondary	University
0 - 4	1570.2	--	--	--	--
5 - 9	1363.6	--	--	--	--
10 - 14	1041.6	63.0	--	--	--
15 - 19	519.4	360.0	16.3	8.1	--
20 - 24	360.0	293.0	38.6	4.2	--
25 - 29	289.5	268.0	33.8	3.1	--
30 - 34	271.6	233.0	26.4	2.9	--
35 - 39	252.7	200.0	14.4	1.6	--
40 - 44	190.3	159.0	8.6	.8	--
45 - 49	178.2	115.0	5.6	.5	--
50 - 54	186.0	85.0	3.7	.4	--
55 - 59	159.4	60.0	2.5	.3	--
60+	334.2	85.0	4.1	.2	--
Total	6717.4	1921.0	154.1	22.1	--

TABLE I-C SAMPLE DEMOGRAPHIC OUTPUT - LRPM4/PDM

Population by Location, Sex, and Level of Education

Location and Sex	Population	Percent by Level of Educational Attainment (Grades Completed) 0-3	4-7	8-11	12-15	16
Rural Agriculture						
Total	12487903.	58.51	31.60	7.42	2.45	0.02
Male	6311928.	52.04	34.41	10.03	3.49	0.03
Female	6175975.	65.12	28.73	4.74	1.39	0.02
Rural Nonagriculture						
Total	3478671.	57.61	25.98	11.78	4.39	0.24
Male	1717551.	53.18	25.95	14.59	5.97	0.31
Female	1761120.	61.94	26.02	9.05	2.84	0.16
Urban						
Total	15987057.	33.79	21.00	20.54	22.55	2.12
Male	8035511.	32.70	19.39	19.30	25.77	2.84
Female	7951546.	34.90	22.64	21.79	19.29	1.39
Total Population						
Total	31953631.	46.04	25.69	14.46	12.72	1.10
Male	16064990.	42.49	25.99	15.15	14.90	1.47
Female	15888641.	49.64	25.38	13.75	10.51	0.72

BIOGRAPHIES

Joseph E.M. Quinn is the chief of the Socioeconomic Analysis Staff, International Statistical Programs Center, U.S. Bureau of the Census.

Roger Even Bove received a Masters degree in Applied Mathematics in 1961, and a PhD in Economics in 1973 from Harvard University. He taught economics in New York area schools before coming to the Census Bureau in 1974 where he has been primarily involved in development of the LRPM2 system.

Ta-Lin Liau is a statistician with the Bureau of the Census. He received a PhD in statistics from Texas A&M University in 1973.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

A NEW APPROACH TO ACCESSING LARGE STATISTICAL DATA FILES

Gary L. Hill
Data Use and Access Laboratories, Arlington, VA 22209

ABSTRACT

The National Institute of Child Health and Human Development (NICHD/NIH) provided funding to DUALabs for the analysis of unique data processing problems posed by large public data files. One mechanism that resulted from this activity was the CENTS-AID II system, which reduces the cost of generating cross-tabulations by as much as 80%. This high-speed statistical access system is designed for use with large files and enables users to produce complex cross-tabulations consisting of up to eight dimensions. A powerful retrieval language and full set of data transformations and recode capabilities can be used to prepare any table or set of tables required. CENTS-AID II provides access to rectangular, heterogeneous, and hierarchical file structures, allowing simultaneous analysis of multiple record formats and, in hierarchical structures, direct analysis of data relationships of up to thirty different levels.

Key words: Generalized; hierarchical; large; software; statistical; survey; tabulation.

1. ACCESSING PUBLIC DATA: The Problem

The U.S. Government provides a continuous flow of computerized statistical data covering virtually every aspect of American life: science and education, health and safety, manpower and employment, consumer prices and expenditures, characteristics of population and housing, and many others. These large public data files represent a valuable source of information for researchers, planners, scientists, and administrators concerned with the activities of people, the products they use, and the environment in which they live.

Large data producers such as the Census Bureau commonly organize sequential files in a hierarchical, or tree structure, format. This type of file organization provides for the definition of one or more record formats describing different units of analysis. For example, a file may contain one record format to describe the characteristics of neighborhoods, another to describe households, and a third for people. Additional valuable data relationships are defined by arranging the records in a predetermined order (tree structure); person records immediately follow the household record in which they live, and household records follow the neighborhood record in which they reside.

The analytical potential afforded by this type of file structure far exceeds the capacity of the punched card concept of file organization where each file has a single unit of analysis expressed in one record format. Unfortunately, most of the widely used generalized statistical access systems require data to be organized as if they were in punched cards. In order to access public files, data must first be reorganized to suit the unique specifications of the software system being used. This process is not only costly, but often destroys data relationships defined by the original structure of the file. Further, most public data files contain tens-of-thousands, hundreds-of-thousands, or millions of

records whereas most statistical access systems are designed to efficiently analyze a limited number of observations. As increasingly larger volumes of data are processed, computer costs become prohibitive.

2. CENTS-AID II: The Basics

Although CENTS-AID II is simple to learn and easy to use, it does require that the user have a minimum of computer orientation and a basic understanding of the relationship of records within his file. Unlike other generalized systems, most data files do not have to be reformatted in order to be analyzed. CENTS-AID II will process simple and complex sequential file structures whose records are fixed or variable length. In a single application, the system can process up to twenty-six different record formats and a hierarchical structure of up to thirty levels. The more complex the file structure, the more data expertise is required of the user.

There is virtually no limit to the number of tables that can be produced in a single run. However, no single table may exceed 17 columns, nor 999 rows, nor 8008 matrix cells. Matrix cells can be incremented by a simple frequency count (1) or by the values of an observation variable such as income, expenditures, age or quantity. A limited set of descriptive statistics is also available: percentage, mean, median, variance, and chi-square.

The free-form command language of CENTS-AID II relieves users from most of the technical details usually associated with extensive data processing. Users can readily control the content and format of simple and sophisticated tabulations. For example, the following TABLE command defined the six-way tabulation displayed on the succeeding page:

```
TABLE PLACE AND RACE AND INCGRP BY EMPST AND AGEGRP AND SEX
```

The VAR LABEL command was used to supply descriptive labels for each of the six variables.

3. SYSTEM DESIGN CONCEPTS: The Principles

CENTS-AID II is a generative system. The system actually generates an ANSI-COBOL program which processes the data file and subsequently prints the requested tables. This generative approach provides an efficient, cost-effective method of file processing. In a matter of seconds, the system generates a tailor-made solution to the requirements posed by the user. Unlike interpretive systems, the generative characteristics of CENTS-AID II enable it to customize the file processing logic for each application. The cost of file processing is minimized.

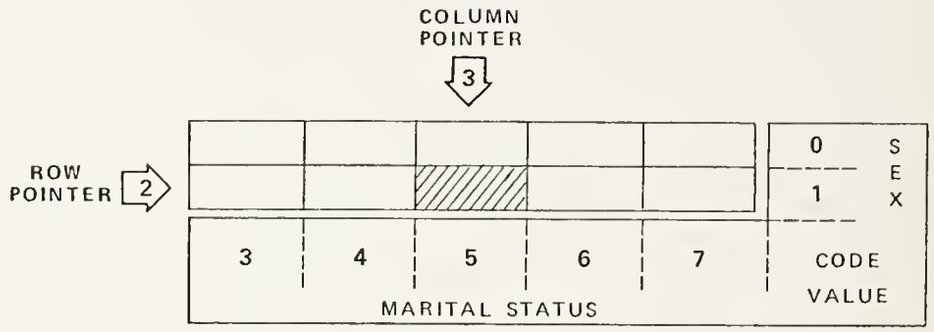
The cost of tabulating data from large files is minimized further by the techniques used within the system to construct and update table matricies. CENTS-AID II constructs a matrix shell for each table prior to the actual processing of the data file. The user must therefore supply the minimum and maximum values of each variable to be included in a table. Simple commands are available to manipulate variables containing alphameric values or non-contiguous coding structures. Since each matrix shell is specifically tailored to accommodate the user's requested tabulations, the system only reserves the amount of core storage actually needed. In many computer billing algorithms, the core storage costs are significant so that by reducing core requirements, computer processing costs can be minimized.

The method used by CENTS-AID II to update, or increment, matrix cells is also a major contributing factor to the efficiency of the system. Instead of continually scanning matrix dimensions to determine the proper matrix cell to increment, CENTS-AID II uses the actual code values from the data file to computer "pointers" into the matrix shell. Simplified, the algorithm used to compute the "pointers" for a two-way table is as follows:

```
(Code Value - Minimum Value) + 1
```

To illustrate the technique, suppose a user has requested the generation of a simple two-way tabulation (Sex by Marital Status); where Sex contains two code values (0 and 1), and Marital Status contains five code values (3, 4, 5, 6, and 7). A record containing a value of 1 for Sex and a value of 5 for Marital Status immediately points to the matrix intersection of (2, 3):

ROW POINTER = (1 - 0) + 1 or 2
COLUMN POINTER = (5 - 3) + 1 or 3



4. PROCESSING EFFICIENCY: A Comparison

CENTS-AID II is engineered to minimize computer processing costs for tabulating data from large statistical files. The techniques employed do not necessarily produce a cost effective mechanism for processing small data files. NICHD and DUALabs decided to conduct a series of benchmark tests designed to generate statistics that would demonstrate the effect of processing increasingly larger volumes of data. Although we feel that it is unrealistic to compare generalized systems that are designed for different purposes, we chose the Statistical Package for the Social Sciences (SPSS) for this comparison because it is so widely used. The benchmarks were not intended to be a comprehensive evaluation of the merits of the two systems. Whereas CENTS-AID II is specifically designed to produce sophisticated tabulations from large data files, SPSS offers a wide range of statistical analysis capabilities that far exceed the current facilities of CENTS-AID II. The benchmark tests were designed by an outside consultant to meet the following specifications: 1) the test must request statistics which both systems could generate; and 2) it must use SPSS as efficiently as possible. The resulting application used the FASTABS option of SPSS version 6.0 with the data files being the 1970 Public Use Samples. The results of the test are presented in the following table:

	BENCHMARK TEST (IBM 360 Model 65)					
	TEST 1		TEST 2		TEST 3	
	SPSS (6.0)	CENTS-AID II	SPSS (6.0)	CENTS-AID II	SPSS (6.0)	CENTS-AID II
Number of Input Records	27,591	27,591	277,723	277,723	2,719,249	2,719,249
Size of Universe	5442	5442	54,741	54,741	537,667	537,667
Number of Variables	9	9	9	9	9	9
CPU * Time (Seconds)	119.59	32.29	1188.17	134.08	11880.00	1113.16
Core Storage	214	94	214	94	214	94
Dollar Cost	$45.99	$10.78	$175.74	$24.48	$1543.04	$111.03

Table T007: PLACE OF RESIDENCE AND RACE AND INCOME GROUP BY EMPLOYED AND AGE GROUP AND SEX

	EMPLOYED								
	YES				NO				
	AGE GROUP				AGE GROUP				
	18 TO 35		OVER 35		18 TO 35		OVER 35		
PLACE OF RESIDENCE / RACE / INCOME GROUP	SEX		SEX		SEX		SEX		
	MALE	FEMALE	MALE	FEMALE	MALE	FEMALE	MALE	FEMALE	T O T A L
URBAN									
WHITE									
$0 TO $4,999	475	549	402	654	264	812	539	1,774	5,469
$5,000 TO $9,999	510	208	676	331	24	16	38	20	1,823
$10,000 AND OVER	281	14	699	53	6	2	20	2	1,077
BLACK									
$0 TO $4,999	75	86	74	121	44	138	80	167	785
$5,000 TO $9,999	63	39	82	44	6	5	8	3	250
$10,000 AND OVER	8	1	28	3	-	-	-	-	40
OTHER									
$0 TO $4,999	14	6	9	8	9	14	2	15	76
$5,000 TO $9,999	5	3	7	7	1	-	-	-	23
$10,000 AND OVER	2	-	4	-	-	-	-	-	6
SUB TOTAL URBAN	1,433	906	1,980	1,221	354	987	687	1,981	9,549
RURAL									
WHITE									
$0 TO $4,999	186	190	317	311	88	316	287	764	2,459
$5,000 TO $9,999	199	46	298	89	1	4	10	3	650
$10,000 AND OVER	66	2	157	7	3	-	6	-	241
BLACK									
$0 TO $4,999	26	16	36	26	11	25	23	59	222
$5,000 TO $9,999	6	1	9	1	1	-	1	-	19
$10,000 AND OVER	-	-	2	1	-	-	-	-	3
OTHER									
$0 TO $4,999	4	3	4	6	3	10	8	9	47
$5,000 TO $9,999	3	3	5	3	-	-	-	1	15
$10,000 AND OVER	3	-	5	-	-	-	-	-	8
SUB TOTAL RURAL	493	261	833	444	107	355	335	836	3,664
T O T A L	1,926	1,167	2,813	1,665	461	1,342	1,022	2,817	13,213

From the comparative statistics generated by the three benchmark tests, it is clear that as the volume of data increases, the computer cost of performing tabulations with ordinary generalized software systems can become almost prohibitive. Subsequent to the execution of the formal benchmarks presented, we undertook a further analysis of the processing efficiencies of the two systems. For example, each system was required to generate multiple tables using various combinations of instructions. Throughout these tests the variation in relative processing efficiencies remained rather consistent with CENTS-AID II applications costing approximately 20% of the SPSS runs. During the testing process, an SPSS SYSTEMS FILE was created which sustantially reduced SPSS tabulation costs. However, the cost of creating such a file can be expensive, and valuable data relationships may be destroyed in the process.

5. SUMMARY: Additional Information

CENTS-AID II is currently installed in over 50 computer sites around the world including the Belgian Archives, University of Heidelberg, New Zealand Department of Statistics, Eastman Kodak Company, Prudential Insurance Company, Congressional Budget Office, Social Security Administration, National Institutes of Health, and the New York State Workmen's Compensation Board. The system is operational on the IBM 360/370 under OS/MFT/MVT/MVS/VS1, as well as IBM 360/370 under DOS/VS. In the fall of 1977, a Honeywell 6000 Series version will become available from DUALabs.

A new statistical generation module is being designed for CENTS-AID II which will minimize or eliminate statistical error caused by accessing very large data files. The module will include facilities for generating correlation matricies, means and standard deviations, sums of squares, sums of cross-products, and variance/covariance matricies. The extended CENTS-AID II system will perform correlation analysis on simple and hierarchical files at a fraction of current costs with an improvement in accuracy compared to other systems.

Arrangements have been established with the National Technical Information Service (NTIS), U.S. Department of Commerce, for distribution of the IBM versions of the CENTS-AID II system at a sale price of $600 domestically and $1,200 for foreign sales. The price includes the User Manual and Programmer's Notebook as well as one year of maintenance and support provided directly by DUALabs. Readers who would be interested in purchasing the IBM 360/370 version of CENTS-AID II should contact Mr. Frank Leibsly, National Technical Information Service, 5285 Port Royal Road, Springfield, Virginia 22161. For additional information concerning the system, contact Gary Hill, Director of Systems, Data Use & Access Laboratories, 1601 North Kent Street, Arlington, Virginia 22209; or call (703) 525-1480.

BIOGRAPHY

Gary Hill received a M.B.A. from Indiana University in 1961 and is the Director of Systems for Data Use and Access Laboratories (DUALabs). For the past seven years, he has been responsible for the development of generalized software systems designed specifically to access large statistical data files. He is a member of the Association of Public Data Users, American Statistical Association, and the Association of Computing Machinery.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

EVALUATION OF NONPARAMETRIC TESTS IN SPSS AND BMDP

F. Kent Kuiper and David L. Nelson
Boeing Computer Services, Inc., Seattle, Washington 98124

ABSTRACT

This paper presents the results of comparisons made between the nonparametric tests contained in the packages SPSS and BMDP. These tests were performed using both IBM and CDC versions of each package. The packages are evaluated for accuracy, readability, machine resources used, appropriateness and completeness of the collection of nonparametric tests.

Key words: BMDP; nonparametric statistical tests; SPSS; statistical program package evaluation; TALENT data.

1. INTRODUCTION

One important tool of statistical hypothesis testing that is beginning to be included in several of the major statistical program packages is a collection of nonparametric, or distribution-free, statistical tests. The tests themselves, which consist of roughly two dozen very commonly used procedures, are employed in a wide variety of fields, including the behavioral and health sciences, econometrics, agronomy and education. For these reasons, we feel that an evaluation of the nonparametric techniques offered by two of the major packages, SPSS (Statistical Package for the Social Sciences), Nie, et al. (1975) and Tuccy (1976), and BMDP (Biomedical Computer Programs), Dixon (1975), is a timely and worthwhile venture.

The evaluation of nonparametric statistical tests reported here was performed on versions of SPSS and BMDP installed on IBM 370 and CDC 6600 computers that are part of the Boeing Computer Services networks. The latest versions of SPSS, along with recent versions of BMDP, were tested. Specifically, SPSS Version 7.0 (IBM) was tested on an IBM 370/168 under OS/VS2 and Version 6.5 (CDC) was tested on CDC 6600/CYBER 74 under KRONOS 2.1. The BMDP tests were performed on these same machines. The CDC version of BMDP is a conversion supplied by the University of Massachusetts.

The evaluation was restricted to those nonparametric tests contained in SPSS procedures NPAR TESTS and NONPAR CORR and the BMDP program BMDP3S. As such, it does not include many nonparametric tests that are associated with contingency tables, which we feel is a topic worthy of separate consideration. Even so, the present evaluation covers 19 tests in SPSS and 8 in BMDP (see Table 1).

Suggested procedures for conducting statistical software evaluation, as outlined in Francis, et al. (1974,1975), have been adhered to as closely as possible. Comments in this article on package performance and suitability have been limited to discussions of the nonparametric tests per se, except in cases where a global package feature has a particularly profound effect on nonparametric procedures.

TABLE 1

TEST	TEST AVAILABLE IN SPSS	TEST AVAILABLE IN BMDP3S	PAGE NUMBER IN SIEGEL FOR TEST DATA
1. Binomial	X		40
2. 1-Sample Chi-Square	X		45
3. K-S 1-Sample	X		50
4. Runs	X		55 and 57
5. McNemar	X		65
6. Sign	X	X	70 and 73
7. Wilcoxon	X	X	79 and 82
8. Cochran Q	X		164
9. Friedman	X	X	171
10. Kendall Coeff. Conc. W	X	X	234
11. Median - 2-Sample	X		114
12. Mann-Whitney U	X	X	119
13. K-S 2-Sample	X		130
14. Wald-Wolfowitz	X		139
15. Moses	X		149
16. Median - K-Sample	X		182
17. Kruskall-Wallis	X	X	190
18. Kendall Rank Corr. Coeff.	X	X	205
19. Spearman Rank Corr. Coeff.	X	X	205

2. EXPERIMENTAL DESIGN

The experiment was designed to allow for the best possible comparisons, not only between the two statistical packages and between the two computers, but also among the nonparametric procedures themselves. Tests were performed on smaller data sets having 3 to 56 data points and on a larger data set of 505 data points. The analyses for each data set and for each package were constructed to be as identical as possible to maximize comparability.

2.1 Data sets. The data sets used for this experiment were chosen because of their appropriateness for the statistical procedures and for their general availability. The smaller data sets were taken from Siegel (1956). These varied in size from roughly 3 observations of 10 variables to 56 observations of 2 variables. Table 1 indicates the page in Siegel where the data set used for each procedure can be found. The larger data set was taken from Cooley and Lohnes (1971), Appendix B. This data set consists of 505 cases with 21 variables, and is referred to as the TALENT data set.

The placement of the test data for the smaller and larger data sets was different. The smaller sets were inserted directly into the SPSS and BMDP command files, following the READ INPUT DATA card and END/ card, respectively. The Cooley-Lohnes TALENT data set, on the other hand, was called from a separate disk file for both BMDP and SPSS.

2.2 Measurement goals and methodology. The primary goals in performing this experiment were to evaluate (1) accuracy of the results, (2) contents of the printed output, (3) cost, (4) documentation, and (5) ease of use. To accomplish this for any given procedure and data set, we wrote SPSS and BMDP code that would be as nearly equivalent as possible. Some of the ground rules we established in analyzing the TALENT data set were:

1) Perform 5 analyses on each run using the same nonparametric procedure.
2) Use the same variable sets and the same number of variables in each analysis.
3) Use variable names rather than indices - an option allowed for in both packages.

4) Exercise the syntactical options of the SPSS or BMDP code as appropriate.
5) Recode the values of variables only when necessary. This was done when categorical data was required, but not available.
6) Assume no missing values in the data sets.

The analyses using the smaller Siegel data sets correspond to the sample runs given in the SPSS update bulletin (Tuccy (1976)).

The tests on each procedure were run as separate jobs for the purpose of comparing costs. In total, there were 76 SPSS runs generated (19 IBM, 19 CDC for each of 2 data sets) and 32 BMDP runs (8 IBM, 8 CDC for each of 2 data sets). All runs were submitted via RJE devices and with the same job queueing priority.

3. RESULTS

3.1 Output formats. Output from both packages is easy to read, although the tabular and somewhat condensed output from SPSS seems to allow somewhat faster identification of relevant numerical information (test statistics, degrees of freedom, significance levels) than BMDP. For certain tests SPSS reports mean ranks for categories while BMDP reports rank sums. This use of rank sums by BMDP led to the overflow of the output format in the Kruskal-Wallis and Mann-Whitney tests when the TALENT data set was used. Similarly, when using this data set the Friedman test statistic overflowed the output format. When the chi-square statistic was calculated and some cell sizes were small, the IBM 370 version of SPSS issued a warning to that effect. This is a valuable addition not found in the 6600 version of SPSS or in either version of BMDP.

3.2 Features. All of the nonparametric tests in BMDP compute and print a table of the mean, standard deviation, minimum and maximum for each variable used. In SPSS, printing of such tables can be ordered optionally by the user through other procedures, although this was not done in the study reported here. Table 2 presents a complete description of the printed output obtained for each nonparametric test in SPSS and BMDP. By referring to this table, the user can quickly determine what information is available in the output from each procedure. This information can be valuable in choosing which package or procedure to use.

3.3 Ease of use. Both SPSS and BMDP are easy to use, regardless of the input medium employed for either command or data files.

3.4 Accuracy. In general, program accuracy did not appear to be a problem for either package or for either computing system. Some results did vary in the last 1-2 decimal places reported, presumably because of differing word length on IBM and CDC computers. In general, more decimal places were reported in the output formats of SPSS on the CDC 6600 than in the IBM 370 version. Test results for the Siegel data generally matched those reported in the text itself. One exception arose in an SPSS run of the Sign Test on the 6600, in which a quantity labeled as a 2-tailed probability was actually the 1-tailed probability. Also in the 6600 version of SPSS, Kendall's W was substantially different from that produced by the other runs using the same data.

The program logic of the tested procedures in both packages seemed to work well to the extent that it was exercised, with the exception of the Friedman test in the CDC 6600 version of BMDP3S. In this test, if n variables are present and the test is requested for any k of them, then the first k are tested. No such problem was encountered in the IBM 370 version of BMDP3S.

3.5 Core requirements. Because of its overlay structure and dynamic storage allocation, the CDC 6600 version of SPSS is able to execute in less than 70K words of core, while the 6600 version of the BMDP programs, which constitute separate entities but which

TABLE 2. STATISTICAL CONTENT OF PRINTED OUTPUT

Numerical codes:
(1) Count of total cases
(2) Count of cases in each category
(3) 2-tailed probability
(4) Significance
(5) Means, standard deviations min., max. of all variables
(6) Degrees of freedom

Test	SPSS	BMDP
Binomial	Hypothesized proportion, (1),(2),(3)	----
Chi-square	1 x n contingency table with expected values, χ^2, (2),(4),(6)	----
Kolmogorov-Smirnov one sample	370 Test distribution with parameters, max positive, negative and absolute differences, K-S Z statistic, (1),(3)	----
	6600 Test distribution with parameters, max difference, K-S Z statistic, (1),(3)	----
Runs	Cut point, number of runs, Z statistic, (1),(3)	----
McNemar	2 x 2 contingency tables χ^2 statistic or exact test, (1),(3)	----
Sign	No. of positive & negative differences, Z statistic or exact test, (1),(3)	No. of non-zero differences, smaller number of like-signed differences, 1-tailed probability, (5)
Wilcoxon	No. of positive & negative ranks with corresponding mean ranks, Z statistic, (1),(3)	No. of non-zero differences, smaller sum of like-signed ranks, 1-tailed probability, (5)
Cochran Q	2 x k contingency table, Q statistic, (1),(4),(6)	----
Friedman	Mean ranks, Friedman χ^2 statistic, (1),(4),(6)	Rank sums, Kendall's coeff. of concordance (W), Friedman χ^2 statistic, (4),(5),(6)
Kendall coeff. of Concordance	Mean ranks, W statistic, χ^2 statistic, (1),(4),(6)	(see Friedman)
Median 2-sample	2 x 2 contingency tables of No. of cases above and below median for each group, χ^2 statistic or exact test, (1),(3)	----
Mann-Whitney	Mean ranks for each category, exact probability for small samples, U statistic, Z statistic, (2),(3)	Rank sums for each category, U statistic and significance, Kruskal-Wallis χ^2 and signif.,(2),(4),(5),(6)
Kolmogorov-Smirnov 2-sample	Max positive, negative and absolute differences, K-S Z statistic, (2),(3)	----
Wald-Wolfowitz	No. of runs, Z statistic, (2),(3)	----
Moses	Span for full data set, 1-tailed probability, span for truncated data set, 1-tailed probability, No. deleted from full data set, (2)	----
Median k-sample	2 x k table of cases above and below median, Median, χ^2 statistic, (1),(4),(6)	----
Kruskal-Wallis	Mean rank for each group, χ^2 and significance, χ^2 and significance corrected for ties, (1),(2),(4)	Rank sum for each group, χ^2, (2),(4),(5),(6)
Kendall rank corr.	τ, (1),(4)	τ, (5)
Spearman rank corr.	r_s, (1),(4)	r_s, (5)

share a common subroutine library, require up to 150K words (110K for BMDP3S). The IBM 370 versions of SPSS and BMDP, on the other hand, require approximately 228K and 152K bytes of storage, respectively, for typical jobs.

SPSS was able to handle the larger TALENT data set on either the CDC 6600 or IBM 370 without increasing the default workspace. The BMDP workspace had to be increased for the TALENT data set for all tests, requiring the user to (1) calculate the additional space required, or (2) make an initial run with inadequate workspace allocation in order to find out how much more space should be requested for the final run.

3.6 Documentation. Documentation of available nonparametric tests is adequate, although referral to a text on nonparametrics is advisable to prevent misuse of some tests. The table of available analyses in Tuccy (1976) was felt to be particularly beneficial.

3.7 Cost. Resource and cost information for the 76 SPSS and 32 BMDP runs examined in this study is available from the authors upon request. The following conclusions can be drawn from that data:

- For smaller Siegel data sets, SPSS was much less costly than BMDP on the 6600, and only slightly less costly on the 370.
- For Siegel data, SPSS was less costly on the 6600 than on the 370, while BMDP showed the reverse.
- For Siegel data sets, a given package on a given machine yielded approximately the same cost, regardless of the nonparametric test used.
- For TALENT data, SPSS was much less costly to run than BMDP on either computer, even though the cost includes a proprietary surcharge for SPSS.
- For TALENT data, SPSS cost about the same for both CDC and IBM versions, while BMDP was slightly more costly on the 6600 than on the 370 for most tests.
- For TALENT data, a given package on a given machine yielded approximately the same cost, regardless of the nonparametric test used, with the following exceptions: (1) computation of the Kendall rank correlation coefficient grew in cost dramatically faster than the other tests in going from a smaller to larger data set; (2) BMDP's Friedman test yielded an unexpectedly high cost on the IBM 370.

3.8 Differences between programs. Aside from a differing collection of nonparametric tests offered by the two packages, many other notable differences arose. One, in the area of missing value treatment, made the exercise of this option in the two packages inappropriate. In BMDP3S, a missing value for any variable listed in the USE= sentence of the VARIABLE paragraph causes deletion of that case. On the other hand, SPSS deletes a case only when a variable actually being tested is missing. Since the VARIABLE paragraph in BMDP3S is outside the inner loop for testing (see Dixon (1975) p. 659), the only way to consider missing data equivalently in both packages would have been to run the BMDP3S TALENT data tests as separate problems. This approach was felt to put an unfair penalty on the BMDP program evaluation.

Several differences were noted in the tests themselves, most of which are pointed out in Table 2. In addition, the evaluation uncovered a difference in the computation of Mann-Whitney U statistics in some instances. BMDP3S assumes the first variable listed is the control variable, while SPSS apparently assumes the variable with the larger mean rank is the control. Thus the U statistic can differ in the two programs, although the significance levels are the same.

Also, some differences were observed between the 6600 and 370 versions of SPSS in the Wald-Wolfowitz and Moses tests. When comparisons were made between the runs performed on the Siegel data no problems were found; however, with the TALENT data, the two versions gave different results. This was probably due to the fact that the TALENT variables contain a large number of ties, which are treated somewhat differently by the two packages (which brings into question the use of these tests for the TALENT variables).

3.9 Needed features. One major conclusion is that it would be desirable if BMDP3S

would offer more nonparametric tests than those currently available, particularly for nominal data and for data comparisons (chi-square test, Kolmogorov-Smirnov, etc.). Both packages would benefit from the inclusion of some graphical capability that particularly applies to nonparametric tests. The K-S test, for example, could compare two cumulative distributions graphically.

3.10 Other packages. Other available packages offer or soon will offer nonparametric tests. STAT/BASIC, an IBM BASIC-language interactive package, has several distribution-free tests. The new version of the Statistical Analysis System, SAS 76.5, will include a procedure NPAR1WAY for one-way rank tests. Comparison of this procedure with corresponding tests in SPSS and BMDP should be included in an expanded version of this article.

5. REFERENCES

COOLEY, W. W. and LOHNES, P. R. (1976). Multivariate Data Analysis. John Wiley, New York.

DIXON, W. J., Ed. (1975). BMDP Biomedical Computer Programs. University of California Press, Berkeley.

FRANCIS, I., HEIBERGER, R. M. and VELLEMAN, P. F. (1974). Report and proposal of the committee on evaluation of program packages. Amer. Statist. Assoc.

FRANCIS, I., HEIBERGER, R. M. and VELLEMAN, P. F. (1975). Criteria and considerations in the evaluation of statistical program packages. The American Statistician 29, 52-56.

NIE, N. H., HULL, C. H., JENKINS, J. G., STEINBRENNER, K. and BENT, D. H. (1975). SPSS: Statistical Package for the Social Sciences, 2nd ed. McGraw-Hill, New York.

SIEGEL, S. (1956). Nonparametric Statistics for the Behavioral Sciences. McGraw-Hill, N.Y.

TUCCY, J. (1976). SPSS Subprogram NPAR Tests: nonparametric statistical tests. Northwestern University Manual No. 324 (Rev. B).

BIOGRAPHIES

F. Kent Kuiper received an M.S. in statistics from the University of Washington in 1971 and is a senior statistician with Boeing Computer Services (BCS). In addition to directing the statistical portions of engineering and marketing projects, he provides evaluation, testing and consultation for the statistical packages maintained by BCS.

David L. Nelson is a senior statistical analyst for BCS. Since receiving his Ph.D. from Texas Tech in 1969, he has been involved in statistical program development and maintenance and statistical consultation. His publications include articles on linear and nonlinear regression, probability distributions, statistical computing and characterizations.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 503
Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface
Held at Nat'l. Bur. of Stds., Gaithersburg, MD, April 14-15, 1977. (Issued February 1978)

CURRENT USE OF COMPUTERS IN THE TEACHING OF STATISTICS

Gary W. Tubb and Larry J. Ringer
Northwestern State University, Natchitoches, LA 71457

ABSTRACT

Conceptually, the current use of computers has taken two forms in the teaching of elementary statistics: integrating the content of statistics with that of computers; and integrating methods of instruction of statistics by use of computers. In the first half of this paper, three computer language textbooks are reviewed. Each uses statistics as a content area presenting programming problems. Also, three textbooks which focus on learning statistics using the computer as an aid are reviewed. The second half of this paper surveys six published articles that evaluate courses employing "hands-on" computer instruction (CAI) and also, many published articles evaluating courses employing a demonstrational mode of instruction. The generation and use of simulated experimental data and interactive vs. non-interactive computerized statistical packages are reviewed. Extensive recommendations for integrating computers into the teaching of statistics are included. The complete paper has been submitted to ERIC with sixty-two references and thirty-four pages.

Key words: Computer assisted instruction; computer, statistical texts; demonstrational statistical methods; simulation; statistical, computer texts; statistical content; statistical instruction; statistical interactive packages; statistical non-interactive packages; teaching statistics.

1. INTRODUCTION

Within the last five years a revolution has occurred in all courses that require calculations. From primary grades to post-doctoral study, the inexpensive electronic pocket calculator has had a pervasive impact upon the curricula. But, just as the introduction of calculators in courses of statistics greatly influenced the development of the analysis of variance and experimental design, so too can we expect the introduction of inexpensive, programmable computers to have a greater influence on the development of statistical theory, practice, and teaching.

The impact of computers on statistical theory is best exemplified by the recent work on matrix decompositions, generalized inverses, and multivariate analysis. Many of the classical, hand-calculator based methods have now become obsolete or have been revised with the advent of computers. The computer has already changed statistical practice. Extensive plotting of data and residuals is quite common. There has been a shift of emphasis from general tables of statistical functions, to direct evaluation of discrete values. It is unusual not to see "p-values" reported in research articles. Whereas, ten years ago ".05", ".01", and "ns" were commonplace. Jack-knifing is an example of a statistical technique whose widespread application would not have been seriously considered before the advent of computers, but it is now included in the curriculum. Evans (1973) gives an excellent review of the influence of computers on modern statistics.

Yet, for all the impact computers have had on the theory and practice of statistics,

only recently have there been attempts to integrate the use of computers with the teaching of elementary statistics. The purpose of this article is to explore the current use of computers in the teaching of elementary statistics. Conceptually, this exploration will be in two forms. First, the various means of integrating computers into the content of elementary statistics will be examined. Criteria for identifying the strengths and weaknesses of published textbooks and computerized statistical packages will be examined from the viewpoint of content relevance. Second, the various methods of implementing the integrated content of statistics and computerized statistical packages will be surveyed. Although the principle emphasis is on an introductory non-calculus course, most of the methods described in the second part of this paper are useful in higher level courses in statistics and research methodology.

2. INTEGRATING THE CONTENT OF STATISTICS AND COMPUTERS

Recent introductory textbooks attempting to integrate statistics with computers appear to take two forms. In the first form, the object is to learn a computer language, where statistics is used as a content area presenting program problems. The second form that many recent introductory statistical textbooks have attempted, focuses on learning statistics using the computer as an aid. In these textbooks, a higher order computer language or a "canned" statistical package is used as a vehicle facilitating rapid computation and/or insight of statistical texts and procedures.

2.1 Introductory computer programming textbooks with statistics. Three introductory programming textbooks using statistics as a vehicle for teaching FORTRAN are: Introductory Statistics with FORTRAN by Kirch (1973); FORTRAN Programming for the Behavioral Sciences by Veldman (1967); and Introduction to Statistics and Computer Programming by Kossack et al. (1975). All three of the textbooks attempt to complement and enhance statistical development. However, each of the three textbooks overwhelmingly emphasize the learning of FORTRAN at the expense of the statistical content. Each of the texts orderly organizes the introduction of FORTRAN from I/O media to program libraries, from simple FORTRAN statements to branching, and from simple manipulation of constants to complex operations upon arrays. The incorporation of previous statistical exercise programs as subroutines of subsequent statistical programs is common to all three texts. Such exercises provide a sense of accomplishment and utility in programming. In reality, statistical programs are built from repetitive meaningful components much in the same way that a statistician's repertoire of designs and analyses originates. However, it is questionable whether the content of such texts should be used in introductory statistics courses. Such texts would better serve the teaching of computer languages.

2.2 Introductory statistical textbooks with computer applications. A classic in the field is Lohnes and Cooley's (1968) Introduction to Statistical Procedures with Computer Exercises. In general though, the content of this text is too advanced for an elementary statistics course. An elementary supplement that follows in the tradition of Lohnes and Cooley is A Computer-assisted Approach to Elementary Statistics: Examples and Problems by Bulgren (1971). The book can be used as a supplement to introductory statistical texts Adler and Roessler (1968), Freund (1967), Hoel (1966), Huntsberger (1967), or Mendenhall (1971). The strong point of Bulgren's supplement is the insight a student can gain through the simulation and manipulative capabilities of the computer. The book consists of exercises to be solved by either writing FORTRAN programs or punching the programs in the appendices. The overlaying of Bulgren's supplement on an elementary text would be a compromise between the strict computer programming texts on statistics and the following texts. Each of the following textbooks use statistical packages or simple "canned" subroutines: Introduction to Statistics and Data Analysis with Computer Applications I & II by Morris and Rolph (1971); Statistics for Education: With Data Processing by White (1973); and Statistical Analysis: A Computer Oriented Approach by Afifi and Azen (1972). The emphasis of the texts is on how and when to use existing statistical techniques. One author states that "computer use replaces theorem proving". The author's statement indicates the

increasing amount of statistical analysis done by researchers with a modest amount of statistical experience. Packages statistical programs have made this possible. Probably none of the books mentioned above would satisfy the needs of every instructor. However, this small number of texts are among the first to recognize the interelationships of statistics and computers. As long as statistical content is not sacrificed, then experimentation of this type has the potential of producing significant improvements upon the quality of elementary statistics courses. Following are guide lines for integrating statistics with computers in textbooks:

1. Cost-efficiency is of prime concern in selecting a general statistical package vs. "canned" programs. Student programming in a low level language is expensive.
2. The emphasis on computers should be on large data set manipulation and visual display (plotting, histograms, residuals).
3. Simulation of data with pseudo-random computer generated numbers is expensive. Alternative non-computer simulations should be used where it is feasible.
4. Statistical techniques antiquated by the computer should be dropped from texts if the techniques contribute nothing to statistical content eg. short-cut approx.
5. I/O of statistical programs should be included in textbooks.
6. Authors should provide machine readable data bases for text and exercises.

Following are reasons for interweaving large statistical packages like BMD, SSP, SPSS, etc. into textbooks:

1. Programs are shorter and easier to write.
2. Programs usually work the first time.
3. Typically such programs produce results of several different types of calculations that a programmer might not have bothered to include in his own program.
4. Writing programs to perform data manipulations in FORTRAN can be tedious.
5. The I/O is fairly uniform from one installation to another.
6. Virtually all analyses are available.
7. Most researchers publish results generated by large statistical packages.

Following are reasons for not interweaving large statistical packages into textbooks.

1. The textbook could not be used at the majority of Universities due to the large computer support system required.
2. Even one run of a program by a student is very expensive.
3. Student fails to grasp theoretical understanding that results from writing his own program.
4. Large statistical packages confuse the student with results that are explained in advanced courses.

Following are reasons for interweaving simple "canned" statistical packages into textbooks:

1. It is unnecessary to learn a computer language.
2. Programs require a small amount of core and can be run at most installations.
3. Saves class time in teaching mechanics.
4. Programs are task specific, hence more efficient and less expensive to use.
5. Provides easy access to standard techniques.

Following are reasons for not interweaving "canned"statistical packages into textbooks:

1. Mindless use of statistical programs replaces the intelligent use of theory.
2. Canned programs can be time consuming if the data output of one program is not compatible with input of other programs.
3. If the data output of one program conforms to the input of another, then data must be stored in a more expensive form than for higher level, more comprehensive statistical packages e.g. computer cards.
4. Canned programs are often machine dependent.

3. METHODS OF INSTRUCTION BY INTEGRATING STATISTICS AND COMPUTERS

As we have seen, the use of computers in statistical content has taken on at least two forms; emphasis on computer language at the expense of statistics or vice versa. In the following discussion it will become clear that the computer can serve many facets in the process of statistical instruction. For convenience, the first half of this section will

deal with published evidence of courses employing a "hands-on" computer mode of instruction (student) and courses employing a demonstrational computer mode of instruction (teacher). The second half of this section will deal with the generation and use of simulated experimental data and interactive vs. non-interactive statistical packages.

3.1 Hands-on. One of the most all encompassing methods of "hands-on" instruction of theoretical material is Computer Assisted Instruction (CAI). Wassertheil (1969) successfully incorporated CAI into the laboratory portion of an introductory statistics course. In her study, the main use of CAI was to individualize instruction. Each student progressed at his own pace. Perhaps the most positive result of Wassertheil's study was that one 75 minute class period per week could be eliminated without deterioration of student performance. The benefit of CAI would be the freeing of the instructor for individual student contact or other duties. Three other studies incorporating the computer to various degrees are reviewed in the complete report of this study. However, an extreme worth mentioning is Skavaril's study (1974) in which the computer is incorporated into all phases of instruction in an introductory service statistics course. In his study, the computer not only provided tutorial CAI support, but also generated statistical exercises and answers, and provided subroutines for complete data analysis. Skavaril used twenty-nine CAI modules, nine exercise generating programs, and twenty-one data analysis programs in his system. A great deal of class time was saved at no expense to learning as measured by the final examination. In addition, the author notes that the exercise-generating and CPS programs, provide additional gains, since the student receives a unique set of data; cribbing is eliminated. Freeing the student of the tedium of calculations allows him to analyze several sets of data and "to build, by comparing statistics between analysis, empirical evidence concerning the underlying distribution of those statistics."

3.2 Demonstrational. In essence, this section simply questions to what extent student involvement with the computer is cost-efficient in the teaching of an elementary statistics course. For example, is it necessary that every student individually simulates the Central Limit Theorem, or individually simulates the meaning of "5%" statistical significance by repeating an experiment 100 times on the computer as described earlier in Bulgren's supplementary textbook? Filming or video taping these computer simulations could provide the same learning at far less cost. Another question is, how cost-efficient is it to generate unique data sets for each individual's homework? These questions truly relate to the merits of the statistical laboratory.

The instructor can do many things with the computer to provide useful information for the statistics classroom or laboratory. A compiled set of statistical problems with computer solutions eliminates expensive student use of the computer and unnecessary learning of the mechanics of programming. Computer graphing of theoretical distributions, populations, samples, or transformations can easily be compiled into booklet form available for student purusal. Wegman and Gere (1972) produced a workbook of problems with computer solutions and a set of forty slides illustrating a variety of distributions, densities, and histograms available at cost. Recent articles by Edgell, Lehman, Starr, and Young (1975), Kanji (1974) Tanis (1973), and Abranovic et al. (1972) offer a large number of methods for the use of the computer or simulating equipment as supplements to a course in statistics. Some of the reasons to use computers to aid in learning or teaching statistics are identified by Andrews (1973).

3.3 Simulation. Not only can the computer eliminate the tedium of computations, it can also eliminate the collection, input, storage, and manipulation of data. A computer can be a fancy random number generator. Statistical designs can be specified for populations of known parameters. An extensive data generation system, EXPERSIM (Main, 1971), is a set of sophisticated computer simulation models for various experimental situations in specific subject areas, eg. imprinting, drug research, motivation. Each simulation includes a complete description of the experimental setting, built in controls, number of variables that can be manipulated, and the sample data. The student may then analyze the experimental data by requesting statistical routines. STEXSIM (W. Thomas, 1972), STATSIM (D. Thomas, 1971), as well as three other simulation packages are reviewed in the complete report of this study.

3.4 Interactive vs. non-interactive statistical packages. Large statistical packages such as BMD, IMSL, SAS, and SPSS are prohibitively expensive (core and external device requirements) for student use in introductory statistics courses. The amount of core and the use of disk and magnetic tape rapidly increases the cost of processing such packages. Smaller packages have been developed too. MINITAB, OMNISHRIMP, OMNITAB, TUSTAT-II, STRAP-I, and STP are reviewed with regard to their interactive nature in the complete report of this study (to be available from ERIC).

4. REFERENCES

ABRANOVIC, W., AGELOFF, R., and FREDRICK,D. (1972). Time-sharing computer systems as a teaching tool. Amer. Stat., 26(1), 34-38.

ANDREWS, D. (1973). Developing examples for learning statistics; data and computing. Intl. Stat. Rev., 41(2), 225-228.

EDGELL, S., LEHMAN, R., STARR, B., and YOUNG, K. (1975). Computer aides in teaching statistics and methodology. Bhv. Res. Meth. & Inst., 7(2), 93-102.

EVANS, D. (1973). Computers in the teaching of statistics. Jour. Royal Stat. Soc., 136, 153-190.

KANJI, G. (1974). The role of the statistical laboratory in the teaching of statistics. Intl. Jour. Math. & Sci. Tech., 5, 53-57.

MAIN, D. (1971). A computer simulation approach for teaching experimental design. Paper presented at APA national meeting 1971.

SKAVARIL, R. (1974). Computer-based instruction of introductory statistics. Jour. Comptr. Based Inst., 1(1), 32-40.

TANIS, E. (1973). A computer laboratory for mathematical probability and statistics. ERIC, ED# 079 985.

THOMAS, D. B. (1971). STATSIM: Exercises in statistics. ERIC, ED# 055 440.

THOMAS, W. H. (1972). The development of a statistical experiment simulator: final report. ERIC, ED# 063 804.

WASSERTHEIL, S. (1969). Computer assistance in statistics. Imprv. Col. & Univ. Tch., 17(4), 264-266.

WEGMAN, E. and GERE, B. (1972). Some thoughts on computers and introductory statistics. Intl. Jour. Math. Ed. & Sci. Tech., 3, 211-221.

BIOGRAPHIES

Gary W. Tubb earned a Ph.D. in EDCI (1974) and completed a post-doctoral Master of Statistics (1976) at Texas A&M University. For the past two years, he has been Director of Educational Research at Northwestern State University. During this time he has implimented a statistical package for the institution.

Larry J. Ringer is a professor of statistics for the Institute of Statistics at Texas A&M University.

LIST OF PARTICIPANTS

Earl C. Abbe
1402 Cola Drive
McLean, VA 22101

Kerry Adkisson
Dept. of Agriculture
SRS
Washington, DC 20250

Cynthia Agard
Bureau of Census
SRD
Suitland, MD 20233

Murray Aitkin
Dept. of Mathematics
Univ. of Lancaster
Lancaster, England
LA1 4YL

James R. Allen
Academic Computing Ctr.
University of Wisconsin
Madison, WI 53706

Scott Allman
Computing Center
University of Colorado
Boulder, CO 80309

John Alman
Boston U. Computing Ctr.
111 Cummington Street
Boston, MA 02146

David Altvater
2714 Terrace Road, S.E.
Apt. B615
Washington, DC 20020

Ingrid A. Amara
Dept. of Biostatistics
U. of North Carolina
Chapel Hill, NC 27514

Gary D. Anderson
26235 33rd Ave. S.
Kent, WA 98031

Ronn Andrusco
Box 468
Postal Station J
Toronto, ON
Canada M4J4Z2

J. Douglas Ashbrook
Nat. Inst. of Health
DCRT, CCB
Bldg. 12, Rm. 2228
Bethesda, MD 20014

Richard Bailey
516 Front Street
Perryville, MD 21903

Philip W. Baker
728 Adams Bldg.
Phillips' Petroleum Co.
Bartlesville, OK 74004

JoAn E. Barnes
Statistics Dept.
Oregon State Univ.
Corvallis, OR 97331

Bruce D. Barnett
ARRADCOM/DOVER
ATTN: DRDAR-MSM
Dover, NJ 07801

John Barone
13 Ontario Way
Trenton, NJ 08648

Anthony J. Barr
SAS Institute Inc.
P.O. Box 10066
Raleigh, NC 27605

Ronald M. Bass
Office of Computer Science
Dept. of the Treasury
1625 I St., N.W.
Washington, DC 20220

Carl B. Bates
1200 Paul Lane
Fredericksburg, VA 22401

Douglas Bates
115 King Street, W.
Kingston, ON
Canada K7L2W6

Leonard R. Bayer
38 Gaslight Lane
Rochester, NY 14610

Lorraine Bayer
38 Gaslight Lane
Rochester, NY 14610

Albert Beaton
Educational Testing Serv.
Princeton, NJ 08540

Karen Becker
2014 Columbia Pike
Apt. #2
Arlington, VA 22204

Richard A. Becker
Bell Laboratories
Murray Hill, NJ 07974

Jay H. Beder
HEW
330 C St., S.W.
Room 2605 MES
Washington, DC 20201

Prem Nath Bhalla
Jackson State Univ.
Jackson, MS 39217

Stephen Bingham
Cooperative Studies Prog.
Coordinating Ctr. (151e)
VA Hospital
Perry Point, MD 21921

David Blaxell
Rm. 506 Corporate Res. Br.
Place Du Portage, Phasell
Ottawa, ON
Canada K1A 0C9

Peter Bloomfield
Princeton University
Department of Statistics
201 Fine Hall
Princeton, NJ 08540

Brent A. Blumenstein
510-T
E. Ponce DeLeon Ave.
Decatur, GA 30030

Paul T. Boggs
U.S. Army Res. Office
P.O. Box 12211
Research Triangle Park
NC 27709

N. R. Bohidar
Merck Sharp & Dohme
Res. Lab.
West Point, PA 19486

Steven R. Borbash, Jr.
14 McLane Avenue
Morgantown, WV 26505

Hubert Boliver
Dept. of Computer Sci.
SUNY
Plattsburgh, NY 12901

Herbert Bown
Image Communications
Communication Res. Centre
Ottawa, ON
Canada

Jere T. Bracey
1060 Ridgewood Drive
Bolingbrook, IL 60439

Douglas B. Bracy
Bu. Economic Analysis
1401 K Street, N.W.
Washington, DC 20230

Laurence R. Brady
2307 S. Lexington Dr.
#308
Mt. Prospect, IL 60056

Jan Bramhall
Applied Physics Lab.
Johns Hopkins Univ.
Johns Hopkins Road
Laurel, MD 20810

William M. Brelsford
Bell Laboratories
Holmdel, NJ 07733

Shirley G. Bremer
A-337 Admin. Bldg.
National Bureau of Stds.
Washington, D.C. 20234

John Brode
23 Berkeley Street
Cambridge, MA 02138

Harold Brodsky
Dept. of Geography
Univ. of Maryland
College Park, MD 20742

Judith Bromberg
Environmental Medicine
MSB213-550 1st Avenue
NYU Medical Center
New York, NY 10016

Robert N. Brown
Biostatistics Center
George Washington Univ.
7979 Old Georgetown Rd.
Bethesda, MD 20014

Richard H. Browne
U. of Texas Health Sci.
Ctr./Medical Comp. Sci.
Dallas, TX 75235

G. Rex Bryce
210 TMCB
Brigham Young Univ.
Provo, UT 84602

Jeff A. Buchanan
1405 Farrell Lane
Richland, WA 99352

Richard K. Buchness
Health Sci. Comp. Fac.
UCLA
Los Angeles, CA 90024

Roald Buhler
Computer Center
87 Prospect Ave.
Princeton, NJ 08540

Shirrell Buhler
Computer Center
87 Prospect Avenue
Princeton, NJ 08540

Laurie Burch
Biostatistics Center
George Washington Univ.
7979 Old Georgetown Rd.
Bethesda, MD 20014

Philip R. Burns
6031 N. Neva
Chicago, IL 60631

David E. Burris
Colgate-Palmolive Co.
Box 175
New Brunswick, NJ 08903

Philip F. Busby, Jr.
204 Short Street
Chapel Hill, NC 27514

Robert H. Byers, Jr.
1271 Oxford Road, N.E.
Atlanta, GA 30306

Gordon R. Caldwell
Center for Demography & Ecology
Univ. of Wisconsin
Madison, WI 53706

Richard T. Campbell
Department of Sociology
Duke University
Durham, NC 27706

William A. Carpenter
Box 3817 University Sta.
Charlottesville, VA 22903

Steven T. Carrier
132 N. Lincoln Street
Pearl River, NY 10965

Janet C. Cassady
Dept. of Biostatistics
Univ. of Miami Med. School
P.O. Box 520875
Miami, FL 33152

David Cavander
Charles River Assoc.
1050 Massachusetts Ave.
Cambridge, MA 02138

J. M. Chambers
Bell Laboratories
Murray Hill, NJ 07901

I-Ming Chang
90 Meyer Road
Apt. 220
Amherst, NY 14226

Steven Chasen
3114 17th Street
Santa Monica, CA 90405

Banvir S. Chaudhary
Room 209, H.I.P.
625 Madison Avenue
New York, NY 10022

Hsiv-Ying Cheng
Geomet, Inc.
15 Firstfield Road
Gaithersburg, MD 20760

J. C. Chetrit
215 Berkeley Pl.
Brooklyn, NY 11217

Dave Christiansen
Polks Landing #91
Chapel Hill, NC 27514

Chang-Jo F. Chung
601 Booth Street
Ottawa, ON
Canada K1A 0E8

Daniel A. Church
9223 Weathervane Pl.
Gaithersburg, MD 20760

Calvin Cillay
Box 1242
Rockville, MD 20850

Faye Citron
Univ. of Chicago
Graduate School of Bus.
5836 South Greenwood
Chicago, IL 60637

Frank C. Clark
Box 8093
Georgia Southern College
Statesboro, GA 30458

James J. Colaianne
Food & Drug Adm.
HFV-105
5600 Fishers Lane
Rockville, MD 20857

James Condie
Federal Reserve Board
Washington, DC 20551

William Conley
Apt. 208
275 Askin Avenue
Windsor, ON
Canada

Richard E. Cooper
Rm. 013, NAL Bldg.
Route 1
Beltsville, MD 20705

Ronald L. Copp
P.O. Box 1125
29 Bay Road
Duxbury, MA 02332

Gerald F. Cotton
NOAA
Silver Spring, MD 20910

Richard W. Coulter
6161 Edsall Road
Apt. T-2
Alexandria, VA 22304

Charles D. Cowan
Bur. of Census
Rm. 3339 FOB #3
Demographic Surveys Div.
Washington, DC 20233

Lawrence H. Cox
Bur. of Census
Suitland, MD 20233

Frances Bardello Craig
R.D. 2
Valencia, PA 16059

Giles L. Crane
73 Philip Drive
Princeton, NJ 08540

David H. Culver
Dept. of Statistics
& Computer Science
Univ. of Georgia
Athens, GA 30601

Gary Cutter
Suite 1114
Coordinating Ctr.
HD & Followup Program
Houston, TX 77030

Leonard P. D'Amato
NATL-CSD-PSB
Patuxent River, MD 20670

Nancy A. David
416 S. Royal Street
Alexandria, VA 22314

Herbert T. Davis
Sandia Labs
Albuquerque, NM 87115

John E. Dennis
Computer Science Dept.
Cornell University
Ithaca, NY 14850

Douglas J. DePriest
ONR
Washington, DC 20375

Kiran A. Desai
100 North First Street
Springfield, IL 62777

Alexander Diament
Federal Reserve Bank
of Philadelphia
100 N. 6th Street
Philadelphia, PA 19105

Peter Dickinson
Ctr. for Demography
and Ecology
Univ. of Wisconsin
Madison, WI 53706

S. R. Divi
Geological Sur. of Canada
601 Booth St., Rm. 122
Ottawa, ON
Canada K1A 0E8

Richard Dosch
Boston U. Comp. Ctr.
111 Cummington St.
Boston, MA 02146

Howard C. Duffield
The MITRE Corp.
METREK Division
1820 Dolley Mad. Blvd.
McLean, VA 22101

Sharon M. Duncan
Rt. 10, Box 364R
Charlotte, NC 28213

Robert M. Dunn
52 McDougal Rd.
Waterloo, ON
Canada N2L 2W5

William Dunn
Congressional Budget Off.
Washington, DC 20515

Nestor Dyhdalo
3020 N. Neenah
Chicago, IL 60634

Churchill Eisenhart
B-268 Metrology Bldg.
National Bur. of Stds.
Washington, DC 20234

Henry Elkins
15 Willow Circle
Bronxville, NY 10708

Daniel L. Elliott
2404 W. Penn. Ave.
Statistical Sci. Dept.
Evansville, IN 47721

Laszlo Engelman
UCLA-HSCF
CHS AV-360
Los Angeles, CA 90274

Andrea G. Fabbri
Geological Sur. of Canada
601 Booth St., Rm. 122
Ottawa, ON
Canada K1A 0E8

Ronald Fairbrother
Charles River Associates
1050 Massachusetts Ave.
Cambridge, MA 02138

Ronald D. Farnan
12804 Hollins Place
Bowie, MD 20715

Stephen Fautman
Federal Reserve Board
Washington, DC 20551

Frances Fazio
Applied Physics Lab.
Johns Hopkins Univ.
Johns Hopkins Rd.
Laurel, MD 20810

Harry Feingold
11801 Prestwick Rd.
Potomac, MD 20854

William Fellner
Appalachian Labs.
Rm. 227
P.O. Box 4292
Morgantown, WV 26505

James J. Filliben
A-337 Admin. Bldg.
National Bur. of Stds.
Washington, DC 20234

Robert H. Finch, Jr.
4618 West Hill Rd.
Ellicott City, MD 21043

I. Fishman
NIH, Bldg. 12A
Room 304
Bethesda, MD 20014

Sylvia Fleisch
Boston U. Comp. Ctr.
111 Cummington Street
Boston, MA 02146

Nancy Flournoy
Dept. of Oncology
FHCRC
1124 Columbia St.
Seattle, WA 98294

James D. Foley
Bureau of the Census
Washington, DC 20233

Roger W. Foster
P.O. Box 33529
AMC Branch
WPAFB, OH 45433

Michael Fox
City of Hope Med. Ctr.
& U.C.L.A.
Duarte, CA 91010

Lewis F. Frain
10081 Maplewood Drive
Ellicott City, MD 21043

Ivor Francis
358 Ives Hall
Cornell University
Ithaca, NY 14853

James W. Frane
Health Science Comp.
Facility, U.C.L.A.
Los Angeles, CA 90024

Barbara Friedman
34 Superior Road
Rochester, NY 14025

E. L. Frome
University of Texas
Austin, TX 78712

A. Ronald Gallant
P.O. Box 5457
Raleigh, NC 27607

Paul H. Geissler
U.S. Fish & Wildlife
Service (MBHRL)
Laurel, MD 20811

James E. Gentle
Statistical Laboratory
Iowa State University
Ames, IA 50011

Jane F. Gentleman
Dept. of Statistics
Univ. of Waterloo
Waterloo, ON
Canada N2L 3G1

James E. George
Los Alamos Scientific
Los Alamos, NM 87544

Thomas M. Gerig
Dept. of Statistics
North Carolina State U.
Raleigh, NC 27607

Michele C. Gerzowski
Room 8A-35
NCHS
5600 Fishers Lane
Rockville, MD 20857

Paul H. Gibbs
18546 Bayleaf Way
Germantown, MD 20767

Roderic D. Gillis
Campground Road
Port de Posit, MD 21904

Carol Glascock
3615 Barcroft View Ter.
Apt. 304
Bailey's Cross Rds., VA
22041

Bruce L. Golden
College of Bus. & Mgmt.
Univ. of Maryland
College Park, MD 20742

Gordon D. Goldstein
Office of Naval Res.
Code 437
Arlington, VA 22217

J. H. Goodnight
SAS Institute
P.O. Box 10066
Raleigh, NC 27605

Paul A. Green
Dept. of Oral Medicine
Univ. of PA Dental School
Philadelphia, PA 19104

Michael Greenberg
2934 Hannah Avenue
A-107
Norristown, PA 19403

Richard L. Greenstreet
Cleveland Clinic
9500 Euclid Avenue
Cleveland, OH 44106

Ronald K. Gress
US Army Computer Systems
Command, STOP C-60
Ft. Belvoir, VA 22060

Patricia E. Griffin
Bur. of the Census
FOB #3
Room 3581
Washington, DC 20233

Joan M. Gurian
P.O. Box 22
Garrett Park, MD 20766

Cathryn L. Gust
HFV-105
5600 Fishers Lane
Rockville, MD 20857

Donald Guthrie
760 Westwood Plaza
Los Angeles, CA 90024

Peter Gutterman
2144 California St., NW
Washington, DC 20008

O. P. Hackney
Mississippi State U.
Dept. of Comp. Science
Mississippi State, MI
39762

Richard A. Hall
5309 Riverdale Road
#302
Riverdale, MD 20840

Dan Hallesy
Economic Research Ser.
Washington, DC 20250

Peggy M. Hamilton
Food & Drug Admin.
Bur. of Radiological Hlth.
5600 Fishers Lane
Rockville, MD 20857

Kenneth A. Hardy
Social Science Stat. Lab.
IRSS, UNC
Chapel Hill, NC 27514

Joseph O. Harrison, Jr.
National Bur. of Stds.
Washington, DC 20234

Douglas Hasslen
Dept. of Agriculture
SRS
Washington, DC 20250

Lee-Ann C. Hayek
Smithsonian Institution
MNH W101
Washington, DC 20560

Roy E. Heatwole
DHEW, NCHS
5600 Fishers Lane
Rockville, MD 20857

Richard Heddingar
Office of Systems & Stds.
Dept. of Labor
Bur. of Labor Statistics
Washington, DC 20212

Richard M. Heiberger
Dept. of Statistics
Wharton School
Univ. of Pennsylvania
Philadelphia, PA 19104

George Heller
1017 Robroy Drive
Silver Spring, MD 20903

William J. Hemmerle
Dept. of Comp. Science &
Exp. Statistics
1A Tyler Hall
Kingston, RI 02881

Donald Henderson
Room 013, NAL Building
Route 1
Beltsville, MD 20705

Gary L. Hensler
Patuxent Wildlife Res.
Center
Laurel, MD 20811

David G. Herr
Math. Dept.
UNC-G
Greensboro, NC 27412

J. Michael Hewitt
3703 Maryland Street
Alexandria, VA 22309

Gary L. Hill
DUALabs
1601 N. Kent Street
Suite 900
Arlington, VA 22209

Norman Hiller
Veterans Administration
(173B)
Washington, DC 20420

Hugh T. Hinman
2337 18th Street, NW
Washington, DC 20009

William Hoagland
Congressional Budget Off.
Washington, DC 20515

David C. Hoaglin
Dept. of Statistics
1 Oxford Street
Cambridge, MA 01776

R. R. Hocking
Dept. of Comp. Science
Mississippi State U.
Mississippi State, MI
39762

Howard J. Hoffman
5523 Northfield Road
Bethesda, MD 20034

David Hogben
A-337 Admin. Bldg.
National Bur. of Stds.
Washington, DC 20234

John Hohwald
Dept. of Statistics
Dietrich Hall
Univ. of Pennsylvania
Philadelphia, PA 19174

Donald A. Holzworth
Battelle Toxicology
Program Office
7405 Colshire Drive
McLean, VA 22101

Samuel A. Hood, Jr.
Federal Reserve Bank
of Philadelphia
100 N. 6th Street
Philadelphia, PA 19105

Wayne Hoover
CSD U.S. Naval Air
Test Center
Patuxent River, MD 20670

Tom Hopper
3703 Adams Drive
Wheaton, MD 20902

David W. Hosmer
Univ. of Massachusetts
Amherst, MA 01003

Trina A. Hosmer
Univ. of Massachusetts
Amherst, MA 01003

Francis Hsuan
40 Gill Lane
Apt. 2E
Iselin, NJ 08830

James Hudson
1731 New Hampshire Ave.
N.W.
Washington, DC 20009

Michael Hunst
Dept. of Agriculture
SRS
Washington, DC 20250

Rex L. Hurst
Applied Stat./Comp. Sci.
Utah State University
Logan, UT 84322

Jerry L. Ivey
Monsanto Res. Corp.
Mound Laboratory
Miamisburg, OH 45342

David Jackson
NIMH
(Mental Health Sty Center)
2340 University Blvd. E.
Adelphi, MD 20783

William E. Jackson
9859 Singleton Drive
Bethesda, MD 20034

Mark T. Jacobson
2365 N. Fillmore St.
Arlington, VA 22207

David Jacobowitz
Biostatistics Lab.
Sloan-Kettering Inst.
New York, NY 10021

Jean G. Jenkins
111 E. Wacker Drive
Suite 1234
Chicago, IL 60601

Robert I. Jennrich
Dept. of Mathematics
Univ. of California
Los Angeles, CA 90024

Gordon L. Jessup
Bur. Radiological Health
Rockville, MD 20857

Pat Johns
104 Brandywine Place
Bel Air, MD 21014

David William Johnson
14 Landsend Drive
Gaithersburg, MD 20760

Douglas M. A. Johnson
Computer Research Ctr.
Univ. of South Florida
Tampa, FL 33620

Wayne Johnson
12117 Village Sq. Terr.
#101
Rockville, MD 20852

Errol W. Jones
61 Warren Hall
Cornell University
Ithaca, NY 14853

Lawrence Jones
ACS-Systems & Data
Processing
Ithaca College
Ithaca, NY 14850

Richard H. Jones
Dept. of Biometrics
Box B-119
Univ. of Colorado Med Ctr.
Denver, CO 80262

Thomas E. Jones
Westat, Inc.
11600 Nebel Street
Rockville, MD 20852

Bruce Junkins
840 Cahill Drive W.
#47
Ottawa, ON
Canada

Lawrence Kaetzel
B-260 Bldg. 226
National Bur. of Stds.
Washington, DC 20234

Roxana Kamen
310 S. Veitch Street
Arlington, VA 22204

Hiromitsu Kanemasu
4620 Southland Avenue
Alexandria, VA 22312

Leon Katz
6102 Summerhill Road
Washington, DC 20031

Linda Kaufman
Bell Laboratories
Murray Hill, NJ 07901

Charles E. Kelly
1245 Park Avenue
New York, NY 10028

William J. Kennedy
Statistical Laboratory
Iowa State University
Ames, IA 50011

Beth A. Kilss
4604 Conwell Drive
Annandale, VA 22003

Harold King
The Urban Institute
2100 M Street, NW
Washington, DC 20037

Lilliam Kingsbury
551 Saratoga Road
King of Prussia, PA
19406

Ernest J. Klotz
Owens Corning Fiberglass
Tech Center
Granville, OH 43023

Robert Kohm
Alcoa Laboratories
Alcoa Center, PA 15069

Robert Kopitske
Teledyne Water-Pik
Fort Collins, CO 80521

John Korbel
Congressional Budget Off.
Washington, DC 20515

John Koval
Dept. of Mathematics
Univ. of Western Ontario
London, ON
Canada N6B 128

James Krupp
126 South Main Street
Middlebury, VT 05753

F. Kent Kuiper
912 111th Pl., S.E.
Bellevue, WA 98004

Michael Kutner
Dept. of Biometry &
Statistics
Emory University
Atlanta, GA 30322

Michael Lackner
United Nations Stat. Off.
United Nations
New York, NY 10017

Leslie Lancaster
5812 Lamont Drive
New Carrollton, MD 20784

Lyle H. Lanier, Jr.
10243 Parkwood Drive
Kensington, MD 20795

John W. Larmer II
1841 Baldwin Drive
McLean, VA 22101

Larry L. Laster
668 Gulph Road
Wayne, PA 19087

Jennie M. Latino
4815 41st Street, NW
Washington, DC 20016

Robert L. Launer
Army Research Office
P.O. Box 12211
Research Triangle Park
NC 27709

Michael V. Lee
2301 Toddsdury Place
Reston, VA 22090

Robert G. Lehnen
6322 Linway Terrace
McLean, VA 22101

Meredith Lesly
111 3rd Avenue
New York, NY 10003

Yvonne Li
6723 Whittier Avenue
Suite 101
McLean, VA 22101

Robert F. Ling
Dept. Math. Sciences
Clemson University
Clemson, SC 29631

David Lawrence Lloyd
1302 Bayliss Drive
Alexandria, VA 22302

James Wildon Longley
8200 Cedar Street
Silver Spring, MD 20910

Gene R. Lowrimore
1007 Indian Trail
Raleigh, NC 27609

Daniel W. Lozier
A-302 Admin. Bldg.
National Bur. of Stds.
Washington, DC 20234

Frances Yu Lu
Biola College
Biola Avenue
La Mirada, CA 90639

Richard E. Lund
Dept. of Mathematics
Montana State Univ.
Bozeman, MT 59715

James A. Lutz
2337 18th Street, NW
Washington, DC 20009

Maureen P. Lynch
Bureau of the Census
FOB #3-3576
Suitland, MD 20233

Linda Lynn
Economic Research Ser.
Dept. of Agriculture
Washington, DC 20250

Paul K. Makens
Statistical Methods Div.
U.S. Bureau of Census
Suitland, MD 20233

Moe Mangad
Social Security Admin.
ORS
1875 Connecticut Ave., NW
Washington, DC 20009

Allan Marcus
Math. Department
University of Maryland
Baltimore County
Catonsville, MD 21228

O. Marrero
Dept. of Mathematics
Francis Marion College
Florence, SC 29501

Paul B. Massell
Battelle-Columbus Labs.
Suite 700
2030 M Street, NW
Washington, DC 20036

Susan E. Mattern
7798 Old Springhouse Rd.
McLean, VA 22101

Michael B. Matthews
C-5 Greenbelt Community
Carrboro, NC 27510

Victor Matthews
The Population Council
245 Park Avenue
New York, NY 10017

Jack McArdle
Academic Computing Ser.
Hofstra University
Hempstead, NY 11550

John L. McCarthy
Survey Research Center
University of California
Berkeley, CA 94720

Pat McCray
G.D. Searle & Co.
Box 1045
Skokie, IL 60076

Bruce J. McDonald
Office of Naval Research
(436)
Arlington, VA 22217

D. H. McElhone
8938 Glenbrook Road
Fairfax, VA 22030

Larry E. McFarling
714 Parkview Drive
California, MD 20619

Donald McLaughlin
American Institute
for Research
1055 Thomas Jefferson St.
Washington, DC 20007

Stanley A. McLeroy
Computer Sciences Corp.
6565 Arlington Blvd.
M.O.B.
Falls Church, VA 22046

Terry Medlin
NIH-NIDR
Bldg. 30, Room B-23
Bethesda, MD 20014

Jeff B. Meeker
428 Foulke Avenue
Ambler, PA 19002

J. J. Mellinger
Army Research Institute
1300 Wilson Boulevard
Arlington, VA 22209

Rudolph C. Mendelssohn
4106 Elizabeth Lane
Fairfax, VA 22030

M. Vijay Menon
Office of Naval Research
536 S. Clark Street
Chicago, IL 60605

James Mergerson
Department of Agriculture
SRS
Washington, DC 20250

J. Philip Miller
Washington U. Med. School
Div. of Biostatistics
700 S. Euclid Avenue
St. Louis, MO 63110

David W. Milne
Bucknell University
Lewisburg, PA 17837

Roy C. Milton
11825 Gainsborough Road
Potomac, MD 20854

George M. Minich
7015 Sea Cliff Road
McLean, VA 22101

Rita G. Minker
National Inst. of Health
Bldg. 12A Room 3051
Bethesda, MD 20014

James M. Minor
DuPont Engg. Louviers
Wilmington, DE 19711

Cleve Moler
Dept. of Mathematics
Univ. New Mexico
Albuquerque, NM 87131

Anil Monga
G.D. Searle & Co.
Box 1045
Skokie, IL 60076

John A. Moore
The Urban Institute
Suite 414
2100 M Street, NW
Washington, DC 20037

Patricia S. Moore
Bucknell University
Freas-Rooke Comp. Ctr.
Lewisburg, PA 17837

Larry R. Muenz
N.I.H.
Bethesda, MD 20014

Mervin E. Muller
5303 Mohican Road
Washington, DC 20016

Peter J. Munson
100 Bonifant Road
Silver Spring, MD 20904

Arthur Nadas
333-165-125
IBM Corp., E.F.
Hopewell Junction, NY
12533

James A. Nash
Interstate Commerce Com.
12th & Constitution Ave.
NW
Washington, DC 20423

John C. Nash
Economics Branch
Agriculture Canada
Ottawa, ON
Canada K1A 0C5

William D. Neal
3636 Carmel Road
Chamblee, GA 30341

David L. Nelson
Org. G-4530 MS 3N-17
Boeing Comp. Ser., Inc.
P.O. Box 24346
Seattle, WA 98124

Richard D. Neumyer
203 Homevale Road
Reisterstown, MD 21136

M. Marvin Newhouse
5989-D Western Run Dr.
Baltimore, MD 21209

Norman H. Nie
111 E. Wacker Drive
Suite 1234
Chicago, IL 60601

Gregory O'Connell
2043 Kirby Road
Falls Church, VA 22043

Robert K. O'Day
Dept. of Stat./Biometry
Emory University
Atlanta, GA 30322

H. Lock Oh
10829 Bocknell Drive
Silver Spring, MD 20902

Julia Dell Oliver
Dept. HEW
Public Health Service
Health Resources Admin.
Rockville, MD 20857

Anthony R. Olsen
Battelle-Northwest
P.O. Box 999
Richland, WA 99352

Terence J. Orchard
O.P.C.S. Titchfield
Fareham, Hants
England PO15 5RR

Beatrice S. Orleans
4501 Connecticut Ave.
NW
Washington, DC 20008

Carol J. Orwant
11305 Ashley Drive
Rockville, MD 20852

Marcello Pagano
1921 Edgewood Drive
Palo Alto, CA 94303

Navin Parekh
Assn. of America
Railroads Tech. Ctr.
3140 South Federal
Chicago, IL 60616

William Parker
900 Elden Street
Herndon, VA 22070

H. McIlvaine Parsons
Executive Director
Inst. for Behavioral Res.
Silver Spring, MD 20910

Chando M. Patel
556 Morris Avenue
CIBA-GEIGY Corp.
Summit, NJ 07901

Charles Pautler
4907 Russett Road
Rockville, MD 20853

Sally T. Peavy
A-337 Admin. Bldg.
National Bur. of Stds.
Washington, DC 20234

Richard A. Penhallegon
7000 Portage
The Upjohn Co.
Kalamazoo, MI 49088

Shien S. Perng
5518 Crossrail Court
Burke, VA 22015

Peter H. Peskun
Dept. of Math., York U.
4700 Keele Street
Downsview, ON
Canada m1W 2V7

Ruthann Piepenburg
300 S. Irving Road
Sterling, VA 22170

David A. Pierce
Federal Reserve Board
Washington, DC 20551

Richard A. Plattsmier
Computer Center
U. of Texas at Austin
Austin, TX 78712

Joanna V. Pomeranz
Population Council
245 Park Avenue
New York, NY 10017

Thomas W. Popham
Southern Forest Exp. Sta.
T-10210 Postal Ser. Bldg.
701 Loyola Avenue
New Orleans, LA 70113

A. Elizabeth Powell
LEAA/NCJISS
Department of Justice
Washington, DC 20531

Kevin Price
463 Cambridge Street
Apt. 405
Ottawa, ON
Canada K1S 5G3

Lloyd Provost
1640 South Stafford St.
Arlington, VA 22204

Clifford Qualls
Dept. of Math. & Stat.
U. New Mexico
Albuquerque, NM 87131

John N. Quiring
9695 South Cedar Drive
West Olive, MI 49460

Richard E. Rader
The Upjohn Company
9601-190-1
Kalamazoo, MI 49001

Lawrence Rafsky
Bell Labs
Holmdel, NJ 07733

P. Rajagopal
Dept. Comp. Sci. & Math.
Atkinson College, York U.
Downsview, ON
Canada M3J 2R7

Anthony Ralston
Dept. of Comp. Science
SUNY Buffalo
4226 Ridge Lea Rd.
Amherst, NY 14226

Mary L. Ralston
1709 Glendon
Los Angeles, CA 90004

Kunj B. Rastogi
Ohio College Lab. Ctr.
1125 Kinnear Road
Columbus, OH 43212

George A. Raub
Office of Comp. Sci.
Dept. of the Treasury
1625 I St., NW, Rm 224
Washington, DC 20220

Joy Reamy
DUALabs
1601 N. Kent Street
Suite 900
Arlington, VA 22209

Norman F. Rehner
Dept. of Math., Stat. &
Comp. Science, M.U.N.
St. John's, Newfoundland
Canada A1C 5S7

David H. Reid
2426 Arlington Blvd.
Apt. G-1
Charlottesville, VA 22903

Bruce Reinhardt
Computing Center
U. of Kentucky
McVey Hall, Rm. 72
Lexington, KY 40506

Charles DeWitt Roberts
5217--42nd Street, NW
Washington, DC 20015

June Roberts
1353 Burr Oak Road
Homewood, IL 60430

Paul L. Roney
2 Surry Ct.
Rockville, MD 20850

Joan R. Rosenblatt
A-337 Admin. Bldg.
National Bur. of Stds.
Washington, DC 20234

Murray Rosenblatt
Dept. of Mathematics
Univ. of California
San Diego
La Jolla, CA 92037

G. J. S. Ross
Rothamsted Experimental
Station
Harpenden, Herts
England AL5 2JQ

Joseph M. Rothberg
Dept. Psychiatry, WRAIR
WRAMC
Washington, DC 20012

Robert Rovinsky
Economic Res. Service
Dept. of Agriculture
Washington, DC 20250

Gail Rowan
1300 Wilson Blvd.
Arlington, VA 22209

Kenneth E. Rowe
3410 NW Roosevelt
Oregon State Univ.
Corvallis, OR 97330

Jack Rower
Economic Research Ser.
Dept. of Agriculture
Washington, DC 20250

Barbara F. Ryan
215 Pond Lab.
University Park, PA
16802

Thomas A. Ryan, Jr.
215 Pond Lab.
University Park, PA
16802

Sidney A. Sachs
Army Research Institute
1300 Wilson Boulevard
Arlington, VA 22209

Gordon Sande
Statistical Services
Statistics Canada
Ottawa, ON
Canada K1A 0T6

S. Sankaran
1663 Clearview Road
Norristown, PA 19403

Janet E. Sargent
22 Moran Drive
Waldorf, MD 20601

Margaret H. Sarner
E.I. duPont de Nemours
& Co., L31E87
Wilmington, DE 19898

John Sau
P.O. Box 10066
Raleigh, NC 27605

Janice Schaefer
1401 K Street, NW
Washington, DC 20230

Sarah E. Schlesselman
11041 Seven Hill Lane
Potomac, MD 20854

Kurt J. Schmucker
3571 Ft. Meade Road
Apt. 519
Laurel, MD 20810

Jack F. Schreckengost
Bur. of Veterinary Med.
HFV-105
5600 Fishers Lane
Rockville, MD 20857

Ronald A. Schwartz
Arnar-Stone Labs., Inc.
601 E. Kensington Avenue
Mount Prospect, IL 60056

Robert J. Sclabassi
Biomed. Eng. Program
Carnegie-Mellon University
Pittsburgh, PA 15213

Stuart Scott
Bur. of Labor Statistics
441 G St., NW
Room 2146
Washington, DC 20212

S. R. Searle
Biometrics Unit
Cornell University
Ithaca, NY 14853

Jeanne L. Sebaugh
P.O. Box 120
Chapman, KS 67431

Murray R. Selwyn
5485 Greathead Court
Columbia, MD 21045

Rena Shampton
Nationwide Insurance
246 N. High Street
Columbus, OH 43216

Eric J. Shangold
Bur. of Radiological Hlth.
HFX 21
5600 Fishers Lane
Rockville, MD 20857

Eduardo N. Siguel
Natl. Inst. on Drug Abuse
11400 Rockville Pike
Rockville, MD 20852

A. Simanis
Canadian Armed Forces
Ottawa, ON
Canada

Anthony P. Simkus
Army Research Office
P.O. Box 12211
Research Triangle Park
NC 27709

David R. Slaby
Room 8-37, NCHS
5600 Fishers Lane
Rockville, MD 20857

Bradford Smith
55 Wheeler Drive
Cambridge, MA 02138

Paul N. Somerville
Dept. of Math & Stat.
Florida Tech. Univ.
P.O. Box 25000
Orlando, FL 32765

Richard A. Soucy
Bur. of Labor Stat.
GAO Bldg. Rm. 2146
Washington, DC 20212

Randall K. Spoeri
Center for Census Use
Studies, Rm 3077-3
Bur. of the Census
Washington, DC 20233

Selig Starr
Brookings SSCC
1775 Mass. Ave., NW
Washington, DC 20036

Leonard Steinberg
11828 Smoketree Road
Rockville, MD 20854

Peter B. Stevens
9412 Holbrook Lane
Potomac, MD 20854

G. W. Stewart
Dept. of Comp. Sciences
Univ. of Maryland
College Park, MD 20740

William R. Stewart, Jr.
University of Maryland
College of Business
College Park, MD 20742

Victor Stotland
Off. of Systems & Stds.
Dept. of Labor
Bur. of Labor Statistics
Washington, DC 20212

Jeanne C. Stringfellow
4626 Conwell Drive
Annandale, VA 22003

Cynthia Struthers
7-414 Hazel Street
Waterloo, ON
Canada N2L 3P8

Robert Stuckart
Bur. Radiological Hlth.
HFX-220
5600 Fishers Lane
Rockville, MD 20857

James P. Summe
Biometrics Division
Stop 23, NMRI, NNMC
Bethesda, MD 20014

Richard W. Swartz
9681 Muirkirk Road
Apt. #B62
Laurel, MD 20811

Kathryn A. Szabat
J526 3901 Locust Walk
Philadelphia, PA 19174

Alan J. Talbert
NIH/NINCDS/OBE
7550 Wisconsin Avenue
Room 7C05
Bethesda, MD 20014

Kunio Tanabe
Dept. of Mathematics
North Carolina State U.
Raleigh, NC 27607

Richard A. Tapia
5723 Partal Drive
Houston, TX 77096

Stephen B. Taubman
Federal Reserve System
Washington, DC 20551

Walter L. Taylor
10704 Phillips Drive
Upper Marlboro, MD 20870

Peeter Teedla
Dept. of Epidemiology
600 W 168th Street
New York, NY 10032

Robert F. Teitel
The Urban Institute
2100 M Street, NW
Washington, DC 20037

D. G. Thomas
NCI, Landow C318
Bethesda, MD 20014

Jerry Thomas
12807 Pt. Pleasant Dr.
Fairfax, VA 22030

Carol B. Thompson
1 Strawberry Court
Clifton Park, NY 12065

James R. Thompson
Dept. of Math. Sciences
Rice University
Houston, TX 77001

Edward J. Timko
3374 Whipple Court
Annandale, VA 22003

Marcia Tolbert
International Futility Program
Research Triangle Park
NC 27709

Lowell H. Tomlinson
1944 Ravenwood Drive
Bethlehem, PA 18018

Jerome D. Toporek
U. of Rochester
Computing Center
727 Elmwood Avenue
Rochester, NY 14620

Marietta Tretter
609 BAB
Penn State Univ.
University Park
PA 16802

Peter V. Tryon
2645 Table Mesa Ct.
Boulder, CO 80303

Chris P. Tsokos
Dept. of Mathematics
Univ. of South Florida
Tampa, FL 33620

C. C. Tu
Bur. of the Census
ISPC
Washington, DC 20233

Joseph Tu
Brookings SSCC
1775 Mass. Ave., NW
Washington, DC 20036

Gary W. Tubb
College of Education
Northwestern State U.
Natchitoches, LA 71457

Sarah Tung
974 Alexandria Drive
Newark, DE 19711

Richard J. Vance
644 John M.
Clawson, MI 48017

William K. Van Hassel
35 New Street
New Hope, PA 18938

John C. Vardy
Syntex Laboratories
3401 Hillview Ave.
Palo Alto, CA 94304

Paul F. Velleman
NY State School of
Industrial & Labor Rel.
356 Ives Hall
Ithaca, NY 14853

Mrs. Raji Vijayraghuan
Ayerst Laboratories
Biostatistics Dept.
685 Third Avenue
New York, NY 10017

C. Wall
U. of Toronto, P.M.&B.
121 St. Joseph Street
Toronto, ON
Canada M5S 2R9

Peter Walsall
Dept. of Biostatistics
Loma Linda University
Loma Linda, CA 92354

Roy H. Wampler
A-337 Admin. Bldg.
National Bur. of Stds.
Washington, DC 20234

Roger Warburton
Univ. of PA.
4744 Larchwood Avenue
Philadelphia, PA 19143

Kenneth R. Waugh
1605 Woodmoor Lane
McLean, VA 22101

William D. Weal
3636 Carmel Road
Chamblee, GA 30341

Richard H. Weaver
Farmland Industries, Inc.
P.O. Box 7305
Kansas City, MO 64116

Arnold L. Weber
Dept. of HEW
Office of the Secretary
Washington, DC 20201

Pamela Weeks
Off. of Systems & Stds.
Dept. of Labor
Bur. of Labor Statistics
Washington, DC 20212

Ray Weingardt
3 Beacon Crescent
St. Albert, Alberta
Canada

Maxine Weinstein
18 Ninth Street, NE
#402
Washington, DC 20002

Roy E. Welsch
50 Memorial Drive
E53-383
Cambridge, MA 02139

Richard A. Wenk
362 Malcolm Avenue
No. Plainfield, NJ 07063

Bernard P. Wess
UMBC Computer Center
5401 Wilkens Avenue
Baltimore, MD 21228

William H. Wetterstrand
Dept. of Math. Sciences
Ball State University
Muncie, IN 47306

James Wheaton
Dept. of Agriculture
SRS
Washington, DC 20250

Kenneth J. White
Dept. of Economics
Rice University
Houston, TX 77001

Mary White
Soc. Sec. Admin.
Metal East Bldg., 3G1
Baltimore, MD 21235

Robert L. White
2619 Lackawanna St.
Adelphi, MD 20783

David E. Whiteman
2506 B 35th Street
Los Alamos, NM 87544

Gary R. Whittle
1807 Walnut Avenue
Baltimore, MD 21222

Clark Wiedmann
Univ. Computing Ctr.
Graduate Research Ctr.
U. of Massachusetts
Amherst, MA 01002

Christopher Wild
Apt. 710
159 University Ave. W
Waterloo, ON
Canada N2L 3E8

A. Martin Wildberger
15811 Pinecroft Lane
Bowie, MD 20716

Graham N. Wilkinson
Bell Laboratories
600 Mountain Avenue
Murray Hill, NJ 07974

G. Williams-Leir
Bldg. M-59
Montreal Road
Ottawa, ON
Canada K1A OR6

Jean F. Williams
Dept. of HEW
Public Health Service
Health Resources Admin.
Rockville, MD 20857

Barbara B. Wolfe
Comp. & Data Proc. Ctr.
Wayne State University
Detroit, MI 48202

Agatha Wolman
6104 Yorkshire Terrace
Bethesda, MD 20014

William Wolman
Federal Highway
Dept. of Transportation
Washington, DC 20590

Yee Wong
Geomet, Inc.
15 Firstfield Road
Gaithersburg, MD 20760

Margaret H. Wright
Operations Research Dept.
Stanford University
Stanford, CA 94305

Robert K. Wright, Jr.
Veterans Admin. Hospital
151 K
Hines, IL 60141

Ronald E. Wyllys
2603 Rogge Lane
Austin, TX 78723

Fred S. Yamada
Rm. 3055, Bldg. 12A
Div. of Comp. Res. &
Technology, NIH
Bethesda, MD 20014

Ervin H. Young
IRSS
Manning Hall 026A
UNC-CH
Chapel Hill, NC 27514

Susan B. Young
U.S.N.R.C.
Washington, DC 20555

H. P. Yule
NUS Corporation
4 Research Pl.
Rockville, MD 20850

James Zum Brunnen
Dept. of Statistics
Colorado State Univ.
Ft. Collins, CO 80523

NBS-114A (REV. 7-73)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET	1. PUBLICATION OR REPORT NO. NBS SP-503	2. Gov't Accession No.	3. Recipient's Accession No.

4. TITLE AND SUBTITLE

SP-503, Computer Science and Statistics: Tenth Annual Symposium on the Interface

5. Publication Date March 1978

6. Performing Organization Code

7. AUTHOR(S) David Hogben and Dennis W. Fife

8. Performing Organ. Report No.

9. PERFORMING ORGANIZATION NAME AND ADDRESS

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20234

10. Project/Task/Work Unit No.

11. Contract/Grant No. MCS77-04441
NR 042-000
ARO 14862-M

12. Sponsoring Organization Name and Complete Address (Street, City, State, ZIP)

National Science Foundation,
Office of Naval Research, and
U.S. Army Research Office
Washington, D.C.

13. Type of Report & Period Covered FINAL

14. Sponsoring Agency Code

15. SUPPLEMENTARY NOTES

16. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)*

The Proceedings of Computer Science and Statistics: Tenth Annual Symposium on the Interface contains 36 invited and 36 contributed poster session papers. The invited papers were presented in six workshops on Evaluation of Statistical Software, Nonlinear Models, Graphics, Large Data Files, Numerical Analysis in Statistics, and Maintenance and Distribution of Statistical Software. The Evaluation of Statistical Software Workshop was divided into two sessions on Statistical Program Packages for Small Computers and Computing Approaches to the Analysis of Variance for Unbalanced Data.

17. KEY WORDS *(six to twelve entries; alphabetical order; capitalize only the first letter of the first key word unless a proper name; separated by semicolons)*

Analysis of variance; computer science; evaluation; graphics; large data files; maintenance and distribution; nonlinear models; numerical analysis; small computers; software; statistical program packages; statistics.

18. AVAILABILITY

- ☐ Unlimited
- ☐ For Official Distribution. Do Not Release to NTIS
- ☐ Order From Sup. of Doc., U.S. Government Printing Office Washington, D.C. 20402, SD Cat. No. C13.10:503
- ☐ Order From National Technical Information Service (NTIS) Springfield, Virginia 22151

19. SECURITY CLASS (THIS REPORT) UNCLASSIFIED

21. NO. OF PAGES 467

20. SECURITY CLASS (THIS PAGE) UNCLASSIFIED

22. Price $6.25

☆ U. S. GOVERNMENT PRINTING OFFICE : 1978 261-238/60

USCOMM-DC 29042-P74

NBS TECHNICAL PUBLICATIONS

PERIODICALS

JOURNAL OF RESEARCH—The Journal of Research of the National Bureau of Standards reports NBS research and development in those disciplines of the physical and engineering sciences in which the Bureau is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology, and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Bureau's technical and scientific programs. As a special service to subscribers each issue contains complete citations to all recent NBS publications in NBS and non-NBS media. Issued six times a year. Annual subscription: domestic $17.00; foreign $21.25. Single copy, $3.00 domestic; $3.75 foreign.

Note: The Journal was formerly published in two sections: Section A "Physics and Chemistry" and Section B "Mathematical Sciences."

DIMENSIONS/NBS

This monthly magazine is published to inform scientists, engineers, businessmen, industry, teachers, students, and consumers of the latest advances in science and technology, with primary emphasis on the work at NBS. The magazine highlights and reviews such issues as energy research, fire protection, building technology, metric conversion, pollution abatement, health and safety, and consumer product performance. In addition, it reports the results of Bureau programs in measurement standards and techniques, properties of matter and materials, engineering standards and services, instrumentation, and automatic data processing.

Annual subscription: Domestic, $12.50; Foreign $15.65.

NONPERIODICALS

Monographs—Major contributions to the technical literature on various subjects related to the Bureau's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NBS, NBS annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

Applied Mathematics Series—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a world-wide program coordinated by NBS. Program under authority of National Standard Data Act (Public Law 90-396).

NOTE: At present the principal publication outlet for these data is the Journal of Physical and Chemical Reference Data (JPCRD) published quarterly for NBS by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements available from ACS, 1155 Sixteenth St. N.W., Wash., D.C. 20056.

Building Science Series—Disseminates technical information developed at the Bureau on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NBS under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The purpose of the standards is to establish nationally recognized requirements for products, and to provide all concerned interests with a basis for common understanding of the characteristics of the products. NBS administers this program as a supplement to the activities of the private sector standardizing organizations.

Consumer Information Series—Practical information, based on NBS research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

Order **above** *NBS publications from: Superintendent of Documents, Government Printing Office, Washington, D.C. 20402.*

Order **following** *NBS publications—NBSIR's and FIPS from the National Technical Information Services, Springfield, Va. 22161.*

Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. Register serves as the official source of information in the Federal Government regarding standards issued by NBS pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NBS Interagency Reports (NBSIR)—A special series of interim or final reports on work performed by NBS for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Services (Springfield, Va. 22161) in paper copy or microfiche form.

BIBLIOGRAPHIC SUBSCRIPTION SERVICES

The following current-awareness and literature-survey bibliographies are issued periodically by the Bureau:

Cryogenic Data Center Current Awareness Service. A literature survey issued biweekly. Annual subscription: Domestic, $25.00; Foreign, $30.00.

Liquified Natural Gas. A literature survey issued quarterly. Annual subscription: $20.00.

Superconducting Devices and Materials. A literature survey issued quarterly. Annual subscription: $30.00. Send subscription orders and remittances for the preceding bibliographic services to National Bureau of Standards, Cryogenic Data Center (275.02) Boulder, Colorado 80302.

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Washington, D.C. 20234

OFFICIAL BUSINESS

Penalty for Private Use, $300

POSTAGE AND FEES PAID
U.S. DEPARTMENT OF COMMERCE
COM-215

SPECIAL FOURTH-CLASS RATE
BOOK