

APR 7 1967



OAK RIDGE NATIONAL LABORATORY
operated by
UNION CARBIDE CORPORATION
NUCLEAR DIVISION
for the
U.S. ATOMIC ENERGY COMMISSION



ORNL-TM-1798

COPY NO. -

116

DATE - March 10, 1967

CESTI PRICES

Multiple Factor Analysis

R. C. Durfee

HQ # 3.00 MN 65

Abstract

The purpose of factor analysis is to identify the dimensions or factors by which a variety of phenomena within a domain are related. Factorial methods were originally developed for the purpose of identifying the principal dimensions or categories of mentality; but the methods are general, so that they have been found useful for other psychological problems and in other sciences as well. Factor analysis is not restricted by assumptions regarding the nature of the factors, whether they be physiological or social, elemental or complex, correlated or uncorrelated.

Thus, three programs have been written which factor analyze a set of measurements in order that the underlying factors which relate the measurements may be identified. These programs are based upon work done by Professor Edward E. Cureton at the University of Tennessee.

THIS DOCUMENT HAS BEEN REVIEWED,
NO INVENTIONS OF PATENT INTEREST
TO THE A.E.C. ARE DISCLOSED THEREIN,
D83 4/10/67

NOTICE This document contains information of a preliminary nature and was prepared primarily for internal use at the Oak Ridge National Laboratory. It is subject to revision or correction and therefore does not represent a final report.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

—
LEGAL NOTICE
—

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

- A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or
- B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

INTRODUCTION

When a particular range of phenomena is to be investigated by means of individual differences, one can proceed in two ways. One can invent a hypothesis regarding the processes that underlie the individual differences, and then set up a laboratory experiment to test the hypothesis. If no promising hypothesis is available, one can represent the domain as adequately as possible in terms of a set of measurements or numerical indices and proceed with a factorial experiment. The factor analysis might reveal an underlying order which would be of great assistance in formulating scientific concepts covering the particular domain. Thus, three computer programs have been written which factor analyze such a set of measurements or indices in order that the underlying factors which relate the measurements may be identified.

Let us consider such a set of measurements to be arranged in a score matrix, S (n by N), where element $s_{i,j}$ would be the score of individual j on test i . Then the observation equation which is the starting point for multiple factor theory would be:

$$\begin{matrix} & N \\ \begin{matrix} n \\ \boxed{s_{i,j}} \end{matrix} & \end{matrix} = \begin{matrix} & q \\ \begin{matrix} n \\ \boxed{a_{i,k}} \end{matrix} & \end{matrix} \times \begin{matrix} & N \\ \begin{matrix} q \\ \boxed{p_{k,j}} \end{matrix} & \end{matrix} \quad (1)$$

S A P

Score Matrix Factor Matrix Population Matrix

where N is the number of individuals taking n tests. This equation expresses the assumption that the scores, $s_{i,j}$, may be accounted for in terms of a smaller number of factors, q , than the number of tests, n . The psychological interpretation of equation (1) is that a subject's performance on a test is determined in part by the abilities that are called for by the test and in part by the degrees to which the subject possesses these abilities. Thus, the factor matrix, A (n by q), describes the abilities which are called for by each test while the population matrix, P (q by N), describes the degree to which the subject possesses each of these abilities. Therefore, the purpose of these three computer programs is to find the q abilities by calculating A , and then rotate A to obtain a simple structure matrix, S , which will give meaning to these abilities.

Actually the factor matrix given in equation (1) is made up of two types of factors: common factors and specific factors. The common factors represent abilities that are involved in two or more tests of a battery. The specific factors represent an ability which is involved in only a single test of a battery. Thus, for every test there is a specific factor which has a factor loading only on that particular test. Since we are not interested in these specific factors, they are removed from the factor matrix and we will no longer consider them as part of the factor matrix, A .

The problem of finding and identifying these q factors, which may be correlated, is broken into two parts: the initial factoring problem and the rotational problem. The initial factoring problem involves the calculation of the orthogonal factor matrix, A (n by q), whose factors

are uncorrelated. This is done in the first computer program, FACTOR ANALYSIS I. The rotational problem then involves an orthogonal rotation followed by an oblique rotation of this factor matrix, A, to obtain the desired simple structure matrix, S, as shown by equation (2). The factors of S are now correlated.

$$\begin{array}{ccc}
 & q & \\
 n & \boxed{s_{i,m}} & = & n & \boxed{a_{i,k}} & \times & q & \boxed{t_{k,m}} & (2) \\
 & S & & A & & & T &
 \end{array}$$

Oblique Simple Structure Matrix = Orthogonal Factor Matrix \times Transformation Matrix

The rotational problem is solved by the last two computer programs, FACTOR ANALYSIS II and FACTOR ANALYSIS III.

The descriptions which follow in this report are only intended to give the user an idea of what each program does, and they do not represent a rigorous description of multiple factor analysis.

GENERAL DESCRIPTION OF THE FACTORIAL METHODS USED

Factor Analysis I

The purpose of this program is to compute a reduced principal axis factor matrix, A (n by q), from the reduced correlation matrix, R (n by n), using the principal axis method.³ Other methods such as the centroid method have been used to calculate factor matrices, but this method seems to be the best for the majority of cases. The reduced correlation matrix, R, is composed of the intercorrelations of the score matrix with the unit diagonal replaced by the communalities of the factor matrix, A. The equation which relates the reduced factor matrix with the reduced correlation matrix is:

$$R = A \cdot A^T + \Delta \quad (3)$$

Thus, the initial factoring problem consists of factoring R in such a manner as to find the number of factors, q, of A, to find the elements of A, and to minimize Δ .

Since the factor matrix communalities, h_i^2 :

$$h_i^2 = \sum_{k=1}^q A_{i,k}^2 \quad (4)$$

are not known at the outset, we insert communality estimates, \hat{h}_i^2 , into the diagonal of R and proceed to calculate an approximation, A_1 , to the factor matrix, A. By using the communalities of A_1 as better estimates, we are then able to repeat the procedure and calculate a better approximation, A_2 , to the factor matrix, A. This iterative procedure is repeated until the communalities stabilize, thus indicating that we have calculated A. The communalities converge more rapidly for large values of n.

It has been shown by Hotelling⁴ that the elements, $a_{i,k}$, of the factor matrix, A, are the elements of the eigenvectors of R associated with the q largest positive eigenvalues, each eigenvector being so scaled that the sum of squares of its elements is equal to the associated eigenvalue. In factor analysis this procedure is termed the principal axis method, because the scaled eigenvectors are the principal axes of ellipsoids of equal density in an n-space geometrical representation of the score matrix.

In actual practice, the determination of the number of significant factors, q, is something of an art. Usually this can be found by examining the eigenvalues. Thus for one set of actual data ($N = 92$; $n = 42$) the eigenvalues were as follows:

Table 1. Eigenvalues and Eigenvalue Differences for a Sample Case

Factor No.	1	2	3	4	5	6	7	8	9	10	11
Eigenvalue	13.01	5.00	2.60	1.68	1.43	1.30	.97	.82	.70	.68	.61
Difference	8.01	2.40	.92	.25	.13	.33	.15	.12	.02	.07	

The eigenvalue differences decrease progressively to factor 6, there is a larger drop from 6 to 7, and the later differences are all small, so we conclude that there are six significant factors. The eigenvalue for the last salient factor is usually somewhere between 1.5 and .5. In addition to examining the eigenvalues, the Bargmann¹ modification of Bartlett's test for the number of significant factors² has been included in the program to help determine q. The relative magnitudes of the factor loadings, $a_{i,k}$, may also be of help in determining q.

When the initial factoring procedure is finished, we have a final unrotated factor matrix, A (n by q), and good reason to suppose that the $(q + 1)$ -th factor is either statistically insignificant or unreliable.

This first computer program has been divided into three sections as can be seen from the FORTRAN listing in Appendix B. The first section computes the correlation matrix from a raw score matrix if the correlation matrix has not been previously calculated. The second section computes the principal axis factor matrix with an excess of factors. The third section uses a Bargmann test to help the user in determining the number of significant factors in the factor matrix.

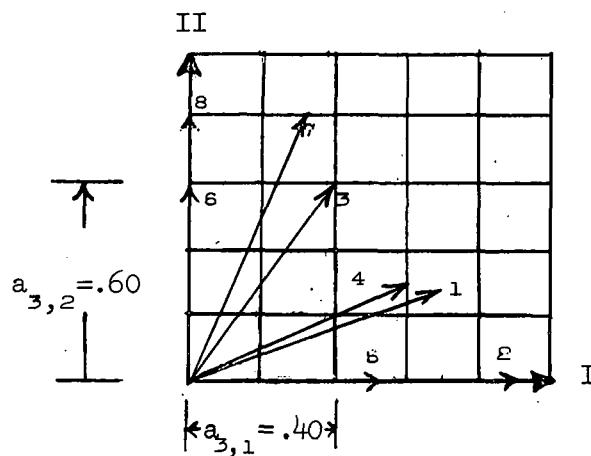
Factor Analysis II

The main purpose of this program is to compute a rotated varimax matrix, V (n by q), by means of an orthogonal normal varimax rotation of the principal axis factor matrix, A (n by q).⁶ In addition, the program also performs several elementary oblique simple structure rotations of the principal axis factor matrix, using the varimax transformation matrix, T (q by q), as an initial trial transformation matrix. The method used in performing the simple structure rotations is an adaptation of the method described by Thurstone.⁸

It is helpful to think of the principal axis factor matrix, A , in terms of a geometrical model. If we let each factor of A represent one arbitrary reference vector in a q dimensional common-factor space, then each of the n tests represents a vector in this space with q components or coordinates. For an exemplary factor matrix with two factors and eight tests the geometrical model would look like:

Factor matrix A

No.	I	II
1.	.70	.30
2.	.90	.00
3.	.40	.60
4.	.60	.30
5.	.50	.00
6.	.00	.60
7.	.30	.80
8.	.00	.80

Fig. 1. Geometrical Representation of the Two-dimensional Factor Matrix, A .

where I and II represent the two orthogonal reference vectors, and the components of each of the eight test vectors are the factor loadings,

$a_{i,k}$, in the factor matrix, A. Note that the scalar product of test vector \underline{i} with test vector \underline{j} is the intercorrelation, $r_{i,j}$, between the two tests. Thus the length of each test vector is h_i , the square root of its communality.

Since the factor matrix is represented with an arbitrary reference frame we now wish to rotate this frame into a preferred or simplifying position. This orthogonal rotation of A is called the normal varimax rotation. The new rotated varimax matrix, V (n by q), is related to the unrotated factor matrix, A (n by q), by the linear transformation matrix, T (q by q), as shown below:

$$\begin{matrix} & q \\ n & \boxed{v_{i,m}} \end{matrix} = \begin{matrix} & q \\ n & \boxed{a_{i,k}} \end{matrix} \times \begin{matrix} & q \\ & \boxed{t_{k,m}} \end{matrix} \quad (5)$$

V A T

Rotated Varimax Matrix \equiv Unrotated Factor Matrix \times Transformation Matrix

The varimax criterion for performing this orthogonal rotation requires that

$$\sum_{k=1}^q \left\{ n \sum_{i=1}^n (a_{i,k}^2 / h_i^2)^2 - (\sum_{i=1}^n (a_{i,k}^2 / h_i^2))^2 \right\} \quad (6)$$

be a maximum, where $a_{i,k}$ is the factor loading for the i th test on the k th factor and h_i^2 is the communality for the i th test. The varimax rotation, in addition to simplifying the representation of the test

vectors, also gives us an initial trial transformation matrix, T, which is needed in simple structure rotations of the factor matrix.

This second computer program, like the first, has been divided into three sections. The first section computes the rotated varimax matrix from the factor matrix as described previously. The second section consists of a single oblique simple structure rotation of the principal axis factor matrix using the varimax transformation matrix as a trial transformation matrix. This section requires that the variables to be used in the rotation along with their respective weights be specified. These quantities are specified by subjective decisions made within the computer. The third section consists of five simple structure rotations using the most recently calculated transformation matrix as a trial matrix. The computer again makes subjective decisions in determining the variables to be used in each rotation, although no weights are used in this section. Although these subjective decisions made by the computer are not optimal, they may yield satisfactory results in some cases. In most cases the user will need to augment these results by using the third program. The discussion of simple structure rotation is given in the last part of this report.

Factor Analysis III

The purpose of this program is to compute an oblique simple structure matrix, S (n by q), and a simple structure transformation matrix, T (q by q), by means of an oblique simple structure rotation of the principal axis factor matrix, A (n by q). As was stated previously, the method is an adaptation of the method described by Thurstone.⁸ When the correct fit has been obtained, this simple structure matrix, S , will identify and give meaning to our q factors which we set out to find. Our oblique simple structure matrix, S , is related to the orthogonal factor matrix, A , by the transformation matrix, T , as shown previously in equation (2). The factors of S will now be correlated (oblique) whereas the factors of A were uncorrelated (orthogonal).

In order to understand the simple structure process, let us again consider the orthogonal factor matrix, A , in terms of its geometrical model in the common-factor space. If we let A consist of three factors and eleven tests, our geometrical model would look like:

	I	II	III
1.	.20	.20	.90
2.	.70	.10	.60
3.	.20	.80	.40
4.	.30	.80	.30
5.	.40	.70	.20
6.	.20	.60	.30
7.	.80	.40	.10
8.	.10	.20	.80
9.	.10	.30	.90
10.	.60	.10	.60
11.	.70	.20	.70

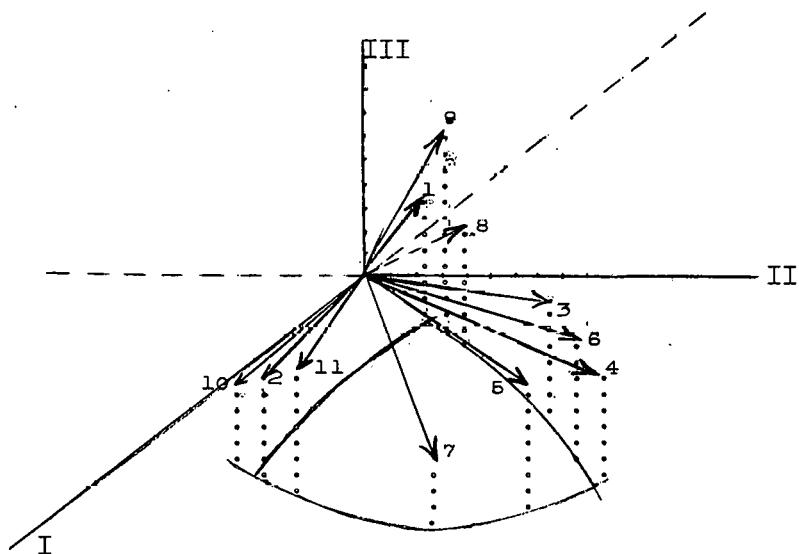


Fig. 2. Geometrical Representation of the Three-dimensional Factor Matrix, A .

where I, II, III represent the three orthogonal reference vectors. The components of the eleven test vectors along each reference vector are just their respective factor loadings, $a_{i,k}$, on that factor. Three dimensional factor matrices can be represented as shown in Figure 2, but it has been found more convenient to represent them on the surface of a sphere as shown below:

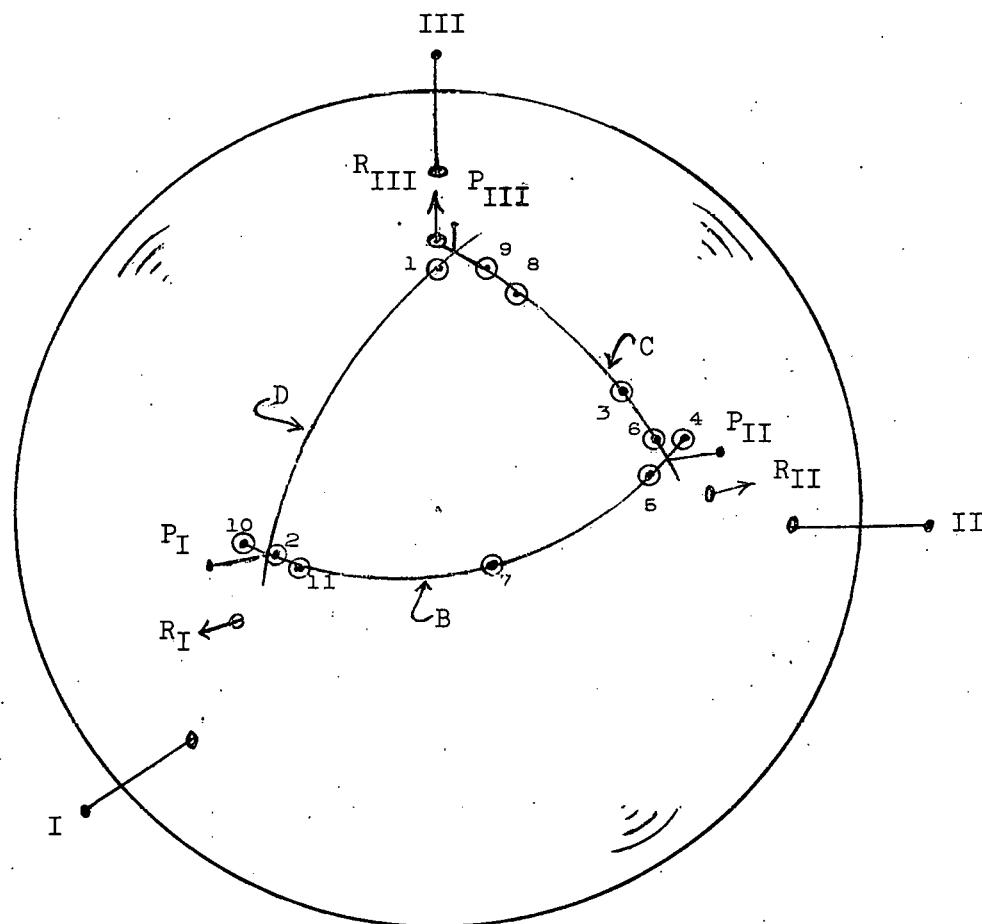


Fig. 3. Spherical Representation of the Three-dimensional Factor Matrix, A.

where the intersection of each test vector with the surface is shown by a dot (with a circle around it) on the surface. Actually it is necessary

to extend the test vectors to unit length for study on a sphere of unit radius. Thus the factor matrix, A, is normalized by dividing each row by h_i , the square root of its communality.

We are now ready to preform the simple structure rotation geometrically. Instead of using the orthogonal reference frame I, II, III which was obtained by solving the initial factoring problem, we now draw a new reference frame as shown by the solid lines B, C, D in figure 3. These solid lines should form the boundary of a spherical triangle such that most of the test vectors terminate in the sides of the triangle. By visualizing a transparent sphere, we should be able to see three bounding hyperplanes which intersect the sphere at the lines B, C, D and then most of the test vectors will lie in or near one of the three hyperplanes. The new reference vectors R_I , R_{II} , R_{III} are the normals to the three bounding hyperplanes and the primary factors P_I , P_{II} , P_{III} are determined by the intersections of the three hyperplanes. Each primary axis represents an "ideal test" which would measure one pure factor. If the new reference vectors were orthogonal, they would coincide with the primary vectors. This technique of fitting a new oblique reference frame to subsets of the test vector termini such that the subsets lie in or near one of the bounding hyperplanes is called oblique simple structure rotation. It is a matter of empirical observation that in large classes of data, the hyperplanes make acute angles with one another, and the reference vectors obtuse angles. If the structure is not acute, it is sometimes difficult to tell whether convergence is toward a bounding hyperplane or toward a hyperplane which cuts across the common-factor space and is therefore not a true factor.

Just as the components of the test vectors along I, II, III made up the factor loadings of the orthogonal factor matrix, A, the components of the test vectors along R_I , R_{II} , R_{III} make up the factor loadings of the oblique factor matrix, S. The coordinates of the new reference frame R_I , R_{II} , R_{III} on the old reference frame I, II, III make up the transformation matrix, T, which transforms A into S. The scalar product of the column vectors of T give the cosines of the angles between the new reference vectors.

In order to (1) find the q bounding hyperplanes, (2) fit them to the subsets of test vectors, and (3) arrive at the transformation matrix, T, it is necessary that three quantities be specified. The first quantity is a trial set of reference vectors, T, which approximate the final reference vectors, R_I , R_{II} , R_{III} . This trial transformation matrix will initially come from Factor Analysis II. The second quantity to be specified will be q sets of remaining test vectors which lie in or near the hyperplane associated with each of the q factors. For a given factor, the remaining test vectors will have near zero factor loadings because they will be almost perpendicular to the new reference vector for that factor. The third quantity which may or may not be specified is a set of weights for each set of the remaining test vectors. These weights tend to move the hyperplane towards the respective test vectors. Higher weights will have a greater effect in moving the hyperplanes.

In actual practice it takes several rotations to determine which direction to go for the best fit and finally produce a satisfactory factor matrix, S, in which the factor loadings of the remaining test vectors are near zero ($\pm .10$). This is why the second program attempts to determine

the three quantities previously mentioned and use them in calculating several preliminary simple structure fits. Perhaps these preliminary fits will give the user a better idea of which direction to go. If the user is unfamiliar with simple structure rotation he will probably be unable to use this third program.

ACKNOWLEDGMENT

The author wishes to express his thanks to Professor Edward E. Cureton whose research at the University of Tennessee forms the basis for these three programs.

Appendix A

INPUT PREPARATION, SAMPLE PROBLEM, AND OUTPUT

General - These three computer programs have been written so that they may be used with the FORTRAN 63 compiler written at the Oak Ridge National Laboratory for the IBM 360 computer or with any standard FORTRAN IV compiler which assumes either a four character word (IBM 360) or a six character word (IBM 7040, 7090).

When using these programs with a six character word machine, the user should find all comment statements in the input/output subroutines which have a "dollar sign" in column 6. The "c" in column 1 and the "dollar sign" in column 6 should then be removed from these cards. The new cards should then replace the card which occurs just before them in the source program. Only those cards associated with DATA statements, FORMAT statements, or other character information will be involved.

The user should realize that the numerical results given in this report were computed with an IBM 360 computer and will not agree exactly with those computed on a six character word machine due to the difference in word size.

The first card of output punched from each of the three programs will be a title card of asterisks and other Hollerith characters. This title card should be removed before the cards are later used as input.

Factor Analysis I

Input - The input variables read by FACTOR ANALYSIS I are described in Table A.1. The sequence in which these variables are read and the format by which they are read is given in Table A.2.

Table A.1 Factor Analysis I Input Variable Description

Input Variable	Description
1. TITLE	Alphanumeric title identification
2. NVAR	Number of variables
3. NOBS	Number of observations
4. NOEXT	Number of factors to be extracted from the correlation matrix
5. FMTX	Input FORMAT statement under which raw score matrix is read
6. FMTR	Input FORMAT statement under which correlation matrix is read
7. SCORE	Raw score matrix
8. R	Correlation matrix
9. FMTC	Input FORMAT statement under which communalities are read
10. NFCOMM	Number of factors in factor matrix (from which communalities are calculated)
11. COMM	Communalities of factor matrix
12. FMAT	Factor matrix
13.	1 to read in a raw score matrix (NOBS x NVAR) - read by rows SSL { 2 to read in a correlation matrix (NVAR x NVAR) - read by upper diagonal rows
14.	1 to read no communalities but use computed SMC values instead SS2 { 2 to read communalities themselves - FMTC read and used 3 to read a factor matrix from which communalities are computed

Table A.2 Factor Analysis I Input Variable Sequence and Format

Input Variables	Card Format
1. TITLE	- FORMAT(13A6,A2)
2. SS1,SS2	- FORMAT(2I2)
3. NVAR,NOBS,NOEXT	- FORMAT(3I4)
4. FMTX (If SS1=1) or FMTR (If SS1=2)	- FORMAT(18A1)
5. SCORE (If SS1=1) or R (If SS1=2)	- FMTX or FMTR
6. FMTC (If SS2=2) or NFCOMM (If SS2=3)	- FORMAT(18A4) or FORMAT(I4)
7. COMM (If SS2=2) or FMAT (If SS2=3)	- FMTC or FORMAT(8X,6F12.8)

Output - The output variables printed by FACTOR ANALYSIS I are described in Table A.3. The sequence in which these variables are printed is given in Table A.4. The printing of some of the variables is controlled by the two sense switch variables, SS1 and SS2, as indicated in Table A.4.

Table A.3 Factor Analysis I Output Variable Description

Output Variable	Description
1. TITLE	Alphanumeric title identification
2. NVAR	Number of variables
3. NOBS	Number of observations
4. AVG	Mean for each variable
5. SIGMA	Standard deviation for each variable
6. R	Correlation matrix
7. DETR	Determinant of correlation matrix
8. SMC	Squared multiple correlations
9. TRACE	Sum of diagonal elements of correlation matrix
10. EIGVAL	Eigenvalues of correlation matrix
11. EVDIFF	Difference between consecutive eigenvalues
12. EVSUM	Sum of eigenvalues
13. PERCT	Percentage of TRACE factored by each successive factor
14. CHISQ	Chi squared values
15. DEV	Deviations (used in Bargmann test for significant factors)
16. FMAT	Factor matrix (will also be punched out for later use)
17. NOSIG	Number of significant factors computed from Bargmann test
18. COMM	Factor matrix communalities using significant factors

Table A.4 Factor Analysis I Output Variable Sequence and Sense Switch Control

Sequence	Output Variable	Sense Switch Control
1.	TITLE	
2.	NVAR,NOBS	
3.	AVG,SIGMA	If SS1 = 1
4.	R	
5.	DETR	
6.	SMC	If SS2 = 1
7.	TRACE	
8.	EIGVAL,EVDIFF,EVSUM,PERCT	
9.	CHISQ,DEV	
10.	FMAT	
11.	NOSIG,COMM	

Sample Problem - A sample problem has been used which is based upon a set of nine tests given to 504 individuals. These tests were designed to measure intelligence factors. The input to Factor Analysis I is given in Table A.5. The correlation matrix had been previously computed so we skipped the first section of the program and read in the correlation matrix. Since we had no communality estimates, computed squared multiple correlations were used for the estimates. The output from Factor Analysis I is given on pages 22 - 32. We are unable to determine the exact number of factors from the eigenvalues although we can see that there will be three or four factors.

The Bargmann test indicates that there are four factors since the first deviation value below 1.96 (95% confidence level) occurs on the fourth factor. But, upon examining the factor matrix we see that there are no factor loadings above .270 in the fourth factor so we conclude that there are really only three significant factors.

In order to obtain a more accurate factor matrix and to stabilize the communalities, we will rerun this first program with communality estimates computed from three significant factors of the previous factor matrix. The new factor matrix (with an excess of factors) from this second run is given on page 32. It will be used as input to Factor Analysis II.

Table A.5 Factor Analysis I Input Data for Sample Problem

Card No.	12345678...	Column No.	...
1.	FACTOR ANALYSIS I - CALCULATION OF SMC FACTOR MATRIX - R. C. DURFEE - TEST CASE		
2.	2 1		
3.	9 504 5		
4.	(F2.0,8F5.3)		
5.	1. .511 .498 .542 .509 .445 .372 .333 .281		
6.	1. .473 .736 .462 .426 .435 .370 .414		
7.	1. .504 .462 .418 .372 .330 .354		
8.	1. .654 .764 .439 .449 .455		
9.	1. .645 .405 .417 .406		
10..	1. .376 .376 .458		
11.	1. .686 .589		
12.	1. .535		
13.	1.		

FACTOR ANALYSIS I - CALCULATION OF SMC FACTOR MATRIX - R. C. DURFEE - TEST CASE

NO. OF VARIABLES = 9
NO. OF OBSERVATIONS = 504

CORRELATION MATRIX (UPPER DIAG. BY ROWS)

DETERMINANT OF REFL. CORR. MATRIX = 0.732264668E-02

VAR. PIVOT ELEMENTS

1	0.1000000E 01
2	0.73887902E 00
3	0.68736857E 00
4	0.40698808E 00
5	0.52196521E 00
6	0.33980942E 00
7	0.73169041E 00
8	0.49552578E 00
9	0.55087537E 00

VAR. SQUARED MULT. CORR.

1	0.41435224
2	0.63503993
3	0.36317712
4	0.80288446
5	0.52686715
6	0.67626745
7	0.56406623
8	0.52019387
9	0.44912457

TRACE = 5.92499628

FACTOR	EIGENVALUE	EIGENVALUE DIFF.	EIGENVALUE SUM	PERCENT VARIANCE	ITERATIONS
1	4.47143269	3.67127228	4.47143269	75.46722412	5
2	0.80016005	0.37582254	5.27159214	88.97203064	9
3	0.42433751	0.16735673	5.69592953	96.13385010	7
4	0.25698878	0.16666502	5.95290947	100.47100830	5
5	0.09031576		6.04322433	101.99536133	8

FACTOR MATRIX

ROW

	1	2	3	4	5
1	.632	.152	.163	-.269	-.049
2	.732	.145	.418	.168	.060
3	.607	.079	.136	-.223	.098
4	.888	.318	.000	.250	-.112
5	.731	.154	-.163	-.180	-.046
6	.760	.250	-.409	.045	.070
7	.675	-.495	.036	-.011	-.040
8	.642	-.460	-.046	.004	-.152
9	.632	-.325	-.088	.097	.175

FACTOR	CHISQ	DEVIATION
1	661.97265625	27.95988464
2	296.13818359	15.91056633
3	162.03767395	9.57592583
4	50.64752197	1.63839436
5	23.24096680	-1.60838413

30'

FACTOR MATRIX COMM. BASED ON 4 SIGNIFICANT FACTORS

ROW COMMUNALITIES

1	0.52101928
2	0.76020306
3	0.44296360
4	0.95234269
5	0.61737323
6	0.80828947
7	0.70249718
8	0.62660819
9	0.52167803

FACTOR ANALYSIS I - CALCULATION OF 3C FACTOR MATRIX - R. C. DURFEE - TEST CASE

FACTOR MATRIX

ROW	1	2	3	4
1	.626	-.151	.156	-.249
2	.729	-.147	.407	.164
3	.603	-.084	.130	-.210
4	.876	-.303	.009	.204
5	.726	-.157	-.143	-.162
6	.761	-.264	-.407	.055
7	.683	.505	.032	-.017
8	.646	.457	-.051	-.004
9	.631	.306	-.088	.100

Factor Analysis II

Input - The input variables read by FACTOR ANALYSIS II are described in Table A.6. The sequence in which these variables are read and the format by which they are read is given in Table A.7.

Table A.6 Factor Analysis II Input Variable Description

Input Variable	Description
1. TITLE	Alphanumeric title identification
2. NVAR	Number of variables
3. NOFACT	Number of factors in the unrotated factor matrix
4. FMAT	Unrotated factor matrix (from Factor Analysis I)

Table A.7 Factor Analysis II Input Variable Sequence and Format

Sequence	Input Variable	FORMAT Statement Used
1.	TITLE	FORMAT(13A6,A2)
2.	NVAR,NOFACT	FORMAT(2I4)
3.	FMAT	FORMAT(8X,6F12.8)

Output - The output variables printed by FACTOR ANALYSIS II are described in Table A.8. The sequence in which these variables are printed is given in Table A.9.

Table A.8 Factor Analysis II Output Variable Description

Output Variable	Description
1. TITLE	Alphanumeric title identification
2. KROW	Indicator of rows reflected in the unrotated factor matrix
3. FCOMM	Communalities of the unrotated factor matrix
4. VCUMM	Communalities of the rotated varimax matrix
5. KTCOL	Indicator of columns reflected in the rotated varimax matrix due to a negative first row transformation element
6. KVCOL	Indicator of columns reflected in the rotated varimax matrix due to a negative column sum in the varimax matrix
7. NOROT	Number of rotations for each iteration of the varimax rotation
8. VARMAT	Rotated varimax matrix (outputted in numerical order by loadings)
9. VTRAN	Varimax transformation matrix
10. NOSTRT NOSTOP }	Specification of variable numbers to be used in next simple structure rotation
11. SSMAT	Rotated simple structure matrix
12. SSTRAN	Simple structure transformation matrix (will also be punched out for later use)

Table A.9 Factor Analysis II Output Variable Sequence

Sequence	Output Variable
1.	TITLE
2.	KROW,FCOMM,VCOMM
3.	KTCOL,KVCOL
4.	NOROT
5.	VARMAT
6.	VTRAN
7.	NOSTRT,NOSTOP
8.	SSORD
9.	SSTRAN

Sample Problem - Using the results from the sample problem in Factor Analysis I we can prepare the input data to Factor Analysis II given in Table A.10. The second factor matrix which was computed and punched out in Factor Analysis I is used as input to this program. Only the first three significant factors were kept, and the remaining cards were discarded.

The output from Factor Analysis II is given on pages 37-45. By examining the varimax factor loadings we can begin to see which tests make up each of the factors. Variables 1, 2, 3, 4 have fairly high loadings on the first factor; variables 7, 8, 9 have high loadings on the second factor; and variables 4, 5, 6 have high loadings on the third factor. By looking at the questions which correspond to each of these

variables, we can identify each of the three factors. The first factor seems to be verbal ability; the second factor seems to be spatial perception; and the third factor seems to be numerical reasoning. Note that the fourth variable contributes to both verbal ability and numerical reasoning. Those factor loadings which do not contribute significantly to a given factor approach zero in the simple structure rotations which follow the varimax rotation. Note that the sequence numbers rather than the variable numbers are used to specify the variables used on each successive simple structure rotation.

Table A.10 Factor Analysis II Input Data for Sample Problem

Card No.	12345678...	Column No.	...
1.			
			FACTOR ANALYSIS II - CALCULATION OF VARIMAX AND SIMPLE STRUCTURE MATRIX - RCD
2.	9	3	
3.	1	0.62642980 0.72580057	0.72937912 0.76098335
4.		0.68274549	0.64598405 0.63112396
5.	2	-0.15120381 -0.15737027	-0.14663714 -0.26391876
6.		0.50468314	0.45681405 0.30555046
7.	3	0.15558910 -0.14286011	0.40688944 -0.40691626
8.		0.03166087	-0.05113751 -0.08781397

FACTOR ANALYSIS II - CALCULATION OF VARIMAX AND SIMPLE STRUCTURE MATRIX - RCD

ROW	FACTOR MATRIX REFLECTION	FACTOR MATRIX COMMUNALITIES	VARI MAX MATRIX COMMUNALITIES
1	1	0.43948478	0.43948448
2	1	0.71905524	0.71905476
3	1	0.38767892	0.38767874
4	1	0.85929626	0.85929549
5	1	0.57196069	0.57196045
6	1	0.81432945	0.81432903
7	1	0.72184604	0.72184545
8	1	0.62858933	0.62858880
9	1	0.49938965	0.49938941

VARI MAX MATRIX COLUMN REFLECTION

COL.	NEGATIVE TRANS. ELEMENT	NEGATIVE COLUMN SUM
1	1	1
2	1	1
3	-1	1

NUMBER OF ROTATIONS
ITR. PER ITERATION

1	2
2	3
3	1
4	0

Figure A.2 (continued)

ORDERED VARIMAX MATRIX

SEQ.	1	2	3
1	2	.787	7
2	4	.652	8
3	1	.548	9
4	3	.492	5
5	5	.401	3
6	6	.274	2
7	7	.253	4
8	9	.217	6
9	8	.199	1
			.215
			7
			.135

TRANSFORMATION MATRIX

ROW	1	2	3
1	.614	.561	.556
2	-.354	.825	-.441
3	.705	-.074	-.705

SEQUENCE NO.S REMAINING
FOR NEXT ROTATION

1	8	-	9
2	5	-	9
3	7	-	9

SPECIAL SIMPLE STRUCTURE MATRIX

SEQ.	1	2	3
1	2	.601	7
2	1	.341	8
3	4	.305	9
4	3	.295	5
5	5	.089	3
6	7	.075	6
7	8	.000	1
8	9	-.000	2
9	6	=.122	4
		=.092	7
			=.038

TRANSFORMATION MATRIX

ROW	1	2	3
1	.252	.270	.348
2	-.251	.951	-.495
3	.935	-.153	-.796

SEQUENCE NO.S REMAINING
FOR NEXT ROTATION

1	5 - 9
2	4 - 9
3	4 - 9

SIMPLE STRUCTURE ROTATED MATRIX (ITR.=1)

SEQ.		1	2	3
1	2	.506	7 .654	6 .653
2	1	.355	8 .610	5 .369
3	4	.340	9 .466	4 .340
4	3	.303	3 .058	9 .092
5	5	.114	5 .053	1 .085
6	7	.030	1 -.005	3 .071
7	9	-.022	6 -.006	8 -.003
8	8	-.037	2 -.006	7 -.087
9	6	-.079	4 -.067	2 -.106

TRANSFORMATION MATRIX

ROW		1	2	3
1		.259	.256	.242
2		-.347	.958	-.446
3		.902	-.129	-.862

SEQUENCE NO.'S REMAINING
FOR NEXT ROTATION

1		5 - 9
2		4 - 9
3		4 - 9

SIMPLE STRUCTURE ROTATED MATRIX (ITR.=2)

SEQ.	1	2	3
1	2	.697	7 .654
2	1	.355	8 .610
3	4	.341	9 .465
4	3	.303	3 .058
5	5	.115	5 .053
6	7	.029	1 -.005
7	9	-.022	2 -.006
8	8	-.038	6 -.006
9	6	-.077	4 =.067
			2 =.112

TRANSFORMATION MATRIX

ROW	1	2	3
1	.259	.256	.240
2	-.350	.958	-.441
3	.900	-.128	-.865

SEQUENCE NO.S REMAINING
FOR NEXT ROTATION

1	5 - 9
2	4 - 9
3	4 - 9

SIMPLE STRUCTURE ROTATED MATRIX (ITR.=3)

SEQ.		1	2	3
1	2	.007	7	.654
2	1	.355	8	.610
3	4	.341	9	.465
4	3	.303	3	.058
5	5	.115	5	.053
6	7	.029	1	-.005
7	9	-.022	2	-.006
8	8	-.038	6	-.006
9	6	-.077	4	-.067
				2
				-.112

TRANSFORMATION MATRIX

ROW		1	2	3
1		.259	.256	.240
2		-.350	.958	-.441
3		.900	-.128	-.865

SEQUENCE NO.'S REMAINING
FOR NEXT ROTATION

1		5	-	9
2		4	-	9
3		4	-	9

SIMPLE STRUCTURE ROTATED MATRIX (ITR.=4)

SEQ.	1	2	3
1	2	.607	7
2	1	.355	8
3	4	.341	9
4	3	.303	3
5	5	.115	5
6	7	.029	1
7	9	-.022	2
8	8	-.038	6
9	6	-.077	4

1	2	.607	7	.654	6	.651
2	1	.355	8	.610	5	.367
3	4	.341	9	.465	4	.336
4	3	.303	3	.058	9	.093
5	5	.115	5	.053	1	.082
6	7	.029	1	-.005	3	.069
7	9	-.022	2	-.006	8	-.002
8	8	-.038	6	-.006	7	-.086
9	6	-.077	4	-.067	2	-.112

TRANSFORMATION MATRIX

ROW	1	2	3
1	.259	.256	.240
2	-.350	.958	-.441
3	.900	-.128	-.865

SEQUENCE NOS. REMAINING
FOR NEXT ROTATION

1	6	9
2	4	9
3	4	9

SIMPLE STRUCTURE ROTATED MATRIX (ITR.=5)

SEQ.		1	2	3
1	2	.538	7	.654
2	4	.410	8	.610
3	1	.393	9	.465
4	3	.336	3	.058
5	5	.172	5	.053
6	7	.031	1	-.005
7	6	.003	2	-.006
8	9	-.004	6	-.006
9	8	-.031	4	-.067
			2	-.112

TRANSFORMATION MATRIX

ROW		1	2	3
1		.315	.256	.240
2		-.418	.958	-.441
3		.852	-.128	-.865

Factor Analysis III

Input - The input variables read by FACTOR ANALYSIS III are described in Table A.11. The sequence in which these variables are read and the format by which they are read is given in Table A.12.

Table A.11 Factor Analysis III Input Variable Description

Input Variable	Description
1. TITLE	Alphanumeric title identification
2. NVAR	Number of variables
3. NOFACT	Number of factors in the unrotated factor matrix
4. FMAT	Unrotated factor matrix (from Factor Analysis I)
5. SSTRAN	Most recently calculated transformation matrix
6. REMAIN CARD	Specification card indicating the variables remaining in the hyperplane for the rotation of that hyperplane
DELETE CARD	Specification card indicating the variables deleted from the hyperplane for the rotation of that hyperplane
WEIGHT CARD	Specification card indicating row weights for those variables remaining in the hyperplane
FINISH CARD	Specification card indicating end of data

Table A.12 Factor Analysis III Input Variable Sequence and Format

Sequence	Input Variable	FORMAT Statement Used
1.	TITLE	FORMAT(13A6,A2)
2.	NVAR,NOFACT	FORMAT(2I4)
3.	FMAT	FORMAT(8X,6F12.8)
4.	SSTRAN	FORMAT(31X,F19.8)
5.	REMAIN CARD	FORMAT(See FORTRAN Listing)
	DELETE CARD	(See FORTRAN Listing)
	WEIGHT CARD	(See FORTRAN Listing)
	FINISH CARD	(See FORTRAN Listing)

For a given factor a REMAIN CARD or a DELETE CARD may be used, but not both. See the FORTRAN listing in Appendix B (pp. 81-82) for a further explanation of the REMAIN, DELETE, WEIGHT, and FINISH CARD variables. A description of the input card format is also given there.

Output - The output variables printed by FACTOR ANALYSIS III are described in Table A.13. The sequence in which these variables are printed is given in Table A.14.

Table A.13 Factor Analysis III Output Variable Description

Output Variable	Description
1. TITLE	Alphanumeric title identification
2. SSMAT	Rotated simple structure matrix (outputted in numerical order by loadings).
3. SSTRAN	Simple structure transformation matrix (will also be punched out for later use)
4. COSMAT	Matrix of cosines of angles between reference vectors

Table A.14 Factor Analysis III Output Variable Sequence

Sequence	Output Variable
1.	TITLE
2.	SSMAT
3.	SSTRAN
4.	COSMAT

Sample Problem - Using the results from the sample problem in Factor Analysis I and Factor Analysis II we can prepare the input data to Factor Analysis III given in Table A.15. We again used the factor matrix computed and punched out in Factor Analysis I. The transformation matrix computed and punched out during the special simple structure rotation in Factor Analysis II is used as our initial trial matrix. The remaining variables were 5, 6, 7, 8, 9 for the first factor; 1, 2, 3, 4, 5, 6 for the second factor; and 1, 2, 3, 7, 8, 9 for the third factor. Instead of using REMAIN CARDS, we could have used DELETE CARDS and specified deleted variables 1, 2, 3, 4 for the first factor; 7, 8, 9 for the second factor; and 4, 5, 6 for the third factor. In order to force some of the loadings of the remaining variables towards zero, we used weights as shown by the WEIGHT CARDS.

The output from Factor Analysis III is given on pages 51-52. We see that a very nice fit has been obtained, with the three factors previously mentioned now being quite evident. The first factor is a verbal ability factor; the second is a spatial perception factor; and the third is a numerical reasoning factor.

As a matter of interest, we also repeated the whole factor analysis using four factors instead of three. This was done because Factor Analysis I suggested that there might possibly be four factors. But, we found in Factor Analysis II that the last two factors converged to the same factor, thus yielding a three factor result.

Table A.15 Factor Analysis III Input Data for Sample Problem

Card No.	12345678...	Column No.	...
1.	FACTOR ANALYSIS III - CALCULATION OF SIMPLE STRUCTURE MATRIX -		
	R. C. DURFEE		
2.	9 3		
3.	1 0.62642980 0.72937912 0.60307741 0.87613046 0.72580057 0.76098335		
4.	0.68274349 0.64598405 0.63112396		
5.	2 -0.15120381 -0.14663714 -0.08351127 -0.30267036 -0.15737027 0.26391876		
6.	0.50468314 0.45681405 0.30555046		
7.	3 0.15558910 0.10688944 0.13039237 0.00908275 -0.14286011 -0.40691626		
8.	0.03166087 -0.05113751 -0.08781397		
9.	1 1 0.37549055		
10.	1 2 0.27023143		
11.	1 3 0.34779316		
12.	2 1 -0.51528352		
13.	2 2 0.95063615		
14.	2 3 -0.49506676		
15.	3 1 0.77038342		
16.	3 2 -0.15253228		
17.	3 3 -0.79621029		
18.	REMAIN 1 5 **** 5 6 7 9 8		
19.	WEIGHT 1 3 **** 5 3.0 6 2.5 8 2.0		
20.	REMAIN 2 6 **** 5 3 6 1 2 4		
21.	REMAIN 3 6 **** 1 3 9 8 2 7		
22.	WEIGHT 3 4 **** 2 2.0 9 1.7 7 1.5 1 1.5		
23.	FINISH		

FACTOR ANALYSIS FIT - CALCULATION OF SIMPLE STRUCTURE MATRIX - R. C. DURFEE

SIMPLE STRUCTURE ROTATED MATRIX

SEQ.		1	2	3		
1	2	.502	7	.654	6	.664
2	1	-.351	8	-.610	5	-.382
3	4	.334	9	.466	4	.362
4	3	.299	3	.058	1	.102
5	5	.109	5	.053	9	.091
6	7	.025	1	-.005	3	.085
7	9	-.027	6	-.006	8	-.009
8	8	-.043	2	-.006	2	-.084
9	6	-.084	4	-.087	7	-.092

TRANSFORMATION MATRIX

ROW		1	2	3		
1		.252		.256		.257
2		-.349		.958		-.477
3		.903		-.129		-.841

COSINE MATRIX

ROW		1	2	3		
1		1.000		-.386		-.528
2		-.386		1.000		-.283
3		-.528		-.283		1.000

Fig. A.3 (continued).

Appendix B

FORTRAN LIST OF THE CODES

Factor Analysis I Computer Program

C FACTOR ANALYSIS I
C THIS PROGRAM COMPUTES AN UNROTATED FACTOR MATRIX
C SPONSOR - DR. CURETON
C PROGRAMMER - RICHARD C. DURFEE
C THIS PROGRAM CONSISTS OF THREE SECTIONS
C 1. CORRELATION MATRIX FROM RAW SCORE MATRIX
C 2. HORST PRINCIPAL AXIS FACTOR MATRIX FROM CORRELATION MATRIX
C 3. BARGMANN TEST FOR NUMBER OF SIGNIFICANT FACTORS
C
C NVAR = NUMBER OF VARIABLES
C NOBS = NUMBER OF OBSERVATIONS
C SCORE = RAW SCORE MATRIX
C AVG = MEAN FOR EACH VARIABLE
C SIGMA = STANDARD DEVIATION FOR EACH VARIABLE
C R = CORRELATION MATRIX
C RINV = INVERSE OF THE CORRELATION MATRIX
C DETR = DETERMINANT OF THE CORRELATION MATRIX
C PIVOT = PIVOT ELEMENTS USED IN INVERSION OF CORRELATION MATRIX
C SMC = SQUARED MULTIPLE CORRELATIONS
C TRACE = SUM OF DIAGONAL ELEMENTS OF THE CORRELATION MATRIX
C COMM = COMMUNALITIES
C NOEXT = NUMBER OF FACTORS TO BE EXTRACTED FROM THE CORRELATION MATRIX
C EIGVAL = EIGENVALUES
C FMAT = FACTOR MATRIX = MATRIX OF EIGENVECTORS
C NOFACT = NUMBER OF FACTORS IN THE FACTOR MATRIX
C EVSUM = SUM OF EIGENVALUES
C EVDIFF = DIFFERENCE OF CONSECUTIVE EIGENVALUES
C PERCT = PERCENTAGE OF TRACE FACTORED BY EACH SUCCESSIVE FACTOR
C DF = DEGREES OF FREEDOM
C CHISQ = CHI SQUARED
C DEV = DEVIATION
C NFCOMM = NUMBER OF FACTORS USED IN CALCULATING COMMUNALITY ESTIMATES
C NOSIG = NUMBER OF SIGNIFICANT FACTORS
C TITLE = ALPHANUMERIC TITLE IDENTIFICATION
C LOGICAL TAPE UNIT L1 IS USED FOR STORING R AND RINV
C THE SENSE SWITCH DEFINITIONS ARE
C*****
C SS1 = 1 TO COMPUTE A CORRELATION MATRIX FROM AN INPUTTED RAW SCORE MATRIX
C SS1 = 2 TO USE AN INPUTTED CORRELATION MATRIX (PREVIOUSLY CALCULATED)
C *****
C SS2 = 1 TO COMPUTE ESTIMATED COMMUNALITIES FROM SMC VALUES FOR R DIAGONAL
C SS2 = 2 TO USE INPUTTED COMMUNALITY ESTIMATES ON DIAGONAL OF R
C*****
C DIMENSION R(100,100),RINV(100,100),FMAT(100,21),AVG(100),
C 1SIGMA(100),SMC(100),COMM(100),EIGVAL(21),EVSUM(21),EVDIFF(21),
C 2PERCT(21),NOITR(21),CHISQ(21),DEV(21),TITLE(14),KR(100),S(100),
C 3AVEC(100),BVEC(100),Y(21,21),PIVOT(100)
C EQUIVALENCE (R,RINV)
C INTEGER SS1,SS2
C DOUBLE PRECISION TITLE
150 L1=L
REWIND L1
C*****BEGIN INPUT
CALL INF1(TITLE,SS1,SS2,NVAR,NOBS,NOEXT,R,AVG,COMM,NFCOMM)
C IF A RAW SCORE MATRIX IS INPUTTED, R IS THE CROSS-PRODUCT MATRIX AND
C AVG IS THE SUM OF OBSERVATIONS FOR EACH VARIABLE (AT THIS POINT)
C*****END INPUT
C*****
C*****BEGIN SECTION 1 - CORRELATION MATRIX FROM RAW SCORE MATRIX

```

SIZE=NOBS
1 IF(ISS1 .EQ. 2) GO TO 4
2 DO 2 I=1,NVAR
3 AVG(I)=AVG(I)/SIZE
4 SIGMA(I)=SQRT((R(I,I)-SIZE*AVG(I)*AVG(I))/(SIZE-1.0))
5 DO 3 I=1,NVAR
6 DO 3 J=1,NVAR
7 R(I,J)=(R(I,J)-SIZE*AVG(I)*AVG(J))/((SIZE-1.0)*SIGMA(I)*SIGMA(J))
C*****END SECTION 1 - CORRELATION MATRIX FROM RAW SCORE MATRIX
C*****BEGIN SECTION 2 - PRINCIPAL AXIS FACTOR MATRIX FROM CORRELATION MATRIX
C*****BEGIN REFLECTION OF CORRELATION MATRIX
8 DO 5 I=1,NVAR
9 KR(I)=1
10 R(I,I)=0.
11 IMIN=0
12 SMIN=0.
13 DO 9 I=1,NVAR
14 S(I)=0.
15 DO 7 J=1,NVAR
16 S(I)=S(I)+R(I,J)
17 IF(S(I) .GE. SMIN) GO TO 9
18 SMIN=S(I)
19 IMIN=I
20 CONTINUE
21 IF(IMIN .EQ. 0) GO TO 12
22 KR(IMIN)=-KR(IMIN)
23 DO 11 I=1,NVAR
24 R(IMIN,I)=-R(IMIN,I)
25 R(I,IMIN)=-R(I,IMIN)
26 GO TO 6
27 DO 121 I=1,NVAR
28 R(I,I)=1.0
C*****END REFLECTION OF CORRELATION MATRIX
C*****BEGIN INVERSION OF CORRELATION MATRIX
29 WRITE(L1) ((R(I,J),J=1,NVAR),I=1,NVAR)
30 DETR=1.0
31 DO 16 K=1,NVAR
32 COM=RINV(K,K)
33 IF(ABS(COM) .LT. 1.E-35) COM=1.E-10
34 PIVOT(K)=COM
35 DETR=DETR*COM
36 RINV(K,K)=1.
37 DO 13 J=1,NVAR
38 RINV(K,J)=RINV(K,J)/COM
39 DO 16 I=1,NVAR
40 IF(I .EQ. K) GO TO 16
41 COM=RINV(I,K)
42 RINV(I,K)=0.
43 DO 15 J=1,NVAR
44 RINV(I,J)=RINV(I,J)-COM*RINV(K,J)
45 CONTINUE
46 WRITE(L1) ((RINV(I,J),J=1,NVAR),I=1,NVAR)
47 REWIND L1
C*****END INVERSION OF CORRELATION MATRIX
C*****BEGIN CALCULATION OF SMC VALUES
48 DO 17 I=1,NVAR
49 SMC(I)=1.0-(1.0/RINV(I,I))
C*****END CALCULATION OF SMC VALUES
C*****BEGIN READ IN OF CORRELATION MATRIX FROM LOGICAL TAPE L1

```

```

READ(L1) ((R(I,J),J=1,NVAR),I=1,NVAR)
DO 18 I=1,NVAR
DO 18 J=I,NVAR
18 R(J,I)=R(I,J)
C*****END READ IN OF CORRELATION MATRIX FROM LOGICAL TAPE L1
C*****BEGIN CALCULATION OF ESTIMATED COMMUNALITIES FROM SMC VALUES
  IF(SSL .EQ. 2) GO TO 27
  SUMSMC=0.
  SUMR=0.
  DO 24 I=1,NVAR
  SUMSMC=SUMSMC+SMC(I)
  R(I,I)=0.
  RMAX=0.
  DO 23 J=1,NVAR
  IF(RMAX .GE. ABS(R(I,J))) GO TO 23
  RMAX=ABS(R(I,J))
23  CONTINUE
24  SUMR=SUMR+RMAX
  CONS=SUMR/SUMSMC
25  DO 26 I=1,NVAR
26  COMM(I)=SMC(I)*CONS
C*****END CALCULATION OF ESTIMATED COMMUNALITIES FROM SMC VALUES
C*****BEGIN CALCULATION OF TRACE
27  TRACE=0.
  DO 29 I=1,NVAR
28  TRACE=TRACE+COMM(I)
C*****END CALCULATION OF TRACE
C*****BEGIN CALCULATION OF EIGENVALUES AND CORRESPONDING FACTOR MATRIX
  CALL SUBFM(NVAR,NOEXT,COMM,R,EIGVAL,EVDIFF,EVSUM,PERCT,NOITR,FMAT)
C*****END CALCULATION OF EIGENVALUES AND CORRESPONDING FACTOR MATRIX
C*****END SECTION 2 - PRINCIPAL AXIS FACTOR MATRIX FROM CORRELATION MATRIX
C*****BEGIN SECTION 3 - BARGMANN TEST FOR NUMBER OF SIGNIFICANT FACTORS
  READ(L1) ((RINV(I,J),J=I,NVAR),I=1,NVAR)
  REWIND L1
  DO 29 I=1,NVAR
  DO 29 J=I,NVAR
29  RINV(J,I)=RINV(I,J)
  NOFACT=NOEXT
  DO 32 ICOL=1,NOFACT
  DO 30 I=1,NVAR
  BVEC(I)=0.
  DO 30 J=1,NVAR
30  BVEC(I)=BVEC(I)+RINV(I,J)*FMAT(J,ICOL)
  DO 31 IROW=1,NOFACT
  Y(IROW,ICOL)=0.
  DO 31 L=1,NVAR
31  Y(IROW,ICOL)=Y(IROW,ICOL)+FMAT(L,IROW)*BVEC(L)
32  CONTINUE
  DO 33 I=1,NOFACT
  DO 33 J=1,NOFACT
  Y(I,J)=-Y(I,J)
33  IF(I .EQ. J) Y(I,J)=1.0+Y(I,J)
  AVEC(1)=Y(1,1)
  DO 37 K=1,NOFACT
  IF(K .EQ. 1) GO TO 34
  AVEC(K)=AVEC(K-1)*Y(K,K)
34  JSET=K+1
  IF(JSET .GT. NOFACT) GO TO 37
  DO 35 J=JSET,NOFACT

```

```

35 BVEC(J)=Y(K,J)/Y(K,K)
DO 36 I=JSET,NOFACT
DO 36 J=1,NOFACT
36 Y(I,J)=Y(I,J)-Y(K,I)*BVEC(J)
CONTINUE
VARNO=NVAR
H=SIZE-(2.0*VARNO-11.0)/6.0
DF=VARNO*(VARNO-1.0)/2.0
IF(DETR .EQ. 0.) DETR=1.E-10
TLOGR=ALOG(ABS(DETR))
ROOT=SQRT(2.0*DF-1.0)
DEVTOL=1.960
C DEVTOL IS THE CONFIDENCE LEVEL USED IN CHOOSING THE SIGNIFICANT FACTORS
DO 38 I=1,NVAR
38 COMM(I)=0.
NOSIG=0
DO 40 K=1,NOFACT
FK=K
GK=0.
BK=H-FK
DO 39 I=1,NVAR
COMM(I)=COMM(I)+FMAT(I,K)**2
COMM1=1.0-COMM(I)
IF(COMM1 .EQ. 0.) COMM1=1.0E-10
GK=GK+ALOG(ABS(COMM1))
39 AVEC(K)=ABS(AVEC(K))
IF(AVEC(K) .EQ. 0.) AVEC(K)=1.0E-10
CHISQ(K)=BK*(GK-ALOG(AVEC(K))-TLOGR)
DEV(K)=SQRT(ABS(2.*CHISQ(K)))-ROOT
IF((DEV(K) .LT. DEVTOL) .AND. (NOSIG .EQ. 0)) NOSIG=K
40 CONTINUE
C*****BEGIN COMMUNALITY CALCULATION FROM SIGNIFICANT FACTORS
IF(NFCOMM .NE. 0) NOSIG=NFCOMM
DO 41 I=1,NVAR
COMM(I)=0.
DO 41 J=1,NOSIG
41 COMM(I)=COMM(I)+FMAT(I,J)**2
C*****END COMMUNALITY CALCULATION FROM SIGNIFICANT FACTORS
C*****END SECTION 3 - BARGMANN TEST FOR NUMBER OF SIGNIFICANT FACTORS
C*****END ****
C*****BEGIN OUTPUT
READ(L1) ((R(I,J),J=I,NVAR),I=1,NVAR)
DO 42 I=1,NVAR
DO 42 J=I,NVAR
R(J,I)=R(I,J)
DO 44 I=1,NVAR
IF(KR(I) .EQ. (+1)) GO TO 44
DO 43 J=1,NVAR
R(I,J)=-R(I,J)
43 R(J,I)=-R(J,I)
44 CONTINUE
CALL OUTFA1(TITLE,SS1,SS2,NVAR,NOBS,NOFACT,NOSIG,AVG,SIGMA,KR,R,
1DETR,PIVOT,SMC,TRACE,EIGVAL,EVIDIFF,EVSUM,PERCT,NOTR,CHISQ,DEV,
2FMAT,COMM)
C*****END OUTPUT
GO TO 150
END

```

```

SUBROUTINE SUBFM(NVAR,NOEXT,COMM,R,EIGVAL,EVDIFF,EVSUM,PERCT,
INOITR,FMAT)
C THIS SUBROUTINE CALCULATES THE EIGENVALUES AND FACTOR MATRIX FROM THE
C CORRELATION MATRIX GIVEN
DIMENSTON COMM(100),R(100,100),EIGVAL(21),EVDIFF(21),EVSUM(21),
1PERCT(21),NOITR(21),FMAT(100,21),AVEC(100),BVEC(100)
SUMCOM=0.
DO 1 I=1,NVAR
1 SUMCOM=SUMCOM+COMM(I)
VARNO=NVAR
EPSLON=.0001
C EPSLON IS THE TOLERANCE USED IN COMPARING CONSECUTIVE EIGENVALUES
DO 16 L=1,NOEXT
NOITR(L)=0
RMAX=0.
IMAX=0.
DO 5 I=1,NVAR
R(I,I)=0.
SUMR=0.
DO 2 J=1,NVAR
2 SUMR=SUMR+ABS(R(I,J))
SUMR=SUMR/(2.0*VARNO-1.0)
IF(COMM(I) .GE. SUMR) GO TO 3
R(I,I)=SUMR
GO TO 4
3 R(I,I)=COMM(I)
4 IF(RMAX .GE. R(I,I)) GO TO 5
RMAX=R(I,I)
IMAX=I
5 CONTINUE
EIG=SQRT(RMAX)
DO 6 I=1,NVAR
6 AVEC(I)=R(I,IMAX)/EIG
ELAST=EIG
NOITR(L)=NOITR(L)+1
SUMAB=0.
DO 9 I=1,NVAR
BVEC(I)=0.
DO 8 J=1,NVAR
8 BVEC(I)=BVEC(I)+R(I,J)*AVEC(J)
SUMAB=SUMAB+BVEC(I)*AVEC(I)
SUMAB=ABS(SUMAB)
EIG=SQRT(SUMAB)
DO 10 I=1,NVAR
10 AVEC(I)=BVEC(I)/EIG
IFI(ABS(EIG-ELAST) .GT. EPSLON) GO TO 7
EIGVAL(L)=EIG
IFI(L .GT. 1) GO TO 11
EVSUM(L)=EIG
EVDIFF(L)=0.
GO TO 12
11 EVSUM(L)=EVSUM(L-1)+EIGVAL(L)
EVDIFF(L)=EIGVAL(L-1)-EIGVAL(L)
12 PERCT(L)=(EVSUM(L)/SUMCOM)*100.
DO 13 I=1,NVAR
13 FMAT(I,L)=AVEC(I)
IFI(L .EQ. NOEXT) GO TO 16
DO 15 I=1,NVAR

```

```
DO 14 J=1,NVAR  
14 R(I,J)=R(I,J)-AVEC(I)*AVEC(J)  
15 COMM(I)=R(I,I)  
16 CONTINUE  
RETURN  
END
```

C SUBROUTINE INFA1(TITLE,SS1,SS2,NVAR,NOBS,NOEXT,R,AVG,COMM,NFCOMM)
 C THIS SUBROUTINE READS AND SUPPLIES ALL INPUT REQUIRED IN FACTOR ANALYSIS I
 C SPONSOR - DR. CURETON
 C PROGRAMMER - RICHARD C. DURFEE
 C TITLE = ALPHANUMERIC TITLE IDENTIFICATION
 C NVAR = NUMBER OF VARIABLES
 C NOBS = NUMBER OF OBSERVATIONS
 C NOEXT = NUMBER OF FACTORS TO BE EXTRACTED FROM THE CORRELATION MATRIX
 C FMTX = INPUT FORMAT STATEMENT UNDER WHICH RAW SCORE MATRIX IS READ
 C SCORE = RAW SCORE MATRIX
 C FMTR = INPUT FORMAT STATEMENT UNDER WHICH CORRELATION MATRIX IS READ
 C R = CORRELATION MATRIX
 C FMTC = INPUT FORMAT STATEMENT UNDER WHICH COMMUNALITIES ARE READ
 C COMM = COMMUNALITIES
 C NFCOMM = NO. OF FACTORS IN FACTOR MATRIX (FROM WHICH COMM ARE COMPUTED)
 C FMAT = FACTOR MATRIX FROM WHICH COMMUNALITIES ARE COMPUTED
 C THE SENSE SWITCH DEFINITIONS ARE
 C *****
 C SS1 = 1 TO READ IN A RAW SCORE MATRIX(NOBS,NVAR) - READ BY ROWS
 C SS1 = 2 TO READ IN A CORRELATION MATRIX(NVAR,NVAR) - UPPER DIAGONAL BY ROW
 C *****
 C SS2 = 1 TO READ NO COMMUNALITIES BUT USE COMPUTED SMC VALUES INSTEAD
 C SS2 = 2 TO READ COMMUNALITIES THEMSELVES - FMTC FORMAT READ AND USED ALSO
 C SS2 = 3 TO READ A FACTOR MATRIX FROM WHICH COMMUNALITIES ARE COMPUTED
 C *****

C INPUT CONSISTS OF

VARIABLES	FORMAT
1. TITLE	- FORMAT(13A6,A2)
2. SS1,SS2	- FORMAT(2I2)
3. NVAR,NOBS,NOEXT	- FORMAT(3I4)
4. FMTX (IF SS1=1) OR FMTR (IF SS1=2)	- FORMAT(18A4)
5. SCORE (IF SS1=1) OR R (IF SS1=2)	- FMTX OR FMTR
6. FMTC (IF SS2=2) OR NFCOMM (IF SS2=3)	- FORMAT(18A4) OR (I4)
7. COMM (IF SS2=2) OR FMAT (IF SS2=3)	- FMTC OR FORMAT(105)

DIMENSION TITLE(14),FMTX(18),SCORE(100),FMTR(18),R(100,100),
 1FMTC(12),COMM(100),FMAT(100),AVG(100)

INTGFR SS1,SS2

DOUBLE PRECISION TITLE

READ 100,(TITLE(I),I=1,14)

100 FORMAT(13A6,A2)

READ 101,SS1,SS2

101 FORMAT(2I2)

READ 102,NVAR,NOBS,NOEXT

102 FORMAT(3I4)

IF(SS1 .EQ. 1) READ 103,(FMTX(I),I=1,18)

C \$IF(SS1 .EQ. 1) READ 103,(FMTX(I),I=1,12)

IF(SS1 .EQ. 2) READ 103,(FMTR(I),I=1,18)

C \$IF(SS1 .EQ. 2) READ 103,(FMTR(I),I=1,12)

103 FORMAT(18A4)

C103 \$FORMAT(12A6)

IF(SS1 .EQ. 2) GO TO 3

DO 1 I=1,NVAR

AVG(I)=0.

DO 1 J=1,NVAR

1 R(I,J)=0.

```

DO 2 L=1,NQBS
READ FMTX,(SCORE(I),I=1,NVAR)
DO 2 I=1,NVAR
AVG(I)=AVG(I)+SCORE(I)
DO 2 J=1,NVAR
2 R(I,J)=R(I,J)+SCORE(I)*SCORE(J)
GO TO 6
3 DO 4 I=1,NVAR
4 READ FMTR,(R(I,J),J=1,NVAR)
DO 5 I=1,NVAR
DO 5 J=I,NVAR
5 R(J,I)=R(I,J)
6 NFCOMM=0
IF(SS2 .EQ. 1) GO TO 10
IF(SS2 .EQ. 2) READ 103,(FMTC(I),I=1,18)
C $IF(SS2 .EQ. 2) READ 103,(FMTC(I),I=1,12)
IF(SS2 .EQ. 3) READ 104,NFCOMM
104 FORMAT(14)
IF(SS2 .EQ. 2) READ FMTC,(COMM(I),I=1,NVAR)
IF(SS2 .EQ. 3) GO TO 7
GO TO 10
7 SS2=2
DO 8 I=1,NVAR
8 COMM(I)=0.
DO 9 L=1,NFCOMM
READ 105,LDUM,(FMAT(I),I=1,NVAR)
105 FORMAT(2X,I3,3X,6F12.8/(8X,6F12.8))
DO 9 I=1,NVAR
9 COMM(I)=COMM(I)+FMAT(I)**2
10 RETURN
END

```

```

SUBROUTINE OUTFA1(TITLE,SS1,SS2,NVAR,NOFACT,NOSIG,Avg,Sigma,
1KR,R,DETR,PIVOT,SMC,TRACE,EIGVAL,EVDIFF,EVSUM,PERCT,NOITR,CHISQ,
2DEV,FMAT,COMM)
C THIS SUBROUTINE PRINTS AND PUNCHES THE OUTPUT FROM FACTOR ANALYSIS I
DIMENSION R(100,100),FMAT(100,21),AVG(100),SIGMA(100),SMC(100),
1COMM(100),EIGVAL(21),EVSUM(21),EVDIFF(21),PERCT(21),NOITR(21),
2CHISQ(21),DEV(21),TITLE(14),KR(100),AST(60),IOUT(3,21),TSIGN(21),
3PIVOT(100)
INTEGER SS1,SS2
DOUBLE PRECISION TITLE,AST
C $DOUBLE PRECISION TITLE
DATA AST/60*6H*****/
DATA PLUS,TMINUS/2H .,2H-./
PRINT 100,(AST(I),I=1,44),(TITLE(I),I=1,14),(AST(I),I=1,44)
100 FORMAT(1H1/1H7/1H2,2(21A6,A5/IX)/26X,13A6,A2//1X,?(21A6,A5/IX))
PRINT 101,NVAR,NOBS
101 FORMAT(1H1,21X,19HNO. OF VARIABLES = ,I3/22X,22HNO. OF OBSERVATION
1S = ,I4//)
IF(SS1 .EQ. 1) PRINT 102,(1,Avg(I),Sigma(I),I=1,NVAR)
102 FORMAT(16X,8HVAR. NO.,6X,4HMEAN,8X,9HSTD. DEV.//(19X,I3,F16.8,
1F15.8))
PRINT 103
103 FORMAT(1H1,40X,40HCORRELATION MATRIX (UPPER DIAG. BY ROWS)//1X,
13HROW,3X,1HK,47X,7HCOLUMNS/)
IF((SS1 .EQ. 1) .AND. (SS2 .EQ. 1)) PUNCH 104,(AST(I),I=1,8)
104 FORMAT(3A6,A2,40HCORRELATION MATRIX (UPPER DIAG. BY ROWS),3A6,A2)
DO 3 I=1,NVAR
R(I,I)=1.0
IF(((NVAR-I)-9) .GE. 0) GO TO 1
I10=NVAR
GO TO 2
1 I10=I+9
2 PRINT 105,(K,K=I,I10)
105 FORMAT(/6X,10I12)
PRINT 106,I,KR(I),(R(I,J),J=I,NVAR)
106 FORMAT(/1X,I3,I4,2X,10F12.8/(10X,10F12.8))
IF((SS1 .EQ. 1) .AND. (SS2 .EQ. 1)) PUNCH 107,I,(R(I,J),J=I,NVAR)
107 FORMAT(2X,I3,3X,6F12.8/(8X,6F12.8))
3 CONTINUE
PRINT 108,DETR,(I,PIVOT(I),I=1,NVAR)
108 FORMAT(1H1,20X,36HDETERMINANT OF REFL. CORR. MATRIX = ,E17.9///,
11X,26X,4HVAR.,7X,15HPIVOT ELEMENTS//(1X,I29,E23.8))
IF(SS2 .EQ. 1) PRINT 109,(I,SMC(I),I=1,NVAR)
109 FORMAT(1H1,26X,4HVAR.,5X,19HSQUARED MULT. CORR.//(1X,I29,F20.8))
PRINT 110,TRACE
110 FORMAT(1H1,24X,7HTRACE =,F19.8)
PRINT 111
111 FORMAT(/5X,6HFACTR,5X,10HEIGENVALUE,5X,16HEIGENVALUE DIFF.,5X,
114HFIGENVALUE SUM,3X,16HPERCENT VARIANCE,5X,10HITERATIONS//)
DO 5 I=1,NOFACT
IF(I .EQ. 1) GO TO 4
PRINT 112,EVDIFF(I)
112 FORMAT(27X,F15.8)
4 PRINT 113,I,EIGVAL(I),EVSUM(I),PERCT(I),NOITR(I)
113 FORMAT(1X,I9,F17.8,17X,2F19.8,I14)
5 CONTINUE
PRINT 114,(K,CHISQ(K),DEV(K),K=1,NOFACT)
114 FORMAT(1H1,15X,6HFACTR,7X,5HCHISQ,10X,9HDEVIATION//(1X,I19,F18.8,

```

```

1F16.8))
PRINT 115,(L,L=1,NOFACT)
115 FORMAT(1H1,28X,14HFACTOR MATRIX/1X,3HROW/3X,9(5X,I1),12(4X,I2))
PRINT 120
PUNCH 116,(AST(I),I=1,12)
116 FORMAT(5A6,A3,14HFACTOR MATRIX,5A6,A3)
DO 6 J=1,NOFACT
6 PUNCH 117,J,(FMAT(I,J),I=1,NVAR)
117 FORMAT(2X,I3,3X,6F12.8/(8X,6F12.8))
DO 9 I=1,NVAR
DO 9 J=1,NOFACT
IF(FMAT(I,J)) 7,9,8
7 FMAT(I,J)=FMAT(I,J)-.0005
GO TO 9
8 FMAT(I,J)=FMAT(I,J)+.0005
9 CONTINUE
DO 13 I=1,NVAR
DO 12 J=1,NOFACT
IF(FMAT(I,J) .GE. 0.) GO TO 10
TSIGN(J)=TMINUS
GO TO 11
10 TSIGN(J)=PLUS
11 ABSFT=ABS(FMAT(I,J))
IOUT(1,J)=ABSFT*10.
IOUT(2,J)=ABSFT*100.-FLOAT(IOUT(1,J))*10.
IOUT(3,J)=ABSFT*1000.-FLOAT(IOUT(1,J))*100.-FLOAT(IOUT(2,J))*10.
12 CONTINUE
13 PRINT 118,I,(TSIGN(J),(IOUT(K,J),K=1,3),J=1,NOFACT)
118 FORMAT(1X,I3,21(1X,A2,3I1))
PRINT 119,NOSIG,(I,COMM(I),I=1,NVAR)
119 FORMAT(1H1,15X,28HFACTOR MATRIX COMM. BASED ON,I3,20H SIGNIFICANT
1FACTORS//28X,3HROW,8X,13HCOMMUNALITIES//(1X,I30,F19.8))
120 FORMAT(1X)
RETURN
END

```

Factor Analysis II Computer Program

```

C   FACTOR ANALYSIS II
C   THIS PROGRAM COMPUTES A VARIMAX MATRIX AND SIMPLE STRUCTURE MATRICES
C   SPONSOR - DR. CURTON
C   PROGRAMMER - RICHARD C. DURFEE
C   THIS PROGRAM CONSISTS OF THREE SECTIONS
C       1. VARIMAX (ORTHOGONAL) ROTATION OF REDUCED FACTOR MATRIX
C       2. SPECIAL SIMPLE STRUCTURE (OBLIQUE) WEIGHTED ROTATION
C          OF REDUCED FACTOR MATRIX
C       3. FIVE SIMPLE STRUCTURE (OBLIQUE) ROTATIONS OF REDUCED
C          FACTOR MATRIX
C
C   NVAR = NUMBER OF VARIABLES (VARIABLE NUMBER = ROW NUMBER)
C   NFACT = NUMBER OF FACTORS IN THE REDUCED FACTOR MATRIX
C   FMAT = REDUCED FACTOR MATRIX FROM FACTOR ANALYSIS I
C   FCOMM = FACTOR MATRIX COMMUNALITIES
C   VARMAT = UNORDERED VARIMAX MATRIX
C   VCÖMM = VARIMAX MATRIX COMMUNALITIES
C   VARORD = NUMERICALLY ORDERED VARIMAX MATRIX
C   VTRAN = VARIMAX TRANSFORMATION MATRIX (ROTATED MATRIX = FMAT * TRAN)
C   SSMAT = UNORDERED SIMPLE STRUCTURE ROTATED MATRIX
C   SSORD = NUMERICALLY ORDERED SIMPLE STRUCTURE ROTATED MATRIX
C   SSTRAN = SIMPLE STRUCTURE TRANSFORMATION MATRIX
C   MINDEX = REORDERED VARIABLE NUMBERS OF THE ORDERED VARIMAX MATRIX
C   INDEX = REORDERED VARIABLE NUMBERS OF THE ORDERED SIMPLE STRUCTURE MATRIX
C   ANGTOL = ANGLE TOLERANCE = 1/4 DEGREE = .0043833231 RADIANS
C   TITLE = ALPHANUMERIC TITLE IDENTIFICATION
C   HICUT = HIGH CUTTING POINT (ALL FACTOR LOADINGS .GT. HICUT ARE DELETED)
C   BLOCUT = LOW CUTTING POINT (ALL FACTOR LOADINGS .LT. BLOCUT ARE DELETED)
C   NOSTRT = VARIABLE NUMBER OF LARGEST LOADING REMAINING IN HYPERPLANE
C   NOSTOP = VARIABLE NUMBER OF SMALLEST LOADING REMAINING IN HYPERPLANE
C   NOIN = TOTAL NUMBER OF ROWS REMAINING IN HYPERPLANE FOR EACH FACTOR
C   INROWS = ROW NUMBERS OF FACTOR LOADINGS REMAINING IN HYPERPLANE
C   WEIGHT = ROW WEIGHTS FOR EACH OF THE INROWS
C   DMAT = SUBSET OF FACTOR MATRIX (COMPOSED OF REMAINING INROWS)
C   INPUT CONSISTS OF
C
C           VARIABLES                   FORMAT
C
C           1. TITLE                   FORMAT(13A6,A2)
C           2. NVAR,NFACT              FORMAT(2I4)
C           3. FMAT                     FORMAT(102)
C
C   DIMENSION FMAT(100,14),FCOMM(100),VARMAT(100,14),VCÖMM(100),
C   1VARORD(100,14),VTRAN(14,14),SSMAT(100,14),SSORD(100,14),
C   2SSTRAN(14,14),TITLE(14),KROW(100),KVCOL(14),KTCOL(14),
C   3NORDT(50),H(100),INDEX(100,14),HIHCUT(14),NOSTRT(14),NOSTOP(14),
C   4NDIN(14),INROWS(100,14),WEIGHT(100,14),MINDEX(100,14)
C   DOUBLE PRECISION TITLE
C **** BEGIN INPUT
150 READ 100,(TITLE(I),I=1,14)
100 FORMAT(13A6,A2)
      READ 101,NVAR,NFACT
101 FORMAT(2I4)
      DO 1 L=1,NFACT
1      READ 102,LDUM,(FMAT(I,L),I=1,NVAR)
102 FORMAT(2X,1B,3X,6F12.8/(8X,6F12.8))
C **** END INPUT
C **** BEGIN SECTION 1 - VARIMAX ROTATION OF REDUCED FACTOR MATRIX

```

```

C**** BEGIN FACTOR MATRIX REFLECTION
DO 3 I=1,NVAR
KROW(I)=1
IF(FMAT(I,I) .GE. 0.) GO TO 3
KROW(I)=-1
DO 2 L=1,NOFACT
2 FMAT(I,L)=-FMAT(I,L)
3 CONTINUE
C**** END FACTOR MATRIX REFLECTION
C**** BEGIN CALCULATION OF FACTOR MATRIX COMMUNALITIES
DO 4 I=1,NVAR
FCOMM(I)=0.
DO 4 L=1,NOFACT
4 FCOMM(I)=FCOMM(I)+FMAT(I,L)**2
C**** END CALCULATION OF FACTOR MATRIX COMMUNALITIES
C**** BEGIN ORTHOGONAL ROTATION
ANGTOL=.43833231E-02
DO 5 I=1,NVAR
DO 5 L=1,NOFACT
5 VARMAT(I,L)=FMAT(I,L)
VARNU=NVAR
NITROT=0
DO 7 I=1,NOFACT
DO 6 J=1,NOFACT
6 VTRAN(I,J)=0.
7 VTRAN(I,I)=1.
DO 8 I=1,NVAR
H(I)=SQRT(FCOMM(I))
DO 8 L=1,NOFACT
8 VARMAT(I,L)=VARMAT(I,L)/H(I)
9 NITROT=NITROT+1
NOROT(NITROT)=0
IN=2
NC1=NOFACT-1
DO 20 J=1,NC1
DO 19 K=IN,NOFACT
SUM3=0.
SUM4=0.
SUM5=0.
SUM6=0.
DO 10 I=1,NVAR
AJ=VARMAT(I,J)
AK=VARMAT(I,K)
DIFFX=AJ*AJ-AK*AK
PRODX=2.*AJ*AK
SUM3=SUM3+(DIFFX+PRODX)*(DIFFX-PRODX)
SUM4=SUM4+DIFFX
SUM5=SUM5+PRODX
10 SUM6=SUM6+2.*PRODX*DIFFX
CNUM=SUM6-2.*SUM4*SUM5/VARNO
DEN=SUM3-((SUM4+SUM5)*(SUM4-SUM5))/VARNO
FRACT=CNUM/DEN
PHI=.25*(AR SIN(FRACT/SQRT(1.+FRACT**2)))
ANG=ABS(PHI)
IF(ANG-ANGTOL) 19,11,11
11 NUROT(NITROT)=NOROT(NITROT)+1
PCOS=COS(PHI)
PSIN=SIN(PHI)
IF(DEN) 13,13,12
12 PSIN=ABS(PSIN)

```

```

      GO TO 14
13   C1=PCOS+PSIN
      C2=PCOS-PSIN
      PCOS=.707107*C1
      PSIN=.707107*C2
14   IF(CNUM) 15,15,16
15   PSIN=-PSIN
16   DO 17 I=1,NVAR
      AK=VARMAT(I,K)
      AJ=VARMAT(I,J)
      VARMAT(I,K)=AJ*(-PSIN)+AK*PCOS
17   VARMAT(I,J)=AJ*PCOS+AK*PSIN
      DO 18 I=1,NOFACT
      TK=VTRAN(I,K)
      TJ=VTRAN(I,J)
      VTRAN(I,J)=TJ*PCOS+TK*PSIN
18   VTRAN(I,K)=TJ*(-PSIN)+TK*PCOS
19   CONTINUE
20   IN=IN+1
      IF(NUROT(NITROT) .NE. 0) GO TO 9
      DO 21 I=1,NVAR
      DO 21 L=1,NOFACT
21   VARMAT(I,L)=VARMAT(I,L)*H(I)
C*****END ORTHOGONAL ROTATION
C*****BEGIN VARIMAX REFLECTION ON NEGATIVE ELEMENT IN FIRST ROW OF TRAN. MATRIX
      DO 24 L=1,NOFACT
      KTCOL(L)=1
      IF(VTRAN(1,L) .GE. 0.) GO TO 24
      KTCOL(L)=-1
      DO 22 I=1,NVAR
22   VARMAT(I,L)=-VARMAT(I,L)
      DO 23 I=1,NOFACT
23   VTRAN(I,L)=-VTRAN(I,L)
24   CONTINUE
C*****END VARIMAX REFLECTION ON NEGATIVE ELEMENT IN FIRST ROW OF TRAN. MATRIX
C*****BEGIN CALCULATION OF VARIMAX MATRIX COMMUNALITIES
      DO 241 I=1,NVAR
      VCOMM(I)=0.
      DO 241 L=1,NOFACT
241  VCOMM(I)=VCOMM(I)+VARMAT(I,L)**2
C*****END CALCULATION OF VARIMAX MATRIX COMMUNALITIES
C*****BEGIN VARIMAX MATRIX REFLECTION ON NEGATIVE COLUMN SUM
      DO 28 L=1,NOFACT
      KVCOL(L)=1
      COLSUM=0.
      DO 25 I=1,NVAR
25   COLSUM=COLSUM+VARMAT(I,L)
      IF(COLSUM .GE. 0.) GO TO 28
      KVCOL(L)=-1
      DO 26 I=1,NVAR
26   VARMAT(I,L)=-VARMAT(I,L)
      DO 27 I=1,NOFACT
27   VTRAN(I,L)=-VTRAN(I,L)
28   CONTINUE
C*****END VARIMAX MATRIX REFLECTION ON NEGATIVE COLUMN SUM
C*****BEGIN NUMERICAL ORDERING OF VARIMAX MATRIX
      CALL ORDER(VARMAT,NVAR,NOFACT,VARORD,MINDEX)
C*****END NUMERICAL ORDERING OF VARIMAX MATRIX
C*****END SECTION 1 - VARIMAX ROTATION OF REDUCED FACTOR MATRIX
C*****=====

```

```

*****BEGIN SECTION 2 - SPECIAL SIMPLE STRUCTURE WEIGHTED ROTATION OF FMAT
*****BEGIN PREPARATION OF SIMPLE STRUCTURE TRANSFORMATION MATRIX
DO 31 L=1,NUFACT
DO 29 I=1,NUFACT
29 SSTRAN(I,L)=VTRAN(I,L)
SSTRAN(1,L)=.75*SSTRAN(1,L)
SUMSQ=0.
DO 30 I=1,NUFACT
30 SUMSQ=SUMSQ+SSTRAN(I,L)**2
SUM=SQRT(SUMSQ)
DO 31 I=1,NUFACT
31 SSTRAN(I,L)=SSTRAN(I,L)/SUM
*****END PREPARATION OF SIMPLE STRUCTURE TRANSFORMATION MATRIX
*****BEGIN PREPARATION OF SSORD MATRIX AND INDEX MATRIX
DO 33 L=1,NUFACT
DO 32 I=1,NVAR
INDEX(I,L)=MINDEX(I,I)
32 SSORD(I,L)=VARORD(I,L)
33 HIHCUT(L)=SSORD(1,L)/3.0
*****END PREPARATION OF SSORD MATRIX AND INDEX MATRIX
*****BEGIN DETERMINATION OF NOSTRT AND NOSTOP
DO 35 L=1,NUFACT
NOSTOP(L)=NVAR
DO 34 I=1,NVAR
IF(SSORD(I,L) .GT. HIHCUT(L)) GO TO 34
NOSTRT(L)=I
GO TO 35
34 CONTINUE
NOSTRT(L)=NVAR
35 CONTINUE
*****END DETERMINATION OF NOSTRT AND NOSTOP
*****BEGIN CALCULATION OF NOIN AND INROWS
DO 36 I=1,NUFACT
NSTART=NOSTRT(L)
NSTOP=NOSTOP(L)
NOIN(L)=NSTOP-NSTART+1
IKNT=0
DO 36 I=NSTART,NSTOP
IKNT=IKNT+1
36 INROWS(IKNT,L)=INDEX(I,L)
*****END CALCULATION OF NOIN AND INROWS
*****BEGIN CALCULATION OF ROW WEIGHTS
DO 37 L=1,NUFACT
NSTART=NOSTRT(L)
NSTOP=NOSTOP(L)
DO 37 I=NSTART,NSTOP
IKNT=INDEX(I,L)
IF(SSORD(I,L) .GT. .1) WEIGHT(IKNT,L)=1.
IF((SSORD(I,L) .LE. .1) .AND. (SSORD(I,L) .GE. (-.1))) WEIGHT(IKNT,L)=2.
IF(SSORD(I,L) .LT. (-.1)) WEIGHT(IKNT,L)=3.
37 CONTINUE
*****END CALCULATION OF ROW WEIGHTS
*****BEGIN OBLIQUE ROTATION
CALL ROTATE(FMAT,SSTRAN,NVAR,NUFACT,NOIN,INROWS,WEIGHT,SSMAT,
1 SSORD,INDEX)
*****END OBLIQUE ROTATION
*****END SECTION 2 - SPECIAL SIMPLE STRUCTURE WEIGHTED ROTATION OF FMAT
*****END *****
*****BEGIN OUTPUT

```

```

      CALL OUTFA2(TITLE,FCOMM,VCOMM,KROW,KCOL,KTCOL,NITROT,NOROT,
      IVARORD,MINDEX,VTRAN,NUSTRT,NOSTOP,SSORD,INDEX,SSTRAN,NVAR,NOFACT)
C*****END OUTPUT
C*****BEGIN SECTION 3 - FIVE SIMPLE STRUCTURE ROTATIONS OF REDUCED FACTOR MATRIX
      DO 73 ITR=1,5
C*****BEGIN DETERMINATION OF NUSTRT AND NOSTOP
      GO TO (38,38,38,63,64),ITR
C*****BEGIN PART 1 - JUDGEMENT CALCULATION OF REMAINING ROWS FOR ITR.S 1,2,3
C      LEAVE IN AT LEAST NOFACT ROWS
C      LEAVE OUT AT LEAST TOP TWO ROWS
C      LEAVE OUT ALL LOADINGS ABOVE +.50
      38 DO 62 L=1,NOFACT
      NVAR2=NVAR-2
      NSTART=2
      NSTOP=NVAR
      C      NO LOWER CUT ALLOWED ON ITR. 1
      IF(ITR .EQ. 1) GO TO 44
C*****BEGIN SUBPART 1 - CALCULATION OF LOWER CUT FOR ITR.S 2,3
C      MAKE LOWER CUT ABOVE ANY NEGATIVE LOADINGS NUMERICALLY LARGER THAN
C      THE LARGEST POSITIVE LOADING
      DO 40 I=1,NVAR2
      J=(NVAR+1)-I
      IF(SSORD(1,L)+SSORD(J,L)) 39,41,41
      39 NSTOP=J-1
      40 CONTINUE
      C      MAKE LOWER CUT FOR LOADINGS BELOW -.20 IF LOADING JUST BELOW CUT IS AT
      C      LEAST TWICE AS GREAT (NUMERICALLY) AS LOADINGS JUST ABOVE CUT
      41 DO 42 I=3,NSTOP
      IF(SSORD(I,L) .GT. (-.20)) GO TO 42
      IF(ABS(SSORD(I,L))-SSORD(I-1,L)) .LT. ABS(SSORD(I-1,L))) GO TO 42
      NSTOP=I-1
      GO TO 43
      42 CONTINUE
      C      LEAVE IN NOFACT ROWS
      43 IF((NSTOP-2) .LT. NOFACT) NSTOP=NVAR
C*****END SUBPART 1 - CALCULATION OF LOWER CUT FOR ITR.S 2,3
C*****BEGIN SUBPART 2 - CALCULATION OF PRELIMINARY UPPER CUT FOR ITR.S 1,2,3
      44 DO 46 I=3,NSTOP
      J=NSTOP-(I-3)
      C      MAKE UPPER CUT ABOVE THE ZERO LOADING
      IF(SSORD(J,L) .LT. 0.) GO TO 46
      C      LEAVE IN AT LEAST NOFACT ROWS
      IF((NSTOP-J+1) .LT. NOFACT) GO TO 46
      C      DO NOT MAKE UPPER CUT ABOVE +.50
      IF(SSORD(J,L) .LT. .50) GO TO 45
      NSTART=J+1
      GO TO 47
      C      MAKE FIRST LOADING (IN) AS POSITIVE AS LAST LOADING (IN) IS NEGATIVE
      45 IF(SSORD(J,L) .LT. ABS(SSORD(NSTOP,L))) GO TO 46
      NSTART=J
      GO TO 47
      46 CONTINUE
C*****END SUBPART 2 - CALCULATION OF PRELIMINARY UPPER CUT FOR ITR.S 1,2,3
C*****BEGIN SUBPART 3 - CALCULATION OF FINAL UPPER CUT FOR ITR.S 1,2,3
C      CALCULATE SUM OF SQUARES OF POSITIVE LOADINGS (SSQ+) AND NEGATIVE LOADINGS
C      (SSQ-) BETWEEN LOWER CUT AND PRELIMINARY UPPER CUT
      47 SUMPOS=0.
      SUMNEG=0.
      DO 49 I=NSTART,NSTOP

```

```

IF(SSORD(I,L) .LT. 0.) GO TO 48
SUMPOS=SUMPOS+SSORD(I,L)**2
GO TO 49
48 SUMNEG=SUMNEG+SSORD(I,L)**2
49 CONTINUE
C CALCULATE DIFF =(SSQ+) - (SSQ-)
DIFF=SUMPOS-SUMNEG
C FIND THE LARGEST GAP FOR ALL POSITIVE LOADINGS (TOP 2 OUT) = GAPMAX
GAPMAX=0.
IROWST=3
NOABF=NSTOP-NOFACT+1
DO 50 I=3,NOABF
IF(SSORD(I,L) .LT. 0.) GO TO 51
GAP=SSORD(I-1,L)-SSORD(I,L)
IF(GAPMAX .GT. GAP) GO TO 50
GAPMAX=GAP
IROWST=I
50 CONTINUE
51 IF(DIFF .GT. 0.) GO TO 53
C IF SSQ+ LESS THAN SSQ- GO UP NVAR/4 LOADINGS AND LOOK FOR LARGEST GAP
HIGAP=0.
IROWHI=3
NOUP=NVAR/4
NOST=NSTART-NOUP
IF(NOST .LT. 3) NOST=3
DO 52 I=NOST,NSTART
GAP=SSORD(I-1,L)-SSORD(I,L)
IF(HIGAP .GT. GAP) GO TO 52
HIGAP=GAP
IROWHI=I
52 CONTINUE
C IF SSQ+ GREATER THAN SSQ- LOOK FOR LARGEST GAP FROM NVAR/8 LOADINGS BELOW
C PRELIMINARY CUT UP TO NVAR/6 LOADINGS ABOVE PRELIMINARY CUT
53 NODN=NVAR/8
NOSP=NSTART-NODN
IF((NSTOP-NOSP+1) .LT. NOFACT) NOSP=NSTOP-NOFACT+1
NOUP=NVAR/6
NOST=NSTART-NOUP
IF(NOST .LT. 3) NOST=3
IF(SSORD(NOSP,L) .GE. 0.) GO TO 56
I=NOSP
54 I=I-1
IF(I .GE. 3) GO TO 55
NOSP=3
GO TO 56
55 IF(SSORD(I,L) .LT. 0.) GO TO 54
NOSP=I
56 HIGAP=0.
IROWHI=3
DO 57 I=NOST,NOSP
GAP=SSORD(I-1,L)-SSORD(I,L)
IF(HIGAP .GT. GAP) GO TO 57
HIGAP=GAP
IROWHI=I
57 CONTINUE
C IF GAPMAX IS GREATER THAN 2*HIGAP THEN MAKE CUT AT GAPMAX - OTHERWISE
C MAKE CUT AT HIGAP
IF(GAPMAX .GT. 2.*HIGAP) GO TO 58
NSTART=IROWHI
GO TO 59

```

```

58  NSTART=IROWST
C   MAKE FINAL UPPER CUT BELOW +.50
59  IF(SSORD(NSTART,L) .LT. .50) GO TO 61
I=NSTART
60  I=I+1
IF(SSORD(I,L) .GE. .50) GO TO 60
NSTART=I
61  NOSTRT(L)=NSTART
62  NOSTOP(L)=NSTOP
GO TO 70
*****END SUBPART 3 - CALCULATION OF FINAL UPPER CUT FOR ITR.S 1,2,3
*****END PART 1 - JUDGEMENT CALCULATION OF REMAINING ROWS FOR ITR.S 1,2,3
*****BEGIN PART 2 - CALC. OF REMAINING ROWS USING HICUT,BLOCUT FOR ITR.S 4,5
63  HICUT=.15
BLOCUT=-.30
GO TO 65
64  HICUT=.10
BLOCUT=-.20
65  DO 69 L=1,NOFACT
DO 66 I=1,NVAR
IF(SSORD(I,L) .GT. HICUT) GO TO 66
NOSTRT(L)=I
GO TO 67
66  CONTINUE
NOSTRT(L)=NVAR
67  NSTART=NOSTRT(L)
DO 68 I=NSTART,NVAR
IF(SSORD(I,L) .GT. BLOCUT) GO TO 68
NOSTOP(L)=I-1
GO TO 69
68  CONTINUE
NOSTOP(L)=NVAR
69  CONTINUE
*****END PART 2 - CALC. OF REMAINING ROWS USING HICUT,BLOCUT FOR ITR.S 4,5
*****END DETERMINATION OF NOSTRT AND NOSTOP
*****BEGIN CALCULATION OF NOIN AND INROWS
70  DO 71 L=1,NOFACT
NSTART=NOSTRT(L)
NSTOP=NOSTOP(L)
NUIN(L)=NSTOP-NSTART+1
IKNT=0
DO 71 I=NSTART,NSTOP
IKNT=IKNT+1
71  INROWS(IKNT,L)=INDEX(I,L)
*****END CALCULATION OF NOIN AND INROWS
*****BEGIN CALCULATION OF ROW WEIGHTS
DO 72 L=1,NOFACT
NSTART=NOSTRT(L)
NSTOP=NOSTOP(L)
DO 72 I=NSTART,NSTOP
IKNT=INDEX(I,L)
72  WEIGHT(IKNT,L)=1.
*****END CALCULATION OF ROW WEIGHTS
*****BEGIN OBLIQUE ROTATION
CALL ROTATE(FMAT,SSTRAN,NVAR,NOFACT,NOIN,INROWS,WEIGHT,SSMAT,
SSORD,INDEX)
*****END OBLIQUE ROTATION
*****BEGIN OUTPUT
CALL OUTSEQ(NoFACT,NOSTRT,NOSTOP)
CALL OUTMAT(SSORD,INDEX,SSTRAN,NVAR,NOFACT,36,36HSIMPLE STRUCTURE

```

1 ROTATED MATRIX , (TR)

C*****END OUTPUT

73 CONTINUE

C*****END SECTION 3 - FIVE SIMPLE STRUCTURE ROTATIONS OF REDUCED FACTOR MATRIX

C**********

GO TO 150

END.

```

SUBROUTINE ROTATE(FMAT,SSTRAN,NVAR,NOFACT,NOIN,INROWS,WEIGHT,
1SSMAT,SSORD,INDEX)
C THIS SUBROUTINE PERFORMS AN OBLIQUE ROTATION ON THE FACTOR MATRIX, FMAT,
C AND LEAVES THE NUMERICALLY ORDERED ROTATED MATRIX IN SSORD (WEIGHTS USED)
C DMAT = SUBSET OF FACTOR MATRIX (COMPOSED OF REMAINING INROWS)
C WTRMAT = TRANSPOSE OF (WEIGHT * DMAT)
C EMAT = WTRMAT * DMAT
C UMAT = SOLUTION TO THE EQUATION, EMAT * UMAT = SSTRAN
C SSTRAN = NORMALIZED UMAT
C SSMAT = FMAT * SSTRAN
C SSORD = NUMERICALLY ORDERED SSMAT
DIMENSION FMAT(100,14),SSTRAN(14,14),NOIN(14),INROWS(100,14),
1WEIGHT(100,14),SSMAT(100,14),SSORD(100,14),INDEX(100,14)
DIMENSION DMAT(100,14),WTRMAT(14,100),EMAT(14,15),UMAT(14,15)

C*****START ROTATION
DO 10 L=1,NOFACT
NIN=NOIN(L)

C*****BEGIN CALCULATION OF DMAT AND WTRMAT
DO 1 I=1,NIN
INROW=INROWS(I,L)
DO 1 J=1,NOFACT
DMAT(I,J)=FMAT(INROW,J)
1 WTRMAT(J,I)=WEIGHT(INROW,L)*DMAT(I,J)

C*****END CALCULATION OF DMAT AND WTRMAT
C*****BEGIN CALCULATION OF EMAT
DO 2 I=1,NOFACT
DO 2 J=1,NOFACT
EMAT(I,J)=0.
DO 2 K=1,NIN
2 EMAT(I,J)=EMAT(I,J)+WTRMAT(I,K)*DMAT(K,J)

C*****END CALCULATION OF EMAT
C*****BEGIN CALCULATION OF SOLUTION VECTOR, UMAT
DO 3 I=1,NOFACT
UMAT(I,NOFACT+1)=SSTRAN(I,L)
DO 3 J=1,NOFACT
3 UMAT(I,J)=EMAT(I,J)
N1=NOFACT+1
DO 6 K=1,NOFACT
K1=K+1
DO 4 J=K1,N1
4 UMAT(K,J)=UMAT(K,J)/UMAT(K,K)
DO 6 I=1,NOFACT
IF(I .EQ. K) GO TO 6
DO 5 J=K1,N1
5 UMAT(I,J)=UMAT(I,J)-UMAT(I,K)*UMAT(K,J)
6 CONTINUE

C*****END CALCULATION OF SOLUTION VECTOR, UMAT
C*****BEGIN NORMALIZATION OF UMAT TO GIVE A NEW SSTRAN
SUMU2=0.
DO 7 I=1,NOFACT
7 SUMU2=SUMU2+UMAT(I,NOFACT+1)**2
SUMU2=SQRT(SUMU2)
DO 8 I=1,NOFACT
8 SSTRAN(I,L)=UMAT(I,NOFACT+1)/SUMU2

C****END NORMALIZATION OF UMAT TO GIVE A NEW SSTRAN
C*****BEGIN CALCULATION OF SSMAT
DO 9 I=1,NVAR
SSMAT(I,L)=0.

```

```
DO 9 J=1,NOFACT
 9  SSMAT(I,L)=SSMAT(I,L)+FMAT(I,J)*SSTRAN(J,L)
C****END CALCULATION OF SSMAT
10  CONTINUE
C****END ROTATION
C****BEGIN NUMERICAL ORDERING OF THE SIMPLE STRUCTURE ROTATED MATRIX
  CALL ORDER(SSMAT,NVAR,NOFACT,SSORD,INDEX)
C****END NUMERICAL ORDERING OF THE SIMPLE STRUCTURE ROTATED MATRIX
  RETURN
END
```

```
SUBROUTINE ORDER(ARRAY,NVAR,NUFACT,ARRORD,INDEX)
C THIS SUBROUTINE ARRANGES THE MATRIX, ARRAY, IN DESCENDING NUMERICAL ORDER
C AND LEAVES THE ORDERED MATRIX IN ARRORD WITH THE ORDERED INDICES IN INDEX
DIMENSION ARRAY(100,14),ARRORD(100,14),INDEX(100,14)
LOGICAL SWAP
NVAR1=NVAR-1
DO 5 L=1,NUFACT
DO 1 I=1,NVAR
INDEX(I,L)=I
1 ARRORD(I,L)=ARRAY(I,L)
2 SWAP=.FALSE.
DO 4 I=1,NVAR1
IF(ARRORD(I,L)-ARRORD(I+1,L)) 3,4,4
3 STORE=ARRORD(I,L)
ARRORD(I,L)=ARRORD(I+1,L)
ARRORD(I+1,L)=STORE
ISTORE=INDEX(I,L)
INDEX(I,L)=INDEX(I+1,L)
INDEX(I+1,L)=ISTORE
SWAP=.TRUE.
4 CONTINUE
IF(SWAP) GO TO 2
5 CONTINUE
RETURN
END
```

```

SUBROUTINE OUTFA2(TITLE,FCOMM,VCOMM,KROW,KTCOL,NITROT,NOROT,
1VARORD,MINDEX,VTRAN,NOSTRT,NOSTOP,SSORD,INDEX,SSTRAN,NVAR,NOFACT)
C THIS SUBROUTINE PRINTS AND PUNCHES THE OUTPUT FROM FACTOR ANALYSIS II
DIMENSION TITLE(14),FCOMM(100),VCOMM(100),KROW(100),KTCOL(14),
1KTCOL(14),NOROT(50),VARORD(100,14),MINDEX(100,14),VTRAN(14,14),
2NOSTRT(14),NOSTOP(14),SSORD(100,14),INDEX(100,14),SSTRAN(14,14),
3AST(60)
DOUBLE PRECISION TITLE,AST
C $DOUBLE PRECISION TITLE
DATA AST/60*6H*****/
PRINT 100,(AST(I),I=1,44),(TITLE(I),I=1,14),(AST(I),I=1,44)
100 FORMAT(1H1/1H7/1H2,2(21A6,A5/1X)/26X,13A6,A2//1X,2(21A6,A5/1X))
PRINT 101,(I,KROW(I)),FCOMM(I),VCOMM(I),I=1,NVAR)
101 FORMAT(1H1,15X,13HFACTOR MATRIX,5X,13HFACTOR MATRIX,5X,14HVARIMAX
1MATRIX/7X,3HROW,7X,10HREFLECTION,7X,13HCOMMUNALITIES,6X,13HCOMMUNA
2LITIES///(I10,I13,F22.8,F19.8))
PRINT 102,(L,KTCOL(L)),KVCOL(L),L=1,NOFACT)
102 FORMAT(//16X,32HVARIMAX MATRIX COLUMN REFLECTION//18X,8HNEGATIVE,
11OX,8HNEGATIVE/7X,4HCOL.,4X,14HTRANS. ELEMENT,6X,10HCOLUMN SUM//)
2(I10,I13,I18))
PRINT 103,(I,NOROT(I),I=1,NITROT)
103 FORMAT(//13X,19HNUMBER OF ROTATIONS/7X,4HITR.,5X,13HPER ITERATION
1///(I10,I13))
CALL OUTMAT(VARORD,MINDEX,VTRAN,NVAR,NOFACT,24,24HORDERED VARIMAX
1 MATRIX,0)
CALL OUTSEQ(NOFACT,NOSTRT,NOSTOP)
CALL OUTMAT(SSORD,INDEX,SSTRAN,NVAR,NOFACT,36,36HSPECIAL SIMPLE
1STRUCTURE MATRIX ,0)
PUNCH 104,(AST(I),I=1,8),((I,J,SSTRAN(I,J),J=1,NOFACT),I=1,NOFACT)
104 FORMAT(4A6,32HSPECIAL TRANSFORMATION MATRIX,4A6/(I22,I9,F19.8))
RETURN
END

```

```

SUBROUTINE OUTMAT(ARRAY, INDEX, TRAN, NVAR, NOFACT, NCHARS, HOLLER, ITR)
C THIS SUBROUTINE PRINTS THE ORDERED ARRAY WITH INDICES AND THE TRAN. MATRIX
DIMENSION ARRAY(100,14), INDEX(100,14), TRAN(14,14), HOLLER(10),
1DIGIT(5), TSIGN(14), IOUT(3,14)
DOUBLE PRECISION ITER,DIGIT
C $ITER AND DIGIT MUST BE DOUBLE PRECISION FOR THE IBM 360
DATA PLUS,TMINUS/2H+,2H-/
DATA ITER/6H(ITERNO/,DIGIT(1)/6H.=1) /,DIGIT(2)/6H.=2) /,
1DIGIT(3)/6H.=3) /,DIGIT(4)/6H.=4) /,DIGIT(5)/6H.=5) /
C NCHARS MUST BE A MULTIPLY OF 4(IBM 360) OR 6(IBM 7090,7040)
NWORDS=NCHARS/4
C $NWORDS=NCHARS/6
IF(ITR .EQ. 0) PRINT 100,(HOLLER(I),I=1,NWORDS)
IF(ITR .GT. 0) PRINT 100,(HOLLER(I),I=1,NWORDS),ITER,DIGIT(ITR)
100 FORMAT(1H1,21X,12A4)
C100 $FORMAT(1H1,21X,12A6)
1 PRINT 101,(L,L=1,NOFACT)
101 FORMAT(//5H SEQ.,14I9)
PRINT 102
102 FORMAT(1X)
DO 7 I=1,NVAR
DO 6 J=1,NOFACT
ELEM=0.
IF(ARRAY(I,J) .LT. 2,4,3
2 ELEM=ABS(ARRAY(I,J)-.0005)
TSIGN(J)=TMINUS
GO TO 5
3 ELEM=ABS(ARRAY(I,J)+.0005)
4 TSIGN(J)=PLUS
5 IOUT(1,J)=ELEM*10.
IOUT(2,J)=ELEM*100.-FLOAT(IOUT(1,J))*10.
IOUT(3,J)=ELEM*1000.-FLOAT(IOUT(1,J))*100.-FLOAT(IOUT(2,J))*10.
6 CONTINUE
7 PRINT 103,I,(INDEX(I,J),TSIGN(J),(IOUT(K,J),K=1,3),J=1,NOFACT)
1C3 FORMAT(1X,I3,2X,14(I3,1X,A2,3I1))
PRINT 104
104 FORMAT(///22X,24HTRANSFORMATION MATRIX)
PRINT 105,(L,L=1,NOFACT)
1C5 FORMAT(//5H ROW ,14I9)
PRINT 102
DO 13 I=1,NOFACT
DO 12 J=1,NOFACT
ELEM=0.
IF(TRAN(I,J) .LT. 8,10,9
8 ELEM=ABS(TRAN(I,J)-.0005)
TSIGN(J)=TMINUS
GO TO 11
9 ELEM=ABS(TRAN(I,J)+.0005)
10 TSIGN(J)=PLUS
11 IOUT(1,J)=ELEM*10.
IOUT(2,J)=ELEM*100.-FLOAT(IOUT(1,J))*10.
IOUT(3,J)=ELEM*1000.-FLOAT(IOUT(1,J))*100.-FLOAT(IOUT(2,J))*10.
12 CONTINUE
13 PRINT 106,I,(TSIGN(J),(IOUT(K,J),K=1,3),J=1,NOFACT)
1C6 FORMAT(1X,I3,2X,14(4X,A2,3I1))
RETURN
END

```

```
SUBROUTINE OUTSEQ(NOFACT,NOSTRT,NOSTOP)
C THIS SUBROUTINE PRINTS OUT THE SEQUENCE NUMBERS REMAINING IN HYPERPLANE
DIMENSION NOSTRT(14),NOSTOP(14)
PRINT 100,(L,NOSTR(L),NOSTOP(L),L=1,NOFACT)
100 FORMAT(//16X,23HSEQUENCE NO.S REMAINING/1X,6HFACTOR,12X,17HFOR NE
IXT ROTATION//(1X,I3,19X,I3,2H -,I3))
RETURN
END
```

Factor Analysis III Computer Program

```

C   FACTOR ANALYSIS III
C   THIS PROGRAM COMPUTES A SINGLE SIMPLE STRUCTURE MATRIX AND A COSINE MATRIX
C   SPONSOR - DR. CURETON
C   PROGRAMMER - RICHARD C. DURFEE
C   THIS PROGRAM CONSISTS OF TWO SECTIONS
C       1. SIMPLE STRUCTURE (OBLIQUE) ROTATION OF REDUCED FACTOR MATRIX
C       2. COSINE MATRIX FROM SIMPLE STRUCTURE TRANSFORMATION MATRIX
C   NVAR = NUMBER OF VARIABLES (VARIABLE NUMBER = ROW NUMBER)
C   NOFACT = NUMBER OF FACTORS IN THE REDUCED FACTOR MATRIX
C   FMAT = REDUCED FACTOR MATRIX FROM FACTOR ANALYSIS I
C   SSMAT = UNORDERED SIMPLE STRUCTURE ROTATED MATRIX
C   SSORD = NUMERICALLY ORDERED SIMPLE STRUCTURE ROTATED MATRIX
C   SSTRAN = SIMPLE STRUCTURE TRANSFORMATION MATRIX FROM FACTOR ANALYSIS II
C   INDEX = REORDERED VARIABLE NUMBERS OF THE ORDERED SIMPLE STRUCTURE MATRIX
C   NOIN = TOTAL NUMBER OF ROWS REMAINING IN HYPERPLANE FOR EACH FACTOR
C   INROWS = ROW NUMBERS OF FACTOR LOADINGS REMAINING IN HYPERPLANE
C   WEIGHT = ROW WEIGHTS FOR EACH OF THE INROWS
C   COSMAT = MATRIX OF COSINES OF ANGLES BETWEEN REFERENCE VECTORS
C   TITLE = ALPHANUMERIC TITLE IDENTIFICATION
C   DIMENSION FMAT(100,14),SSMAT(100,14),SSORD(100,14),SSTRAN(14,14),
C   INDEX(100,14),NOIN(14),INROWS(100,14),WEIGHT(100,14),TITLE(14),
C   2COSMAT(14,14),DIST(14)
C   DOUBLE PRECISION TITLE
C*****BEGIN INPUT
 150 CALL INFAB3(TITLE,NVAR,NOFACT,FMAT,SSTRAN,NOIN,INROWS,WEIGHT)
C*****END INPUT
C*****BEGIN OBLIQUE ROTATION
  CALL ROTATE(FMAT,SSTRAN,NVAR,NOFACT,NOIN,INROWS,WEIGHT,SSMAT,
  1SSORD,INDEX)
C*****END OBLIQUE ROTATION
C*****BEGIN CALCULATION OF COSINE MATRIX
    DO 2 L=1,NOFACT
      DIST(L)=0.
    DO 1 I=1,NOFACT
      DIST(L)=DIST(L)+SSTRAN(I,L)**2
  1 DIST(L)=SQRT(DIST(L))
    DO 4 L=1,NOFACT
      DO 4 I=L,NOFACT
        SUMLI=0.
        DO 3 J=1,NOFACT
          SUMLI=SUMLI+SSTRAN(J,L)*SSTRAN(J,I)
        COSMAT(L,I)=SUMLI/(DIST(L)*DIST(I))
  4 COSMAT(I,L)=COSMAT(L,I)
C*****END CALCULATION OF COSINE MATRIX
C*****BEGIN OUTPUT
  CALL OUTFA3(TITLE,SSORD,INDEX,SSTRAN,COSMAT,NVAR,NOFACT)
C*****END OUTPUT
  GO TO 150
  END

```

SUBROUTINE INFAB(TITLE,NVAR,NOFACT,FMAT,SSTRAN,NOIN,INROWS,WEIGHT)
C THIS SUBROUTINE READS AND SUPPLIES ALL INPUT NEEDED IN FACTOR ANALYSIS III
C SPONSOR - DR. CURETON
C PROGRAMMER - RICHARD C. DURFEE
C TITLE = ALPHANUMERIC TITLE IDENTIFICATION
C NVAR = NUMBER OF VARIABLES (VARIABLE NUMBER = ROW NUMBER)
C NOFACT = NUMBER OF FACTORS IN THE REDUCED FACTOR MATRIX
C FMAT = REDUCED FACTOR MATRIX FROM FACTOR ANALYSIS I
C SSTRAN = SIMPLE STRUCTURE TRANSFORMATION MATRIX FROM FACTOR ANALYSIS II
C NOIN = TOTAL NUMBER OF ROWS REMAINING IN HYPERPLANE FOR EACH FACTOR
C INROWS = ROW NUMBERS OF FACTOR LOADINGS REMAINING IN HYPERPLANE
C WEIGHT = ROW WEIGHTS FOR EACH OF THE INROWS
C INPUT CONSISTS OF

C	VARIABLES	FORMAT
C	1. TITLE	FORMAT(13A6,A2)
C	2. NVAR,NOFACT	FORMAT(2I4)
C	3. FMAT	FORMAT(102)
C	4. SSTRAN	FORMAT(31X,F19.8)
C	5. REMAIN CARD	FORMAT(FREE FORMAT EXPLAINED BELOW)
C	DELETE CARD	FORMAT(FREE FORMAT EXPLAINED BELOW)
C	WEIGHT CARD	FORMAT(FREE FORMAT EXPLAINED BELOW)
C	FINISH CARD	FORMAT(FREE FORMAT EXPLAINED BELOW)

C AN EXAMPLE REMAIN CARD MIGHT LOOK LIKE

C----REMAIN 3 11 **** 1 2 10 7 15 16 18 4 9 20 5

C WHERE THE WORD REMAIN IN COLUMNS 1-6 INDICATES THAT THIS CARD CONTAINS
C 11 VARIABLE NUMBERS (1,2,10,7,15,16,18,4,9,20,5) WHICH ARE TO REMAIN
C IN THE HYPERPLANE DURING THE ROTATION ON FACTOR 3.

C AN EXAMPLE DELETE CARD MIGHT LOOK LIKE

C----DELETE 3 9 ***** 3 6 11 13 12 8 14 17 19

C WHERE THE WORD DELETE IN COLUMNS 1-6 INDICATES THAT THIS CARD CONTAINS
C 9 VARIABLE NUMBERS (3,6,11,13,12,8,14,17,19) WHICH ARE TO BE DELETED
C FROM THE HYPERPLANE DURING THE ROTATION ON FACTOR 3.

C AN EXAMPLE WEIGHT CARD MIGHT LOOK LIKE

C----WEIGHT 3 4*** 1 2.0 10 1.5 15 3.2 5 4.0

C WHERE THE WORD WEIGHT IN COLUMNS 1-6 INDICATES THAT THIS CARD CONTAINS
C 4 VARIABLE NUMBERS (1,10,15,5) WHICH ARE TO BE WEIGHTED WITH THEIR
C RESPECTIVE WEIGHTS(2.0,1.5,3.2,4.0) DURING THE ROTATION ON FACTOR 3.

C AN EXAMPLE FINISH CARD SHOULD LOOK LIKE

C----FINISH

C WHERE THE WORD FINISH IN COLUMNS 1-6 INDICATES THAT THIS CARD IS
C THE LAST DATA CARD.

C*****

C ALL DATA ON THESE FOUR TYPES OF CARDS MAY BE FREE FORM AS LONG AS BLANKS
C OR ASTERISKS ARE PLACED BETWEEN THE NUMBERS. THE ONLY EXCEPTION INVOLVES
C COLUMNS 1-6 WHICH MUST CONTAIN ONE OF THE FOUR GIVEN WORDS. ALSO THE
C FACTOR NUMBER MUST OCCUR FIRST FOLLOWED BY THE TOTAL NUMBER OF VARIABLES
C TO BE READ. THE ASTERISKS ARE NOT MANDATORY BUT MAY BE USED FOR CLARITY.

C ANY CHARACTER IN COLUMNS 7-80 WHICH IS NOT A DIGIT, DECIMAL POINT,
 C BLANK OR ASTERISK WILL GIVE UNPREDICTABLE RESULTS. NOTE ALSO THAT WHEN
 C ROW WEIGHTS ARE USED THE DECIMAL POINT MUST BE INCLUDED IN THE WEIGHTS.
 C IF ALL THE DATA FOR A GIVEN FACTOR CANNOT BE PLACED ON ONE CARD THEN
 C THE DATA MAY BE CONTINUED IN COLUMNS 1-80 OF THE NEXT CARD. OF COURSE
 C A SINGLE NUMBER SHOULD NOT BE SPLIT BETWEEN TWO CARDS. NOTE IN THE
 C EXAMPLE ABOVE THAT THE REMAIN CARD AND THE DELETE CARD BOTH RESULT IN
 C THE SAME ROTATION FOR A 20 VARIABLE PROBLEM. THUS IT WOULD BE PREFERABLE
 C TO USE THE DELETE CARD SINCE FEWER VARIABLE NUMBERS WOULD NEED TO BE
 C PUNCHED.

C*****

DIMENSION TITLE(14),FMAT(100,14),SSTRAN(14,14),NOIN(14),
 1INROWS(100,14),WEIGHT(100,14),IMAGE(80),KWORD(4),ITEMP(100)

DOUBLE PRECISION TITLE

DATA KWORD/4HREMA,4HDELE,4HWEIG,4HFINI/

C \$DATA KWURD/6HREMAIN,6HDELETE,6HWEIGHT,6HFINISH/

DATA IMAGE/80*4H /

C \$DATA IMAGE/80*6H /

READ 100,(TITLE(I),I=1,14)

100 FORMAT(13A6,A2)

READ 101,NVAR,NOFACT

101 FORMAT(2I4)

DO 1 L=1,NOFACT

1 READ 102,LDUM,(FMAT(I,L),I=1,NVARI)

102 FORMAT(2X,I3,3X,6F12.8/(8X,6F12.8))

READ 103,((SSTRAN(I,J),J=1,NOFACT),I=1,NOFACT)

103 FORMAT(31X,F19.8)

C**** BEGIN REMAIN,DELETE,WEIGHT,FINISH CARD INPUT

DO 2 L=1,NOFACT

NOIN(L)=1

INROWS(1,L)=1

DO 2 I=1,NVAR

2 WEIGHT(I,L)=1.

3 READ 104,KTYPE,(IMAGE(I),I=7,80)

104 FORMAT(A4,2X,74A1)

C104 \$FORMAT(A6,74A1)

IPOINT=7

DO 4 I=1,4

IF(KTYPE .NE. KWORD(I)) GO TO 4

GO TO (5,7,11,13),I

4 CONTINUE

CALL BADONE(KTYPE,IMAGE)

RETURN

C**** BEGIN REMAIN CARD ANALYSIS

5 CALL GETNO(IPOINT,IMAGE,NFACT,DUMMY)

CALL GETNO(IPOINT,IMAGE,NOITEM,DUMMY)

NOIN(NFACT)=NOITEM

DO 6 I=1,NOITEM

6 CALL GETNU(IPOINT,IMAGE,INROWS(I,NFACT),DUMMY)

GO TO 3

C**** END REMAIN CARD ANALYSIS

C**** BEGIN DELETE CARD ANALYSIS

7 CALL GETNO(IPOINT,IMAGE,NFACT,DUMMY)

CALL GETNO(IPOINT,IMAGE,NOITEM,DUMMY)

NOIN(NFACT)=NVAR-NOITEM

DO 8 I=1,NOITEM

8 CALL GETND(IPOINT,IMAGE,ITEMP(I),DUMMY)

IKNT=0

DO 10 I=1,NVAR

DO 9 K=1,NOITEM

```
9 IF(ITEMP(K) .EQ. 1) GO TO 10
IKNT=IKNT+1
INROWS(IKNT,NFACT)=I
10 CONTINUE
GO TO 3
C*****END DELETE CARD ANALYSIS
C*****BEGIN WEIGHT CARD ANALYSIS
11 CALL GETNO(IPOINT,IMAGE,NFACT,DUMMY)
CALL GETNO(IPOINT,IMAGE,NOITEM,DUMMY)
DO 12 I=1,NOITEM
CALL GETNO(IPOINT,IMAGE,NROW,DUMMY)
12 CALL GETNO(IPOINT,IMAGE,1DUMMY,WEIGHT(NROW,NFACT))
GO TO 3
C*****END WEIGHT CARD ANALYSIS
C*****BEGIN FINISH CARD ANALYSIS
13 RETURN
C*****END FINISH CARD ANALYSIS
C*****END REMAIN,DELETE,WEIGHT,FINISH CARD INPUT
END
```

```

SUBROUTINE GETNO(IPPOINT, IMAGE, NVAL, WT)
C THIS SUBROUTINE GETS THE NEXT NUMBER FROM THE CARD IMAGE
DIMENSION IMAGF(80), MIGITS(10), NUMBER(10)
DATA MIGITS/40H1 2 3 4 5 6 7 8 9 0 /
C $DATA MIGITS/60H1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
C $1 0 /
DATA MLANK/4H    /, MECPT/4H,   /, MSTER/4H*   /
C $DATA MLANK/6H    /, MEGPT/6H,   /, MSTER/6H*   /
WT=0.
IF(IPPOINT .GT. 80) CALL INCREM(IPPOINT, IMAGE)
1 KCHAR=IMAGE(IPPOINT)
IF((KCHAR .NE. MLANK) .AND. (KCHAR .NE. MSTER)) GO TO 2
CALL INCREM(IPPOINT, IMAGE)
GO TO 1
2 NUMKNT=0
IPOWER=-1
3 IF((KCHAR .EQ. MLANK) .OR. (KCHAR .EQ. MSTER)) GO TO 9
IF(KCHAR .NE. MECPT) GO TO 5
IF(IPOWER .EQ. (-1)) GO TO 4
CALL BADONE(4H2DEC, IMAGE)
GO TO 11
4 IPOWER=NUMKNT+1
IPPOINT=IPPOINT+1
IF(IPPOINT .GT. 80) GO TO 9
KCHAR=IMAGE(IPPOINT)
5 IF(KCHAR .NE. MIGITS(10)) GO TO 6
NUMKNT=NUMKNT+1
NUMBER(NUMKNT)=0
GO TO 8
6 DO 7 I=1,9
IF(KCHAR .NE. MIGITS(I)) GO TO 7
NUMKNT=NUMKNT+1
NUMBER(NUMKNT)=I
GO TO 8
7 CONTINUE
CALL BADONE(4HNONE, IMAGE)
GO TO 11
8 IPPOINT=IPPOINT+1
IF(IPPOINT .GT. 80) GO TO 9
KCHAR=IMAGE(IPPOINT)
GO TO 3
9 NVAL=0
MULT=1
IUP=NUMKNT+1
DO 10 I=1,NUMKNT
IUP=IUP-1
NVAL=NUMBER(IUP)*MULT+NVAL
10 MULT=MULT#10
IF(IPOWER .EQ. (-1)) GO TO 11
PROD=10** (NUMKNT-IPOWER+1)
FNVAL=NVAL
WT=FNVAL/PROD
11 IPPOINT=IPPOINT+1
RETURN
END

```

```
SUBROUTINE INCREM(IPOINT,IMAGE)
C THIS SUBROUTINE INCREMENTS IPOINT AND READS NEXT CARD IF NECESSARY
DIMENSION IMAGE(80)
IF(IPOINT .LT. 80) GO TO 1
READ 100,(IMAGE(I),I=1,80)
100 FORMAT(80A1)
IPOINT=0
1 IPOINT=IPOINT+1
RETURN
END
```

```
SUBROUTINE BADONE(KTYPE, IMAGE)
C THIS SUBROUTINE PRINTS OUT AN ERROR MESSAGE BECAUSE OF A BAD CARD
DIMENSION IMAGE(80)
PRINT 100,KTYPE,(IMAGE(I),I=7,80)
100 FORMAT(1H1,77HBECAUSE OF THIS BAD DATA ON YOUR INPUT CARD ANY FURT
1HUR OUTPUT IS MEANINGLESS//A4,2X,74A1/1H1)
      RETURN
      END
```

```

SUBROUTINE ROTATE(FMAT,SSTRAN,NVAR,NOFACT,NOIN,INROWS,WEIGHT,
1SSMAT,SSORD,INDEX)
C THIS SUBROUTINE PERFORMS AN OBLIQUE ROTATION ON THE FACTOR MATRIX, FMAT,
C AND LEAVES THE NUMERICALLY ORDERED ROTATED MATRIX IN SSORD (WEIGHTS USED)
C DMAT = SUBSET OF FACTOR MATRIX (COMPOSED OF REMAINING INROWS)
C WTRMAT = TRANSPOSE OF (WEIGHT * DMAT)
C EMAT = WTRMAT * DMAT
C UMAT = SOLUTION TO THE EQUATION, EMAT * UMAT = SSTRAN
C SSTRAN = NORMALIZED UMAT
C SSMAT = FMAT * SSTRAN
C SSORD = NUMERICALLY ORDERED SSMAT
DIMENSION FMAT(100,14),SSTRAN(14,14),NOIN(14),INROWS(100,14),
1WEIGHT(100,14),SSMAT(100,14),SSORD(100,14),INDEX(100,14)
DIMENSION UMAT(100,14),WTRMAT(14,100),EMAT(14,15),UMAT(14,15)

*****START ROTATION
DO 10 L=1,NOFACT
NIN=NOIN(L)

*****BEGIN CALCULATION OF DMAT AND WTRMAT
DO 1 I=1,NIN
INROW=INROWS(I,L)
DO 1 J=1,NOFACT
DMAT(I,J)=FMAT(INROW,J)
1 WTRMAT(J,I)=WEIGHT(INROW,L)*DMAT(I,J)
*****END CALCULATION OF DMAT AND WTRMAT
*****BEGIN CALCULATION OF EMAT
DO 2 I=1,NOFACT
DO 2 J=1,NOFACT
EMAT(I,J)=0.
DO 2 K=1,NIN
2 EMAT(I,J)=EMAT(I,J)+WTRMAT(I,K)*DMAT(K,J)
*****END CALCULATION OF EMAT
*****BEGIN CALCULATION OF SOLUTION VECTOR, UMAT
DO 3 I=1,NOFACT
UMAT(I,NOFACT+1)=SSTRAN(I,L)
DO 3 J=1,NOFACT
3 UMAT(I,J)=EMAT(I,J)
N1=NOFACT+1
DO 6 K=1,NOFACT
K1=K+1
DO 4 J=K1,N1
4 UMAT(K,J)=UMAT(K,J)/UMAT(K,K)
DO 6 I=1,NOFACT
IF(I .EQ. K) GO TO 6
DO 5 J=K1,N1
5 UMAT(I,J)=UMAT(I,J)-UMAT(I,K)*UMAT(K,J)
6 CONTINUE
*****END CALCULATION OF SOLUTION VECTOR, UMAT
*****BEGIN NORMALIZATION OF UMAT TO GIVE A NEW SSTRAN
SUMU2=0.
DO 7 I=1,NOFACT
7 SUMU2=SUMU2+UMAT(I,NOFACT+1)**2
SUMU2=SQRT(SUMU2)
DO 8 I=1,NOFACT
8 SSTRAN(I,L)=UMAT(I,NOFACT+1)/SUMU2
*****END NORMALIZATION OF UMAT TO GIVE A NEW SSTRAN
*****BEGIN CALCULATION OF SSMAT
DO 9 I=1,NVAR
SSMAT(I,L)=0.

```

```
DO_9 J=1,NOFACT
9    SSMAT(I,L)=SSMAT(I,L)+FMAT(I,J)*SSTRAN(J,L)
C*****END CALCULATION OF SSMAT
10   CONTINUE
C*****END ROTATION
C*****BEGIN NUMERICAL ORDERING OF THE SIMPLE STRUCTURE ROTATED MATRIX
CALL ORDER(SSMAT,NVAR,NOFACT,SSORD,INDEX)
C*****END NUMERICAL ORDERING OF THE SIMPLE STRUCTURE ROTATED MATRIX
RETURN
END
```

C SUBROUTINE ORDER(ARRAY,NVAR,NUFACT,ARRORD,INDEX)
 C THIS SUBROUTINE ARRANGES THE MATRIX, ARRAY, IN DESCENDING NUMERICAL ORDER
 C AND LEAVES THE ORDERED MATRIX IN ARRORD WITH THE ORDERED INDICES IN INDEX.
 DIMENSTON ARRAY(100,14),ARRORD(100,14),INDEX(100,14)

LOGICAL SWAP

NVAR1=NVAR-1

DO 5 L=1,NUFACT

DO 1 I=1,NVAR

INDEX(I,L)=I

1 ARRORD(I,L)=ARRAY(I,L)

2 SWAP=.FALSE.

DO 4 I=1,NVAR1

IF(ARRORD(I,L)-ARRORD(I+1,L)) 3,4,4

3 STORE=ARRORD(I,L)

ARRORD(I,L)=ARRORD(I+1,L)

ARRORD(I+1,L)=STORE

ISTORE=INDEX(I,L)

INDEX(I,L)=INDEX(I+1,L)

INDEX(I+1,L)=ISTORE

SWAP=.TRUE.

4 CONTINUE

IF(SWAP) GO TO 2

5 CONTINUE

RETURN

END

```

C   SUBROUTINE QUTFAB(TITLE,SSORD,INDEX,SSTRAN,COSMAT,NVAR,NOFACT)
C   THIS SUBROUTINE PRINTS AND PUNCHES THE OUTPUT FROM FACTOR ANALYSIS III
C   DIMENSION TITLE(14),SSORD(100,14),INDEX(100,14),SSTRAN(14,14),
C   1COSMAT(14,14),AST(60),TSIGN(14),IOUT(3,14)
C   DOUBLE PRECISION TITLE,AST
C   $DOUBLE PRECISION TITLE
      DATA AST/60*6H*****/
      DATA PLUS,TMINUS,ONE/2H .,2H-.,2H1./
      PRINT 100,(AST(I),I=1,44),(TITLE(I),I=1,14),(AST(I),I=1,44)
100   FORMAT(1H1/1H7/1H2,2(21A6,A5/1X)/26X,13A6,A2//1X,2(21A6,A5/1X))
      CALL OUTMAT(SSORD,INDEX,SSTRAN,NVAR,NOFACT,36,36HSIMPLE STRUCTURE
1    ROTATED MATRIX ,0)
      PRINT 101
101   FORMAT(//22X,16HCOSINE MATRIX)
      PRINT 102,(L,L=1,NOFACT)
102   FORMAT(//5H ROW ,14I9)
      PRINT 103
103   FORMAT(1X)
      DO 6 I=1,NUFACT
      DO 5 J=1,NUFACT
         ELEM=0.
         IF(COSMAT(I,J)) 1,3,2
1     ELEM=ABS(COSMAT(I,J)-.0005)
         TSIGN(J)=TMINUS
         GO TO 4
2     ELEM=ABS(COSMAT(I,J)+.0005)
3     TSIGN(J)=PLUS
4     IF(ELEM .GE. 1.0) ELEM=.999
         IOUT(1,J)=ELEM*10.
         IOUT(2,J)=ELEM*100.-FLOAT(IOUT(1,J))*10.
         IOUT(3,J)=ELEM*1000.-FLOAT(IOUT(1,J))*100.-FLOAT(IOUT(2,J))*10.
5     CONTINUE
         TSIGN(I)=ONE
         IOUT(1,I)=0
         IOUT(2,I)=0
         IOUT(3,I)=0
6     PRINT 104,I,(TSIGN(J),(IOUT(K,J),K=1,3),J=1,NOFACT)
104   FORMAT(1X,13,2X,14(4X,A2,B1I))
      PUNCH 105,(AST(I),I=1,10),((I,J,SSTRAN(I,J),J=1,NOFACT),I=1,NOFACT
1)
105   FORMAT(4A6,A5,22HTRANSFORMATION MATRIX,4A6,A5/(I22,I9,F19.8))
      RETURN
      END

```

```

SUBROUTINE OUTMAT(ARRAY, INDEX, TRAN, NVAR, NOFACT, NCHARS, HOLLER, ITR)
C THIS SUBROUTINE PRINTS THE ORDERED ARRAY WITH INDICES AND THE TRAN. MATRIX
DIMENSION ARRAY(100,14), INDEX(100,14), TRAN(14,14), HOLLER(10),
1DIGIT(5), TSIGN(14), IOUT(3,14)
DOUBLE PRECISION ITER,DIGIT
C $ITER AND DIGIT MUST BE DOUBLE PRECISION FOR THE IBM 360
DATA PLUS,TMINUS/2H .,2H-./
DATA ITER/6H(1TRNO/,DIGIT(1)/6H.=1) /,DIGIT(2)/6H.=2) /,
1DIGIT(3)/6H.=3) /,DIGIT(4)/6H.=4) /,DIGIT(5)/6H.=5) /
C NCHARS MUST BE A MULTIPLY OF 4(IBM 360) OR 6(IBM 7090,7040)
NWORDS=NCHARS/4
C $NWORDS=NCHARS/6
IF(ITR .EQ. 0) PRINT 100,(HOLLER(I),I=1,NWORDS)
IF(ITR .GT. 0) PRINT 100,(HOLLER(I),I=1,NWORDS),ITER,DIGIT(ITR)
100 FORMAT(1H1,21X,12A4)
C100 $FORMAT(1H1,21X,12A6)
1 PRINT 101,(L,L=1,NOFACT)
101 FORMAT(//5H SEQ.,14I9)
PRINT 102
102 FORMAT(1X)
DO 7 I=1,NVAR
DO 6 J=1,NOFACT
ELEM=0.
IF(ARRAY(I,J)) 2,4,3
2 ELEM=ABS(ARRAY(I,J)-.0005)
TSIGN(J)=TMINUS
GO TO 5
3 ELEM=ABS(ARRAY(I,J)+.0005)
4 TSIGN(J)=PLUS
5 IOUT(1,J)=ELEM*10.
IOUT(2,J)=ELEM*100.-FLOAT(IOUT(1,J))*10.
IOUT(3,J)=ELEM*1000.-FLOAT(IOUT(1,J))*100.-FLOAT(IOUT(2,J))*10.
6 CONTINUE
7 PRINT 103,I,(INDEX(I,J),TSIGN(J),(IOUT(K,J),K=1,3),J=1,NOFACT)
103 FORMAT(1X,13,2X,14(13,1X,A2,3I1))
PRINT 104
104 FORMAT(///22X,24HTRANSFORMATION MATRIX)
PRINT 105,(L,L=1,NOFACT)
105 FORMAT(//5H ROW ,14I9)
PRINT 102
DO 13 J=1,NOFACT
DO 12 J=1,NOFACT
ELEM=0.
IH(TRAN(I,J)) 8,10,9
8 ELEM=ABS(TRAN(I,J)-.0005)
TSIGN(J)=TMINUS
GO TO 11
9 ELEM=ABS(TRAN(I,J)+.0005)
10 TSIGN(J)=PLUS
11 IOUT(1,J)=ELEM*10.
IOUT(2,J)=ELEM*100.-FLOAT(IOUT(1,J))*10.
IOUT(3,J)=ELEM*1000.-FLOAT(IOUT(1,J))*100.-FLOAT(IOUT(2,J))*10.
12 CONTINUE
13 PRINT 106,I,(TSIGN(J),(IOUT(K,J),K=1,3),J=1,NOFACT)
106 FORMAT(1X,13,2X,14(4X,A2,3I1))
RETURN
END

```

References

- (1) Bargmann, R. A. Study of independence and dependence in multivariate normal analysis. University of North Carolina, Institute of Statistics, Mimeograph Series No. 186, December, 1957.
- (2) Bartlett, M. S. Test of significance in factor analysis. Brit. J. statist. Psychol., 1950, 3, 77-85.
- (3) Horst, P. Factor Analysis of Data Matrices. Holt, Rinehart and Winston, Inc., 1965.
- (4) Hotelling, H. Analysis of a complex of statistical variables into principal components. J. Educ. Psychol., 24, 417-41, 498-520, 1933.
- (5) Kaiser, H. F. The varimax criterion for analytic rotation in factor analysis. Psychometrika, 1958, 23, 187-200.
- (6) Kaiser, H. F. Computer program for varimax rotation in factor analysis. Education and Psychological Measurement, Vol. XIX, No. 3, 1959.
- (7) Thurstone, L. L. Multiple-Factor Analysis. Chicago, Illinois: The University of Chicago Press, 1949.
- (8) Thurstone, L. L. An analytic method for simple structure. Psychometrika, 1954, 19, 173-182.

Internal Distribution

- 1-2. Central Research Library
- 3. Document Reference Section
- 4-8. Laboratory Records
- 9. Laboratory Records - Record Copy
- 10-59. Mathematics Division
- 60-62. Davis Bobrow
- 63-65. Al Wilcox
- 66. A. S. Householder
- 67. Harvey Carter
- 68. Manuel Feliciano
- 69. C. W. Nestor
- 70. Paul Nelson
- 71. Don Arnurius
- 72. John Carpenter
- 73. John Fulton
- 74-88. R. C. Durfee
- 89. R. K. Adams
- 90. E. D. Arnold
- 91. C. E. Bettis
- 92. W. R. Burrus
- 93. C. C. Cogdon
- 94. R. R. Dickison
- 95. R. A. Dory
- 96. W. E. Foster
- 97. C. D. Goodman
- 98. T. D. Jones
- 99. W. H. Jordan
- 100. H. W. Joy
- 101. M. A. Kastenbaum
- 102. H. A. Levy
- 103. T. A. Lincoln
- 104. P. D. Miller, Jr.
- 105. H. H. Nichol

106. C. E. Parker
107. H. Postma
108. M. T. Robinson
109. W. S. Snyder
110. J. G. Sullivan
111. C. D. Susano
112. M. L. Tobias
113. D. R. Vondy
114. G. M. Watson
115. C. S. Williams

External Distribution

- 116-130. Division of Technical Information Extension
- 131-135. Dr. Edward E. Cureton
136. University of Tennessee Computing Center
137. Dr. Gordon R. Sherman
138. Div. Research and Development, ORO