

University of North Florida
UNF Digital Commons

**UNF Graduate Theses and Dissertations** 

Student Scholarship

1989

# Identifying Outliers in a Random Effects Model For Longitudinal Data

Tamarah Crouse Dishman University of North Florida

Follow this and additional works at: https://digitalcommons.unf.edu/etd

Part of the Mathematics Commons

#### **Suggested Citation**

Dishman, Tamarah Crouse, "Identifying Outliers in a Random Effects Model For Longitudinal Data" (1989). *UNF Graduate Theses and Dissertations*. 191. https://digitalcommons.unf.edu/etd/191

This Master's Thesis is brought to you for free and open access by the Student Scholarship at UNF Digital Commons. It has been accepted for inclusion in UNF Graduate Theses and Dissertations by an authorized administrator of UNF Digital Commons. For more information, please contact Digital Projects. © 1989 All Rights Reserved



### INDENTIFYING OUTLIERS IN A

#### RANDOM EFFECTS MODEL FOR LONGITUDINAL DATA

by

Tamarah Crouse Dishman

A thesis submitted to the Department of Mathematics and Statistics in partial fulfillment of the requirements for the degree of

Master of Arts in Mathematical Sciences

University of North Florida College of Arts and Sciences

December, 1989

The thesis of Tamarah Crouse Dishman is approved:

Signature deleted

Signature deleted

Signature deleted

12/14/54

12/14/89 12/14/89

Committee Chairperson

Signature deleted

Ý

Accepted for The Department:

Chairperson

Accepted for the College:

Signature deleted

Dean

Accepted for the University:

Signature deleted

Interim Vice-President for Academic Affairs

12/14/89

12/14/89

I extend my sincere appreciation to Graduate Director Dr. Donna Mohr, the advisor of this project, for her invaluable guidance and support. I am also grateful to the Faculty, Staff and Students of this University for their interest and contributions.

I also wish to thank my parents for encouraging and nurturing my scholastic endeavors and especially to my husband for his loving support of my professional goals.

# TABLE OF CONTENTS

-----

Acknowledgements	(iii)
Abstract	(vi)
Chapter 1 - Introduction Section 1 - Random effects model for longitudinal data Section 2 - Estimation of parameters	2 3
Chapter 2 - Method of Identifying Non-Trackers Section 1 - Method of Identification Section 2 - Explanation of computer algorithm	11 14
Chapter 3 - Conclusion	20
Appendix 1	27
Appendix 2	29
Appendix 3	30
References	46
Vita	48

List of Tables and Figures

Table 1 - Parameters for Trackers	15
Table 2 - Simulations Run with only Trackers Present	16
Figure 1 - Graph of Expected Values for Non-Trackers	17
Table 3 - Simulations Run with Non-Trackers Present	18
Figure 2 - Outline for Computer Algorithm	19
Table 4 - Results with Trackers Only	23
Table 5 - Parameter Estimates with Trackers Only	24
Table 6 - Results with Non-Trackers Present	25
Table 7 - Parameter Estimates with Non-Trackers Present	26

#### Abstract

Identifying non-tracking individuals in a population of longitudinal data has many applications as well as complications. The analysis of longitudinal data is а special study in itself. There are several accepted methods, of those we chose a two-stage random effects model coupled with the Estimation Maximization Algorithm (E-M Algorithm). Our project consisted of first estimating population parameters using the previously mentioned The Mahalanobis distance methods. then was used to sequentially identify and eliminate non-trackers from the population. Computer simulations were run in order to measure the algorithm's effectiveness.

Our results show that the average specificity for the repetitions for each simulation remained at the 99% level. The sensitivity was best when only a single non-tracker was present with a very different parameter  $\alpha$ . The sensitivity of the program decreased when more than one tracker was present, indicating our method of identifying a non-tracker is not effective when the estimates of the population parameters are contaminated.

#### Chapter 1 - Introduction

According to Ware (1984) longitudinal studies can be loosely defined as studies in which the response of each individual is observed on two or more occasions. There are obviously many applications of longitudinal studies in the medical and social fields. The objectives of studies of this type are to characterize patterns of response and change over time. This motivates the definition of tracking given by Ware and Wu (1981) as the prediction of future values based on repeated measurements of the same characteristic obtained over time for each of a cohort chart of individuals. In this thesis, non-trackers will be defined as individuals whose longitudinal observations do not seem to belong to the same distribution as the rest of the tracking population.

In the remainder of this chapter a popular model for analyzing longitudinal data called the random effects model (Laird and Ware, 1982) will be introduced and explained. The derivations of the equations from Diem and Liukkonen (1988) for fitting the model will be given in detail. Chapter 2 will include the criteria for distinguishing trackers from non-trackers and conclude with a description of the computer simulation of the method. The computer program will be tested for its specificity (defined as its behavior when no non-trackers are present) as well as its sensitivity (measured by its ability to detect non-trackers when they are present). Results of the simulations and overall conclusions appear in Chapter 3.

# Section 1: Random effects model for longitudinal data

Laird and Ware (1982) introduced a two stage model for the analysis of the highly unbalanced data sets obtained from longitudinal studies. In the first stage, the distribution of the characteristics being measured has the same form for each individual, but the parameters vary over individuals. The second stage describes the distribution of these individual parameters or random effects.

Stage 1 for each unit i

$$\mathbf{y}_{\mathbf{i}} = \mathbf{X}_{\mathbf{i}}\boldsymbol{\alpha} + \mathbf{Z}_{\mathbf{i}}\mathbf{b}_{\mathbf{i}} + \mathbf{e}_{\mathbf{i}}$$
(1)

where  $y_i$  is the vector of  $n_i$  observations from individual i, a is a px1 vector of the unknown population parameters,  $X_i$ is a known design matrix linking a to  $y_i$  for each individual,  $b_i$  is the kx1 vector of individual effects and  $Z_i$  is the known design matrix linking  $b_i$  to  $y_i$  for each individual. The  $e_i$  vectors are distributed N( $0, R_i$ ) and assumed to be independent while a is considered fixed and  $b_i$ is a random vector as described in stage 2. Throughout the rest of our work we take  $R_i = \sigma^2 I$ .

#### <u>Stage 2</u>

The  $\mathbf{b_i}$  are distributed as N(0,D), independently of each other and of the  $\mathbf{e_i}$ . D is a kxk positive definite covariance matrix. The population parameters,  $\boldsymbol{\alpha}$ , are treated as fixed effects.

The  $y_i$  are independent and distributed N( $x_i \alpha$ ,  $z_i D z_i^T + \sigma^2 I$ ). The main disadvantage of this model is the strong assumption made about the structure of the covariance matrix of the  $y_i$  given above.

# Section 2: Estimation of parameters

In this section, equations for estimating  $a, \sigma^2$  and **D** will be developed. Since there are no closed form solutions we will derive the iterative solutions from maximum likelihood estimates using the Estimation Maximization Alorithm (comprised of E-step and M-step and denoted E-M Algorithm) given by Dempster et al (1977). We apply the E-M Algorithm to the random effects model following Diem and Liukkonen (1988). The derivations omitted by them are included in this paper as well as the equations. The idea behind the E-M Algorithm is very simple:

1. In the E-step, the  $\mathbf{b}_{i}$  are treated as missing values and are replaced by estimates of  $\mathbf{b}_{i}$ ,  $\mathbf{\hat{b}}_{i}$ . This estimate is calculated using current estimates of  $\boldsymbol{\alpha}, \sigma^{2}, \mathbf{D}$ .

2. In the M-step, parameters  $a, \sigma^2$  and D are estimated using the  $y_i$  and  $b_j$ .

The algorithm is repeated until convergence is obtained or the maximum allowed iterations is reached. The derivation of the equations is as follows.

#### <u>E-Step</u>

First note that the joint probability distribution for  $\mathbf{y}_{i}$  and  $\mathbf{b}_{i}$  given  $\theta = (\boldsymbol{a}, \sigma^{2}, \mathbf{D})$  is given by:  $f(\mathbf{y}_{i}, \mathbf{b}_{i} | \theta) = f(\mathbf{y}_{i} | \mathbf{b}_{i}, \theta) \cdot f(\mathbf{b}_{i} | \theta)$ 

$$= \mathbf{c_1} \cdot \frac{1}{\det |\sigma^2 \mathbf{I}|^{1/2}} \exp \left\{ \frac{-1}{2\sigma^2} (\mathbf{y_i} - \mathbf{x_i} \mathbf{a} - \mathbf{z_i} \mathbf{b_i})^{\mathrm{T}} (\mathbf{y_i} - \mathbf{x_i} \mathbf{a} - \mathbf{z_i} \mathbf{b_i}) \right\}$$
$$\cdot \frac{1}{\det |\mathbf{D}|^{1/2}} \exp \left\{ \frac{-1}{2} (\mathbf{b_i^T D^{-1} b_i}) \right\}$$

where  $c_1$  is a constant

NOTE:  

$$(\mathbf{y}_{i} - \mathbf{X}_{i} \boldsymbol{\alpha} - \mathbf{Z}_{i} \mathbf{b}_{i})^{\mathrm{T}} (\mathbf{y}_{i} - \mathbf{X}_{i} \boldsymbol{\alpha} - \mathbf{Z}_{i} \mathbf{b}_{i}) = [(\mathbf{y}_{i} - \mathbf{X}_{i} \boldsymbol{\alpha})^{\mathrm{T}} - (\mathbf{Z}_{i} \mathbf{b}_{i})^{\mathrm{T}}] [(\mathbf{y}_{i} - \mathbf{X}_{i} \boldsymbol{\alpha}) - (\mathbf{Z}_{i} \mathbf{b}_{i})]$$

$$= (\mathbf{y}_{i} - \mathbf{X}_{i} \boldsymbol{\alpha})^{\mathrm{T}} (\mathbf{y}_{i} - \mathbf{X}_{i} \boldsymbol{\alpha}) - 2 (\mathbf{y}_{i} - \mathbf{X}_{i} \boldsymbol{\alpha})^{\mathrm{T}} \mathbf{Z}_{i} \mathbf{b}_{i} + \mathbf{b}_{i}^{\mathrm{T}} \mathbf{Z}_{i}^{\mathrm{T}} \mathbf{Z}_{i} \mathbf{b}_{i}$$

Now what we need is the conditional pdf of  $\mathbf{b}_{\mathbf{i}}$  given  $\theta$ 

$$f(\mathbf{b}_{i} | \mathbf{y}_{i}, \theta) = \frac{f(\mathbf{b}_{i}, \mathbf{y}_{i} | \theta)}{f(\mathbf{y}_{i} | \theta)}$$

Note that the denominator above is a constant with respect to  $\mathbf{b_i}$ . We collect all of the  $\mathbf{b_i}$  terms in  $f(\mathbf{b_i}, \mathbf{y_i} | \theta)$  and let the remaining terms become one constant,  $C_2$ . Therefore,

$$f(\mathbf{b}_{i} | \mathbf{y}_{i}, \theta) = C_{2} \exp \left\{ \frac{1}{\sigma^{2}} (\mathbf{y}_{i} - \mathbf{X}_{i} \alpha)^{\mathrm{T}} \mathbf{Z}_{i} \mathbf{b}_{i} \right\}$$
  

$$\cdot \exp \left\{ -\frac{1}{2\sigma^{2}} (\mathbf{b}_{i}^{\mathrm{T}} \mathbf{Z}_{i}^{\mathrm{T}} \mathbf{Z}_{i} \mathbf{b}_{i}) - \frac{1}{2} (\mathbf{b}_{i}^{\mathrm{T}} \mathbf{D}^{-1} \mathbf{b}_{i}) \right\}$$

$$= C_{2} \exp\left\{\frac{1}{2\sigma^{2}} \left[2(\mathbf{y}_{1}-\mathbf{x}_{1}\boldsymbol{\alpha})^{\mathrm{T}}\mathbf{z}_{1}\mathbf{b}_{1} - \mathbf{b}_{1}^{\mathrm{T}}(\mathbf{z}_{1}^{\mathrm{T}}\mathbf{z}_{1}+\mathbf{D}^{-1}\sigma^{2})\mathbf{b}_{1}\right]\right\}$$

$$= C_{2} \exp \left\{ \frac{-1}{2} \left[ \frac{-2 \left( \mathbf{y}_{1} - \mathbf{X}_{1} \boldsymbol{\alpha} \right)^{\mathrm{T}} \mathbf{Z}_{1} \mathbf{b}_{1} + \mathbf{b}_{1}^{\mathrm{T}} \left( \mathbf{Z}_{1}^{\mathrm{T}} \mathbf{Z}_{1} + \mathbf{D}^{-1} \sigma^{2} \right) \mathbf{b}_{1} }{\sigma^{2}} \right] \right\}$$

$$= \mathbf{C}_{2} \exp \left\{ \frac{-1}{2} \left[ \mathbf{b}_{1}^{\mathrm{T}} \mathbf{A} \mathbf{b}_{1} - 2 \left( \mathbf{y}_{1} - \mathbf{X}_{1} \mathbf{a} \right)^{\mathrm{T}} \mathbf{Z}_{1} \mathbf{b}_{1} / \sigma^{2} \right] \right\}$$

where  $\mathbf{A} = (\mathbf{Z}_{i}^{T}\mathbf{Z}_{i} + \mathbf{D}^{-1}\sigma^{2})/\sigma^{2}$ . This can be recognized as the general form of the multivariate normal distribution. The variance is found directly by

$$\operatorname{Var}(\mathbf{b}_{1} | \mathbf{y}_{1}, \theta) = \mathbf{A}^{-1} = \left( \frac{\mathbf{z}_{1}^{T} \mathbf{z}_{1} + \mathbf{D}^{-1} \sigma^{2}}{\sigma^{2}} \right)^{-1} = \sigma^{2} \left( \mathbf{z}_{1}^{T} \mathbf{z}_{1} + \mathbf{D}^{-1} \sigma^{2} \right)^{-1}$$

From Appendix 1, it follows that:

$$\mathbb{E}\left(\mathbf{b}_{i} \mid \mathbf{y}_{i}, \theta\right) = \left(\mathbf{z}_{i}^{\mathrm{T}} \mathbf{z}_{i} + \mathbf{D}^{-1} \sigma^{2}\right)^{-1} \mathbf{z}_{i}^{\mathrm{T}} \left(\mathbf{y}_{i} - \mathbf{x}_{i} \boldsymbol{a}\right) = \hat{\mathbf{b}}_{i}$$
(1)

# <u>M-Step</u>

All sums below are over i=1,m.

Using the values  $\boldsymbol{\hat{b}}_{i}$  calculated in the E-step we want to maximize

 $H(\theta) = E\{\ln[f(y_i, b_i | \theta)] | y_i, \theta\}, \text{ ignoring constants it follows}$ that:

$$= E \begin{cases} m \left[ \frac{-n_{i}}{2} \ln(\sigma^{2}) - \frac{1}{2\sigma^{2}} \left( \mathbf{y}_{i} - \mathbf{X}_{i} \boldsymbol{\sigma} - \mathbf{Z}_{i} \mathbf{b}_{i} \right)^{T} \left( \mathbf{y}_{i} - \mathbf{X}_{i} \boldsymbol{\sigma} - \mathbf{Z}_{i} \mathbf{b}_{i} \right) - \frac{1}{2} \ln(\det \mathbf{D}) \end{cases}$$

$$- \frac{1}{2} \left( \mathbf{b}_{i}^{T} \mathbf{D}^{-1} \mathbf{b}_{i} \right) \right] \right\} =$$

$$\sum_{i=1}^{m} \left[ \frac{-n_{i}}{2} \ln \left( \sigma^{2} \right) - \frac{1}{2\sigma^{2}} \left( \mathbf{y}_{i} - \mathbf{x}_{i} \boldsymbol{\alpha} \right)^{T} \left( \mathbf{y}_{i} - \mathbf{x}_{i} \boldsymbol{\alpha} \right)_{+} \frac{1}{\sigma^{2}} \left( \mathbf{y}_{i} - \mathbf{x}_{i} \boldsymbol{\alpha} \right)^{T} \mathbf{z}_{i} \mathbf{E} \left( \mathbf{b}_{i} | \mathbf{y}_{i}, \theta \right) - \frac{1}{2\sigma^{2}} \mathbf{E} \left( \mathbf{b}_{i}^{T} \mathbf{z}_{i}^{T} \mathbf{z}_{i} \mathbf{b}_{i} | \mathbf{y}_{i}, \theta \right) - \frac{1}{2} \ln \left( \det \mathbf{D} \right) - \frac{1}{2} \mathbf{E} \left( \mathbf{b}_{i}^{T} \mathbf{D}^{-1} \mathbf{b}_{i} | \mathbf{y}_{i}, \theta \right) \right]$$

$$= \frac{-N}{2} \ln \sigma^{2} - \frac{1}{2\sigma^{2}} \sum_{i=1}^{m} (\mathbf{y}_{i} - \mathbf{x}_{i} \boldsymbol{\alpha})^{T} (\mathbf{y}_{i} - \mathbf{x}_{i} \boldsymbol{\alpha}) + \frac{1}{\sigma^{2}} \sum_{i=1}^{m} (\mathbf{y}_{i} - \mathbf{x}_{i} \boldsymbol{\alpha})^{T} \mathbf{z}_{i} \mathbf{z}_{i} (\mathbf{b}_{i} | \mathbf{y}_{i}, \theta) ]$$

$$= -\frac{m}{2} \ln (\det \mathbf{D}) - \frac{1}{2\sigma^{2}} \sum_{i=1}^{m} (\mathbf{b}_{i}^{T} (\mathbf{z}_{i}^{T} \mathbf{z}_{i} + \sigma^{2} \mathbf{D}^{-1}) \mathbf{b}_{i} | \mathbf{y}_{i}, \theta]$$

$$= -\frac{N}{2} \ln \sigma^{2} - \frac{m}{2} \ln (\det \mathbf{D}) - \frac{1}{2\sigma^{2}} \sum_{i=1}^{m} (\mathbf{y}_{i} - \mathbf{x}_{i} \boldsymbol{\alpha})^{T} (\mathbf{y}_{i} - \mathbf{x}_{i} \boldsymbol{\alpha}) + \frac{1}{\sigma^{2}} \sum_{i=1}^{m} (\mathbf{y}_{i} - \mathbf{x}_{i} \boldsymbol{\alpha})^{T} \mathbf{z}_{i} \mathbf{b}_{i}$$

$$- \frac{1}{2\sigma^{2}} \sum_{i=1}^{m} (\operatorname{tr} [(\mathbf{z}_{i}^{T} \mathbf{z}_{i} + \sigma^{2} \mathbf{D}^{-1}) \nabla (\mathbf{b}_{i} | \mathbf{y}_{i}, \theta)] + \mathbf{b}_{i}^{T} (\mathbf{z}_{i}^{T} \mathbf{z}_{i} + \sigma^{2} \mathbf{D}^{-1}) \mathbf{b}_{i})$$
(2)

Note: for the above equation  $N=\Sigma n_i$ 

Now, we use H( $\theta$ ) to derive expressions for  $\hat{a}$ ,  $\hat{\sigma}^2$  and  $\hat{D}$ .

By differentiating  $H(\theta)$  with respect to each variable, setting the expression equal to zero and solving for the given variable, a maximum is obtained. First consider **â**:

Note:

$$-\frac{1}{2\sigma^2}\sum_{\alpha}^{\mathbf{m}} (\mathbf{y}_{\mathbf{i}} - \mathbf{X}_{\mathbf{i}} \boldsymbol{\alpha})^{\mathrm{T}} (\mathbf{y}_{\mathbf{i}} - \mathbf{X}_{\mathbf{i}} \boldsymbol{\alpha}) = \frac{-1}{2\sigma^2}\sum_{\alpha}^{\mathbf{m}} \mathbf{y}_{\mathbf{i}}^{\mathrm{T}} \mathbf{y}_{\mathbf{i}} + \frac{1}{\sigma^2}\sum_{\alpha}^{\mathbf{m}} \mathbf{y}_{\mathbf{i}}^{\mathrm{T}} \mathbf{x}_{\mathbf{i}} \boldsymbol{\alpha} + \frac{-1}{2\sigma^2}\sum_{\alpha}^{\mathbf{m}} \boldsymbol{\alpha}^{\mathrm{T}} \mathbf{x}_{\mathbf{i}}^{\mathrm{T}} \mathbf{x}_{\mathbf{i}} \boldsymbol{\alpha}$$

and,

$$\frac{1}{\sigma^2} \sum_{\alpha}^{m} (\mathbf{y}_{\mathbf{i}} - \mathbf{x}_{\mathbf{i}} \boldsymbol{\alpha})^{\mathrm{T}} \mathbf{z}_{\mathbf{i}} \hat{\mathbf{b}}_{\mathbf{i}} = \frac{1}{\sigma^2} \sum_{\alpha}^{m} \mathbf{y}_{\mathbf{i}}^{\mathrm{T}} \mathbf{z}_{\mathbf{i}} \hat{\mathbf{b}}_{\mathbf{i}} + \frac{1}{\sigma^2} \sum_{\alpha}^{m} \boldsymbol{\alpha}^{\mathrm{T}} \mathbf{x}_{\mathbf{i}}^{\mathrm{T}} \mathbf{z}_{\mathbf{i}} \hat{\mathbf{b}}_{\mathbf{i}}$$

therefore,

$$\frac{\partial \mathbf{H}(\theta)}{\partial \alpha} = -\frac{1}{\sigma^2} \sum_{i=1}^{m} \mathbf{x}_{i}^{T} \mathbf{y}_{i} - \frac{1}{\sigma^2} \sum_{i=1}^{m} \mathbf{x}_{i}^{T} \mathbf{x}_{i} \mathbf{\alpha} - \frac{1}{\sigma^2} \sum_{i=1}^{m} \mathbf{x}_{i}^{T} \mathbf{z}_{i} \mathbf{\hat{b}}_{i} = \mathbf{0}$$

it follows that,

$$\sum_{i=1}^{m} \mathbf{x}_{i}^{\mathrm{T}} \mathbf{x}_{i} \boldsymbol{\alpha} = \sum_{i=1}^{m} \mathbf{x}_{i}^{\mathrm{T}} \mathbf{y}_{i} - \sum_{i=1}^{m} \mathbf{x}_{i}^{\mathrm{T}} \mathbf{z}_{i} \mathbf{\hat{b}}_{i} = \sum_{i=1}^{m} \mathbf{x}_{i}^{\mathrm{T}} (\mathbf{y}_{i} - \mathbf{z}_{i} \mathbf{\hat{b}}_{i})$$

and,

$$\hat{\boldsymbol{\alpha}} = \left( \sum_{i}^{m} \mathbf{x}_{i}^{T} \mathbf{x}_{i} \right)^{-1} \sum_{i}^{m} \mathbf{x}_{i}^{T} \left( \mathbf{y}_{i} - \mathbf{z}_{i} \ \hat{\mathbf{b}}_{i} \right)$$
(3)

Now consider  $\hat{\sigma}^2$ :

$$\frac{\partial \mathbf{H}(\theta)}{\partial \sigma^{2}} = \frac{-N}{2\sigma^{2}} + \frac{\sum_{\mathbf{z}} \left[ \left( \mathbf{y}_{i} - \mathbf{x}_{i} \boldsymbol{\alpha} \right)^{\mathrm{T}} \left( \mathbf{y}_{i} - \mathbf{x}_{i} \boldsymbol{\alpha} \right) \right]}{2\sigma^{4}} - \frac{\sum_{\mathbf{z}} \left[ \left( \mathbf{y}_{i} - \mathbf{x}_{i} \boldsymbol{\alpha} \right)^{\mathrm{T}} \mathbf{z}_{i} \hat{\mathbf{b}}_{i} \right]}{\sigma^{4}}$$

+ 
$$\frac{\sum_{i=1}^{m} \left[ \mathbf{z}_{i}^{T} \mathbf{z}_{i} \quad \nabla(\mathbf{b}_{i} | \mathbf{y}_{i}, \theta) \right]}{2\sigma^{4}} + \frac{\sum_{i=1}^{m} \mathbf{b}_{i} \mathbf{z}_{i}^{T} \mathbf{z}_{i} \hat{\mathbf{b}}_{i}}{2\sigma^{4}} = 0$$

it follows that,

$$\sigma^{2} \mathbf{N} - \sum_{i=1}^{m} \left[ \left( \mathbf{y}_{i} - \mathbf{x}_{i} \boldsymbol{\alpha} \right)^{\mathrm{T}} \left( \mathbf{y}_{i} - \mathbf{x}_{i} \boldsymbol{\alpha} \right) \right] + 2 \sum_{i=1}^{m} \left( \mathbf{y}_{i} - \mathbf{x}_{i} \boldsymbol{\alpha} \right)^{\mathrm{T}} \mathbf{z}_{i} \hat{\mathbf{b}}_{i}$$
$$-\sum_{i=1}^{m} \left[ \mathbf{z}_{i}^{\mathrm{T}} \mathbf{z}_{i} \quad \nabla(\mathbf{b}_{i} | \mathbf{y}_{i}, \theta) \right] - \sum_{i=1}^{m} \hat{\mathbf{b}}_{i}^{\mathrm{T}} \mathbf{z}_{i}^{\mathrm{T}} \mathbf{z}_{i} \hat{\mathbf{b}}_{i} = 0$$

therefore,

$$\hat{\sigma}^{2} = \frac{1}{N} \sum_{i=1}^{M} \{ (\mathbf{y}_{i} - \mathbf{x}_{i} \boldsymbol{\alpha})^{T} (\mathbf{y}_{i} - \mathbf{x}_{i} \boldsymbol{\alpha}) - 2 (\mathbf{y}_{i} - \mathbf{x}_{i} \boldsymbol{\alpha})^{T} \mathbf{z}_{i} \hat{\mathbf{b}}_{i} + \operatorname{tr} [\mathbf{z}_{i}^{T} \mathbf{z}_{i} \ \forall (\mathbf{b}_{i} | \mathbf{y}_{i}, \theta)] + \hat{\mathbf{b}}_{i}^{T} \mathbf{z}_{i}^{T} \mathbf{z}_{i} \hat{\mathbf{b}}_{i} \}$$

$$(4)$$

Finally consider  $\hat{D}$ :

In appendix 2 we present some facts about partial derivatives with respect to D. Using those facts we can show,

$$\frac{\partial \mathbf{H}(\theta)}{\partial \mathbf{D}} = \frac{-\mathbf{m}}{2} \mathbf{D}^{-1} + \frac{1}{2} \mathbf{D}^{-1} \sum_{i=1}^{m} \mathbf{\hat{b}}_{i} \mathbf{\hat{b}}_{i}^{T} \mathbf{D}^{-1} + \frac{1}{2} \mathbf{D}^{-1} \sum_{i=1}^{m} \mathbf{V}(\mathbf{b}_{i} | \mathbf{y}_{i}, \theta) \mathbf{D}^{-1} = 0$$

it follows that,

 $-\mathbf{m}\mathbf{D} + \sum_{i=1}^{m} \mathbf{\hat{b}}_{i} \mathbf{\hat{b}}_{i}^{\mathrm{T}} + \sum_{i=1}^{m} \nabla(\mathbf{b}_{i} | \mathbf{y}_{i}, \theta) = 0$ 

$$\Rightarrow \qquad \mathbf{\hat{D}} = \frac{1}{m} \sum_{m=1}^{m} [\mathbf{\hat{b}}_{i} \mathbf{\hat{b}}_{i}^{T} + V(\mathbf{b}_{i} | \mathbf{y}_{i}, \theta)] \qquad (5)$$

We have now verified the equations given by Diem and Liukkonen (1988).

Chapter 2 - Method of Identifying Non-Trackers In order to test the method discussed in Chapter 1, we used the equations 1-5 and implemented them in a computer program. In order to make computation easier, only the balanced case was addressed. What follows is an explanation of the criterion used in the algorithm for identifying nontrackers, a flow chart of the program and a list of the various simulations that were run.

#### Section 1 : Method of Identification

As mentioned in the introduction, non-trackers will be identified as those individuals whose observations do not seem to belong to the distribution of the tracking population. The criterion we have selected to make this determination is called the Mahalanobis distance and is defined as follows:

for each individual i,

$$\mathcal{D}_{i} = (\mathbf{y}_{i} - \mu)^{T} (\text{var } \mathbf{y})^{-1} (\mathbf{y}_{i} - \mu)$$

$$= (\mathbf{y}_{\mathbf{i}} - \mathbf{X}_{\mathbf{i}} \hat{\boldsymbol{\alpha}})^{\mathrm{T}} (\mathbf{Z}_{\mathbf{i}} \hat{\mathbf{D}} \mathbf{Z}_{\mathbf{i}}^{\mathrm{T}} + \hat{\sigma}^{2} \mathbf{I})^{-1} (\mathbf{y}_{\mathbf{i}} - \mathbf{X}_{\mathbf{i}} \hat{\boldsymbol{\alpha}})$$
(6)

since we assume that each individual is normally distributed with mean  $X_i a$  and variance  $Z_i D Z_i^T + \sigma^2 I$ . If a,  $\sigma^2$  and Dwere known and used in place of their estimates in equation (6), clearly,  $D_1$  would have a chi-square distribution with n degrees of freedom where n is the dimension of **y**.

In order to "weed out" non-trackers, we will first find the individual with the largest Mahalanobis distance. The p-value is calculated for that individual and compared to a previously determined significance level (denoted "signif"). If the p-value is less than the significance level, the individual is considered a non-tracker and eliminated from the tracking population. New population parameters are calculated and the process is repeated until the p-value of the maximum  $\vartheta_i$  in the current iteration is not less than the significance level. At that time the parameters of the tracking population are given as well as the number of nontrackers.

Due to our approximation of the  $D_i$ 's being independently distributed as chi-square each with n degrees of freedom, we arrived at our calculation of the p-value by using order statistics. Each time through the "weeding out" process we are interested in the individual with the maximum  $D_i$ . Note that from equation 6, this is a measurement of the observation with the maximum distance from the normal distribution with paramenters calculated from equations 1-5. Examination of the probability distribution of the maximum order statistic for this type of distribution leads to a pvalue expressed as

p-value=1- F(dmax)<sup>m</sup>

where dmax stands for maximum  $D_i$  from  $X^2(n)$  and F(dmax) equals the cumulative distribution function. The program for computing F(dmax) is taken from Press et al (1986).

In order to test the sensitivity (probability that an individual is identified as a non-tracker given that they are really a non-tracker) and specificity (probability that an individual is identified as a tracker given that they really are a tracker) of our algorithm, several simulations will be run. Combinations of the number of non-trackers present, the number of individuals, the number of observations per individual, the magnitude of  $\sigma^2$ , significance levels and values used for X, Z and  $\alpha$  are listed in tables at the end of this chapter. The results of the simulations described above appear in Chapter 3.

In an attempt to clarify the relationship between trackers, non-trackers and their parameters a pictorial representation appears in Figure 1 at the end of this chapter. Trackers and non-trackers are sketched on the same axis with respect to  $X\alpha$ , their expected values, for n=5 and n=10. Both case A and case B for non-trackers are shown. In case A,  $\alpha = (5, -4, 2)$ , the opposite slopes for the sketches indicate that the non-trackers are drastically different from the trackers. For case B,  $\alpha = (7, 0, 1)$ , there is only a slight difference between the non-trackers and trackers. Given this information, we would expect case A type of nontracker to be easier to identify.

#### Section 2: Explanation of Computer Algorithm

As sketched in Figure 2, the calling program is RANCOEF2. It begins by getting parameters for trackers and non-trackers after which it generates **y**<sub>i</sub> for each. It then calls the subroutine FITRCB2 which is designed to first make initial estimates for the population parameters, then improve these estimates using the E-M algorithm. Next, the subroutine FIND is called to "weed out" the non-trackers. If any individual is eliminated FITRCB2 is called to recalculate the parameter estimates of the tracking population then FIND is called again. After all nontrackers have been "weeded out" we return to the main program where parameter information is recorded. There are 50 repetitions of each simulation and overall statistics for each type of simulation are calculated and given in Tables 4 and 6 of Chapter 3. The random number generator was adopted from Press et al (1986) and LINPACK routines were used for matrix manipulations.

# Table 1- Parameters for Trackers

$$\frac{n=5}{X} = \begin{pmatrix} 1 & -2 & 0 \\ 1 & -1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 2 & 1 \end{pmatrix} \qquad Z = \begin{pmatrix} 1 & -2 \\ 1 & -1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{pmatrix} \qquad D = \begin{pmatrix} 0.5 & 0.0 \\ 0.0 & 0.5 \end{pmatrix}$$
  
The two settings for  $\sigma^2$  are 0.5 and 2.0.

<u>n=10</u>

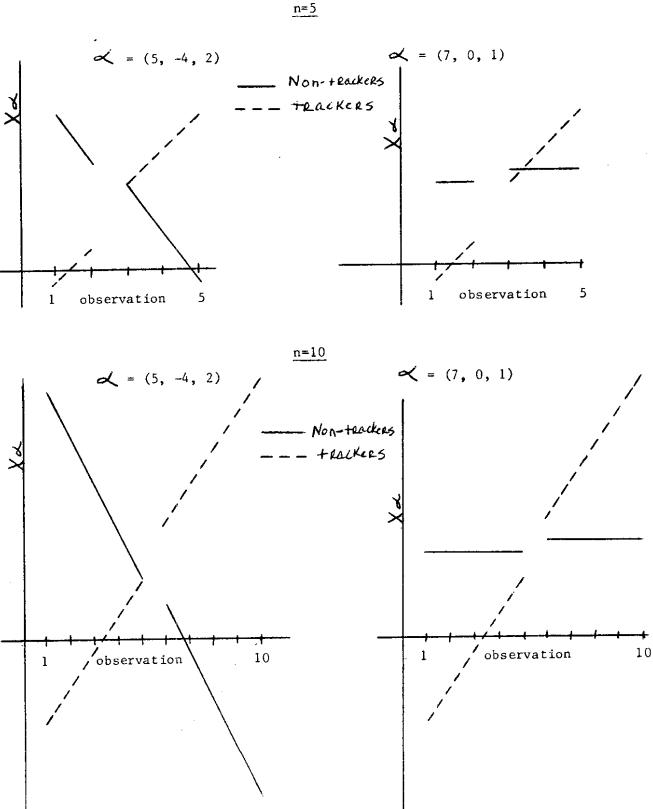
$$\mathbf{X} = \begin{pmatrix} 1 & -4 & 0 \\ 1 & -3 & 0 \\ 1 & -2 & 0 \\ 1 & -1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 3 & 1 \\ 1 & 4 & 1 \\ 1 & 5 & 1 \end{pmatrix} \qquad \mathbf{Z} = \begin{pmatrix} 1 & -4 \\ 1 & -3 \\ 1 & -2 \\ 1 & -1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \end{pmatrix} \qquad \mathbf{D} = \begin{pmatrix} 0.5 & 0.0 \\ 0.0 & 0.5 \end{pmatrix}$$

The two settings for  $\sigma^2$  are 2.0 and 8.0

m	n	σ <sup>2</sup>	signif
25	5	0.5	.10
25	5	0.5	.15
25	5	2.0	.10
25	5	2.0	.15
25	10	2.0	.10
25	10	2.0	.15
25	10	8.0	.10
25	10	8.0	.15
50	5	0.5	.10
50	5	0.5	.15
50	5	2.0	.10
50	5	2.0	.15
50	10	2.0	.10
50	10	2.0	.15
50	10	8.0	.10
50	10	8.0	.15

Table 2 -Simulations Run with only Trackers Present

Figure 1 - Graph of Expected Values for Non-Trackers



m	n	ntr	σ <sup>2</sup>	a
25 25 25 25 25 25 25 25 25 25	5 5 5 5 5 5 5 5 10	1 1 1 2 2 2 2 1	0.5 0.5 2.0 2.0 0.5 0.5 2.0 2.0 2.0	А В А В А В А В
25 25 25 25 25 25 25 25 25	10 10 10 10 10 10 10	1 1 1 2 2 2 2 2	2.0 2.0 8.0 2.0 2.0 8.0 8.0	A B A B A B A B
50 50 50 50 50 50 50 50	5 5 5 5 5 5 5 5 5	1 1 1 3 3 3	0.5 0.5 2.0 2.0 0.5 0.5 2.0	A B A B A B A
50 50 50 50 50 50 50 50 50 50	5 10 10 10 10 10 10 10 10	3 1 1 1 3 3 3 3 3	2.0 2.0 2.0 8.0 8.0 2.0 2.0 8.0 8.0	B A B A B A B A B

Table 3 - Simulations Run with Non-Trackers Present

The significance level used was 0.10. A indicates  $\alpha = (5, -4, 2)$ ; B indicates  $\alpha = (7, 0, 1)$ . Ntr stands for the number of non-trackers present.

#### Figure 2 - Outline for Computer Algorithm

RANCOEF2

Read parameters  $\rightarrow$  Begin a simulation Generate y for trackers Generate y for non-trackers Call FITRCB2 Initial estimate of population parameters E-M algorithm to improve above estimates using previously determined convergence criterion ⇒Call FIND Calculate  $D_i$ Is max  $D_i$  large? YES ⇒)NO Throws out the non-tracker FITRCB2 to Call recalculate parameter estimates of tracking population Record parameter estimates for the simulation Last simulation? No≮ YÉS Calculate overall parameter estimates, sensitivity and specificity for each simulation

#### Chapter 3 - Conclusion

The first observation, obvious by looking at Table 4, is that when no non-trackers are present the algorithm is The average specificity among repetitions stays excellent. above 99% for each type of simulation indicating that the program has no problem identifying a tracker when it really is a tracker. From Table 5 we see that the parameter estimates of  $\alpha$ ,  $\sigma^2$  and D are very accurate. When signif is changed from 0.10 to 0.15 while all other parameters are held constant it is true that specificity is slightly better at the 0.10 level. More interesting is the fact that the percentage of repetitions that throw out a tracker increases as signif increases. The overall average for this percentage at the 0.10 level is 0.0775 and for 0.15 it is 0.10. This is what we would expect to happen since our pvalue for max  $\vartheta_i$  is being compared to the signif level to identify non-trackers and possibly eliminate them.

When non-trackers are included the results are much more interesting. From Table 6 we see that again the average specificity among repetitions remains above 99% for each type of simulation. When we examine the sensitivity, it is easy to see that non-trackers were correctly identified with best accuracy when only one non-tracker was present and when it was significantly different from the tracking population. It is also obvious that increasing the number of non-trackers significantly lowers the sensitivity of the program. This indicates that identifying non-trackers is very difficult when the parameter estimates are very contaminated. As we would expect, throughout the simulations, the sensitivity levels were higher for case A, a = (5, -4, 2), than case B, a = (7, 0, 1), (recall that case A non-trackers are very different from the trackers where case B non-trackers are only slightly different). Signif was held constant at 0.10 for all of the simulations in which non-trackers were included. This level was used in order to reduce the number of trackers incorrectly identified as non-trackers. Changing  $\sigma^2$  while holding everything else constant results in only a slight change in the level of sensitivity.

As shown in Table 7, the estimate of D,  $\hat{D}$ , is affected significantly by the presence of more than one non-tracker. Specifically, the entries of  $\hat{D}$  are larger than D. We know that when  $\hat{D}$  is large, it's inverse is small and therefore by the relationship given in equation 6,  $D_1$  is smaller than it should be. As a result, the power of the Mahalanobis distance is being reduced. This in turn reduces the power of our algorithm. Our research did not include this, but, other suggestions such as eliminating two non-trackers at a time could be studied for the applicability to this problem.

We began this project with the intentions of designing a computer algorithm for identifying non-trackers present in a population from a balanced set of data. Although theoretically sound, some algorithms do not attain the practical application desired. For professionals, this is not discouraging but rather a way of opening other areas of study. An investigation of the influence function seems to be a logical alternative.

- - - -

n	<sub>σ</sub> 2	signif	spec	std dev

m

Table 4 - Results with Trackers Only

25	5	0.5	.10	.9976	.0096
25	5	0.5	.15	.996	.0121
25	5	2.0	.10	.9968	.0110
25	5	2.0	.15	.9936	.0169
25	10	2.0	.10	.996	.0121
25	10	2.0	.15	.996	.0121
25	10	8.0	.10	.9968	.0110
25	10	8.0	.15	.9944	.0162
50	5	0.5	.10	.9984	.0055
50	5	0.5	.15	.998	.0061
50	5	2.0	.10	.9984	.0055
50	5	2.0	.15	.998	.0061
50	10	2.0	.10	.9988	.0048
50	10	2.0	.15	.9988	.0048
50	10	8.0	.10	.9988	.0048
50	10	8.0	.15	.9988	.0048

Table 5 - Average Parameter Estimates with Trackers Only (Selected Cases) <u>Case 1</u>: m=25 n=5  $\sigma^2=2.0$  $\hat{\sigma}^2$  = 1.8791 with std dev = 0.2628  $\mathbf{\hat{D}} = \begin{pmatrix} 0.4755 & 0.0244 \\ 0.0244 & 0.4934 \end{pmatrix}$  $\hat{\boldsymbol{\alpha}} = \begin{pmatrix} \frac{a v q}{4.9823} \\ 3.0287 \\ 1.9332 \end{pmatrix} \begin{pmatrix} \frac{5 L u}{0.2869} \\ 0.2538 \\ 0.5535 \end{pmatrix}$ <u>Case 2</u>: m=50 n=5  $\sigma^2$ =2.0  $\hat{\sigma}^2$  = 1.9458 with std dev = 0.1986  $\hat{\mathbf{b}} = \begin{pmatrix} 0.4949 & 0.0149 \\ 0.0149 & 0.5041 \end{pmatrix}$  $\hat{\alpha} = \begin{pmatrix} \frac{avg}{5.0120} \\ 3.0294 \\ 1.9257 \end{pmatrix} \begin{pmatrix} \frac{std \ dev}{0.1964} \\ 0.1598 \\ 0.3833 \end{pmatrix}$ <u>Case 3</u>: m=25 n=10  $\sigma^2$ =8.0  $\hat{\sigma}^2 = 7.9065$  with std dev = 0.8343  $\hat{\mathbf{b}} = \begin{pmatrix} 0.4964 & -0.0025 \\ -0.0025 & 0.4803 \end{pmatrix}$  $\hat{\boldsymbol{\alpha}} = \begin{pmatrix} \frac{3}{4}, 9264 \\ 3, 0210 \\ 2, 0334 \end{pmatrix} \begin{pmatrix} 0, 4290 \\ 0, 1702 \\ 0, 7609 \end{pmatrix}$ <u>Case 4</u>: m=50 n=10  $\sigma^2=2.0$  $\hat{\sigma}^2 = 1.9827$ with std dev = 0.1313 $\mathbf{\hat{D}} = \begin{pmatrix} 0.4897 & -0.0217 \\ -0.0217 & 0.4970 \end{pmatrix}$  $\hat{\boldsymbol{\alpha}} = \begin{pmatrix} 4.9841 \\ 2.0805 \\ 2.0104 \end{pmatrix} \begin{pmatrix} 0.1870 \\ 0.1085 \\ 0.2676 \end{pmatrix}$ True parameter values for the above are: signif = 0.10 $\begin{array}{c} \text{gnif} = 0.10 \\ = \begin{pmatrix} 0.5 & 0.0 \\ 0.0 & 0.5 \end{pmatrix} \quad \alpha = \begin{pmatrix} 5 \\ 3 \\ 2 \end{pmatrix} \end{array}$ **D** =

m	n	ntr	σ <sup>2</sup>	α	spec	sens
25	5	1	0.5	A	.99583	1.000
25	5	1	0.5	B	.99667	0.300
25	5	1	2.0	A	.99750	0.960
25	5	1	2.0	B	.99667	0.220
25	5	2	0.5	A	.99826	0.260
25	5	2	0.5	B	.99913	0.100
25	5	2	2.0	A	.99652	0.160
25	5	2	2.0	B	.99730	0.080
25	10	1	2.0	A	.99250	0.580
25	10	1	2.0	B	.99583	0.080
25	10	1	8.0	A	.99583	0.640
25	10	1	8.0	B	.99667	0.160
25	10	2	2.0	A	.99391	0.180
25	10	2	2.0	B	.99739	0.030
25	10	2	8.0	A	.99739	0.120
25	10	2	8.0	B	.99826	0.000
50	5	1	0.5	A	.99796	1.000
50	5	1	0.5	B	.99837	0.600
50	5	1	2.0	A	.99878	1.000
50	5	1	2.0	B	.99959	0.360
50	5	3	0.5	A	.99745	0.620
50	5	3	0.5	B	.99830	0.267
50	5	3	2.0	A	.99872	0.627
50	5	3	2.0	B	.99915	0.093
50	10	1	2.0	A	.99714	1.000
50	10	1	2.0	B	.99796	0.320
50	10	1	8.0	A	.99510	1.000
50	10	1	8.0	B	.99959	0.240
50	10	3	2.0	A	.99915	0.227
50	10	3	2.0	B	.99787	0.067
50	10	3	8.0	A	.99872	0.293
50	10	3	8.0	B	.99787	0.047

The significance level used was 0.10. A indicates a=(5, -4, 2); B indicates a=(7,0,1). Ntr stands for the number of non-trackers present.

#### Table 6 - Results with Non-Trackers Present

Table 7 - Average Parameter Estimates with Non-Trackers Present (Selected Cases from non-tracker  $\alpha = (5, -4, 2)$ ) Case 1: m=25 n=5 NTR=2  $\sigma^2$ =2.0  $\hat{\sigma}^2$  = 1.8667 with std dev = 0.3681  $\hat{\mathbf{D}} = \begin{pmatrix} 0.5048 & -0.0211 \\ -0.0211 & 3.5484 \end{pmatrix} \qquad \boldsymbol{\alpha} = \begin{pmatrix} \frac{avq}{4.9932} \\ 2.5188 \\ 2.06585 \end{pmatrix} \begin{pmatrix} \frac{502}{0.2935} \\ 0.2923 \\ 0.6356 \end{pmatrix}$ Case 2: m=50 n=5 NTR=3  $\sigma^2=2.0$  $\hat{\sigma}^2$  = 1.8538 with std dev = 0.2443  $\mathbf{\hat{D}} = \begin{pmatrix} 0.5280 & 0.0032 \\ 0.0032 & 1.4190 \end{pmatrix} \qquad \mathbf{\alpha} = \begin{pmatrix} \frac{avq}{4.9827} \\ 2.8683 \\ 1.9940 \end{pmatrix} \begin{pmatrix} \frac{std dev}{0.2001} \\ 0.2693 \\ 0.3849 \end{pmatrix}$ Case 3: m=25 n=10 NTR=2  $\sigma^2$ =8.0  $\hat{\sigma}^2 = 7.8287$  with std dev = 0.7751  $\mathbf{\hat{D}} = \begin{pmatrix} 0.4989 & -0.0948 \\ -0.0948 & 3.6280 \end{pmatrix} \qquad \mathbf{\alpha} = \begin{pmatrix} \frac{avg}{5.0112} \\ 2.5055 \\ 1.9160 \end{pmatrix} \begin{pmatrix} \frac{5ta}{0.3302} \\ 0.2887 \\ 0.6536 \end{pmatrix}$ Case 4: m=50 n=10 NTR=3  $\sigma^2$ =2.0  $\hat{\sigma}^2$  = 1.9294 with std dev = 0.1603  $\mathbf{\hat{D}} = \begin{pmatrix} 0.5008 & 0.0145 \\ 0.0145 & 2.6450 \end{pmatrix} \quad \mathbf{\alpha} = \begin{pmatrix} \frac{\text{avg}}{4.9727} \\ 2.6678 \\ 2.0182 \end{pmatrix} \begin{pmatrix} \frac{\text{std dev}}{0.1605} \\ 0.2053 \\ 0.2584 \end{pmatrix}$ True parameter values for the above are: signif = 0.10 $\mathbf{D} = \begin{pmatrix} 0.5 & 0.0 \\ 0.0 & 0.5 \end{pmatrix} \quad \mathbf{a} = \begin{pmatrix} 5 \\ 3 \\ 2 \end{pmatrix}$ 

26

# Appendix 1

In order to find the mean of our multivariate normal distribution we set the general form of this distribution, where  $\mathbf{b}_i$  is the random variable, equal to our distribution, substitute for  $\mathbf{A}$  and solve for  $\mu$  directly.

Since,

$$\frac{-\left[\left(\mathbf{b}_{1}-\boldsymbol{\mu}\right)^{\mathrm{T}}\mathbf{A}\left(\mathbf{b}_{1}-\boldsymbol{\mu}\right)\right]}{2} = \frac{-\left[\mathbf{b}_{1}^{\mathrm{T}}\mathbf{A}\mathbf{b}_{1}-2\boldsymbol{\mu}^{\mathrm{T}}\mathbf{A}\mathbf{b}_{1}+\boldsymbol{\mu}^{\mathrm{T}}\mathbf{A}\boldsymbol{\mu}\right]}{2}$$

and we have,

$$\mathbf{A} = \frac{\left(\mathbf{z}_{i}^{\mathrm{T}} \mathbf{z}_{i} + \mathbf{D}^{-1} \sigma^{2}\right)}{\sigma^{2}}$$

Now, setting the "linear term" from above equal to the "linear term" of our distribution

$$-2\mu^{\mathrm{T}}\mathbf{A}\mathbf{b}_{\mathbf{i}} = \frac{-2\left(\mathbf{y}_{\mathbf{i}} - \mathbf{x}_{\mathbf{i}}\boldsymbol{\alpha}\right)^{\mathrm{T}} \mathbf{z}_{\mathbf{i}}\mathbf{b}_{\mathbf{i}}}{\sigma^{2}}$$

$$\Rightarrow \frac{\mu^{\mathrm{T}}\left(\mathbf{z}_{\mathbf{i}}^{\mathrm{T}}\mathbf{z}_{\mathbf{i}} + \mathbf{D}^{-1}\sigma^{2}\right)}{\sigma^{2}} = \frac{\left(\mathbf{y}_{\mathbf{i}} - \mathbf{x}_{\mathbf{i}}\boldsymbol{\alpha}\right)^{\mathrm{T}} \mathbf{z}_{\mathbf{i}}}{\sigma^{2}}$$

$$\Rightarrow \mu^{\mathrm{T}} = \left(\mathbf{y}_{\mathbf{i}} - \mathbf{x}_{\mathbf{i}}\boldsymbol{\alpha}\right)^{\mathrm{T}} \mathbf{z}_{\mathbf{i}} \left(\mathbf{z}_{\mathbf{i}}^{\mathrm{T}}\mathbf{z}_{\mathbf{i}} + \mathbf{D}^{-1}\sigma^{2}\right)^{-1}$$

$$\Rightarrow \mu = \left(\mathbf{z}_{\mathbf{i}}^{\mathrm{T}}\mathbf{z}_{\mathbf{i}} + \mathbf{D}^{-1}\sigma^{2}\right)^{-1} \mathbf{z}_{\mathbf{i}} \left(\mathbf{y}_{\mathbf{i}} - \mathbf{x}_{\mathbf{i}}\boldsymbol{\alpha}\right)$$

#### Appendix 2

We must show that  $\frac{\partial}{\partial D} \mathbf{b}_{i}^{T} \mathbf{D}^{-1} \mathbf{b}_{i} = - \mathbf{D}^{-1} \mathbf{b}_{i} \mathbf{b}_{i}^{T} \mathbf{D}$ We use the fact that  $\mathbf{b}_{i}^{T} \mathbf{D}^{-1} \mathbf{b}_{i} = \operatorname{tr} \mathbf{D}^{-1} \mathbf{b}_{i} \mathbf{b}_{i}^{T} = \operatorname{tr} \mathbf{D}^{-1} \mathbf{A}$  where  $\mathbf{A} = \mathbf{b}_{i} \mathbf{b}_{i}^{T}$ . Both  $\mathbf{D}^{-1}$  and  $\mathbf{A}$  are symmetric. We know that  $\frac{\partial \mathbf{d}_{rc}^{ij}}{\partial \mathbf{d}_{rc}} = - \mathbf{d}^{ir} \mathbf{d}^{cj}$  where  $\mathbf{d}_{rc}$  is the r, c<sup>th</sup> entry in  $\mathbf{D}$  and  $\mathbf{d}^{ir}$  is the i, r<sup>th</sup> entry in  $\mathbf{D}^{-1}$ . Then  $\frac{\partial}{\partial \mathbf{d}_{rc}} \mathbf{b}_{i}^{T} \mathbf{D}^{-1} \mathbf{b}_{i} = \frac{\partial}{\partial \mathbf{d}_{rc}} \operatorname{tr} \mathbf{D}^{-1} \mathbf{A} = \frac{\partial}{\partial \mathbf{d}_{rc}} \sum_{i=1}^{r} \sum_{i=1}^{r} \mathbf{d}^{i1} \mathbf{a}_{1i}$  $= -\sum_{i=1}^{r} \sum_{i=1}^{r} \mathbf{d}^{ir} \mathbf{d}^{c1} \mathbf{a}_{1i} = -\sum_{i=1}^{r} \mathbf{d}^{ir} (\sum_{i=1}^{r} \mathbf{d}^{c1} \mathbf{a}_{1i})$ 

The sum in parentheses in the last expression is the  $c,i^{th}$  entry in  $D^{-1}A$ , or the i, $c^{th}$  entry in  $AD^{-1}$  using the symmetry of A and  $D^{-1}$ . Replacing  $d^{ir}$  by  $d^{ri}$  using symmetry, we have

$$= -\sum_{i} d^{ri} (AD^{-1})_{ic}$$

Since this last expression is just the  $r^{th}$  row of  $D^{-1}$  times the  $c^{th}$  column of  $AD^{-1}$ , we have that

$$\frac{\partial}{\partial D} \mathbf{b}_{i}^{T} \mathbf{D}^{-1} \mathbf{b}_{i} = - \mathbf{D}^{-1} \mathbf{b}_{i} \mathbf{b}_{i}^{T} \mathbf{D}^{-1}$$

## Appendix 3 - Programs for Computer Simulations

с	CALLING PROGRAM TO GENERATE DATA WITH TRACKERS AND NONTRACKERS.	RAN0001
С	PROGRAM THEN CALLS ROUTINE FITRCE TO FIT RANDOM COEFFICIENT	RAN0002
C	LINEAR MODELS AS DESCRIBED BY LAIRD AND WARE, BIOMETRICS, 1982.	RAN0003
С	THE ACTUAL ALGORITHM FOLLOWS THE DESCRIPTION GIVEN BY DIEM AND	RAN0004
С	LIUKKONEN IN STATISTICS IN MEDICINE, 1988.	<b>RAN0005</b>
С	AFTER THE INITIAL FIT, THE PROGRAM USES THE ROUTINE FINDEM	RAN0006
С	TO FIND NONTRACKERS.	RAN0007
с	AT THE END, PROGRAM PRINTS ESTIMATED PARAMETER VALUES FOR	RAN0008
с	TRACKERS, AND A LIST OF ID NUMBERS FOR SUSPECTED NONTRACKERS.	RAN0009
с	****** BALANCED DATA DESIGNS ******	RAN0010
	IMPLICIT REAL*8 (A-H,Q-Z)	RAN0011
	INTEGER N,M,P,K,PN,KN,MTR,MNTR	RAN0012
	DIMENSION X(20,6),Z(20,6)	RAN0013
	DIMENSION Y(20,200), ALPH(6), B(6,200), D(6,6)	RAN0014
	DIMENSION XN(20,6), ZN(20,6)	RAN0015
	DIMENSION ALFBAR(6), DBAR(6,6), ALFDEV(6)	RAN0016
	INTEGER IUSE(200)	RAN0017
	COMMON /DATAS/X,Z,Y	RAN0018
	COMMON /PARAMS/ ALPH, B, SIGMA2, D	RAN0019
	COMMON /ITCON/MAXIT	RAN0020
	COMMON /USEME/IUSE	RAN0021
	DIMENSION TALPH(6),TD(6,6),TQ(6,6),XA(20),BT(6)	RAN0022
	DIMENSION TALPHN(6), TDN(6,6), TON(6,6), XAN(20)	RAN0023
	DIMENSION WORK(6), JTV(6)	<b>RAN0024</b>
С	ALL INPUT READ FROM FILE ON CHANNEL 3	RAN0025
С	ALL INPUT READ FROM FILE ON CHANNEL 3 IUSE(200) IS AN INTEGER VECTOR WHERE IUSE(I)=1 MEANS	RAN0026
С	THE INDIVIDUAL I SHOULD BE USED IN FITTING	RAN0027
С	IUSE(I)=0 MEANS DO NOT USE INDIV. I IN FITTING MODEL	RAN0028
С	READ IN DIMENSIONS N=# OBSERVATIONS PER INDIVIDUAL	RAN0029
С	M = # OF INDIVIDUALS	RAN0030
с	M = # OF INDIVIDUALS P = DIMENSION OF ALPHA (FIXED EFFECTS) K = DIMENSION OF B (RANDOM EFFECTS)	RAN0031
Ç	K = DIMENSION OF B (RANDOM EFFECTS)	RAN0032
С	PNEDIM OF ALPHA KOP NONTRACKERS	RAN0033
С	KN=DIM OF B FOR NONTRACKERS	RAN0034
С	MTR = NUM OF TRACKERS	RAN0035
С		RAN0036
С	*** GET PARAMETERS FOR TRACKERS AND NONTRACKERS ***	RAN0037
С		RAN0038
	READ (3,*) N,M,P,K,PN,KN,MTR	RAN0039
	MNTR=M-MTR	RAN0040
С	READ NUMITR = NUMBER OF REPETITIONS OF SIMULATION	RAN0041
	WRITE (9,*) ' N,M,P,K,MTR',N,M,P,K,MTR	RAN0042
		RAN0043
	READ (3,*) MAXIT, CONV	RAN0044
С	READ (3,*) NUMITR READ (3,*) MAXIT, CONV GET MATRIX X FOR TRACKERS, XN FOR NONTRACKERS DO 10 I=1,N	RAN0045
	DO 10 I=1.N	RAN0046
	READ $(3, *)$ $(X(I,J), J=1, P), (XN(I,J), J=1, PN)$	RAN0047
10	CONTINUE	RAN0048
С	GET MATRIX Z FOR TRACKERS, ZN FOR NONTRACKERS	RAN0049
	DO 15 I=1,N	RAN0050
	READ $(3, *)$ $(Z(I, J), J=1, K), (ZN(I, J), J=1, KN)$	RAN0051

30

		31
15	CONTINUE	RAN0052
с	GET TRUE VALUES OF ALPHA, STORED IN TALPH	RAN0053
	READ $(3, *)$ (TALPH(I), I=1, P), (TALPHN(I), I=1, PN)	RAN0054
	WRITE (9,*) ' ALPH FOR NONTR'	RAN0055
	WRITE(9,16) (TALPHN(I), I=1, PN) -	RAN0056
16	FORMAT(3(1X,F10.4))	RAN0057
С	GET TRUE VALUE OF MEASUREMENT VARIANCE, SIGMA2, STORED AS TSIG2	RAN0058
	READ (3,*) TSIG2,TSIG2N	RAN0059
	TSIG=DSQRT(TSIG2)	RAN0060
	TSIGN=SQRT(TSIG2N)	RAN0061
c	GET TRUE VALUE OF COVARIANCE MATRIX D FOR RANDOM EFFECTS	RAN0062
	DO 20 I=1,K	RAN0063
	READ $(3,*)$ $(TD(I,J),J=1,K)$	RAN0064
20	CONTINUE	RAN0065
	DO 25 I=1,KN	RAN0066
	$READ(3, \star)  (TDN(I,J), J=1, KN)$	RAN0067
25	CONTINUE	RAN0068
	READ (3,*) SIGNIF	RAN0069
	READ (3,*) IDUM	RAN0070
	WRITE (9,*) ' TSIG2, SIGNIF'	RAN0071
	WRITE(9,30) TSIG2, SIGNIF	RAN0072
30	FORMAT(2(1X,F10.4))	RAN0073
	XX=RAN3(IDUM)	RAN0074
C	TRAN	RAN0075
с	CALL CHOLESKY DECOMPOSITION TO FACTOR TD=(TQ)*(TQ)	RAN0076 RAN0077
	DO 75 I=1,K	RAN0078
	DO 72 $J=1, K$	RAN0078
	TQ(I,J) = TD(I,J)	RAN0079
72	CONTINUE	RAN0081
75	CONTINUE	RAN0082
	JOB=0	RAN0083
	LDA=6 CALL DCHDC(TQ,LDA,K,WORK,JPV,JOB,INFO)	RAN0084
с		RAN0085
L.	WRITE (6,*) INFO DO 78 I=1,K-1	RAN0086
	DO 77 J=I+1, K	RAN0087
	TQ(J,I) = TQ(I,J)	RAN0088
	TQ(I,J) = 0.D0	RAN0089
77	CONTINUE	RAN0090
78	CONTINUE	RAN0091
c	TRAN	RAN0092
č	CALL CHOLESKY DECOMPOSITION TO FACTOR TDN=(TQN)*(TQN)	RAN0093
C	DO 95 I=1,KN	RAN0094
	DO 92 J=1,KN	RAN0095
	TQN(I,J) = TDN(I,J)	RAN0096
92	CONTINUE	RAN0097
95	CONTINUE	RAN0098
	JOB=0	RAN0099
	LDA=6	RAN0100
	CALL DCHDC (TQN, LDA, KN, WORK, JPV, JOB, INFO)	RAN0101
с	WRITE (6,*) INFO	RAN0102
-	DO 98 $I=1, KN-1$	RAN0103
	DO 97 J=I+1, KN	RAN0104
	TQN(J,I) = TQN(I,J)	RAN0105
	TQN(I,J) = 0.D0	RAN0106
97	CONTINUE	RAN0107
98	CONTINUE	RAN0108
č		RAN0109
ē	STORE MEAN VECTOR X*ALPHA FOR TRACKERS	RAN0110
	DO 150 I=1,N	RAN0111

		32
	$X\lambda(I)=0,D0$	RAN0112
	DO 145 J=1, P	RAN0113
	XA(I) = XA(I) + X(I,J) + TALPH(J)	RAN0114
145	CONTINUE	RAN0115
150	CONTINUE	RAN0116
С		RAN0117
С	STORE MEAN VECTOR XN*ALPHAN FOR NONTRACKERS	RAN0118
	DO 190 I=1,N	RAN0119
	XAN(I) = 0.D0	RAN0120
	DO 185 $J=1$ , PN	RAN0121 RAN0122
185	XAN(I) = XAN(I) + XN(I,J) + TALPHN(J) CONTINUE	RAN0122 RAN0123
190	CONTINUE	RAN0123
100	CONTINUE	RAN01250
C .	++++++++++ BEGIN REPETITIONS, CREATING DATA FOR +++++++++	RAN0126
с	++++++++++ TRACKERS, NONTRACKERS; FITTING MODEL ++++++++++	RAN0127
С	++++++++++ AND FINDING NON-TRACKERS	RAN0128
С		RAN0129
	SIGBAR=0.0D0	RAN0130
	SIGDEV=0.0D0	RAN0131
	DO 200 I=1,P	RAN0132
	ALFBAR(I)=0.0D0	RAN0133
200	ALFDEV(I)=0.0D0 DO 210 I=1.K	RAN0134 RAN0135
	DO 205 J=1,K	RAN0135
	DBAR(I,J) = 0.0D0	RAN0137
205	CONTINUE	RAN0138
210	CONTINUE	RAN0139
	SUMSN =0.0D0	RAN0140
	SUMSN2=0.0D0	RAN0141
	SUMSP =0.0D0	RAN0142
	SUMSP2=0.0D0	RAN0143
с	DO 3000 III=1,NUMITR	RANO144 RANO145
C	DO 4 I=1,M	RAN0146
4	IUSE(I) = 1	RAN0147
С	+++++ MANUFACTURE Y FOR TRACKERS II=1,MTR ++++++	RAN0148
		RAN01490
	DO 300 II=1,MTR	RAN0150
C	MANUFACTURE B FOR THE II INDIVIDUAL	RAN0151
	CALL MULTNO (TQ, 6, K, BT, IDUM)	RAN0152
	DO 270 I=1,K	RAN0153
270	B(I,II)=BT(I) CONTINUE	RAN0154 RAN0155
270	DO 290 I=1.N	RAN0155
	Y(I,II)=XA(I)+GASDEV(IDUM)*TSIG	RAN0157
	DO 280 $J=1,K$	RAN0158
	Y(I,II) = Y(I,II) + Z(I,J) * BT(J)	RAN0159
280	CONTINUE	RAN0160
290	CONTINUE	RAN0161
300	CONTINUE	RAN0162
С		RAN0163
С	+++++ MANUFACTURE Y FOR NONTRACKERS II=MTR+1,M ++++++	RAN0164
	IF (MTR.GE.M) GO TO 410	RAN01650 RAN0166
	DO 400 II= $MTR+1, M$	RAN0168 RAN0167
с	MANUFACTURE B FOR THE II INDIVIDUAL	RAN0168
-	CALL MULTNO (TQN, 6, KN, BT, IDUM)	RAN0169
	DO 370 $I=1, KN$	RAN0170
	B(I,II) = BT(I)	RAN0171

		22
		33 RAN0172
370	CONTINUE	RANO172 RANO173
	DO 390 $I=1,N$	RAN0174
	Y(I,II)=XAN(I)+GASDEV(IDUM)*TSIG DO 380 J=1,KN	RAN0175
	Y(I,II) = Y(I,II) + 2N(I,J) + BT(J)	RAN0176
380	CONTINUE	RAN0177
390	CONTINUE	RAN0178
400	CONTINUE	RAN0179
410	CONTINUE	RAN0180
С		RAN0181
С		RAN0182
С	CALL FITTING ROUTINE FOR BALANCED DATA	RAN0183
	CALL FITRCB(N, M, P, K)	RAN0184
	CALL FINDEM(N, M, P, K, SIGNIF)	RANO185 RANO186
	WRITE(9, *) 'THE FOLLOWING INDIVIDUALS WERE IDENTIFIED AS	RAN0185
	+NON-TRACKERS'	RAN0188
	NUMTR=0 DO 510 I=1,M	RAN0189
	NUMTR=NUMTR+IUSE(I)	RAN0190
510	IF (IUSE(I) .EQ. 0) WRITE( $9,*$ ) I	RAN0191
	WRITE (9,*) ' NUMBER OF TRACKERS ', NUMTR	RAN0192
	IDCTR=0	RAN0193
	IDCNTR=0	RAN0194
	DO 515 I=1,MTR	RAN0195
515	IDCTR=IDCTR + IUSE(I)	RAN0196
	IF (MTR+1 .GT. M) THEN	RAN0197
	SENS=-1.	RAN0198 RAN0199
	ELSE	RAN0200
520	DO 520 I=MTR+1,M IDCNTR=IDCNTR + (1-IUSE(I))	RAN0201
520	SENS= DFLOAT(IDCNTR)/DFLOAT(M-MTR)	RAN0202
	SUMSN=SUMSN + SENS	RAN0203
	SUMSN2=SUMSN2 + (SENS**2)	RAN0204
	ENDIF	RAN0205
600	SPEC=DFLOAT(IDCTR)/DFLOAT(MTR)	RAN0206
	SUMSP=SUMSP + SPEC	RAN0207
	SUMSP2=SUMSP2 + (SPEC**2)	RAN0208
С		RAN0209 RAN0210
	WRITE (9,*) ' ESTIMATED ALPHAS '	RAN0210 RAN0211
	DO 605 I=1,P MOTINE (A COA) ALDER(T)	RAN0212
604	WRITE (9,604) ALPH(I) Format (1X,F10.4)	RAN0213
605	CONTINUE	RAN0214
	WRITE (9,611) SIGMA2	RAN0215
611	FORMAT (' SIGMA2', F10.5)	RAN0216
	DO 615 I=1,K	RAN0217
	WRITE (9,617) (D(I,J),J=1,K)	RAN0218
617	FORMAT (' D',6(1X,F9.4))	RAN0219
615	CONTINUE	RAN0220
c		RAN0221
c	THIS WILL CALC OVERALL PARAMS FOR EACH SIMULATION	RANO222 RANO223
C	CTCDAD-CTCDAD + CTCMA2	RAN0224
	SIGBAR≂SIGBAR + SIGMA2 SIGDEV=SIGDEV + (SIGMA2**2)	RAN0225
	DO 620 I=1,P	RAN0226
	ALFBAR(I) = ALFBAR(I) + ALPH(I)	RAN0227
620	ALFDEV(I) = ALFDEV(I) + (ALPH(I) **2)	RAN0228
~ - •	DO 650 I=1,K	RAN0229
	DO 640 J=1,K	RAN0230
	DBAR(I,J) = DBAR(I,J) + D(I,J)	RAN0231

34 640 CONTINUE **RAN0232** 650 CONTINUE RAN0233 **RAN0234** С 3000 CONTINUE **RAN0235** 3000 IS END OF REPETITION LOOP ++++++ RAN0236 C ¢ RAN0237 Ç CALCULATE STATS FOR SENSITIVITY AND SPECIFITY **RAN0238** С **RAN0239** RNITR=DFLOAT (NUMITR) RAN0240 SPMEAN=SUMSP/RNITR RAN0241 SPSIG=SQRT(((RNITR\*SUMSP2)-(SUMSP\*\*2))/(RNITR\*(RNITR-1.))) RAN0242 IF(MTR+1.GT.M) GO TO 750 RAN0243 SNMEAN=SUMSN/RNITR **RAN0244** SNSIG=SQRT(((RNITR\*SUMSN2)-(SUMSN\*\*2))/(RNITR\*(RNITR-1.))) RAN0245 WRITE(9,736) SNMEAN, SNSIG RAN0246 FORMAT (' SENSITIVITY MEAN ', F8.5,' STD DEV ', F8.5) 736 **RAN0247** WRITE(9,737) SPMEAN, SPSIG 750 **RAN0248** 737 FORMAT (' SPECIFICITY MEAN ', F8.5,' STD DEV ', F8.5) RAN0249 RAN0250 Ċ WRITE OUT PARAMS FOR THE SIMULATION С RAN0251 С RAN0252 WRITE(9,\*)'OVERALL ESTIMATE OF SIGMA-SQUARED' RAN0253 SIGDEV=SQRT(((RNITR\*SIGDEV)-(SIGBAR\*\*2))/(RNITR\*(RNITR-1.))) RAN0254 SIGBAR=SIGBAR/RNITR **RAN0255** WRITE(9,604) SIGBAR RAN0256 WRITE(9,\*)'WITH STANDARD DEVIATION' **RAN0257** WRITE(9,604) SIGDEV **RAN0258** WRITE(9,\*)'OVERALL ESTIMATED ALPHA' **RAN0259** DO 800 I=1,P RAN0260 ALFDEV(I)=SQRT(((RNITR\*ALFDEV(I))-(ALFBAR(I)\*\*2)) RAN0261 +/(RNITR\*(RNITR-1.))) RAN0262 ALFBAR(I)=ALFBAR(I)/RNITR **RAN0263** 800 WRITE(9,604) ALFBAR(I) RAN0264 WRITE(9,\*)'WITH STANDARD DEVIATION' RAN0265 DO 805 I=1,P **RAN0266** WRITE(9,604) ALFDEV(1) 805 RAN0267 WRITE(9,\*)'OVERALL ESTIMATE OF D' RAN0268 DO 820 I=1,K RAN0269 RAN0270 DO 810 J=1,K DBAR(I,J)=DBAR(I,J)/RNITR RAN0271 810 CONTINUE RAN0272 820 CONTINUE RAN0273 DO 830 I=1,K RAN0274 WRITE(9,617) (DBAR(I,J),J=1,K) RAN0275 830 CONTINUE RAN0276 1000 STOP RAN0277 END RAN0278 RAN02790 С RAN0280 С SUBROUTINE TO PRODUCE MULTIVARIATE NORMAL VECTOR BT WITH RAN0281 COVARIANCE MATRIX GIVEN BY TQ\*TRAN(TQ) DEFINED LENGTH OF VECTOR IS LDA. USED LENGTH IS С RAN0282 С RAN0283 к. SUBROUTINE MULTNO (TQ, LDA, K, BT, IDUM) RAN0284 IMPLICIT REAL\*8 (A-H,O-Z) **RAN0285** DIMENSION TQ(LDA,LDA), BT(LDA) RAN0286 DIMENSION Z(20) **RAN0287** DO 10 I=1,K **RAN0288** Z(I) = GASDEV(IDUM) **RAN0289** 10 CONTINUE RAN0290 DO 20 I=1,K RAN0291

		25
		35 RAN0292
	BT(I)=0.D0 DO 15 J=1,I	RAN0293
	BT(I) = BT(I) + TQ(I,J) * Z(J)	RAN0294
15	CONTINUE	RAN0295
20	CONTINUE	RAN0296
	RETURN	RAN0297
	END	RAN0298
С	FUNCTION GASDEV PRODUCES A STANDARD NORMAL DEVIATE	RAN0299 RAN0300
	FUNCTION GASDEV (IDUM)	RAN0301
	IMPLICIT REAL*8 (A-H,O-Z)	RAN0302
	DATA ISET/0/ IF (ISET.EQ.0) THEN	RAN0303
1	V1=2.*RAN3(IDUM)-1.	<b>RAN0304</b>
-	V2=2.*RAN3(IDUM)-1.	RAN0305
	R=V1**2+V2**2	RAN0306
	IF(R.GE.1.)GO TO 1	RAN0307
	FAC=DSQRT(-2.*DLOG(R)/R)	RAN0308
	GSET=V1*FAC	RAN0309 RAN0310
	GASDEV=V2*FAC	RAN0311
	ISET=1	RAN0312
	ELSE GASDEV=GSET	RAN0313
	ISET=0	RAN0314
	ENDIF	RAN0315
	RETURN	RAN0316
	END	RAN0317
		RAN03180
с	FUNCTION RAN3 PRODUCES A UNIFORM (0,1) RANDOM DEVIATE	RAN0319 RAN0320
	FUNCTION RAN3(IDUM)	RAN0321
с	IMPLICIT REAL*8 (A-H,O-Z) IMPLICIT REAL*4(M)	RAN0322
c	PARAMETER (MBIG=4000000.,MSEED=1618033.,MZ=0.,FAC=2.5E-7)	RAN0323
•	PARAMETER (MBIG=1000000000, MSEED=161803398, MZ≈0, FAC=1.E-9)	RAN0324
	DIMENSION MA(55)	RAN0325
	DATA IFF /0/	RAN0326
	IF(IDUM.LT.O.OR.IFF.EQ.O)THEN	RAN0327
	IFF=1	RANO328 RANO329
	MJ=MSEED-IABS(IDUM)	RAN0329
	MJ=MOD(MJ,MBIG)	RAN0331
	MA(55)=MJ MK=1	RAN0332
	DO 11 $I=1,54$	RAN0333
	II=MOD(21*I,55)	RAN0334
	MA(II)=MK	RAN0335
	MK=MJ-MK	RAN0336
	IF(MK.LT.MZ)MK=MK+MBIG	RAN0337
	MJ=MA(II)	RAN0338 RAN0339
11	CONTINUE	RAN0333
	DO 13 K=1,4 DO 12 I=1,55	RAN0341
	MA(I) = MA(I) - MA(1 + MOD(I + 30, 55))	RAN0342
	IF(MA(I).LT.MZ)MA(I)=MA(I)+MBIG	RAN0343
12	CONTINUE	RAN0344
13	CONTINUE	RAN0345
	INEXT=0	RAN0346
	INEXTP=31	RAN0347
	IDUM=1	RAN0348 RAN0349
	ENDIF INEXT=INEXT+1	RAN0349
	INEXT-INEXT+1 IF(INEXT, EQ, 56) INEXT=1	RAN0351

INEXTP=INEXTP+1 IF(INEXTP.EQ.56)INEXTP=1 MJ=MA(INEXT)-MA(INEXTP) IF(MJ.LT.MZ)MJ=MJ+MBIG MA(INEXT)=MJ RAN3=MJ\*FAC RETURN END 36 RAN0352 RAN0353 RAN0354 RAN0355 RAN0356 RAN0357 RAN0358 RAN0359

FITRCB IS THE ROUTINE FOR FITTING THE POPULATION С С PARAMETERS FOR TRACKERS. THE EQUATIONS ARE TAKEN С FROM DIEM AND LIUKKONEN (1988) AND THEIR DERIVATIONS C C APPEAR IN CHAPTER 1 OF THIS PAPER. С THE FIRST PART OF THIS PROGRAM CALCULATES THE Ċ INITIAL ESTIMATES OF THE B'S, ALPHA, SIGMA SQUARED, С AND THE D'S. THE E-M ALGORITHM IS THEN IMPLEMENTED WITH THE ABOVE ESTIMATES IN ORDER TO ITERATIVELY С IMPROVE THE ESTIMATES OF THE PARAMETERS. Ċ SUBROUTINE FITRCB(N,M,P,K) IMPLICIT REAL \*8 (A-H,Q-Z) INTEGER N,M,P,K INTEGER IPVT(6) DIMENSION X(20,6),Z(20,6),Y(20,200) DIMENSION D(6,6), B(6,200), ALPH(6), DET(2) DIMENSION XTX(6,6), ZTZ(6,6), ZTZI(6,6), ZZZ(6,20) DIMENSION ZT(6,20), YSUM(20), DSUM(6,6), DOLD(6,6) DIMENSION XT(6,20), XALPH(20), DIFF(20) DIMENSION SUMI(6,6), SUMIZT(6,20), YDIFF(20,200) INTEGER IUSE(200) COMMON/DATAS/X,Z,Y COMMON/PARAMS/ALPH, B, SIGMA2, D COMMON/ITCON/MAXIT, IFLAG, CONV COMMON/USEME/IUSE DATA IFRST/1/ COMMON /FIND/ZT, XALPH IFLAG=9 С Ç FIRST CALCULATE USEFUL QUANTITIES Ċ С THE FOLLOWING GIVES TRANS(X) \*X С DO 10 I=1,P DO 5 J=1,P XTX(I,J) = DDOT(N, X(1, I), 1, X(1, J), 1)CONTINUE 5 CONTINUE 10 С THE FOLLOWING GIVES TWO COPIES OF TRANS(Z) \*Z С Ç DO 20 I=1,K DO 15 J=1,K ZTZ(I,J) = DDOT(N, Z(1, I), 1, Z(1, J), 1)ZTZI(I,J) = ZTZ(I,J)CONTINUE 15 20 CONTINUE ¢ С NEED TO CALCULATE INV(TRANS(Z)\*Z)\*TRANS(Z). FIRST NEED TO GET TRANS (Z) THEN USE LINPACK С С FACTOR AND SOLVER. ABOVE WILL BE STORED IN ZZZ С DO 30 I=1,K DO 25 J=1,N  $\operatorname{ZT}(I,J) = \operatorname{Z}(J,I)$ ZZZ(I,J) = ZT(I,J)25 CONTINUE 30 CONTINUE CALL DGEFA(ZTZI,6,K,IPVT,INFO)

FIT0001 FIT0002 FIT0003 FIT0004 FIT0005 FIT0006 FIT0007 FIT0008 FIT0009 FIT0010 FIT00110 FIT0012 FIT0013 FIT0014 FIT0015 FIT0016 FIT0017 FIT0018 FIT0019 FIT0020 FIT0021 FTT0022 FIT0023 FIT0024 FIT0025 FIT0026 FIT0027 FIT0028 FIT0029 FIT0030 FIT0031 **FIT0032** FIT0033 FIT0034 FIT0035 FIT0036 FIT0037 FIT0038 FIT0039 FIT0040 FTT0041 FIT0042 FIT0043 FIT0044 FIT0045 FIT0046 FIT0047 FIT0048 FIT0049 FIT0050 FIT0051 FIT0052 FIT0053 FIT0054 FIT0055 FTT0056 FIT0057 FIT0058

FIT0059

FIT0060

37

		20
	IF(INFO.NE. 0) GO TO 500	
	DO 35 I=1,N CALL DGESL(ZTZI,6,K,IPVT,ZZZ(1,I),0)	
35 C	CONTINUE	
c c	NOW CALCULATE THE INVERSE	
	CALL DGEDI(ZTZI,6,K,IPVT,DET,WORK,1)	
C 38	DM=0.D0	
	DO 39 J=1,M	
39	DM=DM+DFLOAT(IUSE(J)) IDM=INT(DM)	
с		
c	SUM THE Y'S	
C	DO 45 I=1,N	
	YSUM(I)=0.0D0 D0 40 J=1,M	
	IF (IUSE(J).EQ.0) GO TO 40	
	YSUM(I) = YSUM(I) + Y(I,J)	
40	CONTINUE	
45	CONTINUE	
с с	INITIALIZE DSUM, DIFF AND SIGSUM TO ZERO	
č	INTITALIZE DOW, DIFT AND DIGDOM TO BENO	
-	DO 55 I=1,K	
	DO 50 J=1,K	
	DSUM(I,J)=0.0D0	
50 55	CONTINUE	
55	CONTINUE DO 57 I=1,N	
	DIFF(I)=0.0D0	
57	CONTINUE	
	SIGSUM=0.0D0	
С		
с	THIS ROUTINE CALCULATES THE INITIAL EST OF ALPHA	
c c	THE SOLUTION IS STORED IN ALPH	
C	DO 60 I=1,P	
	ALPH(I) = DDOT(N, X(1, I), 1, YSUM, 1)/DM	
60	CONTINUE	
	CALL DGEFA(XTX, 6, P, IPVT, INFO)	
	IF (INFO.NE.0) GO TO 501	
с	CALL DGESL(XTX,6,P,IPVT,ALPH,0)	
č	CALCULATE AND STORE THE INITIAL EST OF X*ALPH	
č		
	DO 70 I=1,P	
	DO 65 J=1,N	
65	XT(I,J) = X(J,I)	
65 70	CONTINUE	
,,,	DO 75 I=1,N	
	XALPH(I)=DDOT(P,XT(1,I),1,ALPH,1)	
75	CONTINUE	
с		
C	THIS CALCULATES THE INITIAL B'S	
с	DO 120 J=1,M	
	DO 120 0	

38

FIT0061 FIT0062 FIT0063 FIT0064 FIT0065 FIT0066 FIT0067 FIT0068 FIT0069 FIT0070 FIT0071 FIT0072 FIT0073 FIT0074 FIT0075 FIT0076 FIT0077 FIT0078 FIT0079 FIT0080 FIT0081 FIT0082 FIT0083 FIT0084 FIT0085 FIT0086 FIT0087 FIT0088 FIT0089 FIT0090 FIT0091 FIT0092 FIT0093 FIT0094 FIT0095 FIT0096 FIT0097 FIT0098 FIT0099 FIT0100 FIT0101 FIT0102 FIT0103 **FIT0104** FIT0105 FIT0106 FIT0107 FIT0108 FIT0109 FIT0110 FIT0111 **FIT0112** FIT0113 FIT0114 FIT0115 FIT0116 FIT0117 FIT0118 FIT0119 FIT0120

.

		20
		39 FIT0121
	IF (IUSE(J).EQ.0) GO TO 120 DO 90 I=1,N	FIT0122
	DIFF(I) = Y(I,J) - XALPH(I)	FIT0123
90	CONTINUE	FIT0124
20	DO 95 I=1,K	FIT0125
	B(I,J)=0.0D0	FIT0126
	DO 92 L=1,N	FIT0127
92	B(I,J) = B(I,J) + 222(I,L) * DIFF(L)	FIT0128
95	CONTINUE	FIT0129 FIT0130
	DO 105 I≂1,K	F170130
	DO 100 L=1,K P(T, T) = P(T, T) + (P(T, T) + P(T, T))	FIT0132
100	DSUM(I,L) = DSUM(I,L) + (B(I,J) * B(L,J)) CONTINUE	FIT0133
105	CONTINUE	FIT0134
700	DO 115 I=1,N	FIT0135
	DO 110 L=1,K	FIT0136
	DIFF(I) = DIFF(I) - (Z(I,L) * B(L,J))	FIT0137
110	CONTINUE	FIT0138
115	CONTINUE	FIT0139
	SIGSUM=SIGSUM+DDOT(N,DIFF,1,DIFF,1)	FITO140 FITO141
120	CONTINUE	F170142
C C	THE INITIAL EST OF SIGMA SQUARED IS IN SIGMA2	FIT0143
c	THE INITIAL EST OF SIGHA SQUARED IS IN SIGHAL	FIT0144
<u> </u>	SIGMA2=SIGSUM/DFLOAT((IDM*N)-P-(K*IDM)+K)	FIT0145
С		FIT0146
С	CALCULATE AND STORE THE INITIAL EST OF D	FIT0147
C		FIT0148
	DO 130 I=1,K	FIT0149
	DO 125 $J=1,K$	FIT0150 FIT0151
105	D(I,J) = (DSUM(I,J)/(DM-1.) - (SIGMA*ZTZI(I,J)))	F170152
125 130	CONTINUE CONTINUE	FIT0153
C	CONTINOE	FIT0154
č	THE E-M ALGORITHM	FIT0155
č		FIT0156
С	E-STEP	FIT0157
С	FIRST STORE USEFUL QUANTITIES	FIT0158
С		FIT0159
	ITER=0	FIT0160
1000	ITER=ITER+1	FIT0161 FIT0162
	SIGOLD=SIGMA2 DO 140 I=1,K	FIT0163
	DO 135 $J=1,K$	FIT0164
	DSUM(I,J)=0.0D0	FIT0165
	DOLD(I,J) = D(I,J)	FIT0166
135	CONTINUE	FIT0167
140	CONTINUE	FIT0168
	SIGSUM=0.0D0	FIT0169
	CALL DGEFA(D, 6, K, IPVT, INFO)	FIT0170
	IF (INFO.NE.O) GO TO 502	FIT0171
~	CALL DGEDI(D,6,K,IPVT,DET,WORK,1)	FIT0172 FIT0173
с с	D NOW CONTAINS INV(D)	FIT0174
c	D HOR CONTETUD THA(D)	FIT0175
~	DO 150 I=1,K	FIT0176
	DO 145 J=1,K	FIT0177
	SUMI(I,J) = ZTZ(I,J) + (SIGMA2 * D(I,J))	FIT0178
145	CONTINUE	FIT0179
150	CONTINUE	FITO180

. . . . . . .

	CALL DGEFA(SUMI,6,K,IPVT,INFO)
	IF(INFO.NE.0) GO TO 503
	CALL DGEDI (SUMI, 6, K, IPVT, DET, WORK, 1)
	DO 160 I=1,K
	DO 155 $J=1,N$
155	SUMIZT(I,J)=DDOT(K,SUMI(1,I),1,ZT(1,J),1)
155 160	CONTINUE
ĉ	001111102
c	CALCULATE THE IMPROVED EST OF THE B'S
с	
	DO 190 J≕1,M IF (IUSE(J).EQ.0) GO TO 190
	DO 165 I=1,N
	DIFF(I) = Y(I, J) - XALPH(I)
165	CONTINUE
	DO 175 I=1,K
	B(I,J)=0.0D0 D0 170 L=1,N
	B(I,J)=B(I,J)+(SUMIZT(I,L)*DIFF(L))
170	CONTINUE
175	CONTINUE
	DO 185 I=1,K DO 180 L≕1,K
	DSUM(I,L) = DSUM(I,L) + (B(I,J) * B(L,J))
180	CONTINUE
185	CONTINUE
190	CONTINUE
c c	
č	M-STEP
С	
c c	RE-CALCULATE THE ALPHAS
C	DO 195 I=1,N
195	YSUM(I)=0.0D0
	DO 205 J=1,M
	IF (IUSE(J).EQ.0) GO TO 205
	DO 200 $I=1,N$ YDIFF(I,J)=Y(I,J)-(DDOT(K,2T(1,I),1,B(1,J),1))
	<pre>IDIT(1,0)=1(1,0) (DEG((x,21(1,1),1,0(1,0),1,0)) YSUM(1)=YSUM(1)+YDIFF(1,J)</pre>
200	CONTINUE
205	CONTINUE
	DO 210 $I=1,P$
210	ALPH(I)=DDOT(N,X(1,I),1,YSUM,1)/DM CONTINUE
210	CALL DGESL(XTX,6,P,IPVT,ALPH,0)
	DO 215 I≈1,N
	XALPH(I) = DDOT(P, XT(1, I), 1, ALPH, 1)
215 C	CONTINUE
c	RE-CALCULATE THE D'S
č	
	DO 225 I=1,K
	DO 220 $J=1,K$ D(I,J)=(DSUM(I,J)/DM)+(SIGMA2*SUMI(I,J))
220	CONTINUE
225	CONTINUE
С	
С	RE-CALCULATE SIGMA-SQUARED

.

40 FIT0181 FIT0182 FIT0183 FIT0184 FIT0185 FIT0186 FIT0187 FIT0188 FIT0189 FIT0190 FIT0191 FIT0192 FIT0193 FIT0194 FIT0195 FIT0196 FIT0197 FIT0198 FIT0199 FIT0200 FIT0201 FIT0202 FIT0203 FIT0204 FIT0205 FIT0206 FIT0207 FIT0208 FIT0209 FIT0210 FIT0211 FIT0212 FIT0213 FIT0214 FIT0215 FIT0216 FIT0217 FIT0218 FIT0219 FIT0220 FIT0221 FIT0222 FIT0223 FIT0224 FIT0225 FIT0226 FIT0227 FIT0228 FIT0229 FIT0230 FIT0231 FIT0232 FIT0233 FIT0234 FIT0235 FIT0236 FIT0237 FIT0238 FIT0239 FIT0240

		41
С		FIT0241
	DO 235 J=1,M	FIT0242
	IF (IUSE(J).EQ.0) GO TO 235	FIT0243
	DO 230 I=1,N	FIT0244
	YDIFF(I,J) = YDIFF(I,J) - XALPH(I)	FIT0245
230	CONTINUE	FIT0246
235	CONTINUE	FIT0247
	DO 240 J=1,M	FIT0248
	IF (IUSE(J).EQ.0) GO TO 240	FIT0249
	SIGSUM=SIGSUM+DDOT(N,YDIFF(1,J),1,YDIFF(1,J),1)	FIT0250
240	CONTINUE	FIT0251
	TRACE=0.0D0	FIT0252
	DO 245 I=1,K	FIT0253
	TRACE=TRACE+DDOT(K, ZTZ(1, I), 1, SUMI(1, I), 1)	FIT0254
245	CONTINUE	FIT0255
	SIGMA2=((SIGSUM)/DFLOAT(M*N))+((SIGMA2*TRACE)/DFLOAT(N))	FIT0256
	U=DABS (SIGMA2-SIGOLD)	FIT0257
	DO 255 I=1,K	FIT0258
	DO 250 J=1,I	FIT0259
	R=DABS(DOLD(I,J)-D(I,J))	FIT0260
250	CONTINUE	FIT0261
255	CONTINUE	FIT0262
	IF (R.GT.U) U=R	FIT0263
	IF (U.GT.CONV) GO TO 1100	FIT0264
	IFLAG=1	FIT0265
	WRITE(20,*) 'CONVERGED IN', ITER, 'ITERATIONS'	FIT0266
1100	IF(ITER.LT.MAXIT) THEN	FIT0267
	GO TO 1000	FIT0268
	ELSE	FIT0269
	WRITE(20,*) 'FAILED TO CONVERGE IN', MAXIT, 'ITERATIONS'	FIT0270
	ENDIF	FIT0271
	RETURN	FIT0272
500	WRITE(20,*)'THE MATRIX TRANS(Z)*Z IS NOT INVERTIBLE'	FIT0273
	STOP	FIT0274
501	WRITE(20,*)'THE MATRIX TRANS(X)*X IS NOT INVERTIBLE'	FIT0275
	STOP	FIT0276
502	WRITE(20,*)' THE MATRIX D IS NOT INVERTIBLE'	FIT0277
	STOP	FIT0278
503	WRITE(20,*) 'THE MATRIX SUMI IS NOT INVERTIBLE'	FIT0279
	STOP	FIT0280
	END	FIT0281

·

000000000	ONCE THE POPULATION PARAMETERS HAVE BEEN CALCULATED BY FITRCB USING ALL OBSERVATIONS FINDEM CALCULATES THE MAXIMUM MAHALNOBIS DISTANCE. THIS NUMBER IS COMPARED TO A P-VALUE CALCULATED BY THE FUNCTION PVCHI AND EITHER ELIMINATED OR KEPT. EACH TIME AN OBSERVATION IS ELIMINATED FITRCB IS CALLED TO RE-CALCULATE THE PARAMETERS FOR THE 'TRACKING' POPULATION.
с	SUBROUTINE FINDEM (N,M,P,K,SIGNIF) IMPLICIT REAL *8(A-H,Q-Z) INTEGER N,M,P,K INTEGER JPVT(20),IPVT(20) DIMENSION X(20,6),Z(20,6),Y(20,200) DIMENSION X(20,6),Z(20,6),Y(20,200) DIMENSION D(6,6),B(6,200),ALPH(6),DET(2) DIMENSION ZT(6,20) DIMENSION ZT(6,20),WORK(20) DIMENSION XALPH(20),WORK(20) DIMENSION XALPH(20),COV(20,20),U(20),DMH(200) INTEGER IUSE(200) DIMENSION YDIFF(20) COMMON/DATAS/X,Z,Y COMMON/DATAS/X,Z,Y COMMON/PARAMS/ALPH,B,SIGMA2,D COMMON/ITCON/MAXIT,IFLAG,CONV COMMON/USEME/IUSE COMMON/FIND/ZT,XALPH
0000	FIRST NEED TO CALCULATE THE COVARIANCE MATRIX THIS CALCULATES INV(2*D*TRANS(2)+SIGMA2*I)
1000	DO 10 I=1,N DO 5 $J \approx 1, K$
5 10	<pre>ZDT(J,I)=DDOT(K,ZT(1,I),1,D(1,J),1) CONTINUE CONTINUE DO 20 I=1,N DO 15 J=1,N COV(I,J)=DDOT(K,ZDT(1,J),1,ZT(1,I),1)</pre>
15	CONTINUE COV(I,I)=COV(I,I)+SIGMA2
20	CONTINUE CALL DGEFA(COV,20,N,IPVT,INFO) IF (INFO.NE.0)THEN WRITE(9,*) 'COV IS NOT INVERTIBLE' ENDIF
С	CALL DGEDI(COV,20,N,IPVT,DET,WORK,1)
c c	USE CHOLESKY DECOMP TO CALC MAHALANOBIS DIST
	DM=0. DMAX=-1.0 DO 30 I=1,M
	IF (IUSE(I).EQ.0) GO TO 30 DM=DM+1. DO 22 J=1,N
	·

42

FIN0006 **FIN0007** FIN0008 FIN0009 FIN00100 FIN00110 FIN00120 FIN0013 **FIN0014** FIN0015 FIN0016 FIN0017 **FIN0018** FIN0019 FIN0020 FIN0021 FIN0022 FIN0023 FIN0024 FIN0025 FIN0026 FIN0027 FIN0028 FIN0029 FIN0030 FIN0031 FIN0032 FIN0033 FIN0034 FIN0035 FIN0036 **FIN0037** FIN0038 FIN0039 FIN0040 FIN0041 FIN0042 FIN0043 FIN0044 FIN0045 FIN0046 FIN0047 FIN0048 FIN0049 FIN0050 FIN0051 FIN0052 FIN0053 FIN0054 FIN0055 FIN0056 FIN0057

		4'3
22	YDIFF(J) = Y(J, I) - XALPH(J)	FIN0058
	DO 25 J=1,N	FIN0059
	U(J) = DDOT (N, COV(1, J), 1, YDIFF(1), 1)	FINO060
25	CONTINUE	FINOO61 FINOO62
	DMH(I) = DDOT(N, U(1), 1, YDIFF(1), 1)	FIN0063
	IF (DMH(I).GT.DMAX) THEN	FIN0064
	DMAX=DMH(I) INDEX=I	FIN0065
	ENDIF	FIN0066
30	CONTINUE	FIN0067
30	PV=PVCHI (DMAX, N, DM)	FIN0068
	IF(PV.LT.SIGNIF) THEN	FIN0069
	IUSE (INDEX)=0	FIN0070
	CALL FITRCB(N,M,P,K)	FIN0071
	GO TO 1000	FIN0072
	ELSE	FIN0073
	RETURN	FIN0074
	ENDIF	FIN0075
	END	FIN0076 FIN00770
		FIN00780
		FIN00790
	FUNCTION PVCHI(DMAX,N,DM)	FIN0080
с	WRITTEN BY D MOHR 10/1/89	FIN0081
c	RETURNS PROB. MAX OF M INDEP CHI-SQUARED VARIATES	FIN0082
č	(EACH WITH N D.F.) WILL BE GREATER THAN D.	FIN0083
•	IMPLICIT REAL*8 (A-H,O-Z)	FINCO84
	INTEGER N	FIN0085
	DATA DL7/356675/	FIN0086
	RN2=DFLOAT(N)/2.	FIN0087
	DM2=DMAX/2.	FINOD88
	PV=GAMMP(RN2,DM2)	FIN0089
	PV=DLOG (PV)	FIN0090
	DL7M=DL7/DM	FINOO91 FINOO92
	IF (DL7M.GT.PV) THEN	FIN0092 FIN0093
	PVCHI=, 3	FIN0094
	ELSE PV=DEXP(PV*DM)	FIN0095
	PVCHI=1PV	FIN0096
	ENDIF	<b>FIN0097</b>
	RETURN	FIN0098
	END	FIN0099
С		FIN0100
	FUNCTION GAMMQ(A,X)	FIN0101
С	FROM 'NUMERICAL RECIPES'	FIN0102
	IMPLICIT REAL*8 (A-H,O-Z)	FIN0103
	IF(X.LT.OOR.A.LE.O.) PAUSE	FIN0104
	IF(X.LT.A+1.)THEN	FIN0105
	CALL GSER (GAMSER, A, X, GLN)	FIN0106 FIN0107
	GAMMQ=1GAMSER	FINO108
	ELSE CALL GCF(GAMMCF, A, X, GLN)	FIN0109
	GAMMQ=GAMMCF	FIN0110
	ENDIF	FINO111
	RETURN	FIN0112
	END	FIN0113
с		FINO114
	SUBROUTINE GSER(GAMSER, A, X, GLN)	FIN0115
с	FROM 'NUMERICAL RECIPES'	FIN0116
	PARAMETER (ITMAX=100,EPS=3.E-7)	FIN0117

		44
	IMPLICIT REAL*8 (A-H,O-Z)	FIN0118
	GLN=GAMMLN(A)	FIN0119
	IF (X.LE.O.) THEN	FIN0120
	IF(X.LT.O.) PAUSE	FIN0121
	GAMSER=0.	FIN0122
	RETURN	FIN0123
	ENDIF	FIN0124
	AP=A	FIN0125
	SUM=1./A	FIN0126
	DEL=SUM	FIN0127
	DO 11 N=1,ITMAX	FIN0128
	AP=AP+1.	FIN0129
	DEL=DEL*X/AP	FINO130 FINO131
	SUM=SUM+DEL	FIN0131
	IF (ABS (DEL).LT.ABS (SUM) *EPS) GO TO 1	FIN0132 FIN0133
11	CONTINUE	FIN0134
	PAUSE 'A TOO LARGE, ITMAX TOO SMALL'	FIN0135
1	GAMSER=SUM*EXP(-X+A*LOG(X)-GLN)	FIN0136
	RETURN END	FIN0137
с	EMD	FIN0138
C	SUBROUTINE GCF(GAMMCF, A, X, GLN)	FIN0139
с	FROM 'NUMERICAL RECIPES'	FIN0140
U U	PARAMETER (ITMAX=100, EPS=3.E-7)	FIN0141
	IMPLICIT REAL*8 (A-H, O-Z)	FIN0142
	GLN=GAMMLN(A)	FIN0143
	GOLD=0.	FIN0144
	A0=1.	FIN0145
	Al=X	FIN0146
	B0=0.	FIN0147
	B1=1.	FIN0148
	FAC=1.	FINO149
	DO 11 N=1, ITMAX	FIN0150
	AN=FLOAT(N)	FINO151 FINO152
	ANA=AN-A	FIN0152
	A0 = (A1 + A0 * ANA) * FAC	FIN0154
	BO = (B1 + B0 * ANA) * FAC	FIN0155
	ANF=AN*FAC A1=X*A0+ANF*A1	FIN0156
	B1=X*B0+ANF*B1	FIN0157
	IF (A1, NE, O.) THEN	FIN0158
	FAC=1./A1	FIN0159
	G=B1*FAC	FIN0160
	IF(ABS((G-GOLD)/G).LT.EPS)GO TO 1	FIN0161
	GOLD=G	FINO162
	ENDIF	FIN0163
11	CONTINUE	FIN0164
	PAUSE 'A TOO LARGE, ITMAX TOO SMALL'	FIN0165
1	GAMMCF=EXP(-X+A*DLOG(X)-GLN)*G	FIN0166
	RETURN	FIN0167
	END	FIN0168
C		FINO169
_	FUNCTION GAMMLN(XX)	FINO170
С	FROM 'NUMERICAL RECIPES'	FIN0171
	IMPLICIT REAL*8 (A-H,O-Z)	FINO172 FINO173
	REAL*8 COF(6), STP, HALF, ONE, FPF, X, TMP, SER	FINO173
	DATA COF, STP/76.18009173D0, -86.50532033D0,24.01409822D0, * -1.231739516D0,.120858003D-2,536382D-5,2.50662827465D0/	FIN0175
	<pre>* -1.231739516D0,.120858003D-2,536382D-5,2.50662827465D0/ DATA HALF,ONE,FPF/0.5D0,1.0D0,5.5D0/</pre>	FINO176
	X=XX-ONE	FIN0177

		45	
	TMP=X+FPF		FIN0178
	TMP = (X + HALF) + LOG (TMP) - TMP		FIN0179
	SER=ONE		FIN0180
	DO 11 J=1,6		FIN0181
	X=X+ONE		<b>FIN0182</b>
	SER=SER+COF(J)/X		FIN0183
11	CONTINUE		FIN0184
	GAMMLN=TMP+LOG (STP*SER)		FIN0185
	RETURN		FIN0186
	END		FIN0187
С			FIN0188
	FUNCTION GAMMP(A,X)		<b>FIN0189</b>
С	FROM 'NUMERICAL RECIPES'		FIN0190
	IMPLICIT REAL*8 (A-H,O-Z)		FIN0191
	IF (X.LT.OOR.A.LE.O.) PAUSE		<b>FIN0192</b>
	IF (X.LT.A+1.) THEN	-	FIN0193
	CALL GSER (GAMSER, A, X, GIN)		FIN0194
	GAMMP=GAMSER		FIN0195
	ELSE		FIN0196
	CALL GCF(GAMMCF, A, X, GLN)		FIN0197
	GAMMP≈1GAMMCF		FIN0198
	ENDIF		FIN0199
	RETURN		FIN0200
	END		FIN0201

## **REFERENCES**

- Dempster, A.P., Laird, N. M. and Rubin, D. B. (1977), "Maximum Likelihood with Incomplete Data Via the E-M Algorithm," Journal of the Royal Statistical Society, 39, pp. 1-38.
- Diem, John E., and Liukkonen, John R. (1988), "A Comparative Study of Three Methods for Analyzing Longitudinal Pulmonary Function Data," Statistics in Medicine, 7, pp. 19-28.
- Laird, Nan M. and Ware, James H. (1982), "Random-Effects Models for Londitudinal Data," Biometrics, 38, pp. 963-974.
- Press, W. H., Flannery, B. P., Teukolsky, S.A., and Vetterling, W. T., <u>Numerical Recipes: The Art of</u> <u>Scientific Computing</u>, Cambridge University Press, Cambridge, 1986

- Ware, James H. (1984), "Linear Models for the Analysis of Longitudinal Studies," The American Statistician, 39,pp. 95-101.
- Ware, James H. and Wu, Margaret C. (1981), "Tracking: Prediction of Future Values from Serial Measurements," Biometrics, 37, pp. 427-437.

## <u>VITA</u>

Tamarah Crouse Dishman

Educational

University of Maryland, Baltimore County Loyola College in Baltimore - B.S. in Mathematics 1985 University of West Florida (UWF) - Graduate Studies University of North Florida (UNF) - Graduate Studies

**Professional** 

UWF - Graduate Teaching Assistantship UNF - Graduate Teaching Assistantship

Honors

National Honor Society - Andover High School Who's Who among American High School Graduates Pi Mu Epsilon Who's Who Among Students in American Universities and Colleges