

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

IDENTIFYING 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

12/14/89

Signature deleted

12/14/89

Signature deleted

12/14/89

Committee Chairperson

Accepted for The Department:

Signature deleted

12/14/89

Chairperson

Accepted for the College:

Signature deleted

12/14/89

Dean

Accepted for the University:

Signature deleted

12/14/89

Interim Vice-President for Academic Affairs

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

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

List of Tables and Figures

	(page)
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 a 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 methods. The Mahalanobis distance was then 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 α . 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

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

where \mathbf{y}_i is the vector of n_i observations from individual i , $\boldsymbol{\alpha}$ is a $p \times 1$ vector of the unknown population parameters, \mathbf{X}_i is a known design matrix linking $\boldsymbol{\alpha}$ to \mathbf{y}_i for each individual, \mathbf{b}_i is the $k \times 1$ vector of individual effects and \mathbf{Z}_i is the known design matrix linking \mathbf{b}_i to \mathbf{y}_i for each individual. The \mathbf{e}_i vectors are distributed $N(\mathbf{0}, \mathbf{R}_i)$ and assumed to be independent while $\boldsymbol{\alpha}$ is considered fixed and \mathbf{b}_i is a random vector as described in stage 2. Throughout the rest of our work we take $\mathbf{R}_i = \sigma^2 \mathbf{I}$.

Stage 2

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

The \mathbf{y}_i are independent and distributed $N(\mathbf{X}_i \boldsymbol{\alpha}, \mathbf{Z}_i \mathbf{D} \mathbf{Z}_i^T + \sigma^2 \mathbf{I})$. The main disadvantage of this model is the strong assumption made about the structure of the covariance matrix of the \mathbf{y}_i given above.

Section 2: Estimation of parameters

In this section, equations for estimating $\boldsymbol{\alpha}, \sigma^2$ and \mathbf{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 Algorithm (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 , $\hat{\mathbf{b}}_i$. This estimate is calculated using current estimates of $\boldsymbol{\alpha}, \sigma^2, \mathbf{D}$.
2. In the M-step, parameters $\boldsymbol{\alpha}, \sigma^2$ and \mathbf{D} are estimated using the \mathbf{y}_i and \mathbf{b}_i .

The algorithm is repeated until convergence is obtained or the maximum allowed iterations is reached. The

derivation of the equations is as follows.

E-Step

First note that the joint probability distribution for \mathbf{y}_i and \mathbf{b}_i given $\theta = (\boldsymbol{\alpha}, \sigma^2, \mathbf{D})$ is given by:

$$\begin{aligned} f(\mathbf{y}_i, \mathbf{b}_i | \theta) &= f(\mathbf{y}_i | \mathbf{b}_i, \theta) \cdot f(\mathbf{b}_i | \theta) \\ &= c_1 \cdot \frac{1}{\det|\sigma^2 \mathbf{I}|^{1/2}} \exp\left\{\frac{-1}{2\sigma^2} (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\alpha} - \mathbf{Z}_i \mathbf{b}_i)^T (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\alpha} - \mathbf{Z}_i \mathbf{b}_i)\right\} \\ &\quad \cdot \frac{1}{\det|\mathbf{D}|^{1/2}} \exp\left\{\frac{-1}{2} (\mathbf{b}_i^T \mathbf{D}^{-1} \mathbf{b}_i)\right\} \end{aligned}$$

where c_1 is a constant

NOTE:

$$\begin{aligned} (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\alpha} - \mathbf{Z}_i \mathbf{b}_i)^T (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\alpha} - \mathbf{Z}_i \mathbf{b}_i) &= [(\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\alpha})^T - (\mathbf{Z}_i \mathbf{b}_i)^T] [(\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\alpha}) - \\ &\quad (\mathbf{Z}_i \mathbf{b}_i)] \\ &= (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\alpha})^T (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\alpha}) - 2 (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\alpha})^T \mathbf{Z}_i \mathbf{b}_i + \mathbf{b}_i^T \mathbf{Z}_i^T \mathbf{Z}_i \mathbf{b}_i \end{aligned}$$

Now what we need is the conditional pdf of \mathbf{b}_i given θ

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

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,

$$\begin{aligned}
f(\mathbf{b}_i | \mathbf{y}_i, \theta) &= c_2 \exp \left\{ \frac{1}{\sigma^2} (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\alpha})^T \mathbf{Z}_i \mathbf{b}_i \right\} \\
&\quad \cdot \exp \left\{ - \frac{1}{2\sigma^2} (\mathbf{b}_i^T \mathbf{Z}_i^T \mathbf{Z}_i \mathbf{b}_i) - \frac{1}{2} (\mathbf{b}_i^T \mathbf{D}^{-1} \mathbf{b}_i) \right\} \\
&= c_2 \exp \left\{ \frac{1}{2\sigma^2} \left[2(\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\alpha})^T \mathbf{Z}_i \mathbf{b}_i - \mathbf{b}_i^T (\mathbf{Z}_i^T \mathbf{Z}_i + \mathbf{D}^{-1} \sigma^2) \mathbf{b}_i \right] \right\} \\
&= c_2 \exp \left\{ \frac{-1}{2} \left[\frac{-2(\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\alpha})^T \mathbf{Z}_i \mathbf{b}_i + \mathbf{b}_i^T (\mathbf{Z}_i^T \mathbf{Z}_i + \mathbf{D}^{-1} \sigma^2) \mathbf{b}_i}{\sigma^2} \right] \right\} \\
&= c_2 \exp \left\{ \frac{-1}{2} \left[\mathbf{b}_i^T \mathbf{A} \mathbf{b}_i - 2(\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\alpha})^T \mathbf{Z}_i \mathbf{b}_i / \sigma^2 \right] \right\}
\end{aligned}$$

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

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

From Appendix 1, it follows that:

$$\mathbb{E}(\mathbf{b}_i | \mathbf{y}_i, \theta) = (\mathbf{z}_i^T \mathbf{z}_i + \mathbf{D}^{-1} \sigma^2)^{-1} \mathbf{z}_i^T (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\alpha}) = \hat{\mathbf{b}}_i \quad (1)$$

M-Step

All sums below are over $i=1, m$.

Using the values $\hat{\mathbf{b}}_i$ calculated in the E-step we want to maximize

$H(\theta) = \mathbb{E}(\ln[f(\mathbf{y}_i, \mathbf{b}_i | \theta)] | \mathbf{y}_i, \theta)$, ignoring constants it follows that:

$$= \mathbb{E} \left\{ \sum_{i=1}^m \left[\frac{-n_i}{2} \ln(\sigma^2) - \frac{1}{2\sigma^2} (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\alpha} - \mathbf{z}_i \mathbf{b}_i)^T (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\alpha} - \mathbf{z}_i \mathbf{b}_i) - \frac{1}{2} \ln(\det \mathbf{D}) - \frac{1}{2} (\mathbf{b}_i^T \mathbf{D}^{-1} \mathbf{b}_i) \right] \right\} =$$

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

$$\begin{aligned}
&= \frac{-N}{2} \ln \sigma^2 - \frac{1}{2\sigma^2} \sum^m (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\alpha})^T (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\alpha}) + \frac{1}{\sigma^2} \sum^m [(\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\alpha})^T \mathbf{z}_i E(\mathbf{b}_i | \mathbf{y}_i, \theta)] \\
&\quad - \frac{m}{2} \ln(\det \mathbf{D}) - \frac{1}{2\sigma^2} E[\mathbf{b}_i^T (\mathbf{z}_i^T \mathbf{z}_i + \sigma^2 \mathbf{D}^{-1}) \mathbf{b}_i | \mathbf{y}_i, \theta] \\
&= \frac{-N}{2} \ln \sigma^2 - \frac{m}{2} \ln(\det \mathbf{D}) - \frac{1}{2\sigma^2} \sum^m (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\alpha})^T (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\alpha}) + \frac{1}{\sigma^2} \sum^m (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\alpha})^T \mathbf{z}_i \hat{\mathbf{b}}_i \\
&\quad - \frac{1}{2\sigma^2} \sum^m \{ \text{tr}[(\mathbf{z}_i^T \mathbf{z}_i + \sigma^2 \mathbf{D}^{-1}) \mathbf{V}(\mathbf{b}_i | \mathbf{y}_i, \theta)] + \hat{\mathbf{b}}_i^T (\mathbf{z}_i^T \mathbf{z}_i + \sigma^2 \mathbf{D}^{-1}) \hat{\mathbf{b}}_i \} \quad (2)
\end{aligned}$$

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

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

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 $\hat{\alpha}$:

Note:

$$\begin{aligned}
 -\frac{1}{2\sigma^2} \sum_{i=1}^m (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\alpha})^T (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\alpha}) &= \frac{-1}{2\sigma^2} \sum_{i=1}^m \mathbf{y}_i^T \mathbf{y}_i + \frac{1}{\sigma^2} \sum_{i=1}^m \mathbf{y}_i^T \mathbf{X}_i \boldsymbol{\alpha} \\
 &+ \frac{-1}{2\sigma^2} \sum_{i=1}^m \boldsymbol{\alpha}^T \mathbf{X}_i^T \mathbf{X}_i \boldsymbol{\alpha}
 \end{aligned}$$

and,

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

therefore,

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

it follows that,

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

and,

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

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

$$\begin{aligned} \frac{\partial H(\theta)}{\partial \sigma^2} &= \frac{-N}{2\sigma^2} + \frac{\sum_{i=1}^m [(\mathbf{y}_i - \mathbf{x}_i \alpha)^T (\mathbf{y}_i - \mathbf{x}_i \alpha)]}{2\sigma^4} - \frac{\sum_{i=1}^m [(\mathbf{y}_i - \mathbf{x}_i \alpha)^T \mathbf{z}_i \hat{\mathbf{b}}_i]}{\sigma^4} \\ &+ \frac{\sum_{i=1}^m \text{tr}[\mathbf{z}_i^T \mathbf{z}_i \text{V}(\mathbf{b}_i | \mathbf{y}_i, \theta)]}{2\sigma^4} + \frac{\sum_{i=1}^m \mathbf{b}_i \mathbf{z}_i^T \mathbf{z}_i \hat{\mathbf{b}}_i}{2\sigma^4} = 0 \end{aligned}$$

it follows that,

$$\begin{aligned} \sigma^{2N} - \sum_{i=1}^m [(\mathbf{y}_i - \mathbf{x}_i \alpha)^T (\mathbf{y}_i - \mathbf{x}_i \alpha)] + 2 \sum_{i=1}^m (\mathbf{y}_i - \mathbf{x}_i \alpha)^T \mathbf{z}_i \hat{\mathbf{b}}_i \\ - \sum_{i=1}^m \text{tr}[\mathbf{z}_i^T \mathbf{z}_i \text{V}(\mathbf{b}_i | \mathbf{y}_i, \theta)] - \sum_{i=1}^m \hat{\mathbf{b}}_i^T \mathbf{z}_i^T \mathbf{z}_i \hat{\mathbf{b}}_i = 0 \end{aligned}$$

therefore,

$$\begin{aligned} \hat{\sigma}^2 = & \frac{1}{N} \sum^m \{ (\mathbf{y}_i - \mathbf{x}_i \boldsymbol{\alpha})^T (\mathbf{y}_i - \mathbf{x}_i \boldsymbol{\alpha}) - 2 (\mathbf{y}_i - \mathbf{x}_i \boldsymbol{\alpha})^T \mathbf{z}_i \hat{\mathbf{b}}_i \\ & + \text{tr}[\mathbf{z}_i^T \mathbf{z}_i \mathbf{v}(\mathbf{b}_i | \mathbf{y}_i, \theta)] + \hat{\mathbf{b}}_i^T \mathbf{z}_i^T \mathbf{z}_i \hat{\mathbf{b}}_i \} \end{aligned} \quad (4)$$

Finally consider $\hat{\mathbf{D}}$:

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

$$\frac{\partial H(\theta)}{\partial \mathbf{D}} = \frac{-m}{2} \mathbf{D}^{-1} + \frac{1}{2} \mathbf{D}^{-1} \sum^m \hat{\mathbf{b}}_i \hat{\mathbf{b}}_i^T \mathbf{D}^{-1} + \frac{1}{2} \mathbf{D}^{-1} \sum^m \mathbf{v}(\mathbf{b}_i | \mathbf{y}_i, \theta) \mathbf{D}^{-1} = 0$$

it follows that,

$$\begin{aligned} -m\mathbf{D} + \sum^m \hat{\mathbf{b}}_i \hat{\mathbf{b}}_i^T + \sum^m \mathbf{v}(\mathbf{b}_i | \mathbf{y}_i, \theta) &= 0 \\ \Rightarrow \hat{\mathbf{D}} = \frac{1}{m} \sum^m [\hat{\mathbf{b}}_i \hat{\mathbf{b}}_i^T + \mathbf{v}(\mathbf{b}_i | \mathbf{y}_i, \theta)] & \quad (5) \end{aligned}$$

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 non-trackers, 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 ,

$$\begin{aligned} D_i &= (\mathbf{y}_i - \mu)^T (\text{var } \mathbf{y})^{-1} (\mathbf{y}_i - \mu) \\ &= (\mathbf{y}_i - \mathbf{X}_i \hat{\alpha})^T (\mathbf{Z}_i \hat{\mathbf{D}} \mathbf{Z}_i^T + \hat{\sigma}^2 \mathbf{I})^{-1} (\mathbf{y}_i - \mathbf{X}_i \hat{\alpha}) \end{aligned} \quad (6)$$

since we assume that each individual is normally distributed with mean $\mathbf{X}_i \alpha$ and variance $\mathbf{Z}_i \mathbf{D} \mathbf{Z}_i^T + \sigma^2 \mathbf{I}$. If α , σ^2 and \mathbf{D} were known and used in place of their estimates in equation

(6), clearly, D_i would have a chi-square distribution with n degrees of freedom where n is the dimension of \mathbf{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 D_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 non-trackers.

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 parameters calculated from equations 1-5. Examination of the probability distribution of the maximum order statistic for this type of distribution leads to a p-value expressed as

$$p\text{-value} = 1 - F(d_{\max})^m$$

where d_{\max} stands for maximum D_i from $\chi^2(n)$ and $F(d_{\max})$ equals the cumulative distribution function. The program for computing $F(d_{\max})$ 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 σ^2 , significance levels and values used for X , Z and α 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 non-tracker 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_i 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 non-trackers 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 Trackersn=5

$$\mathbf{X} = \begin{pmatrix} 1 & -2 & 0 \\ 1 & -1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 2 & 1 \end{pmatrix} \quad \mathbf{Z} = \begin{pmatrix} 1 & -2 \\ 1 & -1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{pmatrix} \quad \boldsymbol{\alpha} = (5, 3, 2)$$

$$\mathbf{D} = \begin{pmatrix} 0.5 & 0.0 \\ 0.0 & 0.5 \end{pmatrix}$$

The two settings for σ^2 are 0.5 and 2.0 .

n=10

$$\mathbf{X} = \begin{pmatrix} 1 & -4 & 0 \\ 1 & -3 & 0 \\ 1 & -2 & 0 \\ 1 & -1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 3 & 1 \\ 1 & 4 & 1 \\ 1 & 5 & 1 \end{pmatrix} \quad \mathbf{Z} = \begin{pmatrix} 1 & -4 \\ 1 & -3 \\ 1 & -2 \\ 1 & -1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \end{pmatrix} \quad \boldsymbol{\alpha} = (5, 3, 2)$$

$$\mathbf{D} = \begin{pmatrix} 0.5 & 0.0 \\ 0.0 & 0.5 \end{pmatrix}$$

The two settings for σ^2 are 2.0 and 8.0

Table 2 -Simulations Run with only Trackers Present

m	n	σ^2	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

Figure 1 - Graph of Expected Values for Non-Trackers

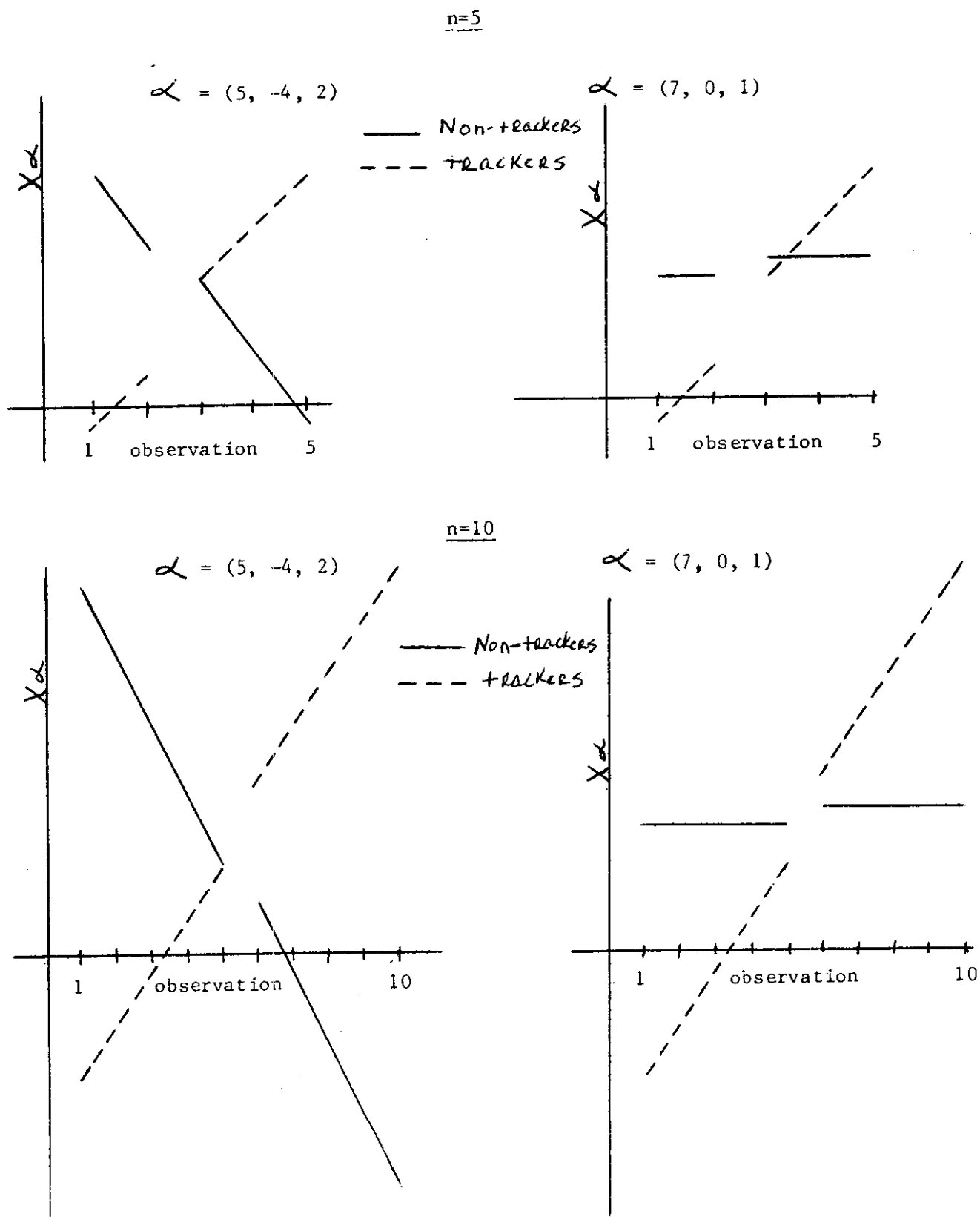


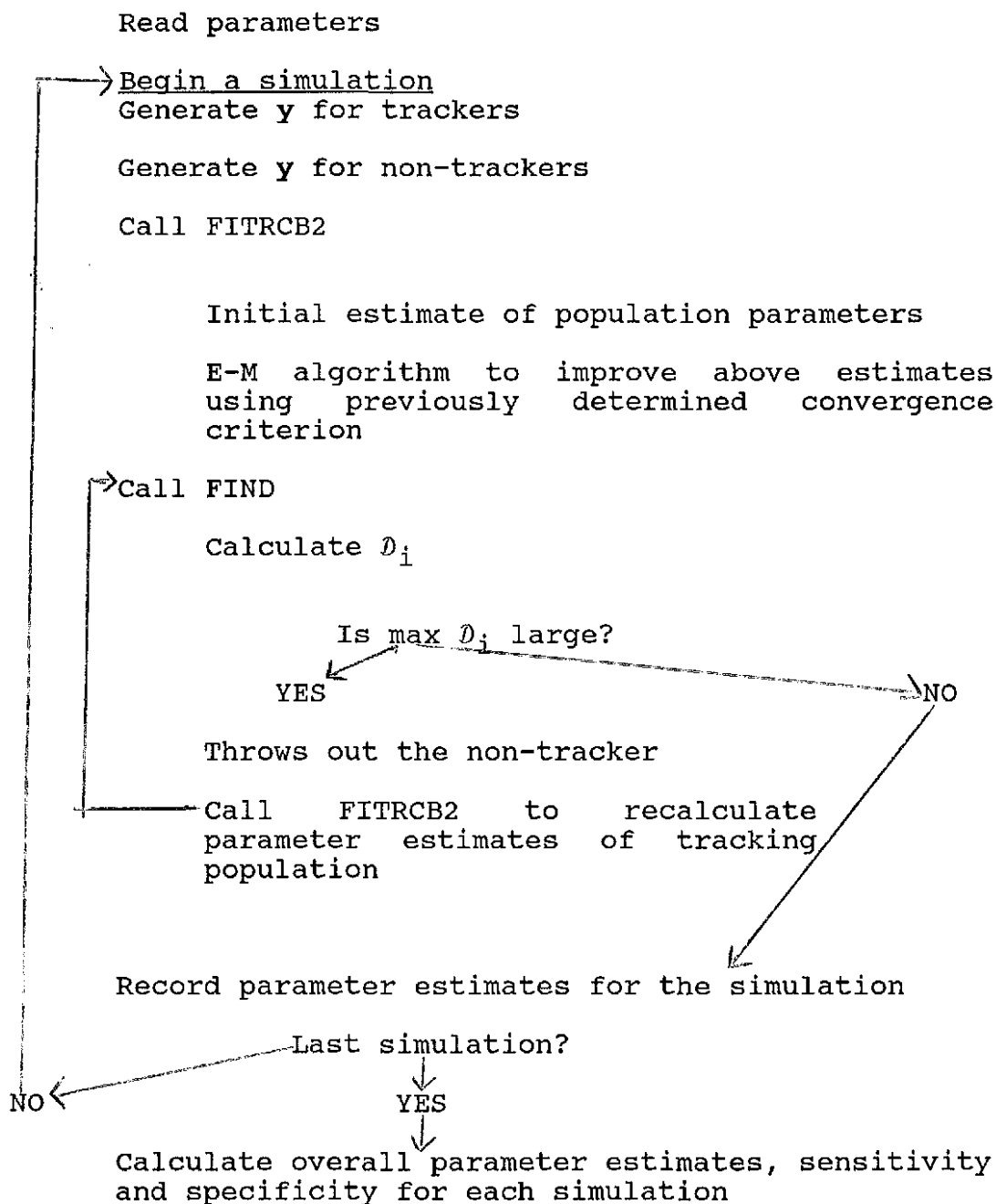
Table 3 - Simulations Run with Non-Trackers Present

m	n	ntr	σ^2	α
25	5	1	0.5	A
25	5	1	0.5	B
25	5	1	2.0	A
25	5	1	2.0	B
25	5	2	0.5	A
25	5	2	0.5	B
25	5	2	2.0	A
25	5	2	2.0	B
25	10	1	2.0	A
25	10	1	2.0	B
25	10	1	8.0	A
25	10	1	8.0	B
25	10	2	2.0	A
25	10	2	2.0	B
25	10	2	8.0	A
25	10	2	8.0	B
50	5	1	0.5	A
50	5	1	0.5	B
50	5	1	2.0	A
50	5	1	2.0	B
50	5	3	0.5	A
50	5	3	0.5	B
50	5	3	2.0	A
50	5	3	2.0	B
50	10	1	2.0	A
50	10	1	2.0	B
50	10	1	8.0	A
50	10	1	8.0	B
50	10	3	2.0	A
50	10	3	2.0	B
50	10	3	8.0	A
50	10	3	8.0	B

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



Chapter 3 - Conclusion

The first observation, obvious by looking at Table 4, is that when no non-trackers are present the algorithm is excellent. The average specificity among repetitions stays 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 α , σ^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 p-value for $\max D_i$ 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, $\alpha=(5,-4,2)$, than case B, $\alpha=(7,0,1)$, (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 σ^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_i 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.

Table 4 - Results with Trackers Only

m	n	σ^2	signif	spec	std dev
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)

Case 1: m=25 n=5 $\sigma^2=2.0$

$\hat{\sigma}^2 = 1.8791$ with std dev = 0.2628

$$\hat{\mathbf{D}} = \begin{pmatrix} 0.4755 & 0.0244 \\ 0.0244 & 0.4934 \end{pmatrix} \quad \hat{\boldsymbol{\alpha}} = \begin{pmatrix} \text{avg} \\ 4.9823 \\ 3.0287 \\ 1.9332 \end{pmatrix} \begin{pmatrix} \text{std dev} \\ 0.2869 \\ 0.2538 \\ 0.5535 \end{pmatrix}$$

Case 2: m=50 n=5 $\sigma^2=2.0$

$\hat{\sigma}^2 = 1.9458$ with std dev = 0.1986

$$\hat{\mathbf{D}} = \begin{pmatrix} 0.4949 & 0.0149 \\ 0.0149 & 0.5041 \end{pmatrix} \quad \hat{\boldsymbol{\alpha}} = \begin{pmatrix} \text{avg} \\ 5.0120 \\ 3.0294 \\ 1.9257 \end{pmatrix} \begin{pmatrix} \text{std dev} \\ 0.1964 \\ 0.1598 \\ 0.3833 \end{pmatrix}$$

Case 3: m=25 n=10 $\sigma^2=8.0$

$\hat{\sigma}^2 = 7.9065$ with std dev = 0.8343

$$\hat{\mathbf{D}} = \begin{pmatrix} 0.4964 & -0.0025 \\ -0.0025 & 0.4803 \end{pmatrix} \quad \hat{\boldsymbol{\alpha}} = \begin{pmatrix} \text{avg} \\ 4.9264 \\ 3.0210 \\ 2.0334 \end{pmatrix} \begin{pmatrix} \text{std dev} \\ 0.4290 \\ 0.1702 \\ 0.7609 \end{pmatrix}$$

Case 4: m=50 n=10 $\sigma^2=2.0$

$\hat{\sigma}^2 = 1.9827$ with std dev = 0.1313

$$\hat{\mathbf{D}} = \begin{pmatrix} 0.4897 & -0.0217 \\ -0.0217 & 0.4970 \end{pmatrix} \quad \hat{\boldsymbol{\alpha}} = \begin{pmatrix} \text{avg} \\ 4.9841 \\ 2.0805 \\ 2.0104 \end{pmatrix} \begin{pmatrix} \text{std dev} \\ 0.1870 \\ 0.1085 \\ 0.2676 \end{pmatrix}$$

True parameter values for the above are:

signif = 0.10

$$\mathbf{D} = \begin{pmatrix} 0.5 & 0.0 \\ 0.0 & 0.5 \end{pmatrix} \quad \boldsymbol{\alpha} = \begin{pmatrix} 5 \\ 3 \\ 2 \end{pmatrix}$$

Table 6 - Results with Non-Trackers Present

m	n	ntr	σ^2	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 $\alpha=(5, -4, 2)$; B indicates $\alpha=(7,0,1)$. Ntr stands for the number of 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

$$\hat{D} = \begin{pmatrix} 0.5048 & -0.0211 \\ -0.0211 & 3.5484 \end{pmatrix} \quad \alpha = \begin{pmatrix} \text{avg} \\ 4.9932 \\ 2.5188 \\ 2.06585 \end{pmatrix} \begin{pmatrix} \text{std dev} \\ 0.2935 \\ 0.2923 \\ 0.6356 \end{pmatrix}$$

Case 2: m=50 n=5 NTR=3 $\sigma^2=2.0$

$\hat{\sigma}^2 = 1.8538$ with std dev = 0.2443

$$\hat{D} = \begin{pmatrix} 0.5280 & 0.0032 \\ 0.0032 & 1.4190 \end{pmatrix} \quad \alpha = \begin{pmatrix} \text{avg} \\ 4.9827 \\ 2.8683 \\ 1.9940 \end{pmatrix} \begin{pmatrix} \text{std dev} \\ 0.2001 \\ 0.2693 \\ 0.3849 \end{pmatrix}$$

Case 3: m=25 n=10 NTR=2 $\sigma^2=8.0$

$\hat{\sigma}^2 = 7.8287$ with std dev = 0.7751

$$\hat{D} = \begin{pmatrix} 0.4989 & -0.0948 \\ -0.0948 & 3.6280 \end{pmatrix} \quad \alpha = \begin{pmatrix} \text{avg} \\ 5.0112 \\ 2.5055 \\ 1.9160 \end{pmatrix} \begin{pmatrix} \text{std dev} \\ 0.3302 \\ 0.2887 \\ 0.6536 \end{pmatrix}$$

Case 4: m=50 n=10 NTR=3 $\sigma^2=2.0$

$\hat{\sigma}^2 = 1.9294$ with std dev = 0.1603

$$\hat{D} = \begin{pmatrix} 0.5008 & 0.0145 \\ 0.0145 & 2.6450 \end{pmatrix} \quad \alpha = \begin{pmatrix} \text{avg} \\ 4.9727 \\ 2.6678 \\ 2.0182 \end{pmatrix} \begin{pmatrix} \text{std dev} \\ 0.1605 \\ 0.2053 \\ 0.2584 \end{pmatrix}$$

True parameter values for the above are:

signif = 0.10

$$D = \begin{pmatrix} 0.5 & 0.0 \\ 0.0 & 0.5 \end{pmatrix} \quad \alpha = \begin{pmatrix} 5 \\ 3 \\ 2 \end{pmatrix}$$

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 μ directly.

Since,

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

and we have,

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

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

$$-2\mu^T \mathbf{A} \mathbf{b}_i = \frac{-2 \left(\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\alpha} \right)^T \mathbf{z}_i \mathbf{b}_i}{\sigma^2}$$

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

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

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

Appendix 2

We must show that $\frac{\partial}{\partial \mathbf{D}} \mathbf{b}_i^T \mathbf{D}^{-1} \mathbf{b}_i = - \mathbf{D}^{-1} \mathbf{b}_i \mathbf{b}_i^T \mathbf{D}$

We use the fact that $\mathbf{b}_i^T \mathbf{D}^{-1} \mathbf{b}_i = \text{tr } \mathbf{D}^{-1} \mathbf{b}_i \mathbf{b}_i^T = \text{tr } \mathbf{D}^{-1} \mathbf{A}$ where $\mathbf{A} = \mathbf{b}_i \mathbf{b}_i^T$. Both \mathbf{D}^{-1} and \mathbf{A} are symmetric. We know that

$$\frac{\partial d^{ij}}{\partial d_{rc}} = - d^{ir} d^{cj} \quad \text{where } d_{rc} \text{ is the } r, c^{\text{th}}$$

entry in \mathbf{D} and d^{ir} is the i, r^{th} entry in \mathbf{D}^{-1} . Then

$$\begin{aligned} \frac{\partial}{\partial d_{rc}} \mathbf{b}_i^T \mathbf{D}^{-1} \mathbf{b}_i &= \frac{\partial}{\partial d_{rc}} \text{tr } \mathbf{D}^{-1} \mathbf{A} = \frac{\partial}{\partial d_{rc}} \sum_i \sum_l d^{il} a_{li} \\ &= - \sum_i \sum_l d^{ir} d^{cl} a_{li} = - \sum_i d^{ir} \left(\sum_l d^{cl} a_{li} \right) \end{aligned}$$

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

$$= - \sum_i d^{ri} (\mathbf{A} \mathbf{D}^{-1})_{ic}$$

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

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

Appendix 3 - Programs for Computer Simulations

```

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

```

		31
15	CONTINUE	RAN0052
C	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
C	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,*) (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
C	CALL CHOLESKY DECOMPOSITION TO FACTOR TD=(TQ)*(TQ)	RAN0075
	DO 75 I=1,K	RAN0076
	DO 72 J=1,K	RAN0077
	TQ(I,J)=TD(I,J)	RAN0078
72	CONTINUE	RAN0079
75	CONTINUE	RAN0080
	JOB=0	RAN0081
	LDA=6	RAN0082
	CALL DCHDC(TQ, LDA, K, WORK, JPV, JOB, INFO)	RAN0083
C	WRITE (6,*) INFO	RAN0084
	DO 78 I=1,K-1	RAN0085
	DO 77 J=I+1,K	RAN0086
	TQ(J,I)=TQ(I,J)	RAN0087
	TQ(I,J)=0.DO	RAN0088
77	CONTINUE	RAN0089
78	CONTINUE	RAN0090
C		TRAN
C	CALL CHOLESKY DECOMPOSITION TO FACTOR TDN=(TQN)*(TQN)	RAN0091
	DO 95 I=1,KN	RAN0092
	DO 92 J=1,KN	RAN0093
	TQN(I,J)=TDN(I,J)	RAN0094
92	CONTINUE	RAN0095
95	CONTINUE	RAN0096
	JOB=0	RAN0097
	LDA=6	RAN0098
	CALL DCHDC(TQN, LDA, KN, WORK, JPV, JOB, INFO)	RAN0099
C	WRITE (6,*) INFO	RAN0100
	DO 98 I=1,KN-1	RAN0101
	DO 97 J=I+1,KN	RAN0102
	TQN(J,I)=TQN(I,J)	RAN0103
	TQN(I,J)=0.DO	RAN0104
97	CONTINUE	RAN0105
98	CONTINUE	RAN0106
C		RAN0107
C	STORE MEAN VECTOR X*ALPHA FOR TRACKERS	RAN0108
	DO 150 I=1,N	RAN0109
		RAN0110
		RAN0111

			32
		XA(I)=0.DO	RAN0112
		DO 145 J=1,P	RAN0113
		XA(I)=XA(I)+X(I,J)*TALPH(J)	RAN0114
145		CONTINUE	RAN0115
150		CONTINUE	RAN0116
C			RAN0117
C		STORE MEAN VECTOR XN*ALPHAN FOR NONTRACKERS	RAN0118
		DO 190 I=1,N	RAN0119
		XAN(I)=0.DO	RAN0120
		DO 185 J=1,PN	RAN0121
		XAN(I)=XAN(I)+XN(I,J)*TALPHN(J)	RAN0122
185		CONTINUE	RAN0123
190		CONTINUE	RAN0124
			RAN01250
C	+++++	BEGIN REPETITIONS, CREATING DATA FOR ++++++	RAN0126
C	+++++	TRACKERS, NONTRACKERS; FITTING MODEL ++++++	RAN0127
C	+++++	AND FINDING NON-TRACKERS	RAN0128
C			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	RAN0134
		DO 210 I=1,K	RAN0135
		DO 205 J=1,K	RAN0136
		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
		DO 3000 III=1,NUMITR	RAN0144
C			RAN0145
		DO 4 I=1,M	RAN0146
4		IUSE(I)=1	RAN0147
C	+++++	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
		B(I,II)=BT(I)	RAN0154
270		CONTINUE	RAN0155
		DO 290 I=1,N	RAN0156
		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
C			RAN0163
C	+++++	MANUFACTURE Y FOR NONTRACKERS II=MTR+1,M ++++++	RAN0164
			RAN01650
		IF (MTR.GE.M) GO TO 410	RAN0166
		DO 400 II=MTR+1,M	RAN0167
C		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

		33
370	CONTINUE	RAN0172
	DO 390 I=1,N	RAN0173
	Y(I,II)=XAN(I)+GASDEV(IDUM)*TSIG	RAN0174
	DO 380 J=1,KN	RAN0175
	Y(I,II)=Y(I,II)+ZN(I,J)*BT(J)	RAN0176
380	CONTINUE	RAN0177
390	CONTINUE	RAN0178
400	CONTINUE	RAN0179
410	CONTINUE	RAN0180
C		RAN0181
C		RAN0182
C	CALL FITTING ROUTINE FOR BALANCED DATA	RAN0183
	CALL FITRCB(N,M,P,K)	RAN0184
	CALL FINDEM(N,M,P,K,SIGNIF)	RAN0185
	WRITE(9,*) 'THE FOLLOWING INDIVIDUALS WERE IDENTIFIED AS	RAN0186
	+NON-TRACKERS'	RAN0187
	NUMTR=0	RAN0188
	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
	ELSE	RAN0199
	DO 520 I=MTR+1,M	RAN0200
520	IDCNTR=IDCNTR + (1-IUSE(I))	RAN0201
	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
C		RAN0209
	WRITE(9,*) ' ESTIMATED ALPHAS '	RAN0210
	DO 605 I=1,P	RAN0211
	WRITE(9,604) ALPH(I)	RAN0212
604	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	RAN0222
C		RAN0223
	SIGBAR=SIGBAR + SIGMA2	RAN0224
	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

640	CONTINUE	RAN0232
650	CONTINUE	RAN0233
C		RAN0234
3000	CONTINUE	RAN0235
C	+++++++ 3000 IS END OF REPETITION LOOP +++++	RAN0236
C		RAN0237
C	CALCULATE STATS FOR SENSITIVITY AND SPECIFICITY	RAN0238
C		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
736	FORMAT (' SENSITIVITY MEAN ',F8.5,' STD DEV ',F8.5)	RAN0247
750	WRITE(9,737) SPMEAN,SPSIG	RAN0248
737	FORMAT (' SPECIFICITY MEAN ',F8.5,' STD DEV ',F8.5)	RAN0249
C		RAN0250
C	WRITE OUT PARAMS FOR THE SIMULATION	RAN0251
C		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
805	WRITE(9,604) ALFDEV(I)	RAN0267
	WRITE(9,*) 'OVERALL ESTIMATE OF D'	RAN0268
	DO 820 I=1,K	RAN0269
	DO 810 J=1,K	RAN0270
	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
C		RAN0280
C	SUBROUTINE TO PRODUCE MULTIVARIATE NORMAL VECTOR BT WITH	RAN0281
C	COVARIANCE MATRIX GIVEN BY TQ*TRAN(TQ)	RAN0282
C	DEFINED LENGTH OF VECTOR IS LDA. USED LENGTH IS K.	RAN0283
	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

			35
		BT(I)=0.DO	RAN0292
		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
C		FUNCTION GASDEV PRODUCES A STANDARD NORMAL DEVIATE	RAN0299
		FUNCTION GASDEV(IDUM)	RAN0300
		IMPLICIT REAL*8 (A-H,O-Z)	RAN0301
		DATA ISET/0/	RAN0302
		IF (ISET.EQ.0) THEN	RAN0303
1		V1=2.*RAN3(IDUM)-1.	RAN0304
		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
		GASDEV=V2*FAC	RAN0310
		ISET=1	RAN0311
		ELSE	RAN0312
		GASDEV=GSET	RAN0313
		ISET=0	RAN0314
		ENDIF	RAN0315
		RETURN	RAN0316
		END	RAN0317
			RAN03180
C		FUNCTION RAN3 PRODUCES A UNIFORM (0,1) RANDOM DEVIATE	RAN0319
		FUNCTION RAN3(IDUM)	RAN0320
		IMPLICIT REAL*8 (A-H,O-Z)	RAN0321
C		IMPLICIT REAL*4(M)	RAN0322
C		PARAMETER (MBIG=4000000.,MSEED=1618033.,MZ=0.,FAC=2.5E-7)	RAN0323
C		PARAMETER (MBIG=1000000000,MSEED=161803398,MZ=0,FAC=1.E-9)	RAN0324
		DIMENSION MA(55)	RAN0325
		DATA IFF /0/	RAN0326
		IF (IDUM.LT.0.OR.IFF.EQ.0) THEN	RAN0327
		IFF=1	RAN0328
		MJ=MSEED-IABS(IDUM)	RAN0329
		MJ=MOD(MJ,MBIG)	RAN0330
		MA(55)=MJ	RAN0331
		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
11		CONTINUE	RAN0339
		DO 13 K=1,4	RAN0340
		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
		ENDIF	RAN0349
		INEXT=INEXT+1	RAN0350
		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
```

```

C      FITRCB IS THE ROUTINE FOR FITTING THE POPULATION          FIT0001
C      PARAMETERS FOR TRACKERS.  THE EQUATIONS ARE TAKEN        FIT0002
C      FROM DIEM AND LIUKKONEN (1988) AND THEIR DERIVATIONS    FIT0003
C      APPEAR IN CHAPTER 1 OF THIS PAPER.                      FIT0004
C                                                                FIT0005
C      THE FIRST PART OF THIS PROGRAM CALCULATES THE           FIT0006
C      INITIAL ESTIMATES OF THE B'S, ALPHA, SIGMA SQUARED,     FIT0007
C      AND THE D'S.  THE E-M ALGORITHM IS THEN IMPLEMENTED     FIT0008
C      WITH THE ABOVE ESTIMATES IN ORDER TO ITERATIVELY        FIT0009
C      IMPROVE THE ESTIMATES OF THE PARAMETERS.                FIT0010
C                                                                FIT0011
C      SUBROUTINE FITRCB(N,M,P,K)                                FIT0012
C      IMPLICIT REAL *8 (A-H,Q-Z)                               FIT0013
C      INTEGER N,M,P,K                                          FIT0014
C      INTEGER IPVT(6)                                          FIT0015
C      DIMENSION X(20,6),Z(20,6),Y(20,200)                    FIT0016
C      DIMENSION D(6,6),B(6,200),ALPH(6),DET(2)                FIT0017
C      DIMENSION XTX(6,6),ZTZ(6,6),ZTZI(6,6),ZZZ(6,20)        FIT0018
C      DIMENSION ZT(6,20),YSUM(20),DSUM(6,6),DOLD(6,6)        FIT0019
C      DIMENSION XT(6,20),XALPH(20),DIFF(20)                   FIT0020
C      DIMENSION SUMI(6,6),SUMIZT(6,20),YDIFF(20,200)          FIT0021
C      INTEGER IUSE(200)                                         FIT0022
C      COMMON/DATAS/X,Z,Y                                        FIT0023
C      COMMON/PARAMS/ALPH,B,SIGMA2,D                           FIT0024
C      COMMON/ITCON/MAXIT,IFLAG,CONV                            FIT0025
C      COMMON/USEME/IUSE                                         FIT0026
C      DATA IFRST/1/                                           FIT0027
C      COMMON /FIND/ZT,XALPH                                    FIT0028
C      IFLAG=9                                                  FIT0029
C                                                                FIT0030
C      FIRST CALCULATE USEFUL QUANTITIES                         FIT0031
C                                                                FIT0032
C      THE FOLLOWING GIVES TRANS(X)*X                            FIT0033
C                                                                FIT0034
C      DO 10 I=1,P                                              FIT0035
C      DO 5 J=1,P                                               FIT0036
C      XTX(I,J)=DDOT(N,X(1,I),1,X(1,J),1)                       FIT0037
C      CONTINUE                                                 FIT0038
5     CONTINUE                                                 FIT0039
10    CONTINUE                                                 FIT0040
C                                                                FIT0041
C      THE FOLLOWING GIVES TWO COPIES OF TRANS(Z)*Z           FIT0042
C                                                                FIT0043
C      DO 20 I=1,K                                              FIT0044
C      DO 15 J=1,K                                               FIT0045
C      ZTZ(I,J)=DDOT(N,Z(1,I),1,Z(1,J),1)                       FIT0046
C      ZTZI(I,J)=ZTZ(I,J)                                       FIT0047
15    CONTINUE                                                 FIT0048
20    CONTINUE                                                 FIT0049
C                                                                FIT0050
C      NEED TO CALCULATE INV(TRANS(Z)*Z)*TRANS(Z).            FIT0051
C      FIRST NEED TO GET TRANS(Z) THEN USE LINPACK              FIT0052
C      FACTOR AND SOLVER.  ABOVE WILL BE STORED IN ZZZ         FIT0053
C                                                                FIT0054
C      DO 30 I=1,K                                              FIT0055
C      DO 25 J=1,N                                               FIT0056
C      ZT(I,J)=Z(J,I)                                           FIT0057
C      ZZZ(I,J)=ZT(I,J)                                         FIT0058
25    CONTINUE                                                 FIT0059
30    CONTINUE                                                 FIT0060
C      CALL DGEFA(ZTZI,6,K,IPVT,INFO)

```

	IF(INFO.NE. 0) GO TO 500	FIT0061
	DO 35 I=1,N	FIT0062
	CALL DGESL(ZTZI,6,K,IPVT,ZZZ(1,I),0)	FIT0063
35	CONTINUE	FIT0064
C		FIT0065
C	NOW CALCULATE THE INVERSE	FIT0066
C		FIT0067
	CALL DGEDI(ZTZI,6,K,IPVT,DET,WORK,1)	FIT0068
C		FIT0069
38	DM=0.DO	FIT0070
	DO 39 J=1,M	FIT0071
39	DM=DM+DFLOAT(IUSE(J))	FIT0072
	IDM=INT(DM)	FIT0073
C		FIT0074
C	SUM THE Y'S	FIT0075
C		FIT0076
	DO 45 I=1,N	FIT0077
	YSUM(I)=0.0DO	FIT0078
	DO 40 J=1,M	FIT0079
	IF (IUSE(J).EQ.0) GO TO 40	FIT0080
	YSUM(I)=YSUM(I)+Y(I,J)	FIT0081
40	CONTINUE	FIT0082
45	CONTINUE	FIT0083
C		FIT0084
C	INITIALIZE DSUM, DIFF AND SIGSUM TO ZERO	FIT0085
C		FIT0086
	DO 55 I=1,K	FIT0087
	DO 50 J=1,K	FIT0088
	DSUM(I,J)=0.0DO	FIT0089
50	CONTINUE	FIT0090
55	CONTINUE	FIT0091
	DO 57 I=1,N	FIT0092
	DIFF(I)=0.0DO	FIT0093
57	CONTINUE	FIT0094
	SIGSUM=0.0DO	FIT0095
C		FIT0096
C	THIS ROUTINE CALCULATES THE INITIAL EST OF ALPHA	FIT0097
C	THE SOLUTION IS STORED IN ALPH	FIT0098
C		FIT0099
	DO 60 I=1,P	FIT0100
	ALPH(I) = DDOT(N,X(1,I),1,YSUM,1)/DM	FIT0101
60	CONTINUE	FIT0102
	CALL DGEFA(XTX,6,P,IPVT,INFO)	FIT0103
	IF (INFO.NE.0) GO TO 501	FIT0104
	CALL DGESL(XTX,6,P,IPVT,ALPH,0)	FIT0105
C		FIT0106
C	CALCULATE AND STORE THE INITIAL EST OF X*ALPH	FIT0107
C		FIT0108
	DO 70 I=1,P	FIT0109
	DO 65 J=1,N	FIT0110
	XT(I,J)=X(J,I)	FIT0111
65	CONTINUE	FIT0112
70	CONTINUE	FIT0113
	DO 75 I=1,N	FIT0114
	XALPH(I)=DDOT(P,XT(1,I),1,ALPH,1)	FIT0115
75	CONTINUE	FIT0116
C		FIT0117
C	THIS CALCULATES THE INITIAL B'S	FIT0118
C		FIT0119
	DO 120 J=1,M	FIT0120

```

IF (IUSE(J).EQ.0) GO TO 120
DO 90 I=1,N
DIFF(I)=Y(I,J)-XALPH(I)
90 CONTINUE
DO 95 I=1,K
B(I,J)=0.0D0
DO 92 L=1,N
92 B(I,J)=B(I,J)+ZZZ(I,L)*DIFF(L)
95 CONTINUE
DO 105 I=1,K
DO 100 L=1,K
DSUM(I,L)=DSUM(I,L)+(B(I,J)*B(L,J))
100 CONTINUE
105 CONTINUE
DO 115 I=1,N
DO 110 L=1,K
DIFF(I)=DIFF(I)-(Z(I,L)*B(L,J))
110 CONTINUE
115 CONTINUE
SIGSUM=SIGSUM+DDOT(N,DIFF,1,DIFF,1)
120 CONTINUE
C
C THE INITIAL EST OF SIGMA SQUARED IS IN SIGMA2
C
SIGMA2=SIGSUM/DFLOAT((IDM*N)-P-(K*IDM)+K)
C
C CALCULATE AND STORE THE INITIAL EST OF D
C
DO 130 I=1,K
DO 125 J=1,K
D(I,J)=(DSUM(I,J)/(DM-1.)-(SIGMA*ZTZI(I,J)))
125 CONTINUE
130 CONTINUE
C
C THE E-M ALGORITHM
C
C E-STEP
C FIRST STORE USEFUL QUANTITIES
C
ITER=0
1000 ITER=ITER+1
SIGOLD=SIGMA2
DO 140 I=1,K
DO 135 J=1,K
DSUM(I,J)=0.0D0
DOLD(I,J)=D(I,J)
135 CONTINUE
140 CONTINUE
SIGSUM=0.0D0
CALL DGEFA(D,6,K,IPVT,INFO)
IF (INFO.NE.0) GO TO 502
CALL DGEDI(D,6,K,IPVT,DET,WORK,1)
C
C D NOW CONTAINS INV(D)
C
DO 150 I=1,K
DO 145 J=1,K
SUMI(I,J)=ZTZ(I,J)+(SIGMA2*D(I,J))
145 CONTINUE
150 CONTINUE

```

```

FIT0121
FIT0122
FIT0123
FIT0124
FIT0125
FIT0126
FIT0127
FIT0128
FIT0129
FIT0130
FIT0131
FIT0132
FIT0133
FIT0134
FIT0135
FIT0136
FIT0137
FIT0138
FIT0139
FIT0140
FIT0141
FIT0142
FIT0143
FIT0144
FIT0145
FIT0146
FIT0147
FIT0148
FIT0149
FIT0150
FIT0151
FIT0152
FIT0153
FIT0154
FIT0155
FIT0156
FIT0157
FIT0158
FIT0159
FIT0160
FIT0161
FIT0162
FIT0163
FIT0164
FIT0165
FIT0166
FIT0167
FIT0168
FIT0169
FIT0170
FIT0171
FIT0172
FIT0173
FIT0174
FIT0175
FIT0176
FIT0177
FIT0178
FIT0179
FIT0180

```

```

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
SUMIZT(I,J)=DDOT(K,SUMI(1,I),1,ZT(1,J),1)
155 CONTINUE
160 CONTINUE
C
C CALCULATE THE IMPROVED EST OF THE B'S
C
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
DO 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
C M-STEP
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,ZT(1,I),1,B(1,J),1))
YSUM(I)=YSUM(I)+YDIFF(I,J)
200 CONTINUE
205 CONTINUE
DO 210 I=1,P
ALPH(I)=DDOT(N,X(1,I),1,YSUM,1)/DM
210 CONTINUE
CALL DGESL(XTX,6,P,IPVT,ALPH,0)
DO 215 I=1,N
XALPH(I)=DDOT(P,XT(1,I),1,ALPH,1)
215 CONTINUE
C
C RE-CALCULATE THE D'S
C
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
C
C RE-CALCULATE SIGMA-SQUARED

```

```

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

```

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


```

C      ONCE THE POPULATION PARAMETERS HAVE BEEN          FIN0001
C      CALCULATED BY FITRCB USING ALL OBSERVATIONS      FIN0002
C      FINDEM CALCULATES THE MAXIMUM MAHALNOBIS        FIN0003
C      DISTANCE. THIS NUMBER IS COMPARED TO A          FIN0004
C      P-VALUE CALCULATED BY THE FUNCTION PVCHI        FIN0005
C      AND EITHER ELIMINATED OR KEPT. EACH TIME       FIN0006
C      AN OBSERVATION IS ELIMINATED FITRCB IS         FIN0007
C      CALLED TO RE-CALCULATE THE PARAMETERS FOR      FIN0008
C      THE 'TRACKING' POPULATION.                     FIN0009
C                                                     FIN0010
C                                                     FIN0011
C                                                     FIN0012
C      SUBROUTINE FINDEM (N,M,P,K,SIGNIF)              FIN0013
C      IMPLICIT REAL *8(A-H,Q-Z)                      FIN0014
C      INTEGER N,M,P,K                                FIN0015
C      INTEGER JPVT(20),IPVT(20)                     FIN0016
C      DIMENSION X(20,6),Z(20,6),Y(20,200)           FIN0017
C      DIMENSION D(6,6),B(6,200),ALPH(6),DET(2)      FIN0018
C      DIMENSION ZT(6,20)                             FIN0019
C      DIMENSION XALPH(20),WORK(20)                  FIN0020
C      DIMENSION ZDT(6,20),COV(20,20),U(20),DMH(200) FIN0021
C      INTEGER IUSE(200)                              FIN0022
C      DIMENSION YDIFF(20)                           FIN0023
C      COMMON/DATAS/X,Z,Y                             FIN0024
C      COMMON/PARAMS/ALPH,B,SIGMA2,D                 FIN0025
C      COMMON/ITCON/MAXIT,IFLAG,CONV                 FIN0026
C      COMMON/USEME/IUSE                              FIN0027
C      COMMON/FIND/ZT,XALPH                           FIN0028
C                                                     FIN0029
C      FIRST NEED TO CALCULATE THE COVARIANCE MATRIX  FIN0030
C      THIS CALCULATES INV(Z*D*TRANS(Z)+SIGMA2*I)    FIN0031
C                                                     FIN0032
C      1000 DO 10 I=1,N                                FIN0033
C           DO 5 J=1,K                                  FIN0034
C             ZDT(J,I)=DDOT(K,ZT(1,I),1,D(1,J),1)    FIN0035
C           5 CONTINUE                                  FIN0036
C      10 CONTINUE                                     FIN0037
C           DO 20 I=1,N                                  FIN0038
C             DO 15 J=1,N                                FIN0039
C               COV(I,J)=DDOT(K,ZDT(1,J),1,ZT(1,I),1) FIN0040
C             15 CONTINUE                               FIN0041
C           COV(I,I)=COV(I,I)+SIGMA2                  FIN0042
C           20 CONTINUE                                 FIN0043
C           CALL DGEFA(COV,20,N,IPVT,INFO)            FIN0044
C             IF (INFO.NE.0)THEN                        FIN0045
C               WRITE(9,*) 'COV IS NOT INVERTIBLE'   FIN0046
C             ENDIF                                     FIN0047
C           CALL DGEDJ(COV,20,N,IPVT,DET,WORK,1)      FIN0048
C                                                     FIN0049
C      USE CHOLESKY DECOMP TO CALC MAHALANOBIS DIST  FIN0050
C                                                     FIN0051
C      DM=0.                                           FIN0052
C      DMAX=-1.0                                       FIN0053
C      DO 30 I=1,M                                       FIN0054
C        IF (IUSE(I).EQ.0) GO TO 30                   FIN0055
C        DM=DM+1.                                       FIN0056
C      DO 22 J=1,N                                       FIN0057

```

```

22      YDIFF(J)=Y(J,I)-XALPH(J)
      DO 25 J=1,N
      U(J)=DDOT(N,COV(1,J),1,YDIFF(1),1)
25      CONTINUE
      DMH(I)=DDOT(N,U(1),1,YDIFF(1),1)
      IF(DMH(I).GT.DMAX) THEN
          DMAX=DMH(I)
          INDEX=I
      ENDIF
30      CONTINUE
      PV=PVCHI(DMAX,N,DM)
      IF(PV.LT.SIGNIF) THEN
          IUSE(INDEX)=0
          CALL FITRCB(N,M,P,K)
          GO TO 1000
      ELSE
          RETURN
      ENDIF
      END

```

FUNCTION PVCHI(DMAX,N,DM)
C WRITTEN BY D MOHR 10/1/89
C RETURNS PROB. MAX OF M INDEP CHI-SQUARED VARIATES
C (EACH WITH N D.F.) WILL BE GREATER THAN D.
IMPLICIT REAL*8 (A-H,O-Z)
INTEGER N
DATA DL7/-.356675/
RN2=DFLOAT(N)/2.
DM2=DMAX/2.
PV=GAMMP(RN2,DM2)
PV=DLOG(PV)
DL7M=DL7/DM
IF(DL7M.GT.PV) THEN
 PVCHI=.3
ELSE
 PV=DEXP(PV*DM)
 PVCHI=1.-PV
ENDIF
RETURN
END

C
FUNCTION GAMMQ(A,X)
C FROM 'NUMERICAL RECIPES'
IMPLICIT REAL*8 (A-H,O-Z)
IF(X.LT.0..OR.A.LE.0.) PAUSE
IF(X.LT.A+1.) THEN
 CALL GSER(GAMSER,A,X,GLN)
 GAMMQ=1.-GAMSER
ELSE
 CALL GCF(GAMMCF,A,X,GLN)
 GAMMQ=GAMMCF
ENDIF
RETURN
END

C
SUBROUTINE GSER(GAMSER,A,X,GLN)
C FROM 'NUMERICAL RECIPES'
PARAMETER (ITMAX=100, EPS=3.E-7)

FIN0058
FIN0059
FIN0060
FIN0061
FIN0062
FIN0063
FIN0064
FIN0065
FIN0066
FIN0067
FIN0068
FIN0069
FIN0070
FIN0071
FIN0072
FIN0073
FIN0074
FIN0075
FIN0076
FIN00770
FIN00780
FIN00790
FIN0080
FIN0081
FIN0082
FIN0083
FIN0084
FIN0085
FIN0086
FIN0087
FIN0088
FIN0089
FIN0090
FIN0091
FIN0092
FIN0093
FIN0094
FIN0095
FIN0096
FIN0097
FIN0098
FIN0099
FIN0100
FIN0101
FIN0102
FIN0103
FIN0104
FIN0105
FIN0106
FIN0107
FIN0108
FIN0109
FIN0110
FIN0111
FIN0112
FIN0113
FIN0114
FIN0115
FIN0116
FIN0117

```

IMPLICIT REAL*8 (A-H,O-Z)
GLN=GAMMLN(A)
IF(X.LE.O.)THEN
  IF(X.LT.O.)PAUSE
  GAMSER=0.
  RETURN
ENDIF
AP=A
SUM=1./A
DEL=SUM
DO 11 N=1,ITMAX
  AP=AP+1.
  DEL=DEL*X/AP
  SUM=SUM+DEL
  IF(ABS(DEL).LT.ABS(SUM)*EPS)GO TO 1
11 CONTINUE
PAUSE 'A TOO LARGE, ITMAX TOO SMALL'
1 GAMSER=SUM*EXP(-X+A*LOG(X)-GLN)
RETURN
END

C
SUBROUTINE GCF(GAMMCF,A,X,GLN)
C FROM 'NUMERICAL RECIPES'
PARAMETER (ITMAX=100,EPS=3.E-7)
IMPLICIT REAL*8 (A-H,O-Z)
GLN=GAMMLN(A)
GOLD=0.
AO=1.
A1=X
BO=0.
B1=1.
FAC=1.
DO 11 N=1,ITMAX
  AN=FLOAT(N)
  ANA=AN-A
  AO=(A1+AO*ANA)*FAC
  BO=(B1+BO*ANA)*FAC
  ANF=AN*FAC
  A1=X*AO+ANF*A1
  B1=X*BO+ANF*B1
  IF(A1.NE.O.)THEN
    FAC=1./A1
    G=B1*FAC
    IF(ABS((G-GOLD)/G).LT.EPS)GO TO 1
    GOLD=G
  ENDIF
11 CONTINUE
PAUSE 'A TOO LARGE, ITMAX TOO SMALL'
1 GAMMCF=EXP(-X+A*DLOG(X)-GLN)*G
RETURN
END

C
FUNCTION GAMMLN(XX)
C FROM 'NUMERICAL RECIPES'
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 COF(6),STP,HALF,ONE,FPF,X,TMP,SER
DATA COF,STP/76.18009173D0,-86.50532033D0,24.01409822D0,
* -1.231739516D0,.120858003D-2,-.536382D-5,2.50662827465D0/
DATA HALF,ONE,FPF/0.5D0,1.0D0,5.5D0/
X=XX-ONE

```

FINO118
FINO119
FINO120
FINO121
FINO122
FINO123
FINO124
FINO125
FINO126
FINO127
FINO128
FINO129
FINO130
FINO131
FINO132
FINO133
FINO134
FINO135
FINO136
FINO137
FINO138
FINO139
FINO140
FINO141
FINO142
FINO143
FINO144
FINO145
FINO146
FINO147
FINO148
FINO149
FINO150
FINO151
FINO152
FINO153
FINO154
FINO155
FINO156
FINO157
FINO158
FINO159
FINO160
FINO161
FINO162
FINO163
FINO164
FINO165
FINO166
FINO167
FINO168
FINO169
FINO170
FINO171
FINO172
FINO173
FINO174
FINO175
FINO176
FINO177

```

      TMP=X+FPF
      TMP=(X+HALF)*LOG(TMP)-TMP
      SER=ONE
      DO 11 J=1,6
        X=X+ONE
        SER=SER+COF(J)/X
11    CONTINUE
      GAMMLN=TMP+LOG(STP*SER)
      RETURN
      END
C
      FUNCTION GAMMP(A,X)
C      FROM 'NUMERICAL RECIPES'
      IMPLICIT REAL*8 (A-H,O-Z)
      IF (X.LT.0..OR.A.LE.0.) PAUSE
      IF (X.LT.A+1.) THEN
        CALL GSER(GAMSER,A,X,GLN)
        GAMMP=GAMSER
      ELSE
        CALL GCF(GAMMCF,A,X,GLN)
        GAMMP=1.-GAMMCF
      ENDIF
      RETURN
      END

```

45

```

FIN0178
FIN0179
FIN0180
FIN0181
FIN0182
FIN0183
FIN0184
FIN0185
FIN0186
FIN0187
FIN0188
FIN0189
FIN0190
FIN0191
FIN0192
FIN0193
FIN0194
FIN0195
FIN0196
FIN0197
FIN0198
FIN0199
FIN0200
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 Longitudinal Data," Biometrics, 38, pp. 963-
974.
- Press, W. H., Flannery, B. P., Teukolsky, S.A., and
Vetterling, W. T., Numerical Recipes: The Art of
Scientific Computing, 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.

VITA

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