1989

Identifying Outliers in a Random Effects Model For Longitudinal Data

Tamarah Crouse Dishman

University of North Florida

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

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

List of Tables and Figures

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

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

where $\mathbf{y}_i$ is the vector of n_i observations from individual i, $\boldsymbol{\alpha}$ is a px1 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 kx1 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 $\mathbf{e}_i$. D is a kxk positive definite covariance matrix. The population parameters, α, are treated as fixed effects.

The $\mathbf{y}_i$ are independent and distributed $N(\mathbf{X}_i\alpha,\ \mathbf{Z}_i\mathbf{D}\mathbf{Z}_i^T + \sigma^2\mathbf{I})$. 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 α,σ^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 $\mathbf{b}_i$, $\hat{\mathbf{b}}_i$. This estimate is calculated using current estimates of $\alpha,\sigma^2,\mathbf{D}$.

2. In the M-step, parameters α,σ^2 and **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:

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

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

where c_1 is a constant

NOTE:

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

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

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,

$$f(\mathbf{b}_i|\mathbf{y}_i,\theta) = c_2 \exp\left\{\frac{1}{\sigma^2}(\mathbf{y}_i-\mathbf{X}_i\boldsymbol{\alpha})^T\mathbf{Z}_i\mathbf{b}_i\right\} \cdot \exp\left\{-\frac{1}{2\sigma^2}(\mathbf{b}_i^T\mathbf{Z}_i^T\mathbf{Z}_i\mathbf{b}_i) - \frac{1}{2}(\mathbf{b}_i^T\mathbf{D}^{-1}\mathbf{b}_i)\right\}$$

$$= c_2 \exp\left\{\frac{1}{2\sigma^2}\left[2(\mathbf{y}_i-\mathbf{X}_i\boldsymbol{\alpha})^T\mathbf{Z}_i\mathbf{b}_i - \mathbf{b}_i^T(\mathbf{Z}_i^T\mathbf{Z}_i+\mathbf{D}^{-1}\sigma^2)\mathbf{b}_i\right]\right\}$$

$$= c_2 \exp\left\{\frac{-1}{2}\left[\frac{-2\left(\mathbf{y}_i-\mathbf{X}_i\boldsymbol{\alpha}\right)^T\mathbf{Z}_i\mathbf{b}_i + \mathbf{b}_i^T\left(\mathbf{Z}_i^T\mathbf{Z}_i +\mathbf{D}^{-1}\sigma^2\right)\mathbf{b}_i}{\sigma^2}\right]\right\}$$

$$= c_2 \exp\left\{\frac{-1}{2}\left[\mathbf{b}_i^T\mathbf{A}\mathbf{b}_i - 2\left(\mathbf{y}_i-\mathbf{X}_i\boldsymbol{\alpha}\right)^T\mathbf{Z}_i\mathbf{b}_i/\sigma^2\right]\right\}$$

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

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

From Appendix 1, it follows that:

$$E\left(\mathbf{b}_i \mid \mathbf{y}_i, \theta\right)=\left(\mathbf{Z}_i^T\mathbf{Z}_i+\mathbf{D}^{-1}\sigma^2\right)^{-1}\mathbf{Z}_i^T\left(\mathbf{y}_i-\mathbf{X}_i\boldsymbol{\alpha}\right)= \hat{\mathbf{b}}_i \qquad (1)$$

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

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

$$=E\left\{\sum^{m}\left[\frac{-n_i}{2}\ln\left(\sigma^2\right)-\frac{1}{2\sigma^2}\left(\mathbf{y}_i-\mathbf{X}_i\boldsymbol{\alpha}-\mathbf{Z}_i\mathbf{b}_i\right)^T\left(\mathbf{y}_i-\mathbf{X}_i\boldsymbol{\alpha}-\mathbf{Z}_i\mathbf{b}_i\right)-\frac{1}{2}\ln(\det\mathbf{D})\right.\right.$$

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

$$\sum^{m}\left[\frac{-n_i}{2}\ln\left(\sigma^2\right)-\frac{1}{2\sigma^2}\left(\mathbf{y}_i-\mathbf{X}_i\boldsymbol{\alpha}\right)^T\left(\mathbf{y}_i-\mathbf{X}_i\boldsymbol{\alpha}\right)+\frac{1}{\sigma^2}\left(\mathbf{y}_i-\mathbf{X}_i\boldsymbol{\alpha}\right)^T\mathbf{Z}_iE\left(\mathbf{b}_i\mid\mathbf{y}_i,\theta\right)\right.$$

$$\left.-\frac{1}{2\sigma^2}E\left(\mathbf{b}_i^T\mathbf{Z}_i^T\mathbf{Z}_i\mathbf{b}_i\mid\mathbf{y}_i,\theta\right)-\frac{1}{2}\ln(\det\mathbf{D})-\frac{1}{2}E\left(\mathbf{b}_i^T\mathbf{D}^{-1}\mathbf{b}_i\mid\mathbf{y}_i,\theta\right)\right]$$

$$= \frac{-N}{2}\ln\sigma^2 - \frac{1}{2\sigma^2}\sum^{m}(y_i - X_i\alpha)^T(y_i - X_i\alpha) + \frac{1}{\sigma^2}\sum^{m}[(y_i - X_i\alpha)^T Z_i E(b_i | y_i, \theta)]$$

$$- \frac{m}{2}\ln(\det D) - \frac{1}{2\sigma^2} E[b_i^T(Z_i^T Z_i + \sigma^2 D^{-1})b_i | y_i, \theta]$$

$$= \frac{-N}{2}\ln\sigma^2 - \frac{m}{2}\ln(\det D) - \frac{1}{2\sigma^2}\sum^{m}(y_i - X_i\alpha)^T(y_i - X_i\alpha) + \frac{1}{\sigma^2}\sum^{m}(y_i - X_i\alpha)^T Z_i \hat{b}_i$$

$$- \frac{1}{2\sigma^2}\sum^{m}\{tr[(Z_i^T Z_i + \sigma^2 D^{-1})V(b_i | y_i, \theta)] + \hat{b}_i^T(Z_i^T Z_i + \sigma^2 D^{-1})\hat{b}_i\} \quad (2)$$

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

Now, we use $H(\theta)$ to derive expressions for $\hat{\alpha}$, $\hat{\sigma}^2$ 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 $\hat{\alpha}$:

Note:

$$-\frac{1}{2\sigma^2}\sum^{m}(y_i-X_i\alpha)^T(y_i-X_i\alpha) = \frac{-1}{2\sigma^2}\sum^{m}y_i^Ty_i + \frac{1}{\sigma^2}\sum^{m}y_i^TX_i\alpha + \frac{-1}{2\sigma^2}\sum^{m}\alpha^TX_i^TX_i\alpha$$

and,

$$\frac{1}{\sigma^2}\sum^{m}(y_i-X_i\alpha)^TZ_i\hat{b}_i = \frac{1}{\sigma^2}\sum^{m}y_i^TZ_i\hat{b}_i + \frac{1}{\sigma^2}\sum^{m}\alpha^TX_i^TZ_i\hat{b}_i$$

therefore,

$$\frac{\partial H(\theta)}{\partial \alpha} = \frac{1}{\sigma^2}\sum^{m}X_i^Ty_i - \frac{1}{\sigma^2}\sum^{m}X_i^TX_i\alpha - \frac{1}{\sigma^2}\sum^{m}X_i^TZ_i\hat{b}_i = 0$$

it follows that,

$$\sum^{m}X_i^TX_i\alpha = \sum^{m}X_i^Ty_i - \sum^{m}X_i^TZ_i\hat{b}_i = \sum^{m}X_i^T(y_i-Z_i\hat{b}_i)$$

and,

$$\hat{\alpha} = \left(\sum^{m} X_i^T X_i \right)^{-1} \sum^{m} X_i^T \left(y_i - Z_i \hat{b}_i \right) \qquad (3)$$

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

$$\frac{\partial H(\theta)}{\partial \sigma^2} = \frac{-N}{2\sigma^2} + \frac{\sum^{m}\left[\left(y_i - X_i\alpha\right)^T\left(y_i - X_i\alpha\right)\right]}{2\sigma^4} - \frac{\sum^{m}\left[\left(y_i - X_i\alpha\right)^T Z_i \hat{b}_i\right]}{\sigma^4}$$

$$+ \frac{\sum^{m} tr\left[Z_i^T Z_i \; V(b_i \mid y_i, \theta)\right]}{2\sigma^4} + \frac{\sum^{m} b_i Z_i^T Z_i \hat{b}_i}{2\sigma^4} = 0$$

it follows that,

$$\sigma^2 N - \sum^{m}\left[\left(y_i - X_i\alpha\right)^T\left(y_i - X_i\alpha\right)\right] + 2\sum^{m}\left(y_i - X_i\alpha\right)^T Z_i \hat{b}_i$$
$$- \sum^{m} tr\left[Z_i^T Z_i \; V(b_i \mid y_i, \theta)\right] - \sum^{m} \hat{b}_i^T Z_i^T Z_i \hat{b}_i = 0$$

therefore,

$$\hat{\sigma}^2 = \frac{1}{N} \sum^{m} \{ (\mathbf{y}_i - \mathbf{X}_i\alpha)^T (\mathbf{y}_i - \mathbf{X}_i\alpha) - 2(\mathbf{y}_i - \mathbf{X}_i\alpha)^T \mathbf{Z}_i \hat{\mathbf{b}}_i + \mathrm{tr}[\mathbf{Z}_i^T \mathbf{Z}_i \, V(\mathbf{b}_i | \mathbf{y}_i, \theta)] + \hat{\mathbf{b}}_i^T \mathbf{Z}_i^T \mathbf{Z}_i \hat{\mathbf{b}}_i \} \qquad (4)$$

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

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

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

it follows that,

$$-m\mathbf{D} + \sum^{m} \hat{\mathbf{b}}_i \hat{\mathbf{b}}_i^T + \sum^{m} V(\mathbf{b}_i | \mathbf{y}_i, \theta) = 0$$

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

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 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,

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

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

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

(6), clearly, $\mathcal{D}_i$ 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 $\mathcal{D}_i$ 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 $\mathcal{D}_i$'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 $\mathcal{D}_i$. 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 p-value expressed as

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

where dmax stands for maximum D_i from $\chi^2(n)$ 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 σ^2, significance levels and values used for **X**, **Z** and $\boldsymbol{\alpha}$ 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 $\mathbf{X}\boldsymbol{\alpha}$, their expected values, for n=5 and n=10. Both case A and case B for non-trackers are shown. In case A, $\boldsymbol{\alpha}$=(5,-4,2), the opposite slopes for the sketches indicate that the non-trackers are drastically different from the trackers. For case B, $\boldsymbol{\alpha}$=(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 Trackers

n=5

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

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

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

n=10

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

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

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

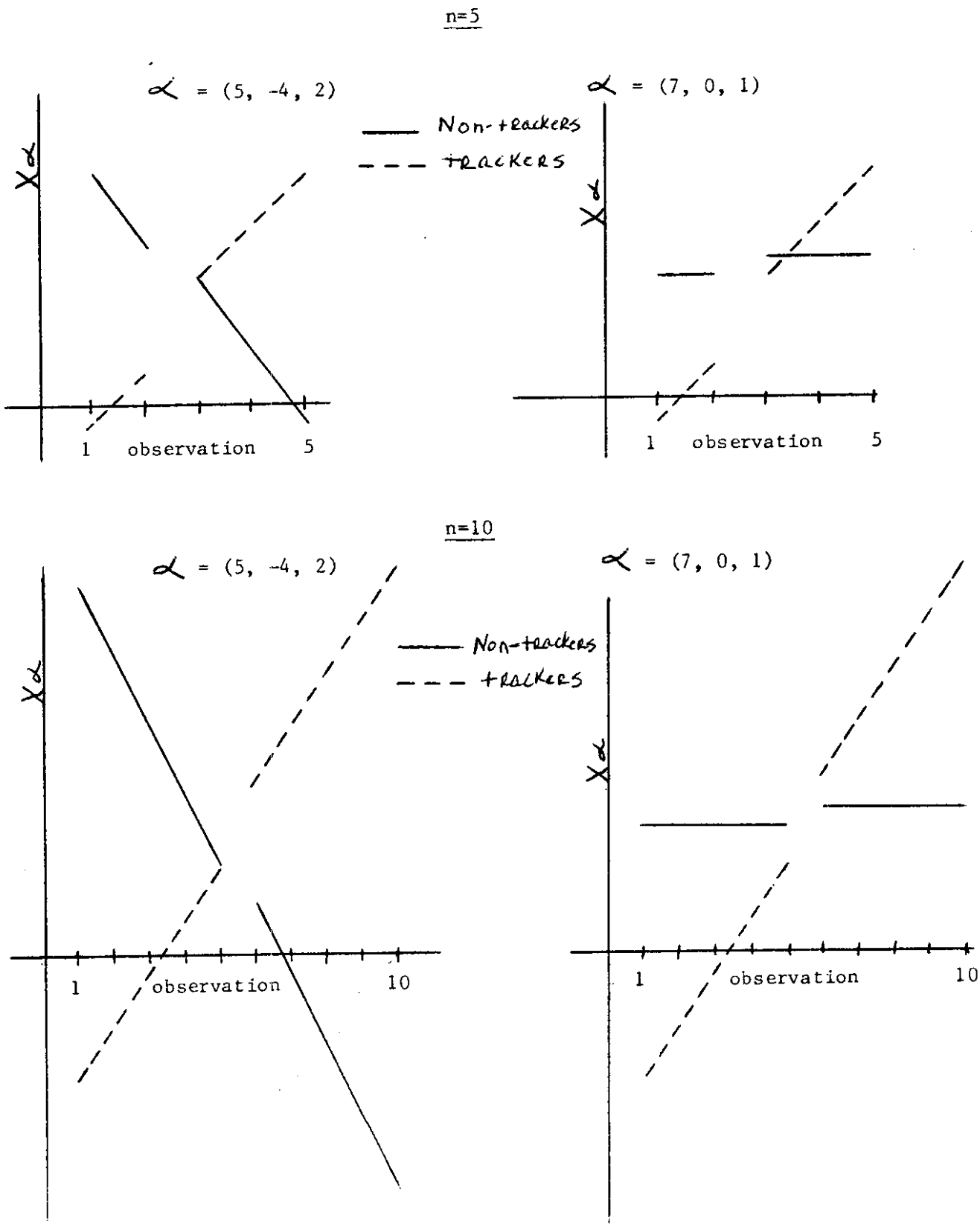
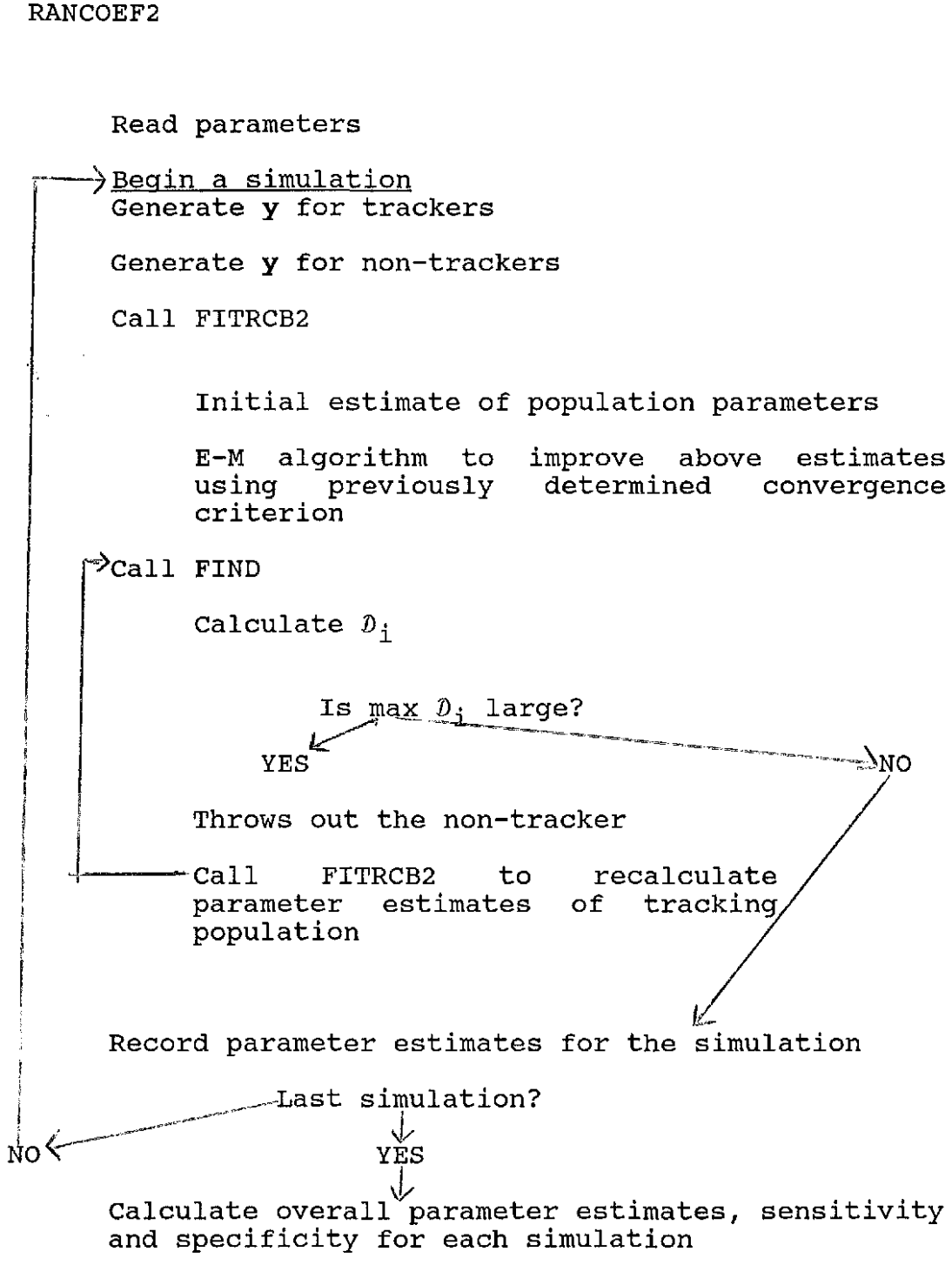


Table 3 - Simulations Run with Non-Trackers Present

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 α=(5, -4, 2); B indicates α=(7,0,1). Ntr stands for the number of non-trackers present.

Figure 2 - Outline for Computer Algorithm



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, α=(5,-4,2), than case B, α=(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**, $\hat{\mathbf{D}}$, is affected significantly by the presence of more than one non-tracker. Specifically, the entries of $\hat{\mathbf{D}}$ are larger than **D**. We know that when $\hat{\mathbf{D}}$ 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 σ^2=2.0

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

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

Case 2: m=50 n=5 σ^2=2.0

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

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

Case 3: m=25 n=10 σ^2=8.0

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

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

Case 4: m=50 n=10 σ^2=2.0

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

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

True parameter values for the above are:

signif = 0.10

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

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 α=(5, -4, 2); B indicates α=(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 α=(5,-4,2))

Case 1: m=25 n=5 NTR=2 σ^2=2.0

$\hat{\sigma}^2$ = 1.8667 with std dev = 0.3681

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

Case 2: m=50 n=5 NTR=3 σ^2=2.0

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

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

Case 3: m=25 n=10 NTR=2 σ^2=8.0

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

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

Case 4: m=50 n=10 NTR=3 σ^2=2.0

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

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

True parameter values for the above are:

signif = 0.10

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

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

$$-2\mu^T \mathbf{A} b_i = \frac{-2\left(y_i - X_i \alpha\right)^T Z_i b_i}{\sigma^2}$$

$$\Rightarrow \quad \frac{\mu^T\left(Z_i^T Z_i + D^{-1}\sigma^2\right)}{\sigma^2} = \frac{\left(y_i - X_i \alpha\right)^T Z_i}{\sigma^2}$$

$$\Rightarrow \quad \mu^T = \left(y_i - X_i \alpha\right)^T Z_i \left(Z_i^T Z_i + D^{-1}\sigma^2\right)^{-1}$$

$$\Rightarrow \quad \mu = \left(Z_i^T Z_i + D^{-1}\sigma^2\right)^{-1} Z_i \left(y_i - X_i \alpha\right)$$

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

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

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

$$\frac{\partial}{\partial d_{rc}} \mathbf{b}_i^T \mathbf{D}^{-1} \mathbf{b}_i = \frac{\partial}{\partial d_{rc}} \text{tr } \mathbf{D}^{-1} \mathbf{A} = \frac{\partial}{\partial d_{rc}} \sum_i \sum_l d^{il} a_{li}$$

$$= - \sum_i \sum_l d^{ir} d^{cl} a_{li} = - \sum_i d^{ir} \left(\sum_l d^{cl} a_{li} \right)$$

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
      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),TQN(6,6),XAN(20)                       RAN0023
      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
      READ (3,*) N,M,P,K,PN,KN,MTR                                        RAN0039
      MNTR=M-MTR                                                          RAN0040
C     READ NUMITR = NUMBER OF REPETITIONS OF SIMULATION                   RAN0041
      WRITE (9,*) ' N,M,P,K,MTR',N,M,P,K,MTR                              RAN0042
      READ (3,*) NUMITR                                                   RAN0043
      READ (3,*) MAXIT,CONV                                               RAN0044
C     GET MATRIX X FOR TRACKERS, XN FOR NONTRACKERS                       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
C      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
```

```
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            RAN0075
C     CALL CHOLESKY DECOMPOSITION TO FACTOR TD=(TQ)*(TQ)                   RAN0076
      DO 75 I=1,K                                                          RAN0077
         DO 72 J=1,K                                                       RAN0078
            TQ(I,J)=TD(I,J)                                                RAN0079
72          CONTINUE                                                       RAN0080
75       CONTINUE                                                          RAN0081
      JOB=0                                                                RAN0082
      LDA=6                                                                RAN0083
      CALL DCHDC(TQ,LDA,K,WORK,JPV,JOB,INFO)                               RAN0084
C     WRITE (6,*) INFO                                                     RAN0085
      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
C     CALL CHOLESKY DECOMPOSITION TO FACTOR TDN=(TQN)*(TQN)                RAN0093
      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
C     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
C                                                                          RAN0109
C     STORE MEAN VECTOR X*ALPHA FOR TRACKERS                               RAN0110
      DO 150 I=1,N                                                         RAN0111
```

```
          XA(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
C                                                                       RAN0117
C       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
              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
```

```
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 SPECIFITY                     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
```

```
          BT(I)=0.D0                                                    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
        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                                                   RAN0352
      IF(INEXTP.EQ.56)INEXTP=1                                          RAN0353
      MJ=MA(INEXT)-MA(INEXTP)                                           RAN0354
      IF(MJ.LT.MZ)MJ=MJ+MBIG                                            RAN0355
      MA(INEXT)=MJ                                                      RAN0356
      RAN3=MJ*FAC                                                       RAN0357
      RETURN                                                            RAN0358
      END                                                               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
                                                                        FIT00110
      SUBROUTINE FITRCB(N,M,P,K)                                        FIT0012
      IMPLICIT REAL *8 (A-H,Q-Z)                                        FIT0013
      INTEGER N,M,P,K                                                   FIT0014
      INTEGER IPVT(6)                                                   FIT0015
      DIMENSION X(20,6),Z(20,6),Y(20,200)                               FIT0016
      DIMENSION D(6,6),B(6,200),ALPH(6),DET(2)                          FIT0017
      DIMENSION XTX(6,6),ZTZ(6,6),ZTZI(6,6),ZZZ(6,20)                   FIT0018
      DIMENSION ZT(6,20),YSUM(20),DSUM(6,6),DOLD(6,6)                   FIT0019
      DIMENSION XT(6,20),XALPH(20),DIFF(20)                             FIT0020
      DIMENSION SUMI(6,6),SUMIZT(6,20),YDIFF(20,200)                    FIT0021
      INTEGER IUSE(200)                                                 FIT0022
      COMMON/DATAS/X,Z,Y                                                FIT0023
      COMMON/PARAMS/ALPH,B,SIGMA2,D                                     FIT0024
      COMMON/ITCON/MAXIT,IFLAG,CONV                                     FIT0025
      COMMON/USEME/IUSE                                                 FIT0026
      DATA IFRST/1/                                                     FIT0027
      COMMON /FIND/ZT,XALPH                                             FIT0028
      IFLAG=9                                                           FIT0029
C                                                                       FIT0030
C     FIRST CALCULATE USEFUL QUANTITIES                                 FIT0031
C                                                                       FIT0032
C     THE FOLLOWING GIVES TRANS(X)*X                                    FIT0033
C                                                                       FIT0034
      DO 10 I=1,P                                                       FIT0035
      DO  5 J=1,P                                                       FIT0036
      XTX(I,J)=DDOT(N,X(1,I),1,X(1,J),1)                                FIT0037
5     CONTINUE                                                          FIT0038
10    CONTINUE                                                          FIT0039
C                                                                       FIT0040
C     THE FOLLOWING GIVES TWO COPIES OF TRANS(Z)*Z                      FIT0041
C                                                                       FIT0042
      DO 20 I=1,K                                                       FIT0043
      DO 15 J=1,K                                                       FIT0044
      ZTZ(I,J)=DDOT(N,Z(1,I),1,Z(1,J),1)                                FIT0045
      ZTZI(I,J)=ZTZ(I,J)                                                FIT0046
15    CONTINUE                                                          FIT0047
20    CONTINUE                                                          FIT0048
C                                                                       FIT0049
C     NEED TO CALCULATE INV(TRANS(Z)*Z)*TRANS(Z).                       FIT0050
C     FIRST NEED TO GET TRANS(Z) THEN USE LINPACK                       FIT0051
C     FACTOR AND SOLVER.  ABOVE WILL BE STORED IN ZZZ                   FIT0052
C                                                                       FIT0053
      DO 30 I=1,K                                                       FIT0054
      DO 25 J=1,N                                                       FIT0055
      ZT(I,J)=Z(J,I)                                                    FIT0056
      ZZZ(I,J)=ZT(I,J)                                                  FIT0057
25    CONTINUE                                                          FIT0058
30    CONTINUE                                                          FIT0059
      CALL DGEFA(ZTZI,6,K,IPVT,INFO)                                    FIT0060
```

```
      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.D0                                                           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)=0.0D0                                                     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)=0.0D0                                                   FIT0089
50    CONTINUE                                                          FIT0090
55    CONTINUE                                                          FIT0091
      DO 57 I=1,N                                                       FIT0092
      DIFF(I)=0.0D0                                                     FIT0093
57    CONTINUE                                                          FIT0094
      SIGSUM=0.0D0                                                      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                                       FIT0121
      DO 90 I=1,N                                                       FIT0122
      DIFF(I)=Y(I,J)-XALPH(I)                                           FIT0123
90    CONTINUE                                                          FIT0124
      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)+ZZZ(I,L)*DIFF(L)                                 FIT0128
95    CONTINUE                                                          FIT0129
      DO 105 I=1,K                                                      FIT0130
      DO 100 L=1,K                                                      FIT0131
      DSUM(I,L)=DSUM(I,L)+(B(I,J)*B(L,J))                               FIT0132
100   CONTINUE                                                          FIT0133
105   CONTINUE                                                          FIT0134
      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)                               FIT0140
120   CONTINUE                                                          FIT0141
C                                                                       FIT0142
C     THE INITIAL EST OF SIGMA SQUARED IS IN SIGMA2                     FIT0143
C                                                                       FIT0144
      SIGMA2=SIGSUM/DFLOAT((IDM*N)-P-(K*IDM)+K)                         FIT0145
C                                                                       FIT0146
C     CALCULATE AND STORE THE INITIAL EST OF D                          FIT0147
C                                                                       FIT0148
      DO 130 I=1,K                                                      FIT0149
      DO 125 J=1,K                                                      FIT0150
      D(I,J)=(DSUM(I,J)/(DM-1.)-(SIGMA*ZTZI(I,J)))                      FIT0151
125   CONTINUE                                                          FIT0152
130   CONTINUE                                                          FIT0153
C                                                                       FIT0154
C     THE E-M ALGORITHM                                                 FIT0155
C                                                                       FIT0156
C     E-STEP                                                            FIT0157
C     FIRST STORE USEFUL QUANTITIES                                     FIT0158
C                                                                       FIT0159
      ITER=0                                                            FIT0160
1000  ITER=ITER+1                                                       FIT0161
      SIGOLD=SIGMA2                                                     FIT0162
      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.0) GO TO 502                                          FIT0171
      CALL DGEDI(D,6,K,IPVT,DET,WORK,1)                                 FIT0172
C                                                                       FIT0173
C     D NOW CONTAINS INV(D)                                             FIT0174
C                                                                       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                                                          FIT0180
```

```
      CALL DGEFA(SUMI,6,K,IPVT,INFO)                                    FIT0181
      IF(INFO.NE.0) GO TO 503                                           FIT0182
      CALL DGEDI(SUMI,6,K,IPVT,DET,WORK,1)                              FIT0183
      DO 160 I=1,K                                                      FIT0184
      DO 155 J=1,N                                                      FIT0185
      SUMIZT(I,J)=DDOT(K,SUMI(1,I),1,ZT(1,J),1)                         FIT0186
155   CONTINUE                                                          FIT0187
160   CONTINUE                                                          FIT0188
C                                                                       FIT0189
C     CALCULATE THE IMPROVED EST OF THE B'S                             FIT0190
C                                                                       FIT0191
      DO 190 J=1,M                                                      FIT0192
      IF (IUSE(J).EQ.0) GO TO 190                                       FIT0193
      DO 165 I=1,N                                                      FIT0194
      DIFF(I)=Y(I,J)-XALPH(I)                                           FIT0195
165   CONTINUE                                                          FIT0196
      DO 175 I=1,K                                                      FIT0197
      B(I,J)=0.0D0                                                      FIT0198
      DO 170 L=1,N                                                      FIT0199
      B(I,J)=B(I,J)+(SUMIZT(I,L)*DIFF(L))                               FIT0200
170   CONTINUE                                                          FIT0201
175   CONTINUE                                                          FIT0202
      DO 185 I=1,K                                                      FIT0203
      DO 180 L=1,K                                                      FIT0204
      DSUM(I,L)=DSUM(I,L)+(B(I,J)*B(L,J))                               FIT0205
180   CONTINUE                                                          FIT0206
185   CONTINUE                                                          FIT0207
190   CONTINUE                                                          FIT0208
C                                                                       FIT0209
C                                                                       FIT0210
C     M-STEP                                                            FIT0211
C                                                                       FIT0212
C     RE-CALCULATE THE ALPHAS                                           FIT0213
C                                                                       FIT0214
      DO 195 I=1,N                                                      FIT0215
195   YSUM(I)=0.0D0                                                     FIT0216
      DO 205 J=1,M                                                      FIT0217
      IF (IUSE(J).EQ.0) GO TO 205                                       FIT0218
      DO 200 I=1,N                                                      FIT0219
      YDIFF(I,J)=Y(I,J)-(DDOT(K,ZT(1,I),1,B(1,J),1))                    FIT0220
      YSUM(I)=YSUM(I)+YDIFF(I,J)                                        FIT0221
200   CONTINUE                                                          FIT0222
205   CONTINUE                                                          FIT0223
      DO 210 I=1,P                                                      FIT0224
      ALPH(I)=DDOT(N,X(1,I),1,YSUM,1)/DM                                FIT0225
210   CONTINUE                                                          FIT0226
      CALL DGESL(XTX,6,P,IPVT,ALPH,0)                                   FIT0227
      DO 215 I=1,N                                                      FIT0228
      XALPH(I)=DDOT(P,XT(1,I),1,ALPH,1)                                 FIT0229
215   CONTINUE                                                          FIT0230
C                                                                       FIT0231
C     RE-CALCULATE THE D'S                                              FIT0232
C                                                                       FIT0233
      DO 225 I=1,K                                                      FIT0234
      DO 220 J=1,K                                                      FIT0235
      D(I,J)=(DSUM(I,J)/DM)+(SIGMA2*SUMI(I,J))                          FIT0236
220   CONTINUE                                                          FIT0237
225   CONTINUE                                                          FIT0238
C                                                                       FIT0239
C     RE-CALCULATE  SIGMA-SQUARED                                       FIT0240
```

```
C                                                                       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
```

```
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
                                                                       FIN00100
                                                                       FIN00110
                                                                       FIN00120
      SUBROUTINE FINDEM (N,M,P,K,SIGNIF)                               FIN0013
      IMPLICIT REAL *8(A-H,Q-Z)                                        FIN0014
      INTEGER N,M,P,K                                                  FIN0015
      INTEGER JPVT(20),IPVT(20)                                        FIN0016
      DIMENSION X(20,6),Z(20,6),Y(20,200)                              FIN0017
      DIMENSION D(6,6),B(6,200),ALPH(6),DET(2)                         FIN0018
      DIMENSION ZT(6,20)                                               FIN0019
      DIMENSION XALPH(20),WORK(20)                                     FIN0020
      DIMENSION ZDT(6,20),COV(20,20),U(20),DMH(200)                    FIN0021
      INTEGER IUSE(200)                                                FIN0022
      DIMENSION YDIFF(20)                                              FIN0023
      COMMON/DATAS/X,Z,Y                                               FIN0024
      COMMON/PARAMS/ALPH,B,SIGMA2,D                                    FIN0025
      COMMON/ITCON/MAXIT,IFLAG,CONV                                    FIN0026
      COMMON/USEME/IUSE                                                FIN0027
      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
1000  DO 10 I=1,N                                                      FIN0033
      DO 5  J=1,K                                                      FIN0034
      ZDT(J,I)=DDOT(K,ZT(1,I),1,D(1,J),1)                              FIN0035
5     CONTINUE                                                         FIN0036
10    CONTINUE                                                         FIN0037
      DO 20 I=1,N                                                      FIN0038
      DO 15 J=1,N                                                      FIN0039
      COV(I,J)=DDOT(K,ZDT(1,J),1,ZT(1,I),1)                            FIN0040
15    CONTINUE                                                         FIN0041
      COV(I,I)=COV(I,I)+SIGMA2                                         FIN0042
20    CONTINUE                                                         FIN0043
      CALL DGEFA(COV,20,N,IPVT,INFO)                                   FIN0044
         IF (INFO.NE.0)THEN                                            FIN0045
         WRITE(9,*) 'COV IS NOT INVERTIBLE'                            FIN0046
         ENDIF                                                         FIN0047
      CALL DGEDI(COV,20,N,IPVT,DET,WORK,1)                             FIN0048
C                                                                      FIN0049
C     USE CHOLESKY DECOMP TO CALC MAHALANOBIS DIST                     FIN0050
C                                                                      FIN0051
      DM=0.                                                            FIN0052
      DMAX=-1.0                                                        FIN0053
      DO 30 I=1,M                                                      FIN0054
      IF (IUSE(I).EQ.0) GO TO 30                                       FIN0055
      DM=DM+1.                                                         FIN0056
      DO 22 J=1,N                                                      FIN0057
```

```
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)                             FIN0060
25    CONTINUE                                                        FIN0061
      DMH(I)=DDOT(N,U(1),1,YDIFF(1),1)                                FIN0062
         IF(DMH(I).GT.DMAX) THEN                                      FIN0063
              DMAX=DMH(I)                                             FIN0064
              INDEX=I                                                 FIN0065
         ENDIF                                                        FIN0066
30    CONTINUE                                                        FIN0067
      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
C     WRITTEN BY D MOHR  10/1/89                                      FIN0081
C     RETURNS PROB. MAX OF M INDEP CHI-SQUARED VARIATES               FIN0082
C     (EACH WITH N D.F.) WILL BE GREATER THAN  D.                     FIN0083
      IMPLICIT REAL*8 (A-H,O-Z)                                       FIN0084
      INTEGER N                                                       FIN0085
      DATA DL7/-.356675/                                              FIN0086
      RN2=DFLOAT(N)/2.                                                FIN0087
      DM2=DMAX/2.                                                     FIN0088
      PV=GAMMP(RN2,DM2)                                               FIN0089
      PV=DLOG(PV)                                                     FIN0090
      DL7M=DL7/DM                                                     FIN0091
      IF (DL7M.GT.PV) THEN                                            FIN0092
          PVCHI=.3                                                    FIN0093
      ELSE                                                            FIN0094
          PV=DEXP(PV*DM)                                              FIN0095
          PVCHI=1.-PV                                                 FIN0096
      ENDIF                                                           FIN0097
      RETURN                                                          FIN0098
      END                                                             FIN0099
C                                                                     FIN0100
      FUNCTION GAMMQ(A,X)                                             FIN0101
C     FROM 'NUMERICAL RECIPES'                                        FIN0102
      IMPLICIT REAL*8 (A-H,O-Z)                                       FIN0103
      IF(X.LT.0..OR.A.LE.0.)PAUSE                                     FIN0104
      IF(X.LT.A+1.)THEN                                               FIN0105
        CALL GSER(GAMSER,A,X,GLN)                                     FIN0106
        GAMMQ=1.-GAMSER                                               FIN0107
      ELSE                                                            FIN0108
        CALL GCF(GAMMCF,A,X,GLN)                                      FIN0109
        GAMMQ=GAMMCF                                                  FIN0110
      ENDIF                                                           FIN0111
      RETURN                                                          FIN0112
      END                                                             FIN0113
C                                                                     FIN0114
      SUBROUTINE GSER(GAMSER,A,X,GLN)                                 FIN0115
C     FROM 'NUMERICAL RECIPES'                                        FIN0116
      PARAMETER (ITMAX=100,EPS=3.E-7)                                 FIN0117
```

```
      IMPLICIT REAL*8 (A-H,O-Z)                                          FIN0118
      GLN=GAMMLN(A)                                                      FIN0119
      IF(X.LE.0.)THEN                                                    FIN0120
        IF(X.LT.0.)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                                                     FIN0130
        SUM=SUM+DEL                                                      FIN0131
        IF(ABS(DEL).LT.ABS(SUM)*EPS)GO TO 1                              FIN0132
11    CONTINUE                                                           FIN0133
      PAUSE 'A TOO LARGE, ITMAX TOO SMALL'                               FIN0134
1     GAMSER=SUM*EXP(-X+A*LOG(X)-GLN)                                    FIN0135
      RETURN                                                             FIN0136
      END                                                                FIN0137
C                                                                        FIN0138
      SUBROUTINE GCF(GAMMCF,A,X,GLN)                                     FIN0139
C     FROM 'NUMERICAL RECIPES'                                           FIN0140
      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
      A1=X                                                               FIN0146
      B0=0.                                                              FIN0147
      B1=1.                                                              FIN0148
      FAC=1.                                                             FIN0149
      DO 11 N=1,ITMAX                                                    FIN0150
        AN=FLOAT(N)                                                      FIN0151
        ANA=AN-A                                                         FIN0152
        A0=(A1+A0*ANA)*FAC                                               FIN0153
        B0=(B1+B0*ANA)*FAC                                               FIN0154
        ANF=AN*FAC                                                       FIN0155
        A1=X*A0+ANF*A1                                                   FIN0156
        B1=X*B0+ANF*B1                                                   FIN0157
        IF(A1.NE.0.)THEN                                                 FIN0158
          FAC=1./A1                                                      FIN0159
          G=B1*FAC                                                       FIN0160
          IF(ABS((G-GOLD)/G).LT.EPS)GO TO 1                              FIN0161
          GOLD=G                                                         FIN0162
        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                                                                        FIN0169
      FUNCTION GAMMLN(XX)                                                FIN0170
C     FROM 'NUMERICAL RECIPES'                                           FIN0171
      IMPLICIT REAL*8 (A-H,O-Z)                                          FIN0172
      REAL*8 COF(6),STP,HALF,ONE,FPF,X,TMP,SER                           FIN0173
      DATA COF,STP/76.18009173D0,-86.50532033D0,24.01409822D0,           FIN0174
     *    -1.231739516D0,.120858003D-2,-.536382D-5,2.50662827465D0/      FIN0175
      DATA HALF,ONE,FPF/0.5D0,1.0D0,5.5D0/                               FIN0176
      X=XX-ONE                                                           FIN0177
```

```
      TMP=X+FPF                                                        FIN0178
      TMP=(X+HALF)*LOG(TMP)-TMP                                        FIN0179
      SER=ONE                                                          FIN0180
      DO 11 J=1,6                                                      FIN0181
        X=X+ONE                                                        FIN0182
        SER=SER+COF(J)/X                                               FIN0183
11    CONTINUE                                                         FIN0184
      GAMMLN=TMP+LOG(STP*SER)                                          FIN0185
      RETURN                                                           FIN0186
      END                                                              FIN0187
C                                                                      FIN0188
      FUNCTION GAMMP(A,X)                                              FIN0189
C     FROM 'NUMERICAL RECIPES'                                         FIN0190
      IMPLICIT REAL*8 (A-H,O-Z)                                        FIN0191
      IF (X.LT.0..OR.A.LE.0.) PAUSE                                    FIN0192
      IF (X.LT.A+1.) THEN                                              FIN0193
         CALL GSER(GAMSER,A,X,GLN)                                     FIN0194
         GAMMP=GAMSER                                                  FIN0195
      ELSE                                                             FIN0196
         CALL GCF(GAMMCF,A,X,GLN)                                      FIN0197
         GAMMP=1.-GAMMCF                                               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., 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