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

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

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 $\mathbf{e}_i$. $\mathbf{D}$ is a kxk positive definite covariance matrix. The population parameters, $\boldsymbol{\alpha}$, are treated as fixed effects.

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

Section 2: Estimation of parameters

In this section, equations for estimating $\boldsymbol{\alpha},\sigma^2$ and $\mathbf{D}$ will be developed. Since there are no closed form solutions we will derive the iterative solutions from maximum likelihood estimates using the Estimation Maximization 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 $\boldsymbol{\alpha},\sigma^2,\mathbf{D}$.

2. In the M-step, parameters $\boldsymbol{\alpha},\sigma^2$ and $\mathbf{D}$ are estimated using the $\mathbf{y}_i$ and $\mathbf{b}_i$.

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

E-Step

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

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

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

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,

$$f(\mathbf{b}_i|\mathbf{y}_i,\theta) = C_2 \exp\left\{\frac{1}{\sigma^2}(\mathbf{y}_i-\mathbf{X}_i\boldsymbol{\alpha})^T\mathbf{Z}_i\mathbf{b}_i\right\}$$

$$\bullet \exp\left\{-\frac{1}{2\sigma^2}(\mathbf{b}_i^T\mathbf{Z}_i^T\mathbf{Z}_i\mathbf{b}_i) - \frac{1}{2}(\mathbf{b}_i^T\mathbf{D}^{-1}\mathbf{b}_i)\right\}$$

$$= C_2 \exp\left\{\frac{1}{2\sigma^2}\left[2(\mathbf{y}_i-\mathbf{X}_i\boldsymbol{\alpha})^T\mathbf{Z}_i\mathbf{b}_i - \mathbf{b}_i^T(\mathbf{Z}_i^T\mathbf{Z}_i+\mathbf{D}^{-1}\sigma^2)\mathbf{b}_i\right]\right\}$$

$$= C_2 \exp\left\{\frac{-1}{2}\left[\frac{-2\left(\mathbf{y}_i-\mathbf{X}_i\boldsymbol{\alpha}\right)^T\mathbf{Z}_i\mathbf{b}_i + \mathbf{b}_i^T\left(\mathbf{Z}_i^T\mathbf{Z}_i+\mathbf{D}^{-1}\sigma^2\right)\mathbf{b}_i}{\sigma^2}\right]\right\}$$

$$= C_2 \exp\left\{\frac{-1}{2}\left[\mathbf{b}_i^T\mathbf{A}\mathbf{b}_i -2\left(\mathbf{y}_i-\mathbf{X}_i\boldsymbol{\alpha}\right)^T\mathbf{Z}_i\mathbf{b}_i/\sigma^2\right]\right\}$$

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

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

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

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

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

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

$$= \frac{-N}{2}\ln\sigma^2 - \frac{1}{2\sigma^2}\sum^m(\mathbf{y}_i-\mathbf{X}_i\boldsymbol{\alpha})^T(\mathbf{y}_i-\mathbf{X}_i\boldsymbol{\alpha}) + \frac{1}{\sigma^2}\sum^m[(\mathbf{y}_i-\mathbf{X}_i\boldsymbol{\alpha})^T\mathbf{Z}_i E(\mathbf{b}_i|\mathbf{y}_i,\theta)]$$

$$- \frac{m}{2}\ln(\det \mathbf{D}) - \frac{1}{2\sigma^2} \quad E\left[\mathbf{b}_i^T(\mathbf{Z}_i^T\mathbf{Z}_i+\sigma^2\mathbf{D}^{-1})\mathbf{b}_i|\mathbf{y}_i,\theta\right]$$

$$= \frac{-N}{2}\ln\sigma^2 - \frac{m}{2}\ln(\det \mathbf{D}) - \frac{1}{2\sigma^2}\sum^m(\mathbf{y}_i-\mathbf{X}_i\boldsymbol{\alpha})^T(\mathbf{y}_i-\mathbf{X}_i\boldsymbol{\alpha}) + \frac{1}{\sigma^2}\sum^m(\mathbf{y}_i-\mathbf{X}_i\boldsymbol{\alpha})^T\mathbf{Z}_i\hat{\mathbf{b}}_i$$

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

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

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

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

First consider $\hat{\boldsymbol{\alpha}}$:

Note:

$$-\frac{1}{2\sigma^2}\sum^m(\mathbf{y}_i-\mathbf{X}_i\boldsymbol{\alpha})^T(\mathbf{y}_i-\mathbf{X}_i\boldsymbol{\alpha}) = \frac{-1}{2\sigma^2}\sum^m\mathbf{y}_i{}^T\mathbf{y}_i + \frac{1}{\sigma^2}\sum^m\mathbf{y}_i^T\mathbf{X}_i\boldsymbol{\alpha}$$
$$+ \frac{-1}{2\sigma^2}\sum^m \boldsymbol{\alpha}^T\mathbf{X}_i^T\mathbf{X}_i\boldsymbol{\alpha}$$

and,

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

therefore,

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

it follows that,

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

and,

$$\hat{\boldsymbol{\alpha}} = \left(\sum^{m} \mathbf{X}_i^T \mathbf{X}_i \right)^{-1} \sum^{m} \mathbf{X}_i^T \left(\mathbf{y}_i - \mathbf{Z}_i \hat{\mathbf{b}}_i \right) \tag{3}$$

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

$$\frac{\partial H(\theta)}{\partial \sigma^2} = \frac{-N}{2\sigma^2} + \frac{\sum^{m}\left[\left(\mathbf{y}_i - \mathbf{X}_i\boldsymbol{\alpha}\right)^T\left(\mathbf{y}_i - \mathbf{X}_i\boldsymbol{\alpha}\right)\right]}{2\sigma^4} - \frac{\sum^{m}\left[\left(\mathbf{y}_i - \mathbf{X}_i\boldsymbol{\alpha}\right)^T \mathbf{Z}_i \hat{\mathbf{b}}_i\right]}{\sigma^4}$$

$$+ \frac{\sum^{m} \mathrm{tr}\left[\mathbf{Z}_i^T \mathbf{Z}_i \; V(\mathbf{b}_i | \mathbf{y}_i, \theta)\right]}{2\sigma^4} + \frac{\sum^{m} \mathbf{b}_i \mathbf{Z}_i^T \mathbf{Z}_i \hat{\mathbf{b}}_i}{2\sigma^4} = 0$$

it follows that,

$$\sigma^2 N - \sum^{m}\left[\left(\mathbf{y}_i - \mathbf{X}_i\boldsymbol{\alpha}\right)^T\left(\mathbf{y}_i - \mathbf{X}_i\boldsymbol{\alpha}\right)\right] + 2\sum^{m}\left(\mathbf{y}_i - \mathbf{X}_i\boldsymbol{\alpha}\right)^T \mathbf{Z}_i \hat{\mathbf{b}}_i$$
$$- \sum^{m} \mathrm{tr}\left[\mathbf{Z}_i^T \mathbf{Z}_i \; V(\mathbf{b}_i | \mathbf{y}_i, \theta)\right] - \sum^{m} \hat{\mathbf{b}}_i^T \mathbf{Z}_i^T \mathbf{Z}_i \hat{\mathbf{b}}_i = 0$$

therefore,

$$\hat{\sigma}^2 = \frac{1}{N} \sum^{m} \{ (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\alpha})^T (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\alpha}) - 2 (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\alpha})^T \mathbf{Z}_i \hat{\mathbf{b}}_i$$

$$+ \operatorname{tr}[\mathbf{Z}_i^T \mathbf{Z}_i \ V(\mathbf{b}_i | \mathbf{y}_i, \theta)] + \hat{\mathbf{b}}_i^T \mathbf{Z}_i^T \mathbf{Z}_i \hat{\mathbf{b}}_i \} \qquad (4)$$

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

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

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

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 $\mathcal{D}_i$ from $X^2(n)$ and F(dmax) equals the cumulative distribution function. The program for computing F(dmax) is taken from Press et al (1986).

In order to test the sensitivity (probability that an individual is identified as a non-tracker given that they are really a non-tracker) and specificity (probability that an individual is identified as a tracker given that they really are a tracker) of our algorithm, several simulations will be run. Combinations of the number of non-trackers present, the number of individuals, the number of observations per individual, the magnitude of σ^2, significance levels and values used for $\mathbf{X}$, $\mathbf{Z}$ 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

$$\mathbf{X} = \begin{pmatrix} 1 & -2 & 0 \\ 1 & -1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 2 & 1 \end{pmatrix} \qquad \mathbf{Z} = \begin{pmatrix} 1 & -2 \\ 1 & -1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{pmatrix} \qquad \boldsymbol{\alpha} = (5,\ 3,\ 2) \qquad \mathbf{D} = \begin{pmatrix} 0.5 & 0.0 \\ 0.0 & 0.5 \end{pmatrix}$$

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

n=10

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

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

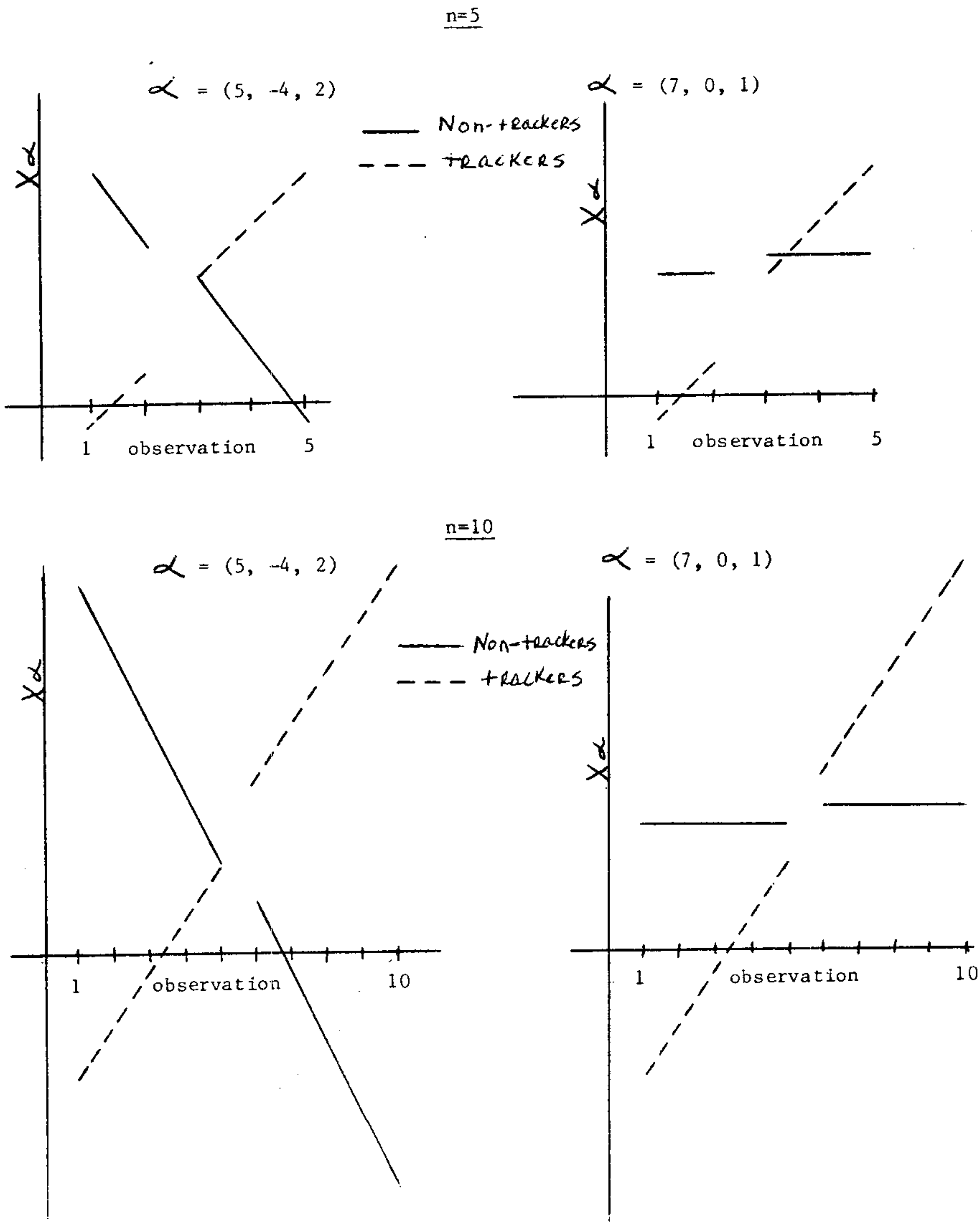

Table 3 - Simulations Run with Non-Trackers Present

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

Figure 2 - Outline for Computer Algorithm

RANCOEF2

Read parameters

Begin a simulation

Generate **y** for trackers

Generate **y** for non-trackers

Call FITRCB2

Initial estimate of population parameters

E-M algorithm to improve above estimates using previously determined convergence criterion

Call FIND

Calculate $\mathcal{D}_i$

Is max $\mathcal{D}_i$ large?

YES

NO

Throws out the non-tracker

Call FITRCB2 to recalculate parameter estimates of tracking population

Record parameter estimates for the simulation

Last simulation?

NO

YES

Calculate overall parameter estimates, sensitivity and specificity for each simulation

Chapter 3 - Conclusion

The first observation, obvious by looking at Table 4, is that when no non-trackers are present the algorithm is 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 $\mathcal{D}_i$ 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, $\boldsymbol{\alpha}$=(5,-4,2), than case B, $\boldsymbol{\alpha}$=(7,0,1), (recall that case A non-trackers are very different from the trackers where case B non-trackers are only slightly different). Signif was held constant at 0.10 for all of the simulations in which non-trackers were included. This level was used in order to reduce the number of trackers incorrectly identified as non-trackers. Changing σ^2 while holding everything else constant results in only a slight change in the level of sensitivity.

As shown in Table 7, the estimate of **D**, $\hat{\mathbf{D}}$, 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

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

Case 2: m=50 n=5 σ^2=2.0

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

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

Case 3: m=25 n=10 σ^2=8.0

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

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

Case 4: m=50 n=10 σ^2=2.0

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

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

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

Table 7 - Average Parameter Estimates with Non-Trackers Present

(Selected Cases from non-tracker $\boldsymbol{\alpha}=(5,-4,2)$)

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

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

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

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

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

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

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

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

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

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

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

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

True parameter values for the above are:

signif = 0.10

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

Appendix 1

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

Since,

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

and we have,

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

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

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

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

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

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

Appendix 2

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

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

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

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

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

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

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

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

Appendix 3 - Programs for Computer Simulations

```
C     CALLING PROGRAM TO GENERATE DATA WITH TRACKERS AND NONTRACKERS.
C     PROGRAM THEN CALLS ROUTINE FITRCB TO FIT RANDOM COEFFICIENT
C     LINEAR MODELS  AS DESCRIBED BY LAIRD AND WARE, BIOMETRICS, 1982.
C     THE ACTUAL ALGORITHM FOLLOWS THE DESCRIPTION GIVEN BY DIEM AND
C     LIUKKONEN IN STATISTICS IN MEDICINE, 1988.
C     AFTER THE INITIAL FIT, THE PROGRAM USES THE ROUTINE FINDEM
C     TO FIND NONTRACKERS.
C     AT THE END, PROGRAM PRINTS ESTIMATED PARAMETER VALUES FOR
C     TRACKERS, AND A LIST OF ID NUMBERS FOR SUSPECTED NONTRACKERS.
C     ********  BALANCED DATA DESIGNS  *******
      IMPLICIT REAL*8 (A-H,Q-Z)
      INTEGER N,M,P,K,PN,KN,MTR,MNTR
      DIMENSION X(20,6),Z(20,6)
      DIMENSION Y(20,200),ALPH(6),B(6,200),D(6,6)
      DIMENSION XN(20,6),ZN(20,6)
      DIMENSION ALFBAR(6),DBAR(6,6),ALFDEV(6)
      INTEGER IUSE(200)
      COMMON /DATAS/X,Z,Y
      COMMON /PARAMS/ ALPH,B,SIGMA2,D
      COMMON /ITCON/MAXIT
      COMMON /USEME/IUSE
      DIMENSION TALPH(6),TD(6,6),TQ(6,6),XA(20),BT(6)
      DIMENSION TALPHN(6),TDN(6,6),TQN(6,6),XAN(20)
      DIMENSION WORK(6),JTV(6)
C     ALL INPUT READ FROM FILE ON CHANNEL 3
C     IUSE(200) IS AN INTEGER VECTOR WHERE IUSE(I)=1 MEANS
C          THE INDIVIDUAL I SHOULD BE USED IN FITTING
C          IUSE(I)=0 MEANS DO NOT USE INDIV. I IN FITTING MODEL
C     READ IN DIMENSIONS  N=# OBSERVATIONS PER INDIVIDUAL
C                         M = # OF INDIVIDUALS
C                         P = DIMENSION OF ALPHA (FIXED EFFECTS)
C                         K = DIMENSION OF B (RANDOM EFFECTS)
C                         PN=DIM OF ALPHA FOR NONTRACKERS
C                         KN=DIM OF B FOR NONTRACKERS
C                         MTR = NUM OF TRACKERS
C
C     ***  GET PARAMETERS FOR TRACKERS AND NONTRACKERS  ***
C
      READ (3,*) N,M,P,K,PN,KN,MTR
      MNTR=M-MTR
C     READ NUMITR = NUMBER OF REPETITIONS OF SIMULATION
      WRITE (9,*) ' N,M,P,K,MTR',N,M,P,K,MTR
      READ (3,*) NUMITR
      READ (3,*) MAXIT,CONV
C     GET MATRIX X FOR TRACKERS, XN FOR NONTRACKERS
      DO 10 I=1,N
         READ (3,*) (X(I,J),J=1,P),(XN(I,J),J=1,PN)
10       CONTINUE
C      GET MATRIX Z FOR TRACKERS, ZN FOR NONTRACKERS
      DO 15 I=1,N
         READ (3,*) (Z(I,J),J=1,K),(ZN(I,J),J=1,KN)
```

```
15        CONTINUE
C      GET TRUE VALUES OF ALPHA, STORED IN TALPH
       READ (3,*) (TALPH(I),I=1,P),(TALPHN(I),I=1,PN)
       WRITE (9,*) ' ALPH FOR NONTR'
       WRITE(9,16) (TALPHN(I),I=1,PN)
16     FORMAT(3(1X,F10.4))
C      GET TRUE VALUE OF MEASUREMENT VARIANCE, SIGMA2, STORED AS TSIG2
       READ (3,*) TSIG2,TSIG2N
       TSIG=DSQRT(TSIG2)
       TSIGN=SQRT(TSIG2N)
C      GET TRUE VALUE OF COVARIANCE MATRIX D FOR RANDOM EFFECTS
       DO 20 I=1,K
          READ (3,*) (TD(I,J),J=1,K)
20        CONTINUE
       DO 25 I=1,KN
          READ(3,*) (TDN(I,J),J=1,KN)
25        CONTINUE
       READ (3,*) SIGNIF
       READ (3,*) IDUM
       WRITE (9,*) ' TSIG2,SIGNIF'
       WRITE(9,30) TSIG2,SIGNIF
30     FORMAT(2(1X,F10.4))
       XX=RAN3(IDUM)
C                                                         TRAN
C      CALL CHOLESKY DECOMPOSITION TO FACTOR TD=(TQ)*(TQ)
       DO 75 I=1,K
          DO 72 J=1,K
             TQ(I,J)=TD(I,J)
72           CONTINUE
75        CONTINUE
       JOB=0
       LDA=6
       CALL DCHDC(TQ,LDA,K,WORK,JPV,JOB,INFO)
C      WRITE (6,*) INFO
       DO 78 I=1,K-1
          DO 77 J=I+1,K
             TQ(J,I)=TQ(I,J)
             TQ(I,J)=0.D0
77           CONTINUE
78        CONTINUE
C                                                           TRAN
C      CALL CHOLESKY DECOMPOSITION TO FACTOR TDN=(TQN)*(TQN)
       DO 95 I=1,KN
          DO 92 J=1,KN
             TQN(I,J)=TDN(I,J)
92           CONTINUE
95        CONTINUE
       JOB=0
       LDA=6
       CALL DCHDC(TQN,LDA,KN,WORK,JPV,JOB,INFO)
C      WRITE (6,*) INFO
       DO 98 I=1,KN-1
          DO 97 J=I+1,KN
             TQN(J,I)=TQN(I,J)
             TQN(I,J)=0.D0
97           CONTINUE
98        CONTINUE
C
C      STORE MEAN VECTOR X*ALPHA FOR TRACKERS
       DO 150 I=1,N
```

```
         XA(I)=0.D0
         DO 145 J=1,P
            XA(I)=XA(I)+X(I,J)*TALPH(J)
145         CONTINUE
150      CONTINUE
C
C     STORE MEAN VECTOR XN*ALPHAN FOR NONTRACKERS
      DO 190 I=1,N
         XAN(I)=0.D0
         DO 185 J=1,PN
            XAN(I)=XAN(I)+XN(I,J)*TALPHN(J)
185         CONTINUE
190      CONTINUE

C     ++++++++++++  BEGIN REPETITIONS, CREATING DATA FOR ++++++++++
C     ++++++++++++  TRACKERS, NONTRACKERS; FITTING MODEL ++++++++++
C     ++++++++++++  AND FINDING NON-TRACKERS
C
      SIGBAR=0.0D0
      SIGDEV=0.0D0
      DO 200 I=1,P
      ALFBAR(I)=0.0D0
200   ALFDEV(I)=0.0D0
      DO 210 I=1,K
      DO 205 J=1,K
      DBAR(I,J)=0.0D0
205   CONTINUE
210   CONTINUE
      SUMSN =0.0D0
      SUMSN2=0.0D0
      SUMSP =0.0D0
      SUMSP2=0.0D0
      DO 3000 III=1,NUMITR
C
      DO 4 I=1,M
4        IUSE(I)=1
C     +++++ MANUFACTURE Y FOR TRACKERS  II=1,MTR ++++++

       DO 300 II=1,MTR
C         MANUFACTURE B FOR THE II INDIVIDUAL
          CALL MULTNO(TQ,6,K,BT,IDUM)
          DO 270 I=1,K
             B(I,II)=BT(I)
270          CONTINUE
          DO 290 I=1,N
             Y(I,II)=XA(I)+GASDEV(IDUM)*TSIG
             DO 280 J=1,K
                Y(I,II)=Y(I,II)+Z(I,J)*BT(J)
280             CONTINUE
290          CONTINUE
300      CONTINUE
C
C     +++++ MANUFACTURE Y FOR NONTRACKERS II=MTR+1,M ++++++

       IF (MTR.GE.M) GO TO 410
       DO 400 II=MTR+1,M
C         MANUFACTURE B FOR THE II INDIVIDUAL
          CALL MULTNO(TQN,6,KN,BT,IDUM)
          DO 370 I=1,KN
             B(I,II)=BT(I)
```

```
370         CONTINUE
         DO 390 I=1,N
            Y(I,II)=XAN(I)+GASDEV(IDUM)*TSIG
            DO 380 J=1,KN
               Y(I,II)=Y(I,II)+ZN(I,J)*BT(J)
380            CONTINUE
390         CONTINUE
400       CONTINUE
410     CONTINUE
C
C
C      CALL FITTING ROUTINE FOR BALANCED DATA
       CALL FITRCB(N,M,P,K)
       CALL FINDEM(N,M,P,K,SIGNIF)
       WRITE(9,*)'THE FOLLOWING INDIVIDUALS WERE IDENTIFIED AS
      +NON-TRACKERS'
       NUMTR=0
       DO 510 I=1,M
          NUMTR=NUMTR+IUSE(I)
510       IF (IUSE(I) .EQ. 0) WRITE(9,*) I
       WRITE (9,*) ' NUMBER OF TRACKERS ',NUMTR
       IDCTR=0
       IDCNTR=0
       DO 515 I=1,MTR
515    IDCTR=IDCTR + IUSE(I)
       IF (MTR+1 .GT. M) THEN
         SENS=-1.
       ELSE
         DO 520 I=MTR+1,M
520      IDCNTR=IDCNTR + (1-IUSE(I))
         SENS= DFLOAT(IDCNTR)/DFLOAT(M-MTR)
         SUMSN=SUMSN + SENS
         SUMSN2=SUMSN2 + (SENS**2)
       ENDIF
600    SPEC=DFLOAT(IDCTR)/DFLOAT(MTR)
       SUMSP=SUMSP + SPEC
       SUMSP2=SUMSP2 + (SPEC**2)
C
       WRITE (9,*) ' ESTIMATED ALPHAS '
       DO 605 I=1,P
          WRITE (9,604) ALPH(I)
604       FORMAT (1X,F10.4)
605       CONTINUE
       WRITE (9,611) SIGMA2
611    FORMAT (' SIGMA2',F10.5)
       DO 615 I=1,K
          WRITE (9,617) (D(I,J),J=1,K)
617       FORMAT (' D',6(1X,F9.4))
615       CONTINUE
C
C      THIS WILL CALC OVERALL PARAMS FOR EACH SIMULATION
C
       SIGBAR=SIGBAR + SIGMA2
       SIGDEV=SIGDEV + (SIGMA2**2)
       DO 620 I=1,P
       ALFBAR(I)=ALFBAR(I)+ALPH(I)
620    ALFDEV(I)=ALFDEV(I)+(ALPH(I)**2)
       DO 650 I=1,K
       DO 640 J=1,K
       DBAR(I,J)=DBAR(I,J)+D(I,J)
```

```
640    CONTINUE
650    CONTINUE
C
3000   CONTINUE
C      +++++++++  3000 IS END OF REPETITION LOOP  ++++++
C
C      CALCULATE STATS FOR SENSITIVITY AND SPECIFITY
C
       RNITR=DFLOAT(NUMITR)
       SPMEAN=SUMSP/RNITR
       SPSIG=SQRT(((RNITR*SUMSP2)-(SUMSP**2))/(RNITR*(RNITR-1.)))
       IF(MTR+1.GT.M) GO TO 750
       SNMEAN=SUMSN/RNITR
       SNSIG=SQRT(((RNITR*SUMSN2)-(SUMSN**2))/(RNITR*(RNITR-1.)))
       WRITE(9,736) SNMEAN,SNSIG
736    FORMAT (' SENSITIVITY MEAN ',F8.5,'  STD DEV ',F8.5)
750    WRITE(9,737) SPMEAN,SPSIG
737    FORMAT (' SPECIFICITY MEAN ',F8.5,' STD DEV ',F8.5)
C
C      WRITE OUT PARAMS FOR THE SIMULATION
C
       WRITE(9,*)'OVERALL ESTIMATE OF SIGMA-SQUARED'
       SIGDEV=SQRT(((RNITR*SIGDEV)-(SIGBAR**2))/(RNITR*(RNITR-1.)))
       SIGBAR=SIGBAR/RNITR
       WRITE(9,604) SIGBAR
       WRITE(9,*)'WITH STANDARD DEVIATION'
       WRITE(9,604) SIGDEV
       WRITE(9,*)'OVERALL ESTIMATED ALPHA'
       DO 800 I=1,P
       ALFDEV(I)=SQRT(((RNITR*ALFDEV(I))-(ALFBAR(I)**2))
      +/(RNITR*(RNITR-1.)))
       ALFBAR(I)=ALFBAR(I)/RNITR
800    WRITE(9,604) ALFBAR(I)
       WRITE(9,*)'WITH STANDARD DEVIATION'
       DO 805 I=1,P
805    WRITE(9,604) ALFDEV(I)
       WRITE(9,*)'OVERALL ESTIMATE OF D'
       DO 820 I=1,K
       DO 810 J=1,K
       DBAR(I,J)=DBAR(I,J)/RNITR
810    CONTINUE
820    CONTINUE
       DO 830 I=1,K
       WRITE(9,617) (DBAR(I,J),J=1,K)
830    CONTINUE
1000   STOP
       END

C
C      SUBROUTINE TO PRODUCE MULTIVARIATE NORMAL VECTOR BT WITH
C      COVARIANCE MATRIX GIVEN BY  TQ*TRAN(TQ)
C      DEFINED LENGTH OF VECTOR IS LDA.  USED LENGTH IS  K.
       SUBROUTINE MULTNO(TQ,LDA,K,BT,IDUM)
       IMPLICIT REAL*8 (A-H,O-Z)
       DIMENSION TQ(LDA,LDA),BT(LDA)
       DIMENSION Z(20)
       DO 10 I=1,K
          Z(I)=GASDEV(IDUM)
10        CONTINUE
       DO 20 I=1,K
```

```
          BT(I)=0.D0
          DO 15 J=1,I
            BT(I)=BT(I)+TQ(I,J)*Z(J)
15          CONTINUE
20        CONTINUE
      RETURN
      END
C     FUNCTION GASDEV PRODUCES A STANDARD NORMAL DEVIATE
      FUNCTION GASDEV(IDUM)
      IMPLICIT REAL*8 (A-H,O-Z)
      DATA ISET/0/
      IF (ISET.EQ.0) THEN
1       V1=2.*RAN3(IDUM)-1.
        V2=2.*RAN3(IDUM)-1.
        R=V1**2+V2**2
        IF(R.GE.1.)GO TO 1
        FAC=DSQRT(-2.*DLOG(R)/R)
        GSET=V1*FAC
        GASDEV=V2*FAC
        ISET=1
      ELSE
        GASDEV=GSET
        ISET=0
      ENDIF
      RETURN
      END

C     FUNCTION RAN3 PRODUCES A UNIFORM (0,1) RANDOM DEVIATE
      FUNCTION RAN3(IDUM)
      IMPLICIT REAL*8 (A-H,O-Z)
C         IMPLICIT REAL*4(M)
C         PARAMETER (MBIG=4000000.,MSEED=1618033.,MZ=0.,FAC=2.5E-7)
      PARAMETER (MBIG=1000000000,MSEED=161803398,MZ=0,FAC=1.E-9)
      DIMENSION MA(55)
      DATA IFF /0/
      IF(IDUM.LT.0.OR.IFF.EQ.0)THEN
        IFF=1
        MJ=MSEED-IABS(IDUM)
        MJ=MOD(MJ,MBIG)
        MA(55)=MJ
        MK=1
        DO 11 I=1,54
          II=MOD(21*I,55)
          MA(II)=MK
          MK=MJ-MK
          IF(MK.LT.MZ)MK=MK+MBIG
          MJ=MA(II)
11      CONTINUE
        DO 13 K=1,4
          DO 12 I=1,55
            MA(I)=MA(I)-MA(1+MOD(I+30,55))
            IF(MA(I).LT.MZ)MA(I)=MA(I)+MBIG
12        CONTINUE
13      CONTINUE
        INEXT=0
        INEXTP=31
        IDUM=1
      ENDIF
      INEXT=INEXT+1
      IF(INEXT.EQ.56)INEXT=1
```

```
      INEXTP=INEXTP+1
      IF(INEXTP.EQ.56)INEXTP=1
      MJ=MA(INEXT)-MA(INEXTP)
      IF(MJ.LT.MZ)MJ=MJ+MBIG
      MA(INEXT)=MJ
      RAN3=MJ*FAC
      RETURN
      END
```

```
C     FITRCB IS THE ROUTINE FOR FITTING THE POPULATION
C     PARAMETERS FOR TRACKERS.  THE EQUATIONS ARE TAKEN
C     FROM DIEM AND LIUKKONEN (1988) AND THEIR DERIVATIONS
C     APPEAR IN CHAPTER 1 OF THIS PAPER.
C
C     THE FIRST PART OF THIS PROGRAM CALCULATES THE
C     INITIAL ESTIMATES OF THE B'S, ALPHA, SIGMA SQUARED,
C     AND THE D'S.  THE E-M ALGORITHM IS THEN IMPLEMENTED
C     WITH THE ABOVE ESTIMATES IN ORDER TO ITERATIVELY
C     IMPROVE THE ESTIMATES OF THE PARAMETERS.

      SUBROUTINE FITRCB(N,M,P,K)
      IMPLICIT REAL *8 (A-H,Q-Z)
      INTEGER N,M,P,K
      INTEGER IPVT(6)
      DIMENSION X(20,6),Z(20,6),Y(20,200)
      DIMENSION D(6,6),B(6,200),ALPH(6),DET(2)
      DIMENSION XTX(6,6),ZTZ(6,6),ZTZI(6,6),ZZZ(6,20)
      DIMENSION ZT(6,20),YSUM(20),DSUM(6,6),DOLD(6,6)
      DIMENSION XT(6,20),XALPH(20),DIFF(20)
      DIMENSION SUMI(6,6),SUMIZT(6,20),YDIFF(20,200)
      INTEGER IUSE(200)
      COMMON/DATAS/X,Z,Y
      COMMON/PARAMS/ALPH,B,SIGMA2,D
      COMMON/ITCON/MAXIT,IFLAG,CONV
      COMMON/USEME/IUSE
      DATA IFRST/1/
      COMMON /FIND/ZT,XALPH
      IFLAG=9
C
C     FIRST CALCULATE USEFUL QUANTITIES
C
C     THE FOLLOWING GIVES TRANS(X)*X
C
      DO 10 I=1,P
      DO  5 J=1,P
      XTX(I,J)=DDOT(N,X(1,I),1,X(1,J),1)
5     CONTINUE
10    CONTINUE
C
C     THE FOLLOWING GIVES TWO COPIES OF TRANS(Z)*Z
C
      DO 20 I=1,K
      DO 15 J=1,K
      ZTZ(I,J)=DDOT(N,Z(1,I),1,Z(1,J),1)
      ZTZI(I,J)=ZTZ(I,J)
15    CONTINUE
20    CONTINUE
C
C     NEED TO CALCULATE INV(TRANS(Z)*Z)*TRANS(Z).
C     FIRST NEED TO GET TRANS(Z) THEN USE LINPACK
C     FACTOR AND SOLVER.  ABOVE WILL BE STORED IN ZZZ
C
      DO 30 I=1,K
      DO 25 J=1,N
      ZT(I,J)=Z(J,I)
      ZZZ(I,J)=ZT(I,J)
25    CONTINUE
30    CONTINUE
      CALL DGEFA(ZTZI,6,K,IPVT,INFO)
```

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

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

```
      CALL DGEFA(SUMI,6,K,IPVT,INFO)
      IF(INFO.NE.0) GO TO 503
      CALL DGEDI(SUMI,6,K,IPVT,DET,WORK,1)
      DO 160 I=1,K
      DO 155 J=1,N
      SUMIZT(I,J)=DDOT(K,SUMI(1,I),1,ZT(1,J),1)
155   CONTINUE
160   CONTINUE
C
C     CALCULATE THE IMPROVED EST OF THE B'S
C
      DO 190 J=1,M
      IF (IUSE(J).EQ.0) GO TO 190
      DO 165 I=1,N
      DIFF(I)=Y(I,J)-XALPH(I)
165   CONTINUE
      DO 175 I=1,K
      B(I,J)=0.0D0
      DO 170 L=1,N
      B(I,J)=B(I,J)+(SUMIZT(I,L)*DIFF(L))
170   CONTINUE
175   CONTINUE
      DO 185 I=1,K
      DO 180 L=1,K
      DSUM(I,L)=DSUM(I,L)+(B(I,J)*B(L,J))
180   CONTINUE
185   CONTINUE
190   CONTINUE
C
C
C     M-STEP
C
C     RE-CALCULATE THE ALPHAS
C
      DO 195 I=1,N
195   YSUM(I)=0.0D0
      DO 205 J=1,M
      IF (IUSE(J).EQ.0) GO TO 205
      DO 200 I=1,N
      YDIFF(I,J)=Y(I,J)-(DDOT(K,ZT(1,I),1,B(1,J),1))
      YSUM(I)=YSUM(I)+YDIFF(I,J)
200   CONTINUE
205   CONTINUE
      DO 210 I=1,P
      ALPH(I)=DDOT(N,X(1,I),1,YSUM,1)/DM
210   CONTINUE
      CALL DGESL(XTX,6,P,IPVT,ALPH,0)
      DO 215 I=1,N
      XALPH(I)=DDOT(P,XT(1,I),1,ALPH,1)
215   CONTINUE
C
C     RE-CALCULATE THE D'S
C
      DO 225 I=1,K
      DO 220 J=1,K
      D(I,J)=(DSUM(I,J)/DM)+(SIGMA2*SUMI(I,J))
220   CONTINUE
225   CONTINUE
C
C     RE-CALCULATE  SIGMA-SQUARED
```

```
C
      DO 235 J=1,M
      IF (IUSE(J).EQ.0) GO TO 235
      DO 230 I=1,N
      YDIFF(I,J)=YDIFF(I,J)-XALPH(I)
230   CONTINUE
235   CONTINUE
      DO 240 J=1,M
      IF (IUSE(J).EQ.0) GO TO 240
      SIGSUM=SIGSUM+DDOT(N,YDIFF(1,J),1,YDIFF(1,J),1)
240   CONTINUE
      TRACE=0.0D0
      DO 245 I=1,K
      TRACE=TRACE+DDOT(K,ZTZ(1,I),1,SUMI(1,I),1)
245   CONTINUE
      SIGMA2=((SIGSUM)/DFLOAT(M*N))+((SIGMA2*TRACE)/DFLOAT(N))
      U=DABS(SIGMA2-SIGOLD)
      DO 255 I=1,K
      DO 250 J=1,I
      R=DABS(DOLD(I,J)-D(I,J))
250   CONTINUE
255   CONTINUE
      IF (R.GT.U) U=R
      IF (U.GT.CONV) GO TO 1100
      IFLAG=1
      WRITE(20,*) 'CONVERGED IN',ITER,'ITERATIONS'
1100  IF(ITER.LT.MAXIT) THEN
         GO TO 1000
      ELSE
         WRITE(20,*) 'FAILED TO CONVERGE IN',MAXIT,'ITERATIONS'
      ENDIF
      RETURN
500   WRITE(20,*)'THE MATRIX TRANS(Z)*Z IS NOT INVERTIBLE'
      STOP
501   WRITE(20,*)'THE MATRIX TRANS(X)*X IS NOT INVERTIBLE'
      STOP
502   WRITE(20,*)' THE MATRIX D IS NOT INVERTIBLE'
      STOP
503   WRITE(20,*) 'THE MATRIX SUMI IS NOT INVERTIBLE'
      STOP
      END
```

```
C      ONCE THE POPULATION PARAMETERS HAVE BEEN
C      CALCULATED BY FITRCB USING ALL OBSERVATIONS
C      FINDEM CALCULATES THE MAXIMUM MAHALNOBIS
C      DISTANCE.  THIS NUMBER IS COMPARED TO A
C      P-VALUE CALCULATED BY THE FUNCTION PVCHI
C      AND EITHER ELIMINATED OR KEPT.  EACH TIME
C      AN OBSERVATION IS ELIMINATED FITRCB IS
C      CALLED TO RE-CALCULATE THE PARAMETERS FOR
C      THE 'TRACKING' POPULATION.



       SUBROUTINE FINDEM (N,M,P,K,SIGNIF)
       IMPLICIT REAL *8(A-H,Q-Z)
       INTEGER N,M,P,K
       INTEGER JPVT(20),IPVT(20)
       DIMENSION X(20,6),Z(20,6),Y(20,200)
       DIMENSION D(6,6),B(6,200),ALPH(6),DET(2)
       DIMENSION ZT(6,20)
       DIMENSION XALPH(20),WORK(20)
       DIMENSION ZDT(6,20),COV(20,20),U(20),DMH(200)
       INTEGER IUSE(200)
       DIMENSION YDIFF(20)
       COMMON/DATAS/X,Z,Y
       COMMON/PARAMS/ALPH,B,SIGMA2,D
       COMMON/ITCON/MAXIT,IFLAG,CONV
       COMMON/USEME/IUSE
       COMMON/FIND/ZT,XALPH
C
C      FIRST NEED TO CALCULATE THE COVARIANCE MATRIX
C      THIS CALCULATES INV(Z*D*TRANS(Z)+SIGMA2*I)
C
1000   DO 10 I=1,N
       DO 5  J=1,K
       ZDT(J,I)=DDOT(K,ZT(1,I),1,D(1,J),1)
5      CONTINUE
10     CONTINUE
       DO 20 I=1,N
       DO 15 J=1,N
       COV(I,J)=DDOT(K,ZDT(1,J),1,ZT(1,I),1)
15     CONTINUE
       COV(I,I)=COV(I,I)+SIGMA2
20     CONTINUE
       CALL DGEFA(COV,20,N,IPVT,INFO)
          IF (INFO.NE.0)THEN
          WRITE(9,*) 'COV IS NOT INVERTIBLE'
          ENDIF
       CALL DGEDI(COV,20,N,IPVT,DET,WORK,1)
C
C      USE CHOLESKY DECOMP TO CALC MAHALANOBIS DIST
C
       DM=0.
       DMAX=-1.0
       DO 30 I=1,M
       IF (IUSE(I).EQ.0) GO TO 30
       DM=DM+1.
       DO 22 J=1,N
```

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



       FUNCTION PVCHI(DMAX,N,DM)
C      WRITTEN BY D MOHR  10/1/89
C      RETURNS PROB. MAX OF M INDEP CHI-SQUARED VARIATES
C      (EACH WITH N D.F.) WILL BE GREATER THAN  D.
       IMPLICIT REAL*8 (A-H,O-Z)
       INTEGER N
       DATA DL7/-.356675/
       RN2=DFLOAT(N)/2.
       DM2=DMAX/2.
       PV=GAMMP(RN2,DM2)
       PV=DLOG(PV)
       DL7M=DL7/DM
       IF (DL7M.GT.PV) THEN
           PVCHI=.3
       ELSE
           PV=DEXP(PV*DM)
           PVCHI=1.-PV
       ENDIF
       RETURN
       END
C
       FUNCTION GAMMQ(A,X)
C      FROM 'NUMERICAL RECIPES'
       IMPLICIT REAL*8 (A-H,O-Z)
       IF(X.LT.0..OR.A.LE.0.)PAUSE
       IF(X.LT.A+1.)THEN
         CALL GSER(GAMSER,A,X,GLN)
         GAMMQ=1.-GAMSER
       ELSE
         CALL GCF(GAMMCF,A,X,GLN)
         GAMMQ=GAMMCF
       ENDIF
       RETURN
       END
C
       SUBROUTINE GSER(GAMSER,A,X,GLN)
C      FROM 'NUMERICAL RECIPES'
       PARAMETER (ITMAX=100,EPS=3.E-7)
```

```
      IMPLICIT REAL*8 (A-H,O-Z)
      GLN=GAMMLN(A)
      IF(X.LE.0.)THEN
        IF(X.LT.0.)PAUSE
        GAMSER=0.
        RETURN
      ENDIF
      AP=A
      SUM=1./A
      DEL=SUM
      DO 11 N=1,ITMAX
        AP=AP+1.
        DEL=DEL*X/AP
        SUM=SUM+DEL
        IF(ABS(DEL).LT.ABS(SUM)*EPS)GO TO 1
11    CONTINUE
      PAUSE 'A TOO LARGE, ITMAX TOO SMALL'
1     GAMSER=SUM*EXP(-X+A*LOG(X)-GLN)
      RETURN
      END
C
      SUBROUTINE GCF(GAMMCF,A,X,GLN)
C     FROM 'NUMERICAL RECIPES'
      PARAMETER (ITMAX=100,EPS=3.E-7)
      IMPLICIT REAL*8 (A-H,O-Z)
      GLN=GAMMLN(A)
      GOLD=0.
      A0=1.
      A1=X
      B0=0.
      B1=1.
      FAC=1.
      DO 11 N=1,ITMAX
        AN=FLOAT(N)
        ANA=AN-A
        A0=(A1+A0*ANA)*FAC
        B0=(B1+B0*ANA)*FAC
        ANF=AN*FAC
        A1=X*A0+ANF*A1
        B1=X*B0+ANF*B1
        IF(A1.NE.0.)THEN
          FAC=1./A1
          G=B1*FAC
          IF(ABS((G-GOLD)/G).LT.EPS)GO TO 1
          GOLD=G
        ENDIF
11    CONTINUE
      PAUSE 'A TOO LARGE, ITMAX TOO SMALL'
1     GAMMCF=EXP(-X+A*DLOG(X)-GLN)*G
      RETURN
      END
C
      FUNCTION GAMMLN(XX)
C     FROM 'NUMERICAL RECIPES'
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 COF(6),STP,HALF,ONE,FPF,X,TMP,SER
      DATA COF,STP/76.18009173D0,-86.50532033D0,24.01409822D0,
     *    -1.231739516D0,.120858003D-2,-.536382D-5,2.50662827465D0/
      DATA HALF,ONE,FPF/0.5D0,1.0D0,5.5D0/
      X=XX-ONE
```

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

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