1975

Program for Missing Data in the Multivariate Normal Distribution

Chi-Ping Lu
Utah State University

PROGRAM FOR MISSING DATA IN THE
MULTIVARIATE NORMAL DISTRIBUTION

by

Chi-Ping Lu

A report submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Applied Statistics

Plan B

Approved:

UTAH STATE UNIVERSITY
Logan, Utah

1975

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to Dr. David White for his guidance and help in fulfilling this report. Thanks also extend to the members of my committee, Dr. Rex L. Hurst for his critical review and much help to solve some important points; and Dr. Donald H. Cooley for his valuable suggestions.

A great deal of thanks go to all professors who contributed to my graduate studies. Moreover, I am highly grateful to my parents for their support and encouragement in my graduate years at Utah State University.

Chi-Ping Lu

TABLE OF CONTENTS

CHAPTER I

INTRODUCTION

Missing data can often cause many problems in research work. Therefore for carrying out analysis, some procedure for obtaining estimates in the presence of missing data should be applied. Various theories and techniques have been developed for different types of problems.

Analysis of the Multivariate Normal Distribution with missing data is one of the areas studied. It has been discussed earlier by Wilkes (1932), Lord (1955), Edgett (1956) and Hartley (1958). They have established some basic concepts and an outline in the way of estimation.

In the last ten years, A. A. Afifi and R. M. Elasfoff also have contributed some important techniques in estimating the parameters respective to mean, variance and covariance. R. R. Hocking, H. H. Oxpring and W. B. Smith are continuously improving it toward a more practical method of calculation. In their paper (1971), they gave the derivation of equations and a numerical example without explaination of the details.

The main purpose of this report is to evaluate the reliability and feasibility of this method by programming it. The procedure will be a general one avaliable for large samples so that research workers can apply it conveniently in estimating the parameters when some observations are missing.

Evaluation of this particular method should consider the following properties:

1. The accuracy of the parameters.
2. The simplicity of the procedure; whether all algorithms needed are adequately described.
3. The generality and flexibility for arbitrary kinds of group classifications.
4. The computing costs.

The method we are discussing can be performed with iterations until all estimators converge within a criteri on. Theoretically, it converges rapidly.

This study includes description of the whole set of theoretical algorithms and the application of a example. A computer program should be able to manipulate the total procedure rapidly and accurately.

Chapter three will provide an outline of the procedure, with each step followed by a numerical example. The construction of the program will also be illustrated.

Chapter four will give the results based on the research done.

Chapter five will bring out the general conclusions and some questionable points of this method.

CHAPTER II

DESCRIPTION OF MAXIMUM LIKELIHOOD ESTIMATION WITH INCOMPLETE NORMAL DATA

2.1 Estimation of mean

The Maximum Likelihood method is a common way for finding point estimates of unknown parameters based on a sample.

Generally, the best statistics satisfy the criteria of "sufficiency", "efficiency", and "consistency" no matter what the distribution is. A maximum likelihood estimator for θ (parameter) can be derived by means of differentiation.

Hocking and Smith have developed a maximum likelihood procedure for estimating parameters in the presence of missing data. This will be the principle concept of this paper. Hocking and Hartley completed a series of equations for the information matrix respective to the estimates of mean and variance/covariance successfully.

We considered the problem of estimating the parameters in multivariate normal population when some of the response vectors were incomplete. Assuming there were N observations taken from a P-variate normal population, which composed the multivariate normal distribution, denoted as $N(u, \Sigma)$, the data could be divided into T groups based on the pattern of incompleteness, the t^{th} group containing N_t observations.

In the classical "Missing Data" problem, we dealt with the vectors as incomplete because not all elements of the P-vector were recorded. We modified the definition by allowing that an incomplete vector may consist of known linear combinations of the original data vector.

The data for this problem can be described as follows:

Y_{ti} for $i = 1, \ldots, n_t$, $t = 1, \ldots, T$

is q_t - variate normal which is $N(u_t, \Sigma_t)$

$$\text{where} \quad u_t = D_t u \tag{1}$$

$$\text{and} \quad \Sigma_t = D_t \, \Sigma \, D_t' \tag{2}$$

u and Σ are the objects of estimation. D_t is a matrix of zeros and ones indicating which observations are recorded, but in general D_t is comprised of known constants. The multivariate - normal distribution function for each t is

$$\frac{1}{(2\pi)^{n_t/2} \, |\Sigma_t|^{1/2}} \, e^{-1/2 (y_{ti} - u_t)' \, \Sigma_t^{-1} \, (y_{ti} - u_t)} \tag{3}$$

Let L_t denote the equation (3).

According to the definition of Maximum Likelihood function

$$L = \prod_{t=1}^{T} L_t$$

Rewrite eq. (3) into

$$\log L_t = C - 1/2 \log|\Sigma_t| - 1/2 \, \mathrm{tr}[\Sigma_t^{-1} \, (y_{ti} - u_t)(y_{ti} - u_t)']$$

$$\log L = C - n_t/2 \log|\Sigma_t| - 1/2 \, \mathrm{tr}(\Sigma_t^{-1} \, A_t) \tag{4}$$

The matrix A_t is given by

$$A_t = \sum_{i=1}^{n_t} (Y_{ti} - u_t)(Y_{ti} - u_t)'$$

$$= \sum_{i=1}^{n_t} (Y_{ti} - \hat{u}_t)(Y_{ti} - \hat{u}_t)' + n_t(\hat{u}_t - u_t)(\hat{u}_t - u_t)'$$

$$= n_t(\hat{\Sigma}_t + H_t) \tag{5}$$

$\hat{\Sigma}_t$ and H_t matrices will be defined by

$$\hat{\Sigma}_t = 1/n_t \; (Y_{ti} - \hat{u}_t)(Y_{ti} - \hat{u}_t)' \qquad (6)$$

$$H_t = (\hat{u}_t - u_t)(\hat{u}_t - u_t)' \qquad (7)$$

where

$$\hat{u}_t = 1/n_t \sum_{i=1}^{n_t} Y_{ti} \qquad (8)$$

Σ is a var/cov matrix shown as:

$$\Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \sigma_{14} \\ & \sigma_{22} & \sigma_{23} & \sigma_{24} \\ & & \sigma_{33} & \sigma_{34} \\ & & & \sigma_{44} \end{bmatrix}$$

For the convenience of calculation, convert the square matrix into a column vector whose elements are in the same order of the column of the Σ matris, under the condition of $(\sigma_{ij}, 1 \leq i \leq j = 1, \ldots, p)$ (9)

p is the number of all variables.

The vector σ has a length of $p(p+1)/2$. The vector σ_t of length $q_t(q_t+1)/2$ represents the corresponding column array of the matrix Σ_t. For instance,

$$\Sigma_2 = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ & \sigma_{22} & \sigma_{23} \\ & & \sigma_{33} \end{bmatrix} \qquad \sigma_2 = [\sigma_{11} \;\; \sigma_{12} \;\; \sigma_{22} \;\; \sigma_{13} \;\; \sigma_{23} \;\; \sigma_{33}]'$$

To relate σ_t with σ, there exists a matrix C_t, analogous to D_t

$$\sigma_t = C_t \sigma \qquad (10)$$

To develop the likelihood equations, the following differentiations have to be done.

$$\frac{\partial \log L_t}{\partial u_t} = \frac{\partial}{\partial u_t}(C - n_t/2 \log|\Sigma_t| - 1/2 \operatorname{tr}(\Sigma_t^{-1} A_t))$$

$$= - n_t \Sigma_t^{-1} (u_t - \hat{u}_t) \tag{11}$$

and

$$\frac{\partial \log L_t}{\partial u} = D_t' \frac{\partial \log L_t}{\partial u_t} \tag{12}$$

Hence

$$\frac{\partial \log L}{\partial u} = \frac{\partial}{\partial u_t} (\prod_{t=1}^{T} \log L_t)$$

$$= \sum_{t=1}^{T} D_t' \frac{\partial \log L_t}{\partial u_t}$$

$$= - n_t \sum_{t=1}^{T} D_t' \Sigma_t^{-1} (D_t u - \hat{u}_t) \tag{13}$$

here $D_t u = u_t$

Differentiate log L_t twice respective to u_t, then

$$\frac{\partial \log L_t}{\partial u_t' \partial u_t} = - n_t \Sigma_t^{-1} \tag{14}$$

We name the portion of the information matrix W_{ut} corresponding to u_t.

$$W_{ut} = n_t \Sigma_t^{-1} \tag{15}$$

We may rewrite eq. (13) as follows:

$$\frac{\partial \log L}{\partial u} = - \sum_{t=1}^{T} D_t' W_{ut} u_t + \sum_{t=1}^{T} D_t' W_{ut} \hat{u}_t$$

$$= - W_u u + \sum_{t=1}^{T} D_t' W_{ut} \hat{u}_t \tag{16}$$

where

$$W_u = \sum_{t=1}^{T} D_t' W_{ut} D_t \tag{17}$$

If we differentiate log L twice with respective to u, the total information matrix for u given W_u can be found.

The estimator of u can be obtained by setting eq.(16) to zero and multiplying by W_u^{-1} under the condition that Σ is known.

$$\tilde{u} = W_u^{-1} \sum_{t=1}^{T} D_t' W_{ut} \hat{u}_t \tag{18}$$

2.2 Estimation of variance/covariance

Develop the likelihood equations for σ.

Let σ_{tij} denote the elements of Σ, and

$$\Sigma_{tij} = \frac{\partial \Sigma_t}{\partial \sigma_{tij}} \tag{19}$$

thus, Σ_{tij} has a one in positions (i, j) and (j,i), and zeros elsewhere. Since we knew

$$\frac{\partial \log|\Sigma_t|}{\partial \sigma_{tij}} = tr\,(\Sigma_t^{-1} \Sigma_{tij}) \tag{20}$$

and

$$\frac{\partial \Sigma_t^{-1}}{\partial \sigma_{tij}} = -\Sigma_t^{-1} \Sigma_{tij} \Sigma_t^{-1} \tag{21}$$

easily solve

$$\frac{\partial \log L_t}{\partial \sigma_{tij}} = \frac{\partial}{\partial \sigma_{tij}} (C - n_t/2 \log|\Sigma_t| - 1/2\, tr(\Sigma_t^{-1} A_t)) \tag{22}$$

by placing eq.(19), (20), and (21) into (22).

$$\frac{\partial \log L_t}{\partial \sigma_{tij}} = -n_t/2\, tr(\Sigma_t^{-1} \Sigma_{tij}) + 1/2\, tr(\Sigma_t^{-1} \Sigma_{tij} \Sigma_t^{-1} A_t) \tag{23}$$

as a likelihood estimate equation of variance.

Concerning the estimate equation of covariance, eq.(23) have to be differentiated once more, easily yielding

$$\frac{\partial \log L_t}{\partial \sigma_{trs}\, \partial \sigma_{tij}} = n_t/2 \, tr(\Sigma_t^{-1} \Sigma_{trs} \Sigma_t^{-1} \Sigma_{tij}) - 1/2 \, tr(\Sigma_t^{-1} \Sigma_{trs} \Sigma_t^{-1} \Sigma_{tij} \Sigma_t^{-1} A_t) - 1/2 \, tr(\Sigma_t^{-1} \Sigma_{tij} \Sigma_t^{-1} \Sigma_{trs} \Sigma_t^{-1} A_t) \quad (24)$$

because $E(A_t) = n_t \Sigma_t$

The expected value of (24) is

$$- n_t/2 \, tr(\Sigma_t^{-1} \Sigma_{tij} \Sigma_t^{-1} \Sigma_{trs}) \quad (25)$$

Logically, we can obtain the likelihood equation of variance by setting eq.(23) to zero, but in fact, this equation are too complicated and difficult to solve. We try another way to simplify it.

We define σ vector from Σ matris as

$$\sigma = (\sigma_{11} \ \sigma_{12} \ \ldots \ \sigma_{1p} \ldots \sigma_{pp})$$

the order of the elements is in the column order of Σ ; same is σ^{ij} of Σ^{-1} displayed as:

$$\sigma^{-1} = (\sigma^{11} \ \sigma^{12} \ \sigma^{22} \ldots \sigma^{1p} \ldots \sigma^{pp})$$

By the definition of σ and σ^{-1}, we can define the following

$$\Delta = \frac{\partial \sigma}{\partial \sigma^{-1}} = \begin{bmatrix} \frac{\partial \sigma_{11}}{\partial \sigma^{11}} & \frac{\partial \sigma_{12}}{\partial \sigma^{11}} & \ldots & \frac{\partial \sigma_{pp}}{\partial \sigma^{11}} \\ \vdots & & & \vdots \\ \frac{\partial \sigma_{11}}{\partial \sigma^{pp}} & \frac{\partial \sigma_{12}}{\partial \sigma^{pp}} & \ldots & \frac{\partial \sigma_{pp}}{\partial \sigma^{pp}} \end{bmatrix} \quad (26)$$

The following relationship exists,

$$\frac{\partial \sigma_{ij}}{\partial \sigma^{uv}} = \begin{cases} -\sigma_{iv}\sigma_{ju} & \text{if } u = v \\ -(\sigma_{iu}\sigma_{jv} + \sigma_{iv}\sigma_{ju}) & \text{if } u \neq v \end{cases} \tag{27}$$

Thus the information matrix defined as $W_{\sigma t}$ can be found, and it is a square matrix of dimension $q_t(q_t+1)/2$.

$$W_{\sigma t} = n_t \ (C_t \ U \ C_t')^{-1} \tag{28}$$

Where U is a square matrix of dimension $p(p+1)/2$, whose components are the products of covariances defined in eq.(27).

Displaying U matrix more explicitly, the element in row(u, v) and column(i, j) for $1 \leqslant u \leqslant v = 1,\dots, p$, $1 \leqslant i \leqslant j = 1,\dots,p$ is expressed as

$$U_{(u,\ v),(i,\ j)} = \sigma_{iu}\sigma_{jv} + \sigma_{iv}\sigma_{ju} \tag{29}$$

The symmetrical U matrix is

$$\begin{bmatrix} \mu_{11,11} & \mu_{11,12} & \mu_{11,22} & \mu_{11,13} & \mu_{11,23} & \mu_{11,33} & \mu_{11,14} & \cdots & \mu_{11,44} \\ \mu_{12,11} & & & & & & & & \mu_{12,44} \\ \mu_{22,11} & & & & & & & & \mu_{22,44} \\ \mu_{13,11} & & & & & & & & \mu_{13,44} \\ \mu_{23,11} & & & & & & & & \mu_{23,44} \\ \mu_{33,11} & & & & & & & & \mu_{33,44} \\ \mu_{14,11} & & & & & & & & \mu_{14,44} \\ \mu_{24,11} & & & & & & & & \mu_{24,44} \\ \mu_{34,11} & & & & & & & & \mu_{34,44} \\ \mu_{44,11} & \mu_{44,12} & \mu_{44,22} & \mu_{44,13} & \mu_{44,23} & \mu_{44,33} & \mu_{44,14} & \cdots & \mu_{44,44} \end{bmatrix}$$

Moreover, the information matrix for likelihood L_t, combined by W_{ut} and $W_{\sigma t}$, is block diagonal

$$W_t = \begin{bmatrix} W_{ut} & 0 \\ 0 & W_{\sigma t} \end{bmatrix} \quad (30)$$

Since we already know

$$\Sigma_t = \sum_{(all\ rs)} \sigma_{trs} \Sigma_{trs} \quad (31)$$

therefore

$$tr\ (\Sigma_t^{-1}\ \Sigma_{tij}) = tr\ (\Sigma_t^{-1}\ \Sigma_{tij}\ \Sigma_t^{-1}\ \Sigma_t)$$

$$= tr\ (\Sigma_t^{-1}\ \Sigma_{tij}\ \Sigma_t^{-1} \sum_{(all\ rs)} \sigma_{trs}\ \Sigma_{trs})$$

$$= \sum_{(all\ rs)} tr\ (\Sigma_t^{-1}\ \Sigma_{tij}\ \Sigma_t^{-1}\ \Sigma_{trs})\ \sigma_{trs} \quad (32)$$

Recalling (23) the estimation equation of covariance is

$$- n_t/2\ tr\ (\Sigma_t^{-1}\ \Sigma_{tij}) + 1/2\ tr\ (\Sigma_t^{-1}\ \Sigma_{tij}\ \Sigma_t^{-1}\ A_t)$$

Compare(32) to the first term on the right-side of eq.(23), it obviously indicating this term is the (i, j)th component of $W_{\sigma t}\cdot\sigma_t$.

Further, we note that the second trem on the right-side of eq. (23) happeded to be

$$1/2\ tr\ (\Sigma_t^{-1}\ \Sigma_{tij}\ \Sigma_t^{-1}\ A_t) = 1/2\ tr\ (\ \Sigma_t^{-1}\ \Sigma_{tij}\ \Sigma_t^{-1}\ n_t(\hat{\Sigma}_t + H_t))$$

$$= W_{\sigma t}\ (\hat{\sigma}_t + h_t) \quad (33)$$

where $\hat{\sigma}_t$ and h_t are in vector forms of $\hat{\Sigma}_t$ and H_t. Therefore

$$\frac{\partial \log L_t}{\partial \sigma} = - W_{\sigma t}\ (\sigma_t - (\hat{\sigma}_t + h_t)) \quad (34)$$

Similar to u, the likelihood equation of σ is given as

$$\frac{\partial \log L_t}{\partial \sigma} = C_t' \frac{\partial \log L_t}{\partial \sigma_t}$$

Since $\sigma_t = C_t \sigma$

$$\frac{\partial \log L}{\partial \sigma} = \prod_{t=1}^{T} \frac{\partial \log L_t}{\partial \sigma}$$

$$= - \sum_{t=1}^{T} C_t' W_{\sigma t} (C_t \sigma - (\hat{\sigma}_t + h_t))$$

$$= - W_{\sigma} \cdot \sigma + \sum_{t=1}^{T} C_t' W_{\sigma t} (\hat{\sigma}_t + h_t) \qquad (35)$$

Where W_σ is the total information matrix for σ shown as

$$W_\sigma = \sum_{t=1}^{T} C_t' W_{\sigma t} C_t \qquad (36)$$

The two likelihood estimation equations of W_u - eq.(16) and W_σ - eq.(35) basically consisted of an identical structure.

It was assumed that all elements of u are estimable. It didn't follow that all elements of σ are estimable too, hence in eq.(35) the vector σ should be interpreted as the vector of estimable parameters from Σ.

In terms of finding the solutions for u and σ based on equation (16) and (35), there are several ways to do, but the simplest one is setting them to zero.

Therefore the two likelihood estimation equations for u and σ are

$$W_u \cdot u = \sum_{t=1}^{T} D_t' W_{ut} \hat{u}_t \qquad (37)$$

$$W_\sigma \cdot \sigma = \sum_{t=1}^{T} C_t' W_{\sigma t} (\hat{\sigma}_t + h_t) \qquad (38)$$

Eventually

$$u = W_u^{-1} \sum_{t=1}^{T} D_t' \, W_{ut} \, \hat{u}_t \tag{39}$$

$$\sigma = W_\sigma^{-1} \sum_{t=1}^{T} C_t' \, W_{\sigma t} \, (\hat{\sigma}_t + h_t) \tag{40}$$

The estimation equations for u and σ are shown in eq.(39) and (40). The estimation of u, σ and h_t started by estimating W_{ut}, $W_{\sigma t}$ and h_t with the initial estimators of u and σ as we assumed previously. After the first time, we repeat the whole process by replacing the initial estimators with the resulting estimates from last iteration. Emperically, after a period of iterations, the modified u and σ will rapidly converge on both of the simulated and actual data.

CHAPTER III

APPLICATION OF NUMERICAL EXAMPLE

3.1 General description

For carrying out the theory, a numerical example was applied to follow the outlined sequence. Every individual step will be explained in detail with actual data so that the whole method can be observed clearly.

The example was the Iris setosa of Multiple Measurements from a taxonomic problem, it was assumed n = 50 observation taken on a 4 - variate normal population, which were divided into 4 groups, according to the pattern of incompleteness. Of the 50 records 31 were complete and 19 were incomplete in different ways.

The original data of 4 groups were shown in appendix 1.

The 4 groups were tabulated as:

P: Presence

M: Missing

Group	X_1	X_2	X_3	X_4	n_i: Obs. in i^{th} grp.	q_t: No of vat.
1	P	P	P	P	$n_1 = 31$	$q_1 = 4$
2	P	P	P	M	$n_2 = 8$	$q_2 = 3$
3	P	P	M	M	$n_3 = 6$	$q_3 = 2$
4	M	M	P	M	$n_4 = 5$	$q_4 = 1$
	Total T = 4				N = 50	

3.2 Procedure

The whole procedure was iterated by means of replacing the estimators of u, Σ and H_t until they converged to constants.

The stepwise sequence in each iteration was following:

1. The $u^{(\ell)}$ (ℓ indicated the number of iteration) would be one of the objectives of estimation. Besides, it also needed an initial value for starting. The best recommended way was using $\hat{u}_1$ as initial estimator of u, denoted as $u^{(0)}$ (0 represented the initial estimator) which came from complete data group. D_t was designed with 0 and 1 indicating the elements of u presented in the u_t vector. Thus $u_t^{(\ell)} = D_t\ u^{(\ell)}$ and $H_t^{(\ell)} = (u_t^{(\ell)} - \hat{u}_t)(u_t^{(\ell)} - \hat{u}_t)'$ could be obtained, where $\hat{u}_t = 1/n_t \sum_{i=1}^{n_t} Y_{ti}$ kept as constants in every iteration. In our case,

$$\hat{u}_1 = u^{(0)} = 1/31 \sum_{i=1}^{31} Y_{ti} = \begin{bmatrix} 4.99 \\ 3.43 \\ 1.45 \\ 0.24 \end{bmatrix} \qquad \hat{u}_2 = 1/8 \sum_{i=1}^{8} Y_{ti} = \begin{bmatrix} 4.90 \\ 3.35 \\ 1.37 \end{bmatrix}$$

$$\hat{u}_3 = 1/6 \sum_{i=1}^{6} Y_{ti} = \begin{bmatrix} 5.12 \\ 3.63 \end{bmatrix} \qquad \hat{u}_4 = 1/5 \sum_{i=1}^{5} Y_{ti} = [\ 1.52\ ]$$

D_t could be defined as:

$$D_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad D_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$D_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \qquad D_4 = [\ 0 \quad 0 \quad 1 \quad 0\ ]$$

Therefore $u_t^{(0)} = D_t\ u^{(0)}$ and $H_t^{(0)}$ were able to be computed.

$$H_1^{(0)} = (\hat{u}_1 - D_t u^{(0)})(\hat{u}_1 - D_t u^{(0)})' = 0$$

$$H_2^{(0)} = \begin{bmatrix} 0.0081 & 0.0072 & 0.0072 \\ & 0.0064 & 0.0064 \\ & & 0.0064 \end{bmatrix}$$

$$H_3^{(0)} = \begin{bmatrix} 0.0169 & 0.0260 \\ & 0.0400 \end{bmatrix} \qquad H_4^{(0)} = [0.0049]$$

2. The σ vector was another objective of estimation whose components were in the column order of Σ matrix. Its best initial estimator used $\hat{\Sigma}_1$as $\Sigma^{(0)}$, where $\hat{\Sigma}_1$ was the covariance matrix of group 1 which was computed by complete data.

$$\hat{\Sigma}_t = 1/n_t \sum_{i=1}^{n_t} (Y_{ti} - \hat{u}_t)(Y_{ti} - \hat{u}_t)'$$

where $\hat{u}_t$ came from step 1. All of $\hat{\Sigma}_t$ would be constants in every iteration.

$$\hat{\Sigma}_1 = 1/31 \sum_{i=1}^{31} (Y_{ti} - \hat{u}_1)(Y_{ti} - \hat{u}_1)'$$

$$= 1/31 \sum_{i=1}^{31} (Y_{ti} - u^{(0)})(Y_{ti} - u^{(0)})'$$

$$= \Sigma^{(0)} = \begin{bmatrix} 0.158 & 0.140 & 0.015 & 0.010 \\ & 0.185 & 0.013 & 0.008 \\ & & 0.026 & 0.004 \\ & & & 0.012 \end{bmatrix}$$

$$\hat{\Sigma}_2 = \begin{bmatrix} 0.070 & 0.021 & 0.027 \\ & 0.053 & -0.009 \\ & & 0.024 \end{bmatrix}$$

$$\hat{\Sigma}_3 = \begin{bmatrix} 0.018 & 0.026 \\ & 0.096 \end{bmatrix} \qquad \hat{\Sigma}_4 = [\ 0.018\ ]$$

3. $W_{\sigma t} = n_t \ (C_t \ U \ C_t')^{-1}$

n_t came from step 1. U was a matrix in rank $p(p+1)/2 = 10$.

The elements of $U^{(\ell)}$ in row(u, v), column(i, j) = $\sigma_{iu}\sigma_{jv} + \sigma_{iv}\sigma_{ju}$

where $u \leq v = 1,\ldots,p$, $i \leq j = 1,\ldots,p$

In the first iteration the value of σ_{ij} came from step 2, otherwise came from step 6. C_t was matrix related σ to σ_t, satisfying $\sigma_t = C_t\sigma$,

which designed with 0 and 1 indicating the elements of σ presented in σ_t. In our case, $C_1 = I_{(10\times10)}$. The length of C_t was $q_t(q_t+1)/2$.

$$C_{2(6\times10)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C_{3(3\times10)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C_{4(1\times10)} = [\ 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0\ ]$$

Therefore

$$\sigma_{1(10\times1)} = C_{1(10\times10)} \times \sigma_{(10\times1)}$$

$$= [\sigma_{11} \quad \sigma_{12} \quad \sigma_{22} \quad \sigma_{13} \quad \sigma_{23} \quad \sigma_{33} \quad \sigma_{14} \quad \sigma_{24} \quad \sigma_{34} \quad \sigma_{44}]'$$

$$\sigma_{2(6\times1)} = C_{2(6\times10)} \times \sigma_{(10\times1)}$$

$$= [\sigma_{11} \quad \sigma_{12} \quad \sigma_{22} \quad \sigma_{13} \quad \sigma_{23} \quad \sigma_{33}]'$$

$$\sigma_{3(3\times1)} = C_{3(3\times10)} \times \sigma_{(10\times1)}$$

$$= [\sigma_{11} \quad \sigma_{12} \quad \sigma_{22}]'$$

$$\sigma_{4(1\times1)} = C_{4(1\times10)} \times \sigma_{(10\times1)}$$

$$= [\sigma_{33}]'$$

4. $W_\sigma = \sum_{t=1}^{T} C_t' W_{\sigma t} C_t$

C_t and $W_{\sigma t}$ came from step 3.

5. $\sigma^{(\ell)} = W_\sigma^{-1} \sum_{t=1}^{T} C_t' W_{\sigma t} (\hat{\sigma}_t + h_t)$

$W_{\sigma t}$ came from step 3.

$\hat{\sigma}_t$ came from step 2.

h_t came from step 1.

6. $\Sigma_t^{(\ell)} = D_t \Sigma^{(\ell)} D_t'$

$\Sigma^{(\ell)}$ came from step 5.

D_t came from step 1.

7. $W_{ut}^{(\ell)} = n_t \Sigma_t^{(\ell)-1}$

$\Sigma_t^{(\ell)}$ came from step 6.

8. $W_u = \sum_{t=1}^{T} D_t' W_{ut}^{(\ell)} D_t$

$W_{ut}^{(\ell)}$ came from step 7.

9. $u^{(\ell)} = W_u^{-1} \sum_{t=1}^{T} D_t' W_{ut}^{(\ell)} \hat{u}_t$

W_u^{-1} came from step 8.

$\hat{u}_t$ came from step 1.

$W_{ut}^{(\ell)}$ came from step 7.

10. $u_t^{(\ell)} = D_t\ u^{(\ell)}$

D_t came from step 1.

$u^{(\ell)}$ came from step 9.

11. $H_t = (\hat{u}_t - u_t^{(\ell)})(\hat{u}_t - u_t^{(\ell)})'$

$\hat{u}_t$ came from step 1.

$u_t^{(\ell)}$ came from step 10.

The new $u^{(\ell)}$ was obtained from step 9, $\sigma^{(\ell)}$ from step 5, $h_t^{(\ell)}$ from step 11.

From the second iteration place the new Σ into step 3, and complete the whole procedure up to step 11, then $u^{(\ell)}$, $\sigma_t^{(\ell)}$, and $h_t^{(\ell)}$ are able to be produced from every iteration. Repeating the whole procedure until $u^{(\ell)}$, $h_t^{(\ell)}$, and $\sigma_t^{(\ell)}$ vectors converged within a given criteria, the eventual estimates of mean, variance and covariance were found.

3.3 Construction of the program

For the convenience of program manipulation, some modifications have been made.

1. D_t and C_t matrices have been shown with different ranks among 4 groups. In the program, all D_t and $C_t (t = 1,\ldots,4)$ were modified into square and symmetrical matrices. All D_t were (4×4) matrices

wherever the data missed filled with 0. So did in $C_t(10\times 10)$. The C_t and D_t matrices were shown in the begining of the output.

2. For the rank comformity of the initial vector u_t to D_t, kept u_t in size (4×1) and filled with 0 when observation missed. h_t was (10×1) comformed to the rank of C_t.

3. The output vector of h and σ were in the following format:

$$h = [h_{11}\ h_{12}\ h_{13}\ h_{14}\ h_{22}\ h_{23}\ h_{24}\ h_{33}\ h_{34}\ h_{44}]'$$

$$\sigma = [\sigma_{11}\ \sigma_{12}\ \sigma_{13}\ \sigma_{14}\ \sigma_{22}\ \sigma_{23}\ \sigma_{24}\ \sigma_{33}\ \sigma_{34}\ \sigma_{44}]'$$

They were in different order from them shown b efore because of rearrangement.

4. Since the initial value of σ and h vector not in the array of column's order but of the row's order of Σ and H matrix. Therefore the related matrices C_t and D_t also were rearranged. However they still satisfied the condition of $\sigma_t = C_t\sigma$ and $u_t = D_t u$.

Above all, those improvements didn't have any influence on the result of estimators.

The basic structure of the program followed the outline shown above.

In the very begining, we started with calculating

$W_{\sigma t} = n_t(C_t\ U\ C_t')^{-1}$ where the elements of U came from initial value of Σ.

Reorganized those steps agreeing to the program sequence:

STEP 1. $\Sigma_t = D_t\ \Sigma\ D_t'$

STEP 2. $W_{ut} = n_t\ \Sigma_t^{-1}$

STEP 3. $W_u = \sum_{t=1}^{T} D_t' W_{ut} D_t$

STEP 4. new $u^{(\ell)} = W_u^{-1} \sum_{t=1}^{T} D_t' W_{ut} \hat{u}_t$

STEP 5. $u_t^{(\ell)} = D_t u^{(\ell)}$

STEP 6. new $H_t^{(\ell)} = (\hat{u}_t - u_t^{(\ell)})(\hat{u}_t - u_t^{(\ell)})'$

STEP 7. $W_{\sigma t} = n_t (C_t U C_t')^{-1}$

The elements of U came from the input $\Sigma^{(0)}$ in the first iteration.

STEP 8. $W_\sigma = \sum_{t=1}^{T} C_t' W_{\sigma t} C_t$

STEP 9. new $\sigma^{(\ell)} = W_\sigma^{-1} \sum_{t=1}^{T} C_t' W_{\sigma t} (\hat{\sigma}_t + h_t)$

h_t came from step 6.

Therefore $u^{(\ell)}$, $H_t^{(\ell)}$, and $\sigma^{(\ell)}$, could be procured from the ℓ^{th} iteration, the $\ell+1^{th}$ iteration started from placing the $\Sigma^{(\ell)}$ to step 1.

C_t, D_t, n_t and $\hat{\sigma}_t$ were a part of input, keeping as constants all the time.

There were some matrix calculation involved, the IBM scientific subroutine could be applied.

1. Transpose a matrix . . . . subroutine WTRA.
2. Subtract two matrices . . . . subroutine MSUB.
3. Products of two matrices . . . . subroutine GMPRD.

4. Multiple a matrix by a scalar . . . . subroutine SMPY.

5. Inverse a matrix (generalized inverse) subroutine DMTSQ supplied by Dr. Hurst.

6. Computing the U matrix . . . . subroutine UMAT.

CHAPTER IV

RESULTS

The conclusions were based on the program output. The u and σ vector converged very fast requiring only 4 iterations to obtain 4 - decimal accuracy. The estimates of u and σ were shown as:

$$u^{(4)} = \begin{bmatrix} 4.9964 \\ 3.4451 \\ 1.4446 \\ 0.2396 \end{bmatrix} \quad \Sigma^{(4)} = \begin{bmatrix} 0.1239 & 0.1055 & 0.0158 & 0.0078 \\ & 0.1518 & 0.0083 & 0.0059 \\ & & 0.0261 & 0.0041 \\ & & & 0.0109 \end{bmatrix}$$

In our case, since we had such a large ratio of complete records (p = 31/50) the estimates of group 1 were good for exploiting as initial estimates.

The estimates of u and Σ were initiated as the estimates of group 1, and modified by the additional information of incomplete data, consequently, the portion of complete data was playing an important role as to the reliability of the estimates.

The parameters of complete data N = 50 should be as following:

$$u = \begin{bmatrix} 5.0041 \\ 3.4265 \\ 1.4633 \\ 0.2469 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 0.1266 & 0.1011 & 0.0168 & 0.0106 \\ & 0.1466 & 0.0120 & 0.0095 \\ & & 0.0307 & 0.0061 \\ & & & 0.0113 \end{bmatrix}$$

Being statistics, the estimates of u and Σ satisfied the property of "unbiaseness" and "consistency" by comparing them with the parameters. The fact was that although there were 3 incomplete groups, none of them contained too much information. Evidence seems to indicate that accuracy increases with a smaller proportion of missing data, and also with smaller variates.

CHAPTER V

CONCLUSIONS

According to the properties of this method, the following conclusions might be made.

1. The accuracy of the estimates seemed dependent on some factors.

A. It was inevitable using a initial estimate of u and Σ, the precision and reliability of eventual estimates were highly affected by the characteristics of this value. Consequently, a larger portion of complete data produced a better estimates of u and Σ. In other words, the initial estimates have to supply sufficient informations of the variables.

B. For the sake that the initial estimates came from complete data, this method definitely was not avaliable for those problems which didn't include any complete data, nor the amount of these data not enough to make $\hat{\Sigma}_1$ to be a non-singular and full rank matrix. For this point, we concerned ourselves with about the "estimability condition" of this method, which meant that $n_t > p$ and $n_t > q_t$ were necessary for any group. Once we failed to satisfy that all $n_t > q_t$, it never yielded the convergent estimates, even in the case of $n_t = p$, or $n_t = q_t$.

C. This method is avaliable for large sample size, but also works for small sample sizes as long as the variable was taken from the normal distribution and met the condition of B.

Sample size also affected the accuracy of the estimates, because a small sample size didn't have as much as information to work as did in large sample size. Therefore, selecting moderate amount of sample size will produce better estimates.

2. The whole procedure could be performed in one program, it was convenient for users by simply placing the initial values for mean, variance/covariance, H_t, C_t, D_t matrices.

3. The possible combinations of missing data types are $2^p - 1$ (p is the number of variables), for large p, the group numbers of C_t and D_t becoming tremendous values so that they are too large to compute. For instance, if P = 10, the maximum group combination could be $2^{10} - 1 = 1023$. It becomes a heavy load for user to prepare the input data, even though it possibly could be done. This method provided absolute flexibility in arbitrary kinds of group classifications, but with small variables it worked more efficiently.

4. The costs of computer time are also high both of storage for the various data and C. P. U. processing time for iteration. Recalling the example of 10 variables and assuming the iteration times were 10, how much it would cost? However, in our example, it cost $3.50 for 4 variables in 4 group with 4 iterations.

For further study, it is suggested that under the conditions of variables $p < 10$, and the classification $T < 10$, this method will be a convenient and powerful one for estimation.

LITERATURE CITED

Afifi, A. A., and R. M. Elashoff. 1966. Missing observations in multivariate statistics I. Journal of the American Statistical Association, Vol. 61, P. 595-604.

Anderson, T. W. 1957. Maximum likelihood estimates for a multivariate normal distribution when some observations are missing. Journal of the American Statistical Association, Vol. 52, P. 200-203.

Hartley, H. O. 1958. Maximum likelihood estimation from incomplete data. Biometrics, Vol. 14, P. 174-194.

Hocking, R. R., and William B. Smith. 1968. Estimation of parameters in the multivariate normal distribution with missing observations. Journal of the American Statistical Association, Vol. 63, P. 159-173.

Hocking, R. R., W. B. Smith, B. R. Waldron and H. H. Oxspring. 1969. Estimation of parameters with incomplete data. Technical Report No. 12, Texas A&M University, project themis.

Hocking, R. R. and H. H. Oxspring. 1971. Maximum likelihood estimation with incomplete multinomial data. Journal of the American Statistical Association.

Hocking, R. R. and H. O. Hartley. 1971. The analysis of incomplete data. Biometrics, Vol. 27, P. 783-823.

Hocking, R. R. and William B. Smith. 1971. Optimum incomplete multinormal samples. Technometrics.

Traxinske, I. M. and R. E. Bargmann. 1964. Maximum likelihood estimation with incomplete multivariate data. Annals of Mathematical Statistics, Vol. 35, P. 647-657.

APPENDIXES

Appendix 1

The Table of 4 Groups of Multiple Measurements in Taxonomic Problem

X: Indicates the data missed

q_t: The number of varible in t^{th} group

n_t: The number of observations in t^{th} group

Group 1. $q_1 = 4$, $n_1 = 31$

	Varible		
X_1	X_2	X_3	X_4
5.1	3.5	1.4	0.2
4.7	3.2	1.3	0.2
4.6	3.1	1.5	0.2
5.0	3.6	1.4	0.2
4.6	3.4	1.4	0.3
4.4	2.9	1.4	0.2
5.4	3.7	1.5	0.2
4.8	3.4	1.6	0.2
4.8	3.0	1.4	0.1
4.3	3.0	1.1	0.1
5.8	4.0	1.2	0.2
5.7	4.4	1.5	0.4
5.4	3.4	1.7	0.2
5.1	3.7	1.5	0.4
4.8	3.4	1.9	0.2
5.0	3.4	1.6	0.4
5.2	3.5	1.5	0.2
5.2	3.4	1.4	0.2
4.8	3.1	1.6	0.2
5.2	4.1	1.5	0.1
5.5	4.2	1.4	0.2
5.0	3.2	1.2	0.2
4.9	3.6	1.4	0.1
4.4	3.0	1.3	0.2
5.1	3.4	1.5	0.2
4.5	2.3	1.3	0.3
5.0	3.5	1.6	0.6
4.8	3.0	1.4	0.3
4.6	3.2	1.4	0.2
5.4	3.9	1.3	0.4
5.7	3.8	1.7	0.3

Group 2. $q_2 = 3$, $n_2 = 8$

	Varible		
X_1	X_2	X_3	X_4
4.9	3.0	1.4	X
5.0	3.4	1.5	X
4.9	3.1	1.5	X
5.1	3.5	1.4	X
4.6	3.6	1.0	X
4.4	3.2	1.3	X
5.3	3.7	1.5	X
5.0	3.3	1.4	X

Group 3. $q_3 = 2$, $n_3 = 6$

	Varible		
X_1	X_2	X_3	X_4
5.4	3.9	X	X
5.1	3.8	X	X
5.0	3.0	X	X
5.0	3.5	X	X
5.1	3.8	X	X

Group 4. $q_4 = 1$, $n_4 = 5$

	Varible		
X_1	X_2	X_3	X_4
X	X	1.7	X
X	X	1.6	X
X	X	1.5	X
X	X	1.5	X
X	X	1.3	X

Appendix 2

Computer Program

The list of all subroutines and main program referred for missing data in the multivariate normal distribution.

```
C
C     ..................................................................
C
C        SUBROUTINE MTRA
C
C        PURPOSE
C           TRANSPOSE A MATRIX
C
C        USAGE
C           CALL MTRA(A,R,N,M,MS)
C
C        DESCRIPTION OF PARAMETERS
C           A - NAME OF MATRIX TO BE TRANSPOSED
C           R - NAME OF OUTPUT MATRIX
C           N - NUMBER OF ROWS OF A AND COLUMNS OF R
C           M - NUMBER OF COLUMNS OF A AND ROWS OF R
C           MS  - ONE DIGIT NUMBER FOR STORAGE MODE OF MATRIX A (AND R)
C                  0 - GENERAL
C                  1 - SYMMETRIC
C                  2 - DIAGONAL
C
C        REMARKS
C           MATRIX R CANNOT BE IN THE SAME LOCATION AS MATRIX A
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           MCPY
C
C        METHOD
C           TRANSPOSE N BY M MATRIX A TO FORM M BY N MATRIX R BY MOVING
C           EACH ROW OF A INTO THE CORRESPONDING COLUMN OF R. IF MATRIX
C           A IS SYMMETRIC OR DIAGONAL, MATRIX R IS THE SAME AS A.
C
C     ..................................................................
C
      SUBROUTINE MTRA(A,R,N,M,MS)
      DIMENSION A(100),R(100)
C
C        IF MS IS 1 OR 2, COPY A
C
      IF(MS) 10,20,10
   10 CALL MCPY(A,R,N,N,MS)
      RETURN
C
C        TRANSPOSE GENERAL MATRIX
C
   20 IR=0
      DO 30 I=1,N
      IJ=I-N
      DO 30 J=1,M
      IJ=IJ+N
      IR=IR+1
   30 R(IR)=A(IJ)
      RETURN
      END
```

```
C
C     ..................................................................
C
C        SUBROUTINE MSUB
C
C        PURPOSE
C           SUBTRACT TWO MATRICES ELEMENT BY ELEMENT TO FORM RESULTANT
C           MATRIX
C
C        USAGE
C           CALL MSUB(A,B,R,N,M,MSA,MSB)
C
C        DESCRIPTION OF PARAMETERS
C           A - NAME OF INPUT MATRIX
C           B - NAME OF INPUT MATRIX
C           R - NAME OF OUTPUT MATRIX
C           N - NUMBER OF ROWS IN A,B,R
C           M - NUMBER OF COLUMNS IN A,B,R
C           MSA - ONE DIGIT NUMBER FOR STORAGE MODE OF MATRIX A
C                  0 - GENERAL
C                  1 - SYMMETRIC
C                  2 - DIAGONAL
C           MSB - SAME AS MSA EXCEPT FOR MATRIX B
C
C        REMARKS
C           NONE
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           LOC
C
C        METHOD
C           STRUCTURE OF OUTPUT MATRIX IS FIRST DETERMINED. SUBTRACTION
C           OF MATRIX B ELEMENTS FROM CORRESPONDING MATRIX A ELEMENTS
C           IS THEN PERFORMED.
C           THE FOLLOWING TABLE SHOWS THE STORAGE MODE OF THE OUTPUT
C           MATRIX FOR ALL COMBINATIONS OF INPUT MATRICES
C                       A                B                R
C                       GENERAL          GENERAL          GENERAL
C                       GENERAL          SYMMETRIC        GENERAL
C                       GENERAL          DIAGONAL         GENERAL
C                       SYMMETRIC        GENERAL          GENERAL
C                       SYMMETRIC        SYMMETRIC        SYMMETRIC
C                       SYMMETRIC        DIAGONAL         SYMMETRIC
C                       DIAGONAL         GENERAL          GENERAL
C                       DIAGONAL         SYMMETRIC        SYMMETRIC
C                       DIAGONAL         DIAGONAL         DIAGONAL
C
C     ..................................................................
C
      SUBROUTINE MSUB(A,B,R,N,M,MSA,MSB)
      DIMENSION A(100),B(100),R(100)
C
C        DETERMINE STORAGE MODE OF OUTPUT MATRIX
C
      IF(MSA-MSB) 7,5,7
    5 CALL LOC(N,M,NM,N,M,MSA)
      GO TO 100
    7 MTEST=MSA*MSB
      MSR=0
      IF(MTEST) 20,20,10
```

```
   10 MSR=1
   20 IF(MTEST-2) 35,35,30
   30 MSR=2
C
C        LOCATE ELEMENTS AND PERFORM SUBTRACTION
C
   35 DO 90 J=1,M
      DO 90 I=1,N
      CALL LOC(I,J,IJR,N,M,MSR)
      IF(IJR) 40,90,40
   40 CALL LOC(I,J,IJA,N,M,MSA)
      AEL=0.0
      IF(IJA) 50,60,50
   50 AEL=A(IJA)
   60 CALL LOC(I,J,IJB,N,M,MSB)
      BEL=0.0
      IF(IJB) 70,80,70
   70 BEL=B(IJB)
   80 R(IJR)=AEL-BEL
   90 CONTINUE
      RETURN
C
C        SUBTRACT MATRICES FOR OTHER CASES
C
  100 DO 110 I=1,NM
  110 R(I)=A(I)-B(I)
      RETURN
      END
```

```
C
C     ..................................................................
C        SUBROUTINE LOC
C        PURPOSE
C           COMPUTE A VECTOR SUBSCRIPT FOR AN ELEMENT IN  A MATRIX OF
C           SPECIFIED STORAGE MODE
C        USAGE
C           CALL LOC (I,J,IR,N,M,MS)
C        DESCRIPTION OF PARAMETERS
C           I - ROW NUMBER OF ELEMENT
C           J - COLUMN NUMBER OF ELEMENT
C           IR - RESULTANT VECTOR OF ROWS IN MATRIX
C           M - NUMBER OF COLUMNS IN MATRIX
C           MS - ONE DIGIT NUMBER FOR STORAGE MODE OF MATRIX
C                 0 - GENERAL
C                 1 - SYNNETRIC
C                 2 - DIAGONAL
C        SUBROUTINES AND FUNCITION SUBPROGRAMS REQUIRED
C           NONE

      SUBROUTINE LOC(I,J,IR,N,M,MS)
      IX=I
      JX=J
      IF(MS-1) 10,20,30
   10 IRX=N*(JX-1)+IX
      GO TO 36
   20 IF(IX-JX) 22,24,24
   22 IRX=IX+(JX*JX-JX)/2
      GO TO 36
   24 IRX=JX+(IX*IX-IX)/2
      GO TO 36
   30 IRX=0
      IF(IX-JX) 36,32,36
   32 IRX=IX
   36 IR=IRX
      RETURN
      END
```

```
C
C     ..................................................................
C
C        SUBROUTINE SMPY
C
C        PURPOSE
C           MULTIPLY EACH ELEMENT OF A MATRIX BY A SCALAR TO FORM A
C           RESULTANT MATRIX
C
C        USAGE
C           CALL SMPY(A,C,R,N,M,MS)
C
C        DESCRIPTION OF PARAMETERS
C           A - NAME OF INPUT MATRIX
C           C - SCALAR
C           R - NAME OF OUTPUT MATRIX
C           N - NUMBER OF ROWS IN MATRIX A AND R
C           M - NUMBER OF COLUMNS IN MATRIX A AND R
C           MS  - ONE DIGIT NUMBER FOR STORAGE MODE OF MATRIX A (AND R)
C                  0 - GENERAL
C                  1 - SYMMETRIC
C                  2 - DIAGONAL
C
C        REMARKS
C           NONE
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           LOC
C
C        METHOD
C           SCALAR IS MULTIPLIED BY EACH ELEMENT OF MATRIX
C
C     ..................................................................
C
      SUBROUTINE SMPY(A,C,R,N,M,MS)
      DIMENSION A(100),R(100)
C
C        COMPUTE VECTOR LENGTH, IT
C
      IF (N-10) 2,10,10
   10 CALL LOC(10,10,IT,10,10,0)
      GO TO 100
    2 CALL LOC(4,4,IT,4,4,0)
  100 DO 1 I=1,IT
C
C        MULTIPLY BY SCALAR
C
    1 R(I)=A(I)*C
      RETURN
      END
```

```
C     ..................................................................
C
C        SUBROUTINE GMPRD
C
C        PURPOSE
C           MULTIPLY TWO GENERAL MATRICES TO FORM A RESULTANT GENERAL
C           MATRIX
C
C        USAGE
C           CALL GMPRD(A,B,R,N,M,L)
C
C        DESCRIPTION OF PARAMETERS
C           A - NAME OF FIRST INPUT MATRIX
C           B - NAME OF SECOND INPUT MATRIX
C           R - NAME OF OUTPUT MATRIX
C           N - NUMBER OF ROWS IN A
C           M - NUMBER OF COLUMNS IN A AND ROWS IN B
C           L - NUMBER OF COLUMNS IN B
C
C        REMARKS
C           ALL MATRICES MUST BE STORED AS GENERAL MATRICES
C           MATRIX R CANNOT BE IN THE SAME LOCATION AS MATRIX A
C           MATRIX R CANNOT BE IN THE SAME LOCATION AS MATRIX B
C           NUMBER OF COLUMNS OF MATRIX A MUST BE EQUAL TO NUMBER OF ROW
C           OF MATRIX B
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           THE M BY L MATRIX B IS PREMULTIPLIED BY THE N BY M MATRIX A
C           AND THE RESULT IS STORED IN THE N BY L MATRIX R.
C
C     ..................................................................
C
      SUBROUTINE GMPRD(A,B,R,N,M,L)
      DIMENSION A(100),B(100),R(100)
C
      IR=0
      IK=-M
      DO 10 K=1,L
      IK=IK+M
      DO 10 J=1,N
      IR=IR+1
      JI=J-N
      IB=IK
      R(IR)=0
      DO 10 I=1,M
      JI=JI+N
      IB=IB+1
   10 R(IR)=R(IR)+A(JI)*B(IB)
      RETURN
      END
```

```
C
C     ...............................................................
C        SUBROUTINE MCPY
C        PURPOSE
C           COPY ENTIRE MATRIX
C        USAGE
C           CALL MCPY
C        DESCRIPTION OF PARAMETERS
C           A - NAME OF INPUT MATRIX
C           R - NAME OF OUTPUT MATRIX
C           N - NUMBER OF ROWS IN A OR R
C           M - NUMBER OF COLUMNS IN A OR R
C           MS - ONE DIGIT NUMBER FOR STORAGE MODE OF MATRIX A (AND R)
C              0-GENERAL
C              1-SYMMETRIC
C              2-DIAGONAL
C        SUBROUTINE REQUIRED LOC

      SUBROUTINE MCPY(A,R,N,M,MS)
      DIMENSION A(100),R(100)
      CALL LOC(N,M,IT,N,M,MS)
      DO 1 I=1,IT
    1 R(I)=A(I)
      RETURN
      END
```

```
C
C      ..................................................................
C      SUBROUTINE DMTSQ

      SUBROUTINE DMTSQ(A,N1,N2,N3,NY,DET,TEST,NI)
C*     ASYMMETRIC MATRIX INVERSION ROUTINE.  INVERTS PORTION BETWEEN N1
C*     AND N2.  RIGHT HAND SIDES START AT N3 AND GO FOR NY COLUMNS.
C*     INVERSE WILL REPLACE ORIGINAL MATRIX AND SOLUTIONS WILL REPLACE
C*     RIGHT HAND SIDES. THE DETERMINANT IS ALSO COMPUTED.  TEST IS A CRI
C*     TO BE USED IN OBTAINING A G-INVERSE, TEST IS A SUITABLE SMALL CONS
C*     ANY DIAGONAL ELEMENT SMALLER THAN TEST WILL BE SET TO ZERO.
C*     THE NUMBER OF RIGHT HAND SIDES MAY BE ZERO.  THE SIZE OO THE ARRAY
C*     IN THE MAIN PROGRAN IS 'NI'.
      DIMENSION A(NI,NI)
      NK=N3+NY-1
      DET=1.0
      DO 608 L=N1,N2
      IF(ABS(A(L,L)).GE.TEST) GO TO 601
      A(L,L)=0.0
      DET=0.0
      GO TO 602
  601 DET=DET*A(L,L)
      A(L,L)=1.0/A(L,L)
  602 CONTINUE
      DO 605 I=N1,N2
      IF(I.EQ.L) GO TO 605
      A(I,L)=A(I,L)*A(L,L)
      DO 603 J=N1,N2
      IF(J.EQ.L) GO TO 603
      A(I,J)=A(I,J)-A(I,L)*A(L,J)
  603 CONTINUE
      IF(NY.LE.0) GO TO 605
      DO 604 J=N3,NK
      A(I,J)=A(I,J)-A(I,L)*A(L,J)
  604 CONTINUE
  605 CONTINUE
      DO 606 J=N1,N2
      IF(J.EQ.L) GO TO 606
      A(L,J)=-A(L,L)*A(L,J)
  606 CONTINUE
      IF(NY.LE.0) GO TO 608
      DO 607 J=N3,NK
      A(L,J)=-A(L,L)*A(L,J)
  607 CONTINUE
  608 CONTINUE
      IF(NY.EQ.0) GO TO 610
      DO 609 I=N1,N2
      DO 609 J=N3,NK
      A(I,J)=-A(I,J)
  609 CONTINUE
  610 RETURN
      END
```

```
C
C     ..............................................................
C       SUBROUTINE UMAT
C       PURPOSE
C       CALCULATE THE U MATRIX
C       DESCRIPTION OF PARAMETERS
C          A - NAME OF INPUT MATRIX
C          B - NAME OF OUTPUT MATRIX
C       USAGE
C          CALL UMAT

      SUBROUTINE UMAT(A,B)
      INTEGER RI,CI
      DIMENSION A(4,4),B(10,10)
      DO 1 I=1,4
      DO 2 J=1,4
      RI=(I-1)*4+J
      IF(RI-5) 11,2,12
   12 IF(RI-8)13,13,14
   13 RI=RI-1
      GO TO 11
   14 IF(RI-10)2,2,15
   15 IF(RI-12) 16,16,17
   16 RI=RI-3
      GO TO 11
   17 IF(RI-16) 2,18,2
   18 RI=10
   11 DO 3 K=1,4
      DO 4 L=1,4
      CI=(K-1)*4+L
      IF(CI-5)21,4,22
   22 IF(CI-8)23,23,24
   23 CI=CI-1
      GO TO 21
   24 IF(CI-10) 4,4,25
   25 IF(CI-12)26,26,27
   26 CI=CI-3
      GO TO 21
   27 IF(CI-16)4,28,4
   28 CI=10
   21 B(RI,CI)=A(K,I)*A(L,J)+A(K,J)*A(L,I)
    4 CONTINUE
    3 CONTINUE
    2 CONTINUE
    1 CONTINUE
      RETURN
      END
```

```
      DIMENSION A(4,4),AA(4,4),BB(10,10),D(10,10),C(10,10),B(10,10),
     #YD(4,4,4),WT(4,1),WD(4,4),WW(4,4),WC(4,1),WI(4,1),CC(4,10,10),
     #TT(4,4),TI(4,4,4),TD(4,4),TA(4,4),WB(4,1),WA(4,1),YR(4,1),YE(4,1),
     #R(10,10),H(10,10),F(10,10),G(4,10,10),HH(10,10),V(10,1),W(10,1),
     #Q(10,10),P(10,10),PP(10,10),QQ(4,10,1),RR(10,1),S(10,1),X(10,10),
     #DD(4,4,4),YT(4,4,1),HK(10,1),DI(4,4),AT(4,4),T(4,4),YI(4),
     #CI(4,10,10),HHT(4,10,1),VVT(4,10,1),YA(10),YB(10),HT(4,10,1),
     #VT(4,10,1),YY(4)
      READ(5,10) ((AA(I,J),J=1,4),I=1,4)

      READ(5,52) (YY(J),((CC(J,II,JJ),JJ=1,10),II=1,10),J=1,4)
      READ(5,70) ((HT(I,J,1),J=1,10),(VT(I,L,1),L=1,10),I=1,4)
      READ (5,14) (((DD(M,I,J),J=1,4),I=1,4),M=1,4)
      READ(5,16) ((YT(M,I,1),I=1,4),M=1,4)
C*******************************************************************
C     PRINT OUT THE INITIAL VALUE FOR D MATRIX, THE ESTIMATED MEAN VECT
C     THE C MATRIX FOR 4 GROUPS RESPECTIVELY, AND THE INITIAL VAR/COV M
      WRITE(6,135)

      WRITE(6,114)
      DO 115 II=1,4
  115 WRITE(6,112)((DD(I,II,JJ),JJ=1,4),I=1,4)
      WRITE(6,116)
      DO 118 II=1,4
  118 WRITE(6,124) (YT(I,II,1),I=1,4)
      DO 125 I=1,4
      WRITE(6,126)I
      DO 125 II=1,10
      WRITE(6,777) (CC(I,II,JJ),JJ=1,10)
  125 CONTINUE
      WRITE(6,123)
      DO 122 II=1,4
  122 WRITE(6,111)(AA(II,JJ),JJ=1,4)
      WRITE(6,127)
      DO 130 J=1,10
  130 WRITE(6,62)(HT(I,J,1),I=1,4)
      WRITE(6,131)
      DO 132 J=1,10
  132 WRITE(6,62)(VT(I,J,1),I=1,4)
      DO 1000 N=1,7
      IF (N-1) 11,11,99
C*******************************************************************
C**   STEP 1.
C     SIG(T)=D(T)*SIG(DT)*DT
C     T= SIG(T)
   99 DO 80 I=1,4
      YK=YY(I)
      DO 81 II=1,4
      DO 81 JJ=1,4
      YR(II,1)=YT(I,II,1)
      DI(II,JJ)=DD(I,II,JJ)
   81 CONTINUE
      CALL GMPRD(DI,AA,AT,4,4,4)
      CALL GMPRD(AT,DI,T,4,4,4)
      CALL DMTSQ (T,1,4,0.0,DET,TEST,4)
C*******************************************************************
C**   STEP 2.
C     W(UT)=N(T)* INV OF SIG(T)
C     TT=W(UT)
```

```
      CALL SMPY(T,YK,TT,4,4,0)
C*********************************************************************
C**   STEP 3.
C     YO=D(T)*W(UT)*U(T)
C     W(U)=SUM OF DT*W(UT)*DT
C     WW=W(U)
      CALL GMPRD (DI,TT,TA,4,4,4)
      CALL GMPRD(TA,DI,TD,4,4,4)
      CALL GMPRD(TA,YR,YE,4,4,1)
      DO 82 II=1,4
      DO 82 JJ=1,4
      YO(I,II,1)=YE(II,1)
      TI(I,II,JJ)=TD(II,JJ)
   82 CONTINUE
   80 CONTINUE
C*********************************************************************
C**   STEP 4.
C     WW= INV OF WU
C     WI= SUM OF U(T)*W(UT)*U(T)
C     U=WW*WI
C     WT=U
      DO 84 II=1,4
      DO 84 JJ=1,4
      WW(II,JJ)=0
      WI(II,1)=0
   84 CONTINUE
      DO 85 I=1,4
      DO 85 II=1,4
      WI(II,1)=WI(II,1)+YO(I,II,1)
      DO 85 JJ=1,4
      WW(II,JJ)=WW(II,JJ)+TI(I,II,JJ)
   85 CONTINUE
      CALL DMTSW (WW,1,4,0,0,DET,TEST,4)
      CALL GMPRD (WW,WI,WT,4,4,1)
      WRITE(6,7) N-1
      WRITE(6,59)
      DO 78 I=1,4
   78 WRITE(6,77) WT(I,1)
C*********************************************************************
C**   STEP 5.
C     U(T)=D(T)*U
C     WA=U(T)
      DO 90 I=1,4
      DO 91 II=1,4
      DO 91 JJ=1,4
      YR(II,1)=YT(I,II,1)
      DI(II,JJ)=DD(I,II,JJ)
   91 CONTINUE
C*********************************************************************
C**   STEP 6.
C     H(T)= ( E(UT)-UT) * TRAN (E(UT)-UT)
C     WD=H(T)
      CALL GMPRD (DI,WT,WA,4,4,1)
      CALL MSUB(YR,WA,WB,4,1,0,0)
      CALL MTRA (WB,WC,4,1,0)
      CALL GMPRD (WB,WC,WD,4,1,4)
      DO 93 M=1,4
   93 HK(M,1)=WD(M,1)
      DO 94 M=2,4
   94 HK(M+3,1)=WD(2,M)
      DO 95 M=3,4
```

```
   95 HK(5+M,1)=WD(3,M)
      HK(10,1)=WD(4,4)
      DO 96 II=1,10
   96 HT(I,II,1)=HK(II,1)
   90 CONTINUE
      WRITE(6,/) N-1
      WRITE(6,79)
      DO 61 II=1,10
   61 WRITE(6,62)  (HT(I,II,1),I=1,4)
C*********************************************************************
C**   STEP 7.
C     W(SIGT)=NT*( INV OF (CT*UU*CT))
C     G=W(SIGT)
      WRITE (6,71)N
   11 CALL UMAT(AA,BB)
      DO 51 I=1,4
      Y=YY(I)
      DO 2 II=1,10
      DO 2 JJ=1,10
    2 C(II,JJ)=CC(I,II,JJ)
   13 CALL GMPRD (C,BB,D,10,10,10)
      CALL GMPRD (D,C,H,10,10,10)
      TEST=0.5*1./(10**10)
      CALL DMTSQ(H,1,10,0,0,DET,TEST,10)
      CALL SMPY(H,Y,F,10,10,1)
      DO 55 J=1,10
      DO 55 K=1,10
   55 G(I,J,K)=F(J,K)
   51 CONTINUE
C*********************************************************************
C**   STEP 8.
C     W= SUM OF W(SIGT)
C     Q=W
      DO 152 I=1,10
      DO 152 J=1,10
  152 Q(I,J)=0
      DO 60 I=1,4
      DO 60 J=1,10
      DO 60 K=1,10
   60 Q(J,K)=Q(J,K)+G(I,J,K)
C*********************************************************************
C**   STEP 9.
C     Q= INV OF W(SIG)
C     W=Q(J,K)
C     RR= SUM OF ( C(T)*W(SIG T)*(SIG T + HT))
C     S=Q*RR
C     S=VAR AND COV MATRIX
      DO 100 II=1,4
      DO 7 JJ=1,10
      DO 7 LL=1,10
    7 C(JJ,LL)=CC(II,JJ,LL)
      DO 8 JJ=1,10
      HH(JJ,1)=HT(II,JJ,1)
      V(JJ,1)=VT(II,JJ,1)
    8 CONTINUE
      DO 72 I=1,10
      J=1
   72 W(I,J)=HH(I,J)+V(I,J)
      DO 20 I=1,10
      DO 20 J=1,10
   20 X(I,J)=G(II,I,J)
```

```
      CALL GMPRD(C,X,P,10,10,10)
      CALL GMPRD (P,W,PP,10,10,1)
      DO 73 K=1,10
      L=1
   73 QQ(II,K,L)=PP(K,L)
  100 CONTINUE
      DO 156 I=1,10
  156 RR(I,1)=0
      DO 75 I=1,4
      DO 75 J=1,10
      RR(J,1)=RR(J,1)+QQ(I,J,1)
   75 CONTINUE
      TEST=0.5*1./(10**10)
      CALL DMTSQ(Q,1,10,0,0,DET,TEST,10)
      CALL GMPRD(Q,RR,S,10,10,1)
      WRITE(6,/) N
      WRITE(6,53)
      DO 76 I=1,10
   76 WRITE(6,77) S(I,1)
      DO 3 I=1,4
      AA(1,1)=S(I,1)
    3 AA(I,1)=S(I,1)
      DO 4 I=2,4
      AA(2,1)=S(I+3,1)
    4 AA(I,2)=S(I+3,1)
      AA(3,3)=S(8,1)
      AA(3,4)=S(9,1)
      AA(4,3)=S(9,1)
      AA(4,4)=S(10,1)
 1000 CONTINUE
C     ****************************************************************
   10 FORMAT(16F5.3)
   14 FORMAT(36I2/28I2)
   16 FORMAT(16F5.3)
   52 FORMAT(I5,37I2,/5X,37I2,/5X,26I2)
   53 FORMAT('    VAR/COV VECTOR IS')
   59 FORMAT('    MEAN VECTOR IS')
   62 FORMAT(4F16.6)
   70 FORMAT(8F9.4,/8F9.4,/4F9.4)
   71 FORMAT(/,'  THE  ',I2,'  ITERATION')
   77 FORMAT(F12.6)
   79 FORMAT(10X,'H(1)',12X,'H(2)',12X,'H(3)',12X,'H(4)')
  111 FORMAT(4F12.6)
  112 FORMAT(4(4I3,6X))
  114 FORMAT('    THE D(T) MATRICES ARE:'/5X,'D(1)',14X,'D(2)',14X,
     #'D(3)',14X,'D(4)')
  116 FORMAT('    THE ESTIMATED MEAN VECTOR FOR U(T) IS :'/10X,'U(1)',
     @ 12X,'U(2)',11X,'U(3)',11X,'U(4)')
  117 FORMAT('    D(',I2,') MATRIX IS')
  123 FORMAT('    THE INITIAL VAR/COV MATRIX IS')
  124 FORMAT(4F15.3)
  126 FORMAT('    C(',I2,') MATRIX IS ')
  127 FORMAT('    THE INITIAL H(T) VECTORS ARE :'/10X,'H(1)',12X,'H(2)',
     #12X,'H(3)',12X,'H(4)')
  131 FORMAT('    THE SIG(T) VECTORS ARE :'/10X,'S(1)',12X,'S(2)',12X,
     #'S(3)',12X,'S(4)')
  135 FORMAT('    THE PROGRAM OUTPUT IS FOLLOWING:'/)
  777 FORMAT(10I3)
      STOP
      END
```

Appendix 3

Program Output

This output included the initial estimators of mean vector, var/cov matrix, C_t, D_t, H_t matrices. Records of estimates contained 6 iterations.

```
THE PROGRAM OUTPUT IS FOLLOWING:

THE D(I) MATRICES ARE:
   D(1)               D(2)               D(3)               D(4)
1  0  0  0         1  0  0  0         1  0  0  0         0  0  0  0
0  1  0  0         0  1  0  0         0  1  0  0         0  0  0  0
0  0  1  0         0  0  1  0         0  0  0  0         0  0  1  0
0  0  0  1         0  0  0  0         0  0  0  0         0  0  0  0
THE ESTIMATED MEAN VECTOR FOR U(T) IS :
       U(1)               U(2)               U(3)               U(4)
       4.990              4.900              5.120              0.000
       3.430              3.350              3.630              0.000
       1.450              1.370              0.000              1.520
       0.240              0.000              0.000              0.000
C( 1) MATRIX IS
1  0  0  0  0  0  0  0  0  0
0  1  0  0  0  0  0  0  0  0
0  0  1  0  0  0  0  0  0  0
0  0  0  1  0  0  0  0  0  0
0  0  0  0  1  0  0  0  0  0
0  0  0  0  0  1  0  0  0  0
0  0  0  0  0  0  1  0  0  0
0  0  0  0  0  0  0  1  0  0
0  0  0  0  0  0  0  0  1  0
0  0  0  0  0  0  0  0  0  1
C( 2) MATRIX IS
1  0  0  0  0  0  0  0  0  0
0  1  0  0  0  0  0  0  0  0
0  0  1  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0
0  0  0  0  1  0  0  0  0  0
0  0  0  0  0  1  0  0  0  0
0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  1  0  0
0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0
C( 3) MATRIX IS
1  0  0  0  0  0  0  0  0  0
0  1  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0
0  0  0  0  1  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0
C( 4) MATRIX IS
0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  1  0  0
0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0
```

THE INITIAL VAR/COV MATRIX IS

0.158000	0.140000	0.015000	0.010000
0.140000	0.185000	0.013000	0.008000
0.015000	0.013000	0.026000	0.004000
0.010000	0.008000	0.004000	0.012000

THE INITIAL H(I) VECTORS ARE :

H(1)	H(2)	H(3)	H(4)
0.000000	0.008100	0.016900	0.000000
0.000000	0.007200	0.026000	0.000000
0.000000	0.007200	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000
0.000000	0.006400	0.040000	0.000000
0.000000	0.006400	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000
0.000000	0.006400	0.000000	0.004900
0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000

THE SIG(T) VECTORS ARE :

S(1)	S(2)	S(3)	S(4)
0.153000	0.070000	0.018000	0.000000
0.136000	0.021000	0.026000	0.000000
0.014000	0.027000	0.000000	0.000000
0.009000	0.000000	0.000000	0.000000
0.179000	0.053000	0.096000	0.000000
0.013000	-0.009000	0.000000	0.000000
0.008000	0.000000	0.000000	0.000000
0.026000	0.024000	0.000000	0.018000
0.004000	0.000000	0.000000	0.000000
0.011000	0.000000	0.000000	0.000000

1,

VAR/COV VECTOR IS

0.123511
0.105525
0.016572
0.007877
0.151909
0.008775
0.005907
0.026335
0.004262
0.010990

1,

MEAN VECTOR IS

4.996597
3.445231
1.444725
0.239568

1,

H(1)	H(2)	H(3)	H(4)
0.000044	0.009331	0.015228	0.000000
0.000100	0.009199	0.022801	0.000000
-0.000035	0.007218	0.000000	0.000000
-0.000003	0.000000	0.000000	0.000000
0.000232	0.009069	0.034139	0.000000
-0.000080	0.007116	0.000000	0.000000
-0.000006	0.000000	0.000000	0.000000
0.000028	0.005584	0.000000	0.005666
0.000002	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000

THE 2 ITERATION

2,

VAR/COV VECTOR IS

0.123355
0.105576
0.015783
0.007908
0.151627
0.008254
0.005977
0.026130
0.004109
0.010966

2,

MEAN VECTOR IS

4.996391
3.445689
1.444645
0.239602

2,

H(1)	H(2)	H(3)	H(4)
0.000041	0.009291	0.015279	0.000000
0.000096	0.009166	0.022857	0.000000
-0.000034	0.007195	0.000000	0.000000
-0.000003	0.000000	0.000000	0.000000
0.000228	0.009042	0.034192	0.000000
-0.000081	0.007098	0.000000	0.000000
-0.000006	0.000000	0.000000	0.000000
0.000029	0.005572	0.000000	0.005678
0.000002	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000

THE 3 ITERATION

3,

VAR/COV VECTOR IS

0.123661
0.105580
0.015683
0.007666
0.151829
0.008321
0.005957
0.026154
0.004119
0.010963

3,

MEAN VECTOR IS

4.996417
3.445108
1.444655
0.239602

3,

H(1)	H(2)	H(3)	H(4)
0.000041	0.009296	0.015273	0.000000
0.000097	0.009170	0.022849	0.000000
-0.000034	0.007198	0.000000	0.000000
-0.000003	0.000000	0.000000	0.000000
0.000228	0.009046	0.034185	0.000000
-0.000081	0.007100	0.000000	0.000000
-0.000006	0.000000	0.000000	0.000000
0.000029	0.005573	0.000000	0.005677
0.000002	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000

THE 4 ITERATION

4,
VAR/COV VECTOR IS
0.123860
0.105579
0.015870
0.007878
0.151829
0.008313
0.005968
0.026151
0.004115
0.010963

4,
MEAN VECTOR IS
4.996414
3.445106
1.444653
0.239603

4,

H(1)	H(2)	H(3)	H(4)
0.000041	0.009296	0.015274	0.000000
0.000097	0.009169	0.022850	0.000000
-0.000034	0.007198	0.000000	0.000000
-0.000003	0.000000	0.000000	0.000000
0.000228	0.009045	0.034186	0.000000
-0.000081	0.007100	0.000000	0.000000
-0.000006	0.000000	0.000000	0.000000
0.000029	0.005573	0.000000	0.005677
0.000002	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000

THE 5 ITERATION

5,
VAR/COV VECTOR IS
0.123861
0.105579
0.015872
0.007874
0.151829
0.008314
0.005965
0.026152
0.004116
0.010[illegible]

5,
MEAN VECTOR IS
4.996414
3.445106
1.444654
0.239603

5,

H(1)	H(2)	H(3)	H(4)
0.000041	0.009296	0.015273	0.000000
0.000097	0.009170	0.022850	0.000000
-0.000034	0.007198	0.000000	0.000000
-0.000003	0.000000	0.000000	0.000000
0.000228	0.009045	0.034186	0.000000
-0.000081	0.007100	0.000000	0.000000
-0.000006	0.000000	0.000000	0.000000
0.000029	0.005573	0.000000	0.005677
0.000002	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000

THE 6 ITERATION

6,
VAR/COV VECTOR IS
0.123861
0.105579
0.015872
0.007875
0.151829
0.008314
0.005966
0.026152
0.004115
0.010963

6,
MEAN VECTOR IS
4.996414
3.445106
1.444654
0.239603

6,

H(1)	H(2)	H(3)	H(4)
0.000041	0.009296	0.015273	0.000000
0.000097	0.009170	0.022850	0.000000
-0.000034	0.007198	0.000000	0.000000
-0.000003	0.000000	0.000000	0.000000
0.000228	0.009045	0.034186	0.000000
-0.000081	0.007100	0.000000	0.000000
-0.000006	0.000000	0.000000	0.000000
0.000029	0.005573	0.000000	0.005677
0.000002	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000

VITA

Chi-Ping Lu

Candidate for the Degree of

Master of Science

Report: The Program of Incomplete Data in Multivariate Normal Distribution

Major Field: Applied Statistics

Biographical Information:

Personal Data: Born at Pei-Ping, China, May 5, 1949, daughter of Hsueh-Lee and Win-Jin Lu.

Education: Graduated from Provincial Taipei Second Girl High School, Taiwan in 1967; received a Bachelor of Science degree from National Chen-Chi University, majoring in Statistics, Taiwan in 1971; completed requirements for the Master of Science degree in Applied Statistics at Utah State University in 1975.

Professional Experience: Sept. 1971 to Feb. 1973, worked in Accounting Dept. of Chinese Petroleum Corporation in Taiwan, Sept. 1973 to present, statistical consultant for research workers on campus at Utah State University.