A

THE GIBBS SAMPLER APPLIED TO MISSING DATA WITH CATEGORICAL, CONTINUOUS AND MIXED DATA TYPES

by

SARAH BOSLAUGH

A dissertation submitted to the Graduate Faculty in Educational Psychology in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York

1996

Approval Page

This manuscript has been read and accepted for the Graduate Faculty in Educational Psychology in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

4/9/96
Date

Chair of Examining Committee

4/9/96
Date

Executive Officer

David Rindskopf, Ph.D

Carol Kehr Tittle, Ph.D

Roger Millsap, Ph.D

Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK

Abstract

THE GIBBS SAMPLER APPLIED TO MISSING DATA
WITH CATEGORICAL, CONTINUOUS AND MIXED DATA TYPES
by
Sarah Boslaugh
Advisor: Professor Alan Gross

In the present research we have investigated the problem of estimating multiple correlations for regression models containing a mix of categorical, continuous and interaction terms given a data set containing missing values. We consider the case where the model has a single binary predictor variable, a single continuous predictor variable, and a cross-product term. A Bayesian approach is used to obtain an interval estimate of the multiple correlation of Y on the predictors (ρ^2) using a Gibbs sampling procedure. Using 5,000 samples from the posterior distribution of ρ^2, we empirically construct .90 highest-density regions (HDR's) for ρ^2. To demonstrate the estimation procedure, 32 data

samples were used; 18 with data missing completely at random (MCAR) and 18 with data missing at random (MAR). Within each set of 18, three sample sizes (30, 50 or 100), three population values for ρ^2 (.10, .25 and .50) and two probabilities of missing data (.271 and .657) were used. In the MCAR case, 17 of the 18 HDR's contained the population ρ^2, while in the MAR cases, 16 of the 18 HDR's contained the population ρ^2. As expected, smaller sample sizes and more missing data produced wider HDR's, and MAR data produced slightly wider HDR's than MCAR data.

Table of Contents

List of Tables

Introduction

The problem of missing data occurs frequently in social science research. In most cases specialized procedures are required to avoid introducing bias into statistical analyses of missing data. While procedures exist to handle some types of missing data, this dissertation will develop a procedure for a type of problem which has not yet been addressed, that of a regression model with missing data on both continuous and categorical variables and which also contains interaction terms, i.e., a nonadditive model. More specifically, it will consider the problem of predicting a continuous variable from one continuous and one categorical predictor, with one interaction term. The goal of the proposed analysis is to obtain an interval estimate of the multiple correlation of Y with the predictor variables. Standard procedures exist to estimate these parameters in the case of complete data, but they are not applicable when data are missing. While various ad hoc procedures exist to make these estimates in the missing data case, the most common procedures produce unsatisfactory results (i.e., they produce biased estimates) in even fairly simple missing data situations. Although a statistical model (the General Location Model) (Little & Rubin, ch. 10) has

been used in problems where there are missing data on both continuous and categorical variables, this model does not allow for the inclusion of interaction terms in the regression equation.

In the present study, a Bayesian approach is used to obtain an interval estimate of the multiple correlation. We first introduce a statistical model for the continuous dependent variable, the continuous predictor, the categorical predictor, and the cross-product term. The parameters of this model are then estimated using an iterative Monte Carlo procedure known as Gibbs sampling. This procedure yields an empirical approximation to the posterior distribution of ρ^2. An interval estimate of ρ^2 is thus obtained in terms of an approximate .90 highest density region (HDR).

The next section provides a literature survey and an explanation of the modern statistical theory of missing data analysis. The methods section describes the proposed model and the Gibbs sampling procedure.

Literature Survey

Types of Missing Data

Missing data may be classified as either missing completely at random (MCAR), missing at random (MAR), or nonignorable (Little & Rubin, 1987). Missing data are MCAR if the probability of a particular variable being missing is independent of both the potential values of the missing data and of the observed data. Missing data are MAR if, given the value of the observed data, the conditional probability of a particular variable being missing is independent of the potential value of that variable. Missing data are nonignorable if, given the observed data, the conditional probability of a particular variable being missing is related to the potential value of that variable.

Missing data patterns may be classified as monotonic or nonmonotonic. Monotonic missing data allow cases to be ranked in order of their completeness, as in the data set in Table 1 below (M = missing):

Table 1

Monotonic Missing Data Pattern

X_1	X_2	X_3	Y
1	1	12	45
1	1	11	50
1	2	8	48
1	2	11	M
2	1	13	M
2	1	M	M
2	2	M	M
2	M	M	M
2	M	M	M

This data set has the monotonic pattern because we know if a variable is missing X_3, it will also be missing Y, and if it is missing X_2, it will also be missing X_3 and Y. The data set in Table 2 below is nonmonotonic because the missing data does not fall into this simple pattern:

Table 2

Nonmonotonic Missing Data Pattern

X_1	X_2	X_3	Y
1	1	12	45
1	M	11	50
1	2	8	48
1	M	M	34
2	1	13	M
2	M	15	M
M	2	9	M
M	1	M	46
2	1	M	M

Ad Hoc Procedures for Analyzing Missing Data

The ad hoc methods for handling missing data have a long history (Little & Rubin, 1987, ch. 3) and require the least computation, but are not satisfactory except in particular limited cases. As the name implies, they are not based on a general theory of statistical estimation but rather make use of what is at hand in a particular situation. The validity of the results they produce depend on the process which caused the data to be missing and what parameters are being estimated, and it is often not intuitively obvious whether results produced through these methods are

valid or not.

There are three basic ad hoc methods: listwise deletion, casewise deletion, and various fill-in procedures. The simplest method is listwise deletion or complete case analysis, in which only cases which have no missing data are analyzed; cases missing even one variable are dropped from all analyses. Statistical analysis packages typically use listwise deletion as the default for regression and factor analysis procedures (Dixon, 1990; SAS Institute, 1990; SPSS, 1988). Since listwise deletion constitutes using the complete cases as if they were a random sample from the original sample, it is valid only if data are MCAR. Listwise deletion can also result in substantial loss of data, since a case missing only one variable is essentially deleted from the data set, i.e. it is not used in any analyses.

Casewise deletion, or available case analysis, uses all cases which have the variables needed for a particular procedure, which may result in different numbers of cases being used in different procedures; for instance, in a correlation matrix, individual pairwise correlations may be based on different numbers of cases. Although this often means that more of the data can be used, it will introduce biases in the estimation of many parameters unless data are MCAR, and can produce statistical anomalies such as a correlation matrix which is not positive definite. Casewise deletion is the default for pairwise correlation and simple data description in many statistical packages (Dixon, 1990; SAS Institute,

1990; SPSS, 1988).

Fill-in procedures constitute a third type of ad hoc method. In these procedures, a summary statistic is used to fill in the missing pieces of data, producing what appears to be a complete data set, which can be then be analyzed as such. The statistic used is usually either the unconditional mean (i.e., the mean as calculated from the available data is substituted for each missing case) or a conditional mean for each missing case, calculated by regressing the missing variable on the observed variables. This method is only possible if there are some cases which are complete in all variables. Fill-in methods will produce valid estimates for some parameters (e.g., the mean of the missing variable) but will always produce invalid results for others (e.g., the variance of the missing variable), even under MCAR. Unconditional mean substitution is available in several regression packages (Dixon, 1990; SAS Institute, 1990; SPSS, 1988), and conditional mean substitution is available in BMDP's program AM (Data Description and Estimation) (Dixon, 1990).

Maximum Likelihood (ML) Methods

Another class of procedures for dealing with missing data use maximum likelihood (ML) estimation methods, i.e. they estimate the parameters as those values which make sample data most likely. ML methods are based on statistical theory and have two major advantages

over ad hoc methods: they produce consistent parameter estimates with MAR data, while the ad hoc produces unbiased estimates only with MCAR data, and, given MAR data and a large sample, ML estimates are known to have desirable properties such as consistency and efficiency (enumerated later).

The procedure for producing MLE's consists of expressing the joint probability (likelihood) of the data, and then choosing the parameter values that maximize that likelihood. In the complete data case, the form of the likelihood functions are known and MLE's may often be calculated in a straightforward manner (Ross, 1976). However, when there are missing data, the functions can become quite complicated and may be impossible to directly maximize. In such a case, an iterative approach such as that of the EM algorithm is necessary to find the MLE's (Dempster, Laird & Rubin, 1977). ML methods are available in BMDP programs AM (Data description and estimation) and 8D (Correlations with missing data); these programs can produce a correlation or covariance matrix using ML estimation, which may be saved and used as inputs for other statistical analyses (Dixon, 1990).

In general, the EM algorithm consists of taking the expected values of the theoretical complete data likelihood over the missing data, given the observed data and current parameter estimates (the E-step). The expected likelihood is then maximized with respect to the unknown parameters (the M-step). Using these maximum likelihood estimates as

new parameter estimates the E-step is repeated and the process is continued until convergence. In the case of variables with a bivariate normal distribution and monotonic missing pattern, the unknown parameters we wish to estimate are μ_x, μ_y, σ_x, σ_y, and σ_{xy}. In this case, the E-step essentially involves taking the expected value of the sufficient statistics (ΣX, ΣY, ΣX^2, ΣY^2, ΣXY) needed to estimate the population parameters. The M-step then involves using those expected values to obtain the MLE's.

ML Estimation (Complete Data Case)

As an example of ML estimation, consider a case where we have complete binomial data and are interested in estimating the population parameter π, i.e. the probability that Y = 1. The likelihood function (L) is known to be:

$$L(\pi) \text{ is proportional to } L(\pi \mid \text{data}) = P(\text{data} \mid \pi) = \pi^{r} (1 - \pi)^{(n-r)} \quad (1)$$

where n is the number of trials, r is the number of trials where Y = 1, and π is the probability in the population that Y = 1.

The goal is to find the value of π which makes the values of r given n most likely, i.e. the value which maximizes $L(\pi \mid \text{data})$. When there are no missing values, the maximum likelihood estimate (MLE) for π is known

to be r/n. This formula can be derived by (a) taking the log of the likelihood function, (b) taking this function's first derivative, and (c) solving for the point where the derivative of the loglikelihood equals 0.

a. $\text{Log } L(\pi) = r \log \pi + (n - r) \log (1 - \pi)$

b. $L'(\pi) = (r/\pi) - (n - r)/(1 - \pi)$

c. $r(1 - \pi) - (n - r)\pi = 0$

$r - r\pi - n\pi + r\pi = 0$

$r - n\pi = 0$

$r = n\pi$

$r/n = \hat{\pi}$

The MLE yields the highest probability for $\underline{r}$ of any π-value, as is also clear from examining neighboring values from a binomial table. For instance, if $\underline{n} = 10$ and $\underline{r} = 4$, then the MLE for π is 4/10 or .4. Values from Table 3 below show that given $\underline{n} = 10$, for the event $\underline{r} = 4$, $\pi = .4$ is the most probable of the π-values, confirming that .4 is the MLE for π for the binomial case with $\underline{n} = 10$, $\underline{r} = 4$.

Table 3

Binomial Probabilities (n = 10, r = 4)

π	P (r = 4 | π)
.37	.2461
.38	.2487
.39	.2503
.40	.2508 **MLE
.41	.2503
.42	.2488
.43	.2462

MLE's have the following properties in the large sample case which make them desirable estimators (Little & Rubin, 1987, ch. 7):

1) Consistency: with increasing n, the MLE converges in probability to the true population parameter value.

2) Asymptotic normality: with increasing n, the sampling distribution of the MLE tends toward normality.

3) Efficiency: MLE's have the smallest sampling variance among the consistent, asymptotically normal estimators.

4) MLE's are a function of the sufficient statistics, if they exist; for instance, in the binomial case, the MLE is a function of r (the number of times Y = 1) rather than of the specific values of Y for each case.

5) An estimate of the asymptotic sampling variance of the

parameter estimate can be estimated by taking the reciprocal of the second derivative of the log likelihood with respect to the unknown parameter (Little & Rubin, 1987, ch. 5).

Maximum Likelihood (Theory for Missing Data)

Consider the extension of ML estimation to the missing data case. Suppose now that we have a data set with five cases, but Y is missing on two of them (M = missing data, R = response variable):

Table 4

ML Example Using a Response Variable

Y	R
1	1
1	1
0	1
M	0
M	0

The observed data now consists of the 3 observed Y-values (1, 1, 0) plus a vector of R-values which denote if a case is missing (1 = complete, 0 = missing, so that $\underline{r}' = [1\ 1\ 1\ 0\ 0]$). The likelihood function in the missing data case must include the additional parameters $\phi' = [\phi_1, \phi_0]$ describing

the conditional probability that R = 1 given Y, i.e. $\phi_0 = P(R = 1 | Y = 0)$ and $\phi_1 = P(R = 1 | Y = 1)$. The probability for the first three cases is a product of the probability of Y given π multiplied by the probability of R given Y and ϕ. For cases 4 and 5, we need only express the marginal probability that R = 0. The loglikelihood function is:

$$\begin{aligned} \text{Log } L(\pi, \phi) &= \log P(Y_1 = 1 | \pi)\,(\phi_1) \\ &+ \log P(Y_2 = 1 | \pi)\,(\phi_1) \\ &+ \log P(Y_3 = 0 | \pi)\,(\phi_0) \\ &+ \log P\,(R_4 = 0 | \pi, \underline{\phi}) \\ &+ \log P\,(R_5 = 0 | \pi, \underline{\phi}) \end{aligned} \qquad (2)$$

$$\begin{aligned} &= 2 \log (\pi)\,(\phi_1) + \log (1 - \pi)\,(\phi_o) \\ &+ 2 \log [(1 - \phi_0)\,(1 - \pi) + (1 - \phi_1)\,(\pi)] \end{aligned}$$

$$\begin{aligned} &= 2 \log (\pi) + \log (1 - \pi) + 2 \log (\phi_1) + \log (\phi_0) \\ &+ 2 \log [(1 - \phi_0)\,(1 - \pi) + (1 - \phi_1)\,(\pi)]. \end{aligned}$$

If the data are not MAR, i.e. ϕ_0 is not equal to ϕ_1, it becomes complicated to calculates the MLE, because we must maximize the likelihood with respect to both π and $\underline{\phi}$. Even when ϕ_0 and ϕ_1 are known, the likelihood is not standard. If we can assume the data are MAR, i.e. that $\phi_0 = \phi_1$, the loglikelihood function can be written so that it is a sum of two terms, one

depending on π and one depending on ϕ: Lik $(\pi, \phi) = f(\pi) + g(\phi)$. In this case, the MLE can be obtained by maximizing the term relating to π only, which is the same function to be maximized in the complete case, i.e. $\pi^{r}(1 - \pi)^{(n-r)}$, where $\underline{n}$ is now the number of complete cases. Thus, the MAR assumption greatly simplifies ML estimation since one need only consider the likelihood in terms of the observed data, and the R variable can be ignored. Given MAR and monotonic missing data patterns, non-iterative procedures can be used to compute the MLE. However, given MAR but a nonmonotonic pattern, the estimates may still be difficult to obtain and require iterative methods. We will consider the monotonic and nonmonotonic cases next.

ML Applications to Monotonic Missing Data Patterns

ML methods have been developed for dealing with missing categorical data (Little & Rubin, ch. 9), missing continuous data (Little & Rubin, 1987, ch. 8), and the mixed case in which both continuous and categorical variables are missing (Little & Rubin, 1987, ch. 10). The analyses are relatively simple for monotonic cases, but, as noted, nonmonotonic cases require iterative calculation. Before discussing the specific case treated in this dissertation, we will discuss some simpler cases involving only categorical, only continuous, and categorical and continuous data without a cross-product term.

Monotonic categorical data.

First, consider monotonic MAR categorical data. Take for example the following 2 x 2 table with complete information on Y but some cases missing on X (i.e., a data set with one supplemental margin). The goal is to estimate the cell or joint probabilities:

$$\pi_{11} = P(Y=1, X=1)$$

$$\pi_{12} = P(Y=1, X=2)$$

$$\pi_{21} = P(Y=2, X=1)$$

$$\pi_{22} = P(Y=2, X=2)$$

Table 5

Monotonic Categorical Data (Observed Data)

		X		
		1	2	
Y	1	20	40	60
	2	30	20	50
		50	60	

Y	1	20	20
	2	30	30
		50	

Since the likelihood can be factored into the product of the marginal distribution of Y and the conditional distribution of X given Y, each of these parameters can be separately estimated, and the joint probability estimated from them. So, the joint probability of two dichotomous variables can be estimated by multiplying the product of the conditional and the marginal: for instance, $P(X = 1, Y = 1) = P(X = 1 \mid Y = 1)\, P(Y = 1)$. In this particular example, $P(X = 1 \mid Y = 1)$ is estimated to be 20/60 and $P(Y = 1)$ is estimated to be 80/160, so the joint probability $P(X = 1, Y = 1)$ is estimated as (20/60) (80/160) or 1/6. The estimates are:

$$\hat{\pi}_{11} = \frac{20}{60} * \frac{80}{160} = \frac{1}{6}$$

$$\hat{\pi}_{12} = \frac{40}{60} * \frac{80}{160} = \frac{2}{6} = \frac{1}{3}$$

$$\hat{\pi}_{21} = \frac{30}{50} * \frac{80}{160} = \frac{3}{10}$$

$$\hat{\pi}_{22} = \frac{20}{50} * \frac{80}{160} = \frac{2}{10}$$

These estimates vary from those calculated using complete data only:

$$\hat{\pi}_{11} = \frac{20}{110} = \frac{2}{11}$$

$$\hat{\pi}_{12} = \frac{4}{11}$$

$$\hat{\pi}_{21} = \frac{3}{11}$$

$$\hat{\pi}_{22} = \frac{2}{11}$$

Using just complete data in this case would yield inconsistent estimates since X is not MCAR. This is clear because the probability that Y = 1 is not independent of the response variable for X (R_x), i.e. the estimate for

$P(Y = 1 \mid R_x = 1) = 6/11$, while the estimate for $P(Y = 1 \mid R_x = 0) = 2/5$.

Applications to Monotonic Missing Continuous Data

Next, consider the case of a monotonic missing pattern with continuous data. To take a simple case of two bivariate normal variables, one complete (X) and the other with missing data (Y), our goal is to estimate the following population parameters: μ_x, σ_{xx}, μ_y, σ_{yy}, and σ_{xy}. With complete data, we would estimate these using the traditional sample statistics, e.g. the MLE for the mean of X is the sample mean of X. With monotonic missing data, assuming that Y is MAR, we could estimate these parameters in a straightforward manner since the incomplete likelihood can be factored into the likelihood based on X (conditioned on μ_x and σ_{xx}) and the conditional likelihood of Y given X (conditioned on β_0, β_1 and $\sigma_{yy|x}$). Simple transformations of these estimates yields MLE's for the mean and variance of Y, and the correlation of X and Y. For example, the mean of Y is estimated as

$$\hat{\beta}_0 + \hat{\beta}_1 \hat{\mu}_x$$

where $\hat{\beta}_0$ and $\hat{\beta}_1$ are the least squares estimates computed from cases complete on X and Y (Little & Rubin, 1987, ch. 6). To demonstrate ML estimation with a monotonic missing data set, consider the data in Table 6 below. We show the MLE's for various parameters together with

estimates obtained using the ad hoc methods of listwise deletion, pairwise deletion, and unconditional mean substitution. These data were generated from a bivariate normal density using an IMSL subroutine (IMSL, 1987). The parameter values for this population were specified as follows: $\mu_X = 10$, $\mu_Y = 15$, $\sigma^2_X = 1$, $\sigma^2_Y = 2$ ($\sigma_y = 1.41$), $\sigma_{xy} = 1$ ($\rho_{xy} = .7072$), $\beta_0 = 5$, $\beta_1 = 1$.

Table 6

Data Set 1: Complete Data

case	X	Y	case	X	Y
1	11.2	16.7	16	9.0	12.2
2	10.4	15.8	17	10.5	17.0
3	9.8	16.8	18	10.8	17.7
4	10.7	15.2	19	10.1	14.9
5	10.3	14.4	20	10.0	14.2
6	8.8	13.1	21	11.4	16.5
7	9.8	14.4	22	9.8	14.3
8	9.9	15.5	23	10.2	15.3
9	8.1	12.7	24	11.1	17.2
10	10.3	15.3	25	9.9	16.7
11	10.4	14.6	26	10.8	15.6
12	10.5	16.8	27	10.3	15.5
13	9.7	14.6	28	11.1	15.5
14	9.3	14.2	29	10.4	15.0
15	10.4	15.0	30	11.4	16.2

Complete sample parameter estimates, from the SPSS-X regression program (SPSS, 1988):

$\hat{\mu}_X = 10.213$

$\hat{\sigma}_X = .752$

$\hat{\mu}_Y = 15.297$

$\hat{\sigma}_Y = 1.326$

$\hat{\rho}_{XY} = .749$

$\hat{\beta}_0 = 1.825$

$\hat{\beta}_1 = 1.319$

This data set can be transformed into a monotonic MAR data set missing on Y only, by applying the following rule: if X is greater than or equal to 10.5, Y is missing. This rule produces the following data set:

Table 7

Data Set 2: Monotonic MAR Data

case	X	Y	case	X	Y
1	11.2	M	16	9.0	12.2
2	10.4	15.8	17	10.5	M
3	9.8	16.8	18	10.8	M
4	10.7	M	19	10.1	14.9
5	10.3	14.4	20	10.0	14.2
6	8.8	13.2	21	11.4	M
7	9.8	14.4	22	9.8	14.3
8	9.9	15.5	23	10.2	15.2
9	8.1	12.7	24	11.1	M
10	10.3	15.3	25	9.9	16.7
11	10.4	14.6	26	10.8	M
12	10.5	M	27	10.3	15.5
13	9.7	14.6	28	11.1	M
14	9.3	14.2	29	10.4	15.5
15	10.4	15.0	30	11.4	M

Now consider the results of the different forms of parameter estimation. Table 8 below shows the MLE's for various parameters (produced using the EM algorithm, as discussed below) together with those estimated using the ad hoc methods of listwise deletion, pairwise deletion, and unconditional mean substitution.

Table 8

Parameter Estimates for Monotonic MAR data

	pairwise deletion	listwise deletion	mean subst.	ML	parameter values
$\hat{\mu}_x$	10.213	9.845	10.213	10.213	10.000
$\hat{\sigma}_x$	.752	.618	.752	.738	1.000
$\hat{\mu}_y$	14.725	14.725	14.475	15.202	15.000
$\hat{\sigma}_y$	1.158	1.158	.938	1.258	1.414
$\hat{\rho}_{xy}$	.679	.679	.451	.763	.707
$\hat{\beta}_0$	4.052	2.194	8.984	1.911	5.000
$\hat{\beta}_1$	1.045	1.273	.562	1.301	1.000

Comparing the various estimates with the parameter values, we can see that mean substitution is the worst method since it underestimates the

variance of Y and the correlation of X and Y; listwise deletion is also ineffective since it underestimates the mean and variance of both X and Y because the rule for making the data monotonic missing removed the larger values of both X and Y. Note also that the estimate of the standard deviation of X is lower using the ML method than when using mean substitution, due to division by n in the former case, and n -1 in the latter. With this relatively small sample size and low proportion of missing data, there are no great differences between the casewise deletion estimates and the MLE's.

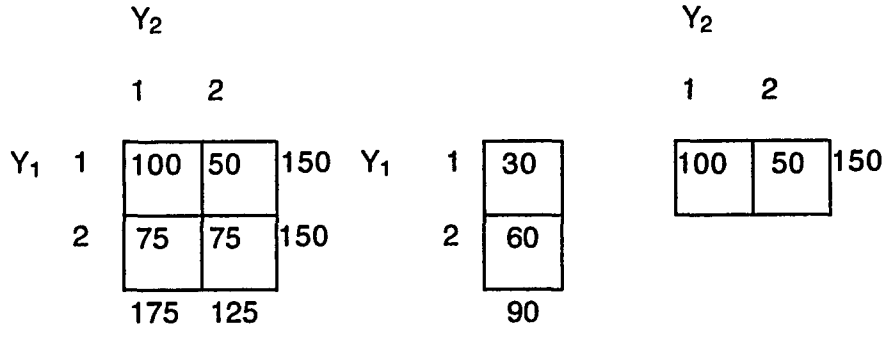
ML Applications to Nonmonotonic Missing Data Patterns

Nonmonotonic missing data (categorical variables).

It is also possible to apply ML estimation to nonmonotonic missing categorical data (taken from Little & Rubin, pp. 183-185); for instance, consider the 2 x 2 table below. This table has some complete data, some data missing on Y_1 and some missing on Y_2, so it has two supplemental margins.

Table 9

Nonmonotonic Categorical Data



The goal, as in the monotonic categorical case above, is to estimate the cell probabilities; however, because this example is nonmonotonic, it requires use of the EM algorithm to maximize the likelihood. Initial estimates of the cell probabilities are taken from the complete data; these are then used to classify the missing portion of the incomplete data. For instance, we can estimate the Y_2 values for the 30 cases in which $Y_1 = 1$ but Y_2 is missing by using the probabilities from the complete data that $P(Y_2 = 1 | Y_1 = 1) = 100/150 = 2/3$, and $P(Y_2 = 2 | Y_1 = 1) = 50/150 = 1/3$. Thus, 2/3 or 20 of these cases will be assigned the condition $(Y_1 = 1, Y_2 = 1)$ and 1/3 or 10 to $(Y_1 = 1, Y_2 = 2)$. The same process is applied to the other partially classified data, and new cell probabilities are computed from the now-complete data. These steps are repeated

until convergence. Final probabilities for the cells using this process (Little & Rubin, 1987, p. 183) are:

$\hat{\pi}_{11} = .28,\ \hat{\pi}_{12} = .17,\ \hat{\pi}_{21} = .24,\ \hat{\pi}_{22} = .31$

as opposed to the complete data probability estimates:

$\hat{\pi}_{11} = .33,\ \hat{\pi}_{12} = .17,\ \hat{\pi}_{21} = .25,\ \hat{\pi}_{22} = .25.$

Nonmonotonic missing data (continuous variables).

A nonmonotonic MAR data set can be produced from the complete data presented in Table 6 by the application of two levels of decision-making. First, each variable in the complete data set is assigned probabilistically to one of three data patterns: complete data, X potentially missing or Y potentially missing. Then, for the second and third patterns, a data point is made missing or not depending on the value of the other variable (fulfilling the MAR condition). Specifically, for data pairs assigned pattern 2, X is missing if Y is less than or equal to 15; for those assigned pattern 3, Y is missing if X is greater than or equal to 9. This produces the data set in Table 10 below:

Table 10

Data Set 3: Nonmonotonic MAR Data

case	X	Y	case	X	Y
1	11.2	16.8	16	M	12.2
2	10.4	15.7	17	10.5	17.0
3	9.8	M	18	10.8	17.7
4	10.7	M	19	10.1	M
5	10.3	M	20	M	14.2
6	8.8	13.1	21	11.4	16.5
7	9.8	M	22	9.8	M
8	9.9	15.5	23	10.2	15.3
9	M	12.7	24	11.1	17.2
10	10.3	15.3	25	9.9	16.7
11	10.4	M	26	10.8	M
12	10.5	16.8	27	10.3	15.5
13	M	14.6	28	11.1	15.5
14	9.3	14.2	29	10.4	15.0
15	10.4	15.0	30	11.4	16.2

If this were a complete data set, we would maximize

$$L = \prod_{i=1}^{n} P(X_i, Y_i \mid \mu_X, \mu_Y, \sigma_X, \sigma_Y, \sigma_{XY}). \quad (3)$$

However, due to the nonmonotonic missing data, there will be three separate data sets: those cases which are complete on both variables, those which are missing on Y, and those which are missing on X. The likelihood is the product of the likelihoods of these three cases:

$$L = \prod_{i=1}^{n_1} P(X_i, Y_i \mid \mu_X, \mu_Y, \sigma_X, \sigma_Y, \sigma_{XY}) \; * \quad (4)$$

$$\prod_{i=n_1+1}^{n_1+n_2} P(X_i \mid \mu_X, \sigma_X) \; *$$

$$\prod_{i=n_1+n_2+1}^{n} P(Y_i \mid \mu_Y, \sigma_Y)$$

where $\underline{n}_1$ cases have complete data, $\underline{n}_2 - \underline{n}_1$ have X only, and $\underline{n} - \underline{n}_1 - \underline{n}_2$ have Y only. There is no straightforward way to maximize this type of likelihood: thus we need to use an iterative procedure, in this case the EM algorithm, to find the MLE. Table 11 below compares ML estimates,

obtained through use of the EM algorithm, with the parameter values and with estimates obtained by the three ad hoc procedures.

Table 11

Estimates for Nonmonotonic MAR data

	pairwise deletion	listwise deletion	mean subst.	MLE	parameter values
$\hat{\mu}_x$	10.369	10.439	10.369	10.266	10.000
$\hat{\sigma}_x$	.618	.690	.573	.662	1.000
$\hat{\mu}_y$	15.395	15.833	15.395	15.369	15.000
$\hat{\sigma}_y$	1.466	1.142	1.248	1.379	1.414
$\hat{\rho}_{xy}$	.709	.709	.485	.722	.707
$\hat{\beta}_0$	-2.070	3.586	4.460	.068	5.000
$\hat{\beta}_1$	1.684	1.173	1.055	1.490	1.000

It is even clearer in this case that mean substitution is the worst method, since it underestimates the variance of both variables and their correlation. The quality of the estimates produced by ML estimation and by the two deletion methods varies as a function of the parameter to be

estimated, again probably due to the small sample size and low percentage of data missing.

ML Estimation Applications to the Mixed Model (Data Missing on Both Continuous and Categorical Variables)

In the previous examples, the variables were of one type only, either continuous or categorical. In this section we consider the problem of missing data in the case where the model includes both continuous and categorical variables. Models with MAR data on both continuous and categorical variables may be obtained within the framework of the General Location Model (GLM), by employing the EM algorithm (Little & Rubin, ch. 10; Little & Schluchter, 1985). In the GLM, cases are classified into different cells depending on their values for the categorical variables, with the continuous variables distributed multivariate normal within each cell. Each cell may have a different mean for each continuous variable, but is assumed to have the same covariance structure. A consequence of this assumption of homogeneity of covariance is that the regression of any continuous variable on the remaining variables will not include any cross-product terms, and therefore only the intercept terms (not the slope)

will change as a function of the categorical variables. The model developed in this dissertation will allow for the possibility of cross-product terms; if the cross-product term is not significantly different from zero, the model is the same as the General Location Model. However, if the cross-product term is necessary, the variance/covariance matrices will be different in some way.

Consider the data set in Table 12 below, which has one continuous variable Y and two dichotomous categorical variables, X_1 and X_2.

Table 12

Data Set 4: Continuous (Y) and Dichotomous Categorical (X1 and X2)

Data

Y	X_1	X_2
5	1	1
M	1	1
5	M	2
3	1	2
4	2	1
8	M	1
7	2	2
4	2	2
M	M	2

Classifying cases by their X_1 and X_2 scores yields a 2 x 2 classification table, with a distribution of Y in each cell. Note that we are missing the Y score for some subjects, and are also missing one of the X scores for some, i.e. we don't know their cell classification. The unknown parameters to be estimated are the 4 cell probabilities, the 4 Y-means, and the common within-cell Y-variance. If the data were complete, the MLE's could be computed in terms of the following sufficient statistics: the cell frequencies, the sum of Y in each cell, and the sum of Y^2 over the

four cells. When some data are missing, the EM algorithm can be used to iteratively obtain the expected values of these sufficient statistics, and the corresponding MLE's. In the E-step, the expected values of the complete-data sufficient statistics are computed; in the M-step, the MLE's are computed with the E-step estimates replacing the complete-data statistics, and the two steps are repeated until the estimates converge (see Little & Rubin, pp. 197-198, for details).

Bayesian Methods

Bayesian methods are well established for complete data problems (Winkler, 1972; Lee, 1989; Box & Tiao, 1973). One advantage of the Bayesian approach is its ability to incorporate prior information which can increase the precision of the parameter estimates. This might be important in missing data problems, because of the loss of information due to an incomplete data set. A second advantage is that Bayesian estimates are appropriate for samples of any size, whereas the desirable properties of ML estimates (e.g., asymptotic normality) are based on the large sample case and may not hold with small samples. As was the case for ML estimation, Bayesian methods can be applied to the missing data case, although with some complications. First let us consider the complete data case, then the missing data case.

Suppose we have a complete data set. The likelihood for the

observed data (O) can be designated as P(O | θ), where this probability of the observed data is viewed as a function of θ, the unknown parameters. The ML approach simply requires maximizing this function to obtain an estimate of θ. In a Bayesian approach to estimating the distribution of θ, we combine the prior distribution, P'(θ), with the likelihood, P(data | θ), and use the prior and the likelihood to calculate the posterior distribution of θ, P''(θ | data). Point estimates for θ can be obtained in terms of the mean or mode of the posterior distribution. Interval estimates are obtained in terms of the highest posterior density region (Winkler, 1972). Specifically, prior beliefs and sample data are combined to produce a posterior distribution through the application of Bayes' theorem. Let P'(θ) be the prior distribution for θ. This can be interpreted as either a quantification of one's prior knowledge concerning θ, or as an empirical distribution. Let P''(θ | data) be the posterior distribution of θ after observing the data. Following Bayes' theorem:

$$P''(\theta \mid \text{data}) = \frac{P(\text{data} \mid \theta)\, P'(\theta)}{\int_{\theta} P(\text{data} \mid \theta)\, P'(\theta)} = \frac{P(\text{data} \mid \theta)\, P'(\theta)}{P(\text{data})} \qquad (5)$$

where P'(θ) = the prior distribution

P(data | θ) = the data likelihood

P''(θ I data) = the posterior distribution
P(data) = the marginal or predictive distribution of the data.

When there is a diffuse state of prior information, i.e. an uninformative prior, the posterior distribution depends almost entirely on the likelihood, and the ML and Bayesian approaches will produce similar results. Since the posterior distribution is proportional to the prior times the likelihood, if the prior is uninformative or diffuse, the likelihood will dominate the posterior. Note that the prior need only be locally diffuse for this condition to be met, i.e. it need only be noninformative in the range in which the likelihood is non-negligible (Winkler, 1972).

For many standard estimation problems, the form of the resulting posterior distribution given the prior and likelihood have been derived (Winkler, 1972; Lee, 1989; Box & Tiao, 1973). For instance, given a Beta prior and binomial likelihood, the posterior will have a Beta distribution whose parameters are simple functions of the parameters of the prior distribution and the data (Winkler, 1972). This straightforward manner of computing the posterior distribution (i.e., the posterior being a member of the same family of distributions as the prior) is generally not possible when missing data occur, even when the missing data are MAR. In the present research we will employ a Monte Carlo process, the Gibbs sampler (Gelfand & Smith, 1990; Dellaportas & Smith, 1993; Smith & Roberts, 1993; MacEachern & Berliner, 1994), to obtain posterior

distributions of the desired parameters.

The Gibbs Sampler

In general, the Gibbs sampler is a Monte Carlo approach to calculating estimates of marginal probability distributions from conditional distributions. It involves repeated generation of unknown parameter values, and the construction of the empirical marginal probability distribution. As an example, assume we have the following conditional densities (i.e., we can readily sample in a Monte Carlo fashion from these conditional distributions):

$$P(\theta_1 \mid \theta_2, \theta_3) \qquad (6)$$
$$P(\theta_2 \mid \theta_1, \theta_3)$$
$$P(\theta_3 \mid \theta_1, \theta_2).$$

The problem is to estimate the joint density of $(\theta_1, \theta_2, \theta_3)$ and then obtain an approximation of the marginal density of some parameter of interest. The Gibbs algorithm first generates a value for θ_1 from the first conditional density, using initial estimates of θ_2 and θ_3. It then generates a value for θ_2 using the θ_1 just generated and the initial estimate for θ_3, then generates a value for θ_3 using the generated values for θ_1 and θ_2. This process is repeated t times, each time substituting the new

generated value for each parameter for the previous estimate, and the generated parameter values at time $\underline{t}$, are regarded as a random sample of size one from the joint distribution of the parameters. This process is repeated $\underline{n}$ times, producing a sample of size $\underline{n}$ which can be used to empirically construct each parameter's marginal density (Dellaportas and Smith, 1993).

In the case of incomplete data, the missing data are treated as additional unknowns, analogous to the unknown parameters which are being estimated (Smith & Roberts, 1993). To take a simple case, suppose we have a data set consisting of a continuous Y-variable and dichotomous X-variable, with data missing on the X-variable only. Suppose that the conditional distribution of Y given X is normal with a linear homoscedastic regression model [$E(Y \mid X) = \beta_0 + \beta_1 X$; $Var(Y \mid X) = \sigma_{yy|x}$]. Given the observed data (O = observed X's and all Y's), the goal is to obtain the posterior distributions of $\underline{\beta}' = [\beta_0, \beta_1]$, $\sigma_{yy|x} = V$, and the probability π that X = 1. The posterior distributions $P''(\underline{\beta} \mid O)$, $P''(V \mid O)$ and $P''(\pi \mid O)$ are generated through the following steps:

1. Read in data and flag missing values.
2. Fill in missing X values, i.e. sample values for M, from

$\pi^* = P(M \mid \underline{\beta}, V, O, \pi)$, a distribution readily sampled from, using initial estimates for the population parameters. More specifically, given the parameter estimates, a simple formula can be derived for

P (X = 1 | $\underline{\beta}$, V, M, O, π) and P (X = 0 | $\underline{\beta}$, V, O, M, π). Then values for X can be generated using a random number generator, from a uniform (0, 1) distribution, using the rule that when the generated number is greater than $\pi*$, X = 0, and when it is less than $\pi*$, X = 1.

3. Sample the β_0 and β_1 parameters from the posterior bivariate normal distribution, P($\underline{\beta}$ | M, V, π, O) using the estimated values for the missing data plus the initial parameter estimates. From standard complete data theory (Box & Tiao, 1973) we know that this posterior distribution (using standard noninformative priors) is bivariate normal with mean $\underline{\beta} = (X'X)^{-1} (X'\underline{Y})$ and variance/covariance matrix $(X'X)^{-1} (\sigma^2)$ where X is the design matrix and $\underline{Y}$ is the vector of Y scores.

4. Sample V from a Gamma distribution, using the new $\underline{\beta}$ estimates plus the previous parameter estimates, i.e. sample from P(V | $\underline{\beta}$, π, M, O).

5. Sample π from a Beta distribution, i.e. sample from P(π | $\underline{\beta}$, V, M, O).

Steps 2-5 are repeated $\underline{t}$ times, in each repetition the previous parameter estimates being replaced with those which have just been calculated. The $\underline{t}$-th repetition is taken as a random sample of size one from the posterior distribution of the parameters given the observed data, i.e. P''($\underline{\beta}$, V, π | O). This process is repeated $\underline{n}$ times so that a sample of parameter values of size $\underline{n}$ is generated which may then be treated as a random sample from the joint posterior distribution. It is then possible to

plot each parameter separately to estimate the marginal posterior distribution of each parameter. These basic principles are applied in this dissertation to a more complicated data set consisting of a continuous Y variable, a dichotomous X_1 variable, a continuous X_2 variable, and the X_1X_2 cross-product term.

This is a subsampling approach to using the Gibbs sampler. Another possibility, and the one used in this dissertation, is to use the entire run of generated estimates, after an initial burn-in period. The merits of both approaches are discussed by MacEachern and Berliner (1994). They both produce reasonable estimates; the subsampling approach has the advantages that it definitely overcomes the problem of serial correlation, may require less computer space (since only the subsample needs to be stored) and provides a simple variance estimate, while the straight-run approach may ultimately be more efficient in the use of computer space since it requires a smaller sample to be generated (because fewer of the generated values are discarded).

Method

This dissertation addresses parameter estimation under MAR conditions for data sets consisting of a continuous Y variable, a dichotomous X_1 variable, a continuous X_2 variable, and a cross-product term (X_1X_2) in the regression of Y on X_1 and X_2. The model for this type of data is:

$$\hat{Y} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2. \quad (7)$$

This model would describe, for instance, an attempt to predict first-year college grades (Y) from SAT scores (X_2) and gender (X_1), assuming an interaction between gender and SAT scores. This model would also occur in any ANCOVA model with two groups and one covariate. There may be missing data on any of the variables and on any combination of them, in either a monotonic or nonmonotonic pattern; however, we will assume that we have some cases with complete data. In this dissertation the focus is on estimating the squared multiple correlation (ρ^2) between Y and X_1, X_2 and X_1X_2; it would also be possible to consider the estimates for the individual regression parameters (which were used to calculate ρ^2) if they were the focus of interest.

The present research uses Bayesian principles to estimate ρ^2, i.e. it combines information from the prior distribution of all unknown

parameters (θ) and the likelihood of the data given θ, to produce a posterior distribution of θ. Since the parameter of interest, ρ^2, is a function of θ, an interval estimate for ρ^2 can be obtained from the marginal posterior highest density region (HDR). Unfortunately, when data are missing, there is no straightforward way to compute the marginal posterior distribution of ρ^2. Therefore we use the Gibbs sampler to generate a posterior distribution for the desired parameters. Two FORTRAN programs which perform the necessary computations have been written (Appendixes 2 and 3). For convenience, we will use the notation $\sigma_{yy|x1,x2,x1x2} = Var(Y \mid X_1,X_2,X_1X_2) = V_1$, and $\sigma_{x2x2|x1} = Var\,(X_2 \mid X_1) = V_2$.

The proposed statistical model is as follows:

$$P(Y \mid X_1, X_2) = N(\beta_0 + \beta_1X_1 + \beta_2X_2 + \beta_3X_1X_2, V_1) \quad (8)$$

$$P(X_2 \mid X_1) = N(\alpha_0 + \alpha_1X_1, V_2) \quad (9)$$

$$P(X_1) = \pi^{x1}\,(1 - \pi)^{(1-x1)} \quad (10)$$

In terms of the model defined in Equations 8-10, one can express the squared multiple correlation between Y and X_1, X_2 and X_1X_2. This squared correlation can be defined as:

$$\rho 2 = 1 - \frac{V_1}{\sigma_{yy}} \qquad (11)$$

where σ_{yy} is the unconditional variance of Y. An expression for σ_{yy} can be constructed by noting the following:

a. The marginal distribution of Y is a normal mixture of the form $(1 - \pi)\, P(Y \mid X_1 = 0) + (\pi)\, P(Y \mid X_1 = 1)$.

b. The distributions of $P(Y \mid X_1 = 0)$ and $P(Y \mid X_1 = 1)$ are normal with means and variances which are functions of the basic parameters.

c. $E(Y^2) = (1 - \pi)\, P(Y^2 \mid X_1 = 0) + (\pi)\, P(Y^2 \mid X_1 = 1)$ (12)

d. $E(Y) = (1 - \pi)\, E(Y \mid X_1 = 0) + (\pi)\, E(Y \mid X_1 = 1)$ (13)

e. The final expression for σ_{yy} is :

$$E(Y^2) - [E(Y)]^2 =$$
$$1 - \pi)\,[V_1 + \beta_2^2\, V_2] + (\pi)\,[V_1 + (\beta_2 + \beta_3)^2\, V_2] \qquad (14)$$
$$+ (\pi)\,(1 - \pi)\,[\beta_1 + \beta_2\, \alpha_1 + \beta_3\,(\alpha_0 + \alpha_1 X_1)]^2$$

We will assume that the parameter sets ($\underline{\beta}$, V_1), ($\underline{\alpha}$, V_2), and (π) are a priori independent in terms of the joint priors. Further, the marginal

prior distributions for ($\underline{\beta}$, V_1), ($\underline{\alpha}$, V_2), and (π) follow the standard noninformative form: $P'(\underline{\beta}, V_1) \propto 1/V_1$, $P'(\underline{\alpha}, V_2) \propto 1/V_2$, and $P'(\pi) = 1$. It should be noted that the model given by Equations 12-14 does not constrain the 2 x 2 variance-covariance matrices of Y and X_2 to be constant for $X_1 = 1$ and $X_1 = 0$. Were these matrices equal, β_3 would equal 0 and this model would be equivalent to the General Location Model.

The known information in this model is the observed data (O); unknowns are the missing data (M), and the parameters $\underline{\beta}$ (the vector of regression weights predicting Y from X_1 and X_2), V_1 (the residual variance for Y given X_1 and X_2), $\underline{\alpha}$ (the vector of regression weights predicting X_2 from X_1), V_2 (the residual variance of X_2 given X_1) and π (the probability that $X_1 = 1$). The problem is to construct the joint posterior distribution of these parameters, given the incomplete observed data (O), i.e. $P''(\underline{\beta}, V_1, \underline{\alpha}, V_2, \pi \mid O)$. Once this distribution is approximated it is straightforward to approximate the posterior distribution of the squared multiple correlation.

The first step in the Gibbs process is to set up the following series of conditional distributions:

$$P''(M \mid \underline{\beta}, V_1, \underline{\alpha}, V_2, \pi, O) \quad (15)$$

$$P''(\underline{\beta} \mid V_1, M, O) \quad (16)$$

$$P''(V_1 \mid \underline{\beta}, M, O) \quad (17)$$

$P''(\underline{\alpha} \mid V_2, M, O)$ (18)

$P''(V_2 \mid \underline{\alpha}, M, O)$ (19)

$P''(\pi \mid M, O)$ (20)

The form of the first conditional distribution varies according to the missing data pattern for each case; this is described in detail in Equations 28-58. Each of the other conditional distributions has a known form; for instance the $\underline{\alpha}$- and $\underline{\beta}$-weights are distributed multivariate normal, V_1 and V_2 have Gamma distributions, and π has a Beta distribution. More specifically,

$$P(\underline{\beta} \mid V_1, M, O) = N_4 (\underline{\hat{\beta}}, (X'X)^{-1} V_1) \tag{21}$$

where $\underline{\hat{\beta}}' = [\beta_0 \ \beta_1 \ \beta_2 \ \beta_3]$

and X is a $\underline{n}$ x 4 matrix consisting of X_0 (a column of 1's) and each subject's X_1, X_2 and cross-product (X_1X_2) scores. Similarly,

$$P(V_1 \mid \underline{\beta}, M, O) = \Gamma (\theta_1, \theta_2) \tag{22}$$

where $\theta_1 = (n) / 2$, where n = the number of subjects, and

$$\theta_2 = [\sum_{i=1}^{n} (Y - \beta_0 - \beta_1 X_1 - \beta_2 X_2 - \beta_3 X_1 X_2)^2] / 2 \tag{23}$$

For the second set of terms,

$$P(\underline{\alpha} \mid V_2) = N_2 (\hat{\underline{\alpha}}, (X_1'X_1)^{-1} V_2) \quad (24)$$

where $\hat{\underline{\alpha}}' = [\alpha_0 \quad \alpha_1]$, and X_1 is a matrix consisting of a column of 1's and a column of each subject's X_1 scores. The second variance term is defined by:

$$P(V_2 \mid \underline{\alpha}, M, O) = \Gamma (\theta_3, \theta_4), \quad (25)$$

where $\theta_3 = n/2$, and $\theta_4 =$

$$[\sum_{i=1}^{n} (X_2 - \alpha_0 - \alpha_1 X_1)^2] / 2 \quad (26)$$

The conditional distribution for π is:

$$P''(\pi \mid M, O) = \text{Beta } (\theta_5, \theta_6) \quad (27)$$

where $\theta_5 = \Sigma X_1 + 1$, and $\theta_6 = n - \Sigma X_1 + 1$

The process begins by setting initial values for $\underline{\alpha}$, $\underline{\beta}$, V_1, V_2 and π and sampling values for the missing data. Given the observed and missing

values, we cycle through Equations 16 to 20 to obtain new values for $\underline{\alpha}$, $\underline{\beta}$, V_1, V_2 and π. Given these new values, we generate new values for the missing data. This process is repeated $\underline{t}$ times. The first $\underline{t}$ - 1 values are discarded as a burn-in period and the $\underline{t}$-th generated set of values for $\underline{\alpha}$, $\underline{\beta}$, V_1, V_2 and π is taken as a sample of size 1 from the joint posterior density $P(\underline{\alpha}, \underline{\beta}, V_1, V_2, \pi \mid O)$. The ρ^2 value is computed (as described in Appendix 1) using the parameter values in the sample, thus providing a sample of size 1 from the marginal posterior distribution of ρ^2. Continuing to generate parameter values and to calculate ρ^2 from these values an additional $\underline{M}$ times yields a sample of $\underline{M}$ ρ^2 values. By plotting these $\underline{M}$ ρ^2 values, one can empirically approximate an $(1 - \alpha)$ highest density region for ρ^2.

Missing Data Patterns

The data set under consideration consists of a continuous Y variable, a dichotomous X_1 variable, and a continuous X_2 variable. As shown in the table below (M = missing, O = observed), there are seven patterns of missing data (the eighth possible pattern, with all variables missing, is not relevant here, since the data are MAR).

Table 13

Missing Data Patterns

Pattern	Y	X_1	X_2	description
P_1	O	O	O	complete data
P_2	O	O	M	X_2 missing
P_3	O	M	O	X_1 missing
P_4	M	O	O	Y missing
P_5	O	M	M	X_1 and X_2 missing
P_6	M	O	M	Y and X_2 missing
P_7	M	M	O	Y and X_1 missing

The problem is to iteratively generate values for the missing data and use both the observed and generated data to iteratively generate values for the parameters of interest. Each pattern of missing data requires a

different set of procedures, which are outlined below. For each of the six incomplete data patterns we consider the probability distribution of the missing data given the observed data and current parameter values. When a continuous variable is missing (Y or X_2), we directly sample from the probability distribution of the missing data given the current parameter values and the observed data. This sampling is done using Monte Carlo algorithms available in the IMSL subroutine package (IMSL, 1987). When the missing data value is dichotomous (X_1), a two-step process is necessary. Given the current value for π [$P(X_1 = 1)$] we generate a uniform random variable (u) between 0 and 1. If u is less than or equal to π, X_1 is assigned the value of 1; if u is greater than π, X_1 is assigned the value of 0.

P2: Y and X1 Observed, X2 Missing

The first missing data pattern is P_2, where Y and X_1 are observed, and we need to estimate X_2. The problem is to define $P(X_2 \mid X_1, Y)$. We know from the original definitions of our variables that $P(Y \mid X_1, X_2)$ and $P(X_2 \mid X_1)$ are normally distributed with the following parameters:

$$P(Y \mid X_1, X_2) = N(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2, \text{Var}(Y \mid X_1, X_2)) \quad (28)$$

$$P(X_2 \mid X_1) = N(\alpha_0 + \alpha_1 X_1, \text{Var}(X_2 \mid X_1)) \quad (29)$$

When $X_1 = 1$:

$$P(Y \mid X_1, X_2) = N(\beta_0 + \beta_1 + (\beta_2 + \beta_3)X_2, \text{Var}(Y \mid X_1, X_2)) \quad (30)$$

$$P(X_2 \mid X_1) = N((\alpha_0 + \alpha_1, \text{Var}(X_2 \mid X_1)). \quad (31)$$

When $X_1 = 0$:

$$P(Y \mid X_1, X_2) = N(\beta_0 + \beta_2 X_2, \text{Var}(Y \mid X_1, X_2)) \quad (32)$$

$$P(X_2 \mid X_1) = N(\alpha_0, \text{Var}(X_2 \mid X_1)). \quad (33)$$

If $P(Y \mid X_1, X_2)$ and $P(X_2 \mid X_1)$ are both normally distributed, then $P(X_2, Y \mid X_1)$ is distributed bivariate normal with parameters:

$$P(X_2, Y \mid X_1) = N_2[\underline{\mu}, \Sigma] \quad (34)$$

where $\underline{\mu}' = [a \ b]$, and

$$\Sigma = \begin{bmatrix} c & d \\ & e \end{bmatrix}$$

with the further definitions that $a = E(X_2 \mid X_1)$, $b = E(Y \mid X_1)$,

$c = Var(X_2 \mid X_1)$, $d = Cov(Y, X_2 \mid X_1)$, and $e = Var(Y \mid X_1)$. Note that a, b, d and e will be different when $X_1 = 0$ and $X_1 = 1$, but c will not change.

The conditional distribution of X_2 given Y and X_1, $P(X_2 \mid X_1, Y)$, will be normal with a standard linear and homoscedastic regression:

$$P(X_2 \mid Y, X_1) = N(a - \frac{d}{e} b + \frac{d}{e} Y, c(1 - \frac{d^2}{ce})) \tag{35}$$

The variable X_1 is included as a predictor in these equations because a, b, d and e will vary depending on whether $X_1 = 1$ or $X_1 = 0$.

We can find the values for the parameters a-e as follows:

$$\begin{aligned} a = E(X_2 \mid X_1) &= \alpha_0 + \alpha_1 X_1 \\ &= \alpha_0 + \alpha_1 \quad (\text{when } X_1 = 1) \\ &= \alpha_0 \quad (\text{when } X_1 = 0). \end{aligned} \tag{36}$$

The value of $b = E(Y \mid X_1)$ can be found by taking $E(Y \mid X_1, X_2)$ and averaging over X_2. Thus $E(Y \mid X_1) = \beta_0 + \beta_1 X_1 + \beta_2[E(X_2 \mid X_1)] + \beta_3 X_1 E(X_2 \mid X_1)]$, where $E(X_2 \mid X_1) = \alpha_0 + \alpha_1 X_1$. In summary:

$$b = E(Y \mid X_1) = \beta_0 + \beta_1 X_1 + (\alpha_0 + \alpha_1 X_1)(\beta_2 + \beta_3 X_1). \tag{37}$$

The first variance component was previously defined as:

$$c = \text{Var}(X_2 \mid X_1) = V_2. \tag{38}$$

The second variance component, $e = \text{Var}(Y \mid X_1)$ can be calculated as:

$$\begin{aligned} e = \text{Var}(Y \mid X_1) &= E\,[\text{Var}\,(Y \mid X_1, X_2)] + \text{Var}\,[E(Y \mid X_1, X_2)\,] \\ &= \text{Var}\,(Y \mid X_1, X_2) + \text{Var}[\beta_0 + \beta_1 X_1 + (\beta_2 + \beta_3 X_1)\, X_2] \end{aligned} \tag{39}$$

V_1 was previously defined as $\text{Var}\,(Y \mid X_1, X_2, X_1X_2)$, so Equation 39 can be written as:

$$e = V_1 + \text{Var}[\beta_0 + \beta_1 X_1 + (\beta_2 + \beta_3 X_1)\, X_2] \tag{40}$$

Since $(\beta_0 + \beta_1 X_1)$ is a constant, Equation 39 can finally be written as

$$e = V_1 + \text{Var}(X_2 \mid X_1)\,(\beta_2 + \beta_3 X_1)^2. \tag{41}$$

An expression for the covariance component, $d = \text{Cov}(Y, X_2 \mid X_1)$, can be computed by noting that the regression of Y on X_2 given X_1 can be written as:

$$E(Y \mid X_1, X_2) = [\beta_0 + \beta_1 X_1] + [\beta_2 + \beta_3 X_1]\, X_2. \tag{42}$$

The second bracketed expression is analogous to a slope coefficient, giving us:

$$\beta_2 + \beta_3 X_1 = \frac{\text{Cov}\,(Y, X_2 \mid X_1)}{\text{Var}\,(X_2 \mid X_1)} \tag{43}$$

Thus: $d = (\beta_2 + \beta_3 X_1)\ \text{Var}(X_2 \mid X_1)$. (44)

In summary, given X_1 and Y, X_2 is normally distributed with mean and variance given by Equation 35, and the terms of the equations can be found as demonstrated above.

P3: Y and X2 Observed, X1 Missing

One can find the probability of X_1, given Y and X_2 as follows:

$$P(X_1 \mid Y, X_2) = \frac{P(Y, X_1, X_2)}{P(Y, X_2)} \tag{45}$$

Writing the joint probabilities as products of the conditionals and marginal, and summing the denominator over X_1 produces:

$$P(X_1 \mid Y, X_2) = \frac{P(Y \mid X_1, X_2)\, P(X_2 \mid X_1)\, P(X_1)}{\sum_{X_1} P(Y \mid X_1, X_2)\, P(X_2 \mid X_1)\, P(X_1)} \tag{46}$$

The necessary terms found as follows (again using the definitions $V_1 = Var(Y \mid X_1, X_2, X_1X_2)$ and $V_2 = Var(X_2 \mid X_1)$):

(47)

$$P(Y \mid X_1,X_2) = \frac{1}{\sqrt{V_1}} \exp \frac{-1}{2V_1} (Y - \beta_0 - \beta_1X_1 - \beta_2X_2 - \beta_3X_1X_2)^2$$

$$P(X_2 \mid X_1) = \frac{1}{\sqrt{V_2}} \exp \frac{-1}{2V_2} (X_2 - \alpha_0 - \alpha_1X_1)^2 \quad (48)$$

Note that the constant term $1/\sqrt{2\pi}$, where π is the constant 3.14. . ., has been omitted from both numerator and denominator since it cancels out.

P4: X_1 and X_2 Observed, Y Missing

By definition of our variables, Y given X_1 and X_2 is distributed normally, as follows (Equation 8) :

$$P(Y \mid X_1, X_2) = N(\beta_0 + \beta_1X_1 + \beta_2X_2 + \beta_3X_1X_2, V_1)$$

Using current parameter estimates for the β-weights and V_1, the missing

Y-scores can be generated.

P5: Y Observed, X_1 and X_2 Missing

First sample X_1 from $P(X_1 \mid Y)$, then sample X_2 from $P(X_2 \mid X_1, Y)$ as described in Equations 28-44. The conditional probability of X_1 given Y is found as follows:

$$P(X_1 \mid Y) = \frac{P(X_1, Y)}{P(Y)} = \frac{\int_{X_2} P(X_1, X_2, Y)}{\sum_{X_1} \int_{X_2} P(X_1, X_2, Y)} \quad (49)$$

The joint distribution of X_1, X_2 and Y can be expressed as:

$$P(Y, X_2 \mid X_1)\, P(X_1). \quad (50)$$

After integrating over X_2, the numerator of Equation 49 is expressed as:

$$= P(Y \mid X_1)\, P(X_1) \quad (51)$$

$$= [N(b, e)]\, [(\pi)^{X1} (1 - \pi)^{(1 - X_1)}] \quad (52)$$

where b and e vary according to whether $X_1 = 1$ or $X_1 = 0$ (i.e.,

$b_1 = E(Y \mid X_1 = 1)$, as given in Equation 37, and $e_1 = Var(Y \mid X_1 = 1)$, as given in Equation 39), and π is $P(X_1 = 1)$.

Therefore, the numerator in Equation 49 is (except for the constant $\sqrt{1/2\pi}$, π here being the constant 3.14. . .):

$$P(Y \mid X_1) = \frac{1}{\sqrt{e_1}} \exp \frac{-1}{2\, e_1} (Y - b_1)^2 (\pi) \tag{53}$$

The denominator of Equation 49 is then: (54)

$$\frac{1}{\sqrt{e_1}} \exp \frac{-1}{2\, e_1} (Y - b_1)^2 (\pi) + \frac{1}{\sqrt{e_0}} \exp \frac{-1}{2\, e_0} (Y - b_0)^2 (1 - \pi)$$

P6: X_1 Observed, X_2 and Y Missing

First values for X_2 are generated by sampling from $N(\alpha_0 + \alpha_1 X_1, V_2)$. Then, using these sampled X_2 values, Y is sampled from the normal distribution $N(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2, V_1)$.

P7: X_2 Observed, Y and X_1 Missing

First sample X_1 from $P(X_1 \mid X_2)$, then sample Y from $P(Y \mid X_1, X_2)$. The second distribution is defined in Equation 8, and the first is outlined

below.

$$P(X_1 \mid X_2) = \frac{P(X_1, X_2)}{P(X_2)} = \frac{P(X_1)\ P(X_2 \mid X_1)}{\sum_{X_1} P(X_1)\, P(X_2 \mid X_1)} \tag{55}$$

When $X_1 = 1$, the numerator of Equation 55 is:

$$(\pi)\frac{1}{\sqrt{V_2}}\ \exp\frac{-1}{2V_2}(X_2 - \alpha_0 - \alpha_1 X_1)^2 \tag{56}$$

and when $X_1 = 0$, the numerator is:

$$(1 - \pi)\frac{1}{\sqrt{V_2}}\ \exp\frac{-1}{2V_2}(X_2 - \alpha_0)^2 \tag{57}$$

The denominator is the sum of these two terms, i.e.: (58)

$$(\pi)\frac{1}{\sqrt{V_2}}\ \exp\frac{-1}{2V_2}(Y - \alpha_0 - \alpha_1 X_1)^2 + (1 - \pi)\frac{1}{\sqrt{V_2}}\ \exp\frac{-1}{2V_2}(Y - \alpha_0)^2$$

Demonstration of the Gibbs Sampling Method

To evaluate the statistical accuracy of the Bayesian interval estimates of ρ^2, a series of sampling experiments was conducted. This process consisted of the following steps:

1. Values were specified for $\underline{\alpha}$, $\underline{\beta}$, V_1, V_2, and π (see Appendix 1 for actual values) such that $\rho^2 = .1, .25$ or $.50$.

2. Sample size was specified as $\underline{n} = 30, 50$ or 100.

3. Using two FORTRAN programs (reproduced in Appendices 2 and 3), a random sample of size $\underline{n}$ was generated from a population using the parameters specified in step 1, and a subset of the data was deleted using a MCAR or MAR process, as described below.

4. Given an incomplete sample drawn from a specified population, a value for ρ^2 was generated from the posterior distribution of ρ^2 given the incomplete data set, using the Gibbs sampling process. This process was repeated 6,000 times; after a burn-in period of 1,000 repetitions, the last 5,000 ρ^2 values were saved in a file.

5. An interval estimate of ρ^2 in terms of a .90 highest density region (HDR) was computed as follows: the 5,000 ρ^2's were rank-ordered, the width of the 500 possible .90 HDR's (each containing 4500 values) was calculated, and the narrowest of these intervals was selected as the HDR. These five steps were repeated twice for each of the 18 specified population/sample size combinations shown in Tables 14 and 15, once

with an MCAR process, once with an MAR process.

Table 14

Bayesian .90 Highest Density Regions for the Squared Population Multiple Correlation (Data MCAR)

sample size	P(missing)	ρ2	complete cases	HDR	HDR width	HDR contains ρ2?	E(ρ2)
100	0.271	0.10	76	.08, .31	0.23	yes	.20
		0.25	75	.10, .35	0.25	yes	.23
		0.50	69	.51, .73	0.22	no	.61
50	0.271	0.10	33	.00, .31	0.31	yes	.16
		0.25	36	.01, .33	0.32	yes	.18
		0.50	36	.35, .68	0.33	yes	.53
30	0.271	0.10	20	.03, .42	0.39	yes	.24
		0.25	22	.23 ,.67	0.44	yes	.46
		0.50	25	.19, .67	0.48	yes	.44
100	0.657	0.10	35	.01, .22	0.21	yes	.22
		0.25	27	.05, .34	0.29	yes	.34
		0.50	35	.45, .72	0.27	yes	.59
50	0.657	0.10	19	.10, .53	0.43	yes	.32
		0.25	21	.16, .60	0.44	yes	.38
		0.50	19	.21, .66	0.45	yes	.43
30	0.657	0.10	12	.05, .55	0.50	yes	.31
		0.25	7	.02, .54	0.52	yes	.29
		0.50	13	.24, .73	0.49	yes	.48

Table 15

Bayesian .90 Highest Density Regions for the Squared Population Multiple Correlation (Data MAR)

sample size	P(missing)	ρ2	complete cases	HDR	width	HDR contains ρ2?	E(ρ2)
100	0.271	0.10	76	.07, .30	0.23	yes	.19
		0.25	74	.15, .44	0.29	yes	.30
		0.50	73	.45, .70	0.25	yes	.57
50	0.271	0.10	31	.03, .68	0.65	yes	.36
		0.25	33	.04, .47	0.43	yes	.26
		0.50	36	.29, .67	0.38	yes	.49
30	0.271	0.10	20	.01, .41	0.40	yes	.23
		0.25	22	.21, .67	0.45	yes	.44
		0.50	26	.25, .74	0.49	yes	.49
100	0.657	0.10	38	.00, .17	0.17	yes	.08
		0.25	29	.01, .33	0.32	yes	.18
		0.50	36	.41, .76	0.35	yes	.61
50	0.657	0.10	34	.13, .79	0.66	no	.45
		0.25	19	.02, .43	0.41	yes	.24
		0.50	16	.21, .71	0.50	yes	.45
30	0.657	0.10	10	.20, .83	0.63	no	.53
		0.25	13	.07, .61	0.54	yes	.35
		0.50	10	.18, .81	0.63	yes	.51

MCAR Case

To create an MCAR data set, the sample data were regarded as n cases each consisting of 3 variables (Y, X_1 and X_2). Each variable was assigned a specified chance of being missing, independently of the others, except for the provision that all three could not be missing; if this occurred, the value of X_1 was restored. Two values for P(missing) were used: .1 and .3. Using P(missing) = .1 yields the following (for any one case):

$P(Y, X_1, X_2 \text{ observed}) = (.9)^3 = .729$

$P(\text{one variable missing}) = 3(.9)^2(.1) = .243$

$P(\text{two variables missing}) = 3(.9)(.1)^2 = .027$

$P(\text{all three missing}) = (.1)^3 = .001$

Since in the last case, X_1 is restored, this yields:

$P(\text{two variables missing}) = .028$

$P(\text{three variables missing} = 0.0$

Therefore, the probability of at least one variable being missing in any particular case is 1 - .729 or .271.

When P(missing) = .3 for any one variable, the following probabilities exist:

P(all observed) = $(.7)^3 = .343$

P(one missing) = $3(.7)^2(.3) = .441$

P(two missing) = $3(.7)(.3)^2 = .189$

P(three missing) = $(.3)^3 = .027$

Applying the rule that if all three variables are missing, X_1 is restored, we have:

P(two missing) = .216

P(three missing) = 0.0

The probability of a particular case having one or more variables missing = 1 - .343 or .657.

MAR Case

The following rules were applied:

1. X_1 is always observed
2. X_2 is MCAR
3. Y is MAR, dependent on the value of X_1.

To set the probability that one or more variables would be missing to

.271, the following probabilities were used:

$P(X_2 \text{ missing}) = .10$

$P(Y \text{ missing} \mid X_1 = 1) = .33$

$P(Y \text{ missing} \mid X_1 = 0) = .05$

To set the probability that one or more variables would be missing to .657, the following probabilities were used:

$P(X_2 \text{ missing}) = .30$

$P(Y \text{ missing} \mid X_1 = 1) = .62$

$P(Y \text{ missing} \mid X_1 = 0) = .40$

In both the MCAR and MAR cases, the decision to make a given variable missing or not was made by generating a random number from a uniform distribution in the range [0 - 1], and making the variable missing if the uniform number was less than the specified probability of being missing for that variable. For instance, in the first MCAR case, if the random number was less than .1, the variable was made missing.

The results in terms of the .90 HDR's are given in Table 14 as a function of sample size, ρ^2 population value, and P(missing) for the MCAR case, and in Table 15 for the MAR case, and histograms for the distribution of ρ^2 in two of the populations are given in Tables 16 and 17.

The results in Tables 14 and 15 showed that the Bayesian .90 interval estimates were quite accurate in both the MCAR and MAR cases. With MCAR data, 17 of the 18 HDR's containing the population ρ^2, while in the MAR case, 16 of the 18 HDR's contained the population ρ^2. Note that 18 different samples were used (this was controlled by changing the seed), and that the same sample was used for both MCAR and MAR calculations for each distinct specification of sample size, P(missing) and ρ^2. A further inspection of tables 14 and 15 shows that for both the MCAR and MAR cases, the smaller sample sizes (e.g., $\underline{n}$ = 30) produces less precise estimates, i.e., the HDR's for these samples are wider than for the larger samples (e.g., $\underline{n}$ = 100). Further, as P(missing) increases (holding $\underline{n}$ constant) from .271 to .657, precision also tends to decrease. In addition, population ρ^2 appears to have no effect on HDR width. Finally, comparing the MCAR and MAR tables, we see that the HDR's computed on MAR data tend to be wider (holding all other factors constant) than the HDR's computed on MCAR data. There are a few odd results in the MAR case which may be due to each estimate being based on only one sample, a situation which could be remedied in future research by taking multiple (I.e., 100) samples from each population, and reporting the average HDR for each population. For instance, note the odd results from the samples from populations with ρ^2 = .10; contrary to expectations, the larger samples do not always result in narrower HDR's. In particular, the case with $\underline{n}$ = 50, P(missing) = .657, and ρ^2 = .10 has more complete

cases than the analogous samples with $\rho^2 = .25$ and $\rho^2 = .50$, yet it has the widest HDR.

As examples, the histograms (of 5,000 sampled values of ρ^2 from two different populations) are presented in Tables 16 and 17. Both approximate normal distributions. The histogram in Table 16, for the MCAR case where $\rho^2 = .10$, P(missing) = .271, and $\underline{n} = 100$ (76 complete cases) is only slightly skewed, with a kurtosis of .173, a mean of .195 and a mode of .134. The histogram in Table 16, representing ρ^2 values generated from the MCAR cases where $\rho^2 = .50$, P(missing) = .271, and $\underline{n} = 30$ (25 complete cases) has a stronger skew, with a kurtosis of -.406, a mean of .441 and a mode of .511.

Table 16.

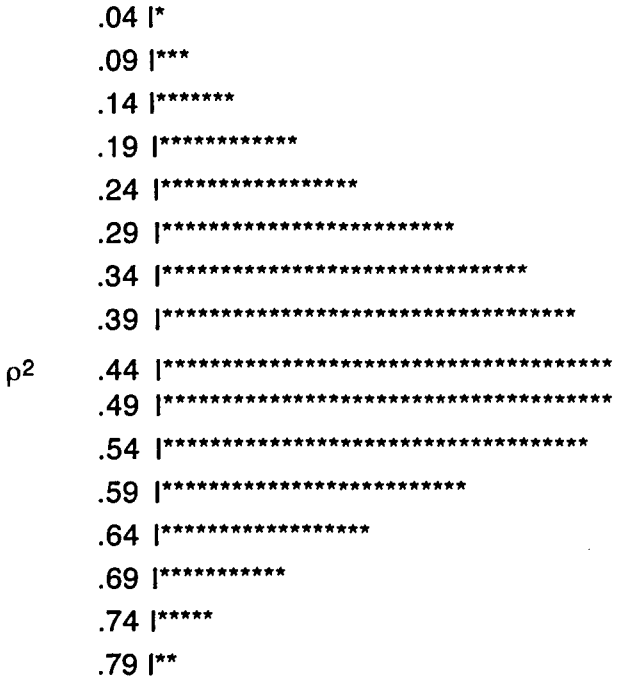
Distribution of 5,000 ρ^2 values : MCAR case where $\rho^2 = .10$, P(missing) = .271, and n = 100 (76 complete cases)

ρ^2		
	.030	I**
	.055	I*****
	.080	I**********
	.105	I**********
	.130	I**********
	.155	I**********
	.180	I**********
	.205	I**********
	.230	I**********
	.255	I**********
	.280	I**********
	.305	I**********
	.330	I********
	.355	I*****
	.380	I**
	.405	I*

One symbol equals approximately 16 occurrences.

Table 17.

Distribution of 5,000 ρ^2 values : MCAR case where $\rho^2 = .50$, P(missing) = .271, and n = 30 (25 complete cases)



One symbol equals approximately 16 occurrences.

Discussion and Summary

In the present research we have investigated the problem of estimating the multiple correlation for regression models containing a mix of categorical, continuous and interaction terms given a data set containing missing values. We consider the case where the model has a single binary predictor variable, a single continuous predictor variable, and one cross-product term. A Bayesian approach is used to obtain an interval estimate of ρ^2 using a Gibbs sampling procedure: using 5,000 samples drawn from the posterior distribution of ρ^2, we empirically construct .90 HDR's for ρ^2. To demonstrate the estimation procedure, 18 cases with MCAR data and 18 cases with MAR data were studied. Within each set of 18, three sample sizes ($\underline{n}$ = 30, 50 or 100), three population values for ρ^2 (.10, .25 and .50) and two probabilities of missing data (.271 and .657) were used. In the MCAR case, 17 of the 18 HDR's contained the population ρ^2, while in the MAR cases, 16 of the 18 HDR's contained the population ρ^2. As expected, smaller sample sizes and more missing data produced wider HDR's, and MAR data produced slightly wider HDR's than MCAR data.

There are several reasons why the research presented here has practical uses. First, it can accommodate models using both continuous and categorical independent variables, and their cross-products as well.

Thus, it represents a generalization of the General Location Model. Second, it provides a procedure for dealing with missing data, without requiring the strong assumption that the data be missing in an MCAR fashion. Only the weaker MAR assumption is required. Third, the method gives an interval estimate (an interval of specified probability) of the parameters being estimated in terms of an HDR.

This research suggests several questions which may be addressed in the future. For instance, this research used noninformative priors; how would the results change if informative priors were incorporated into the estimation procedures? Another avenue of research is the generalization to models with more than 2 independent variables; how would multiple continuous and categorical independent variables be tested, and how would the many potential cross-product terms be tested for inclusion? Finally, how many repetitions of the Gibbs process are necessary for convergence, and how many repetitions are needed for the burn-in period? A burn-in period of 1,000 repetitions and 5,000 saved repetitions were used in this program, and they provided satisfactory results, but might equally good results be gained with fewer repetitions? If a more complicated model was being tested, it could be critical to know the minimum number of repetitions necessary for convergence.

Appendix 1: Population Parameters

$\beta_0 = \beta_1 = \beta_2 = \beta_3 = 1.0$

$\alpha_0 = \alpha_1 = 1.0$

$V_2 = 5.0$

$V_1 = 148.5$, 49.5 or 16.5; $p^2 = ,10, .25$ or .50

$\pi = .5$

According to Equation 11, $\rho^2 = 1 - \frac{V_1}{Var(Y)}$

According to Equation 14, Var(Y) =

$$(1 - \pi)[V_1 + \beta_2^2 V_2] + (\pi)[V_1 + (\beta_2 + \beta_3)^2 V_2]$$
$$+ (\pi)(1 - \pi)[\beta_1 + \beta_2 \alpha_1 + \beta_3(\alpha_0 + \alpha_1 X_1)]^2$$

Substituting the parameter values above, this gives us:

$$\rho^2 = 1 - \frac{V_1}{V_1 + 16.5}$$

so when $V_1 = 148.5$, $\rho^2 = 1 - \frac{148.5}{148.5 + 16.5} = 1 - .9 = .1$

When $V_1 = 49.5$, $\rho^2 = 1 - \frac{49.5}{49.5 + 16.5} = 1 - .75 = .50$

When $V_1 = 16.5$, $\rho^2 = 1 - \frac{16.5}{16.5 + 16.5} = 1 - .50 = .50$

Appendix 2: FORTRAN Program for MCAR Data

```
// JOB TIME=5
// EXEC FORTVCLG,IMSLIF='SYS2.IMSL.SPIF',IMSL='SYS2.IMSL'
//*MAIN LINES=8
//SYSIN DD *
C THIS IS THE MCAR PROGRAM.

      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION DATA(100,4), XTXIY(4,4), TY(4,4), B(1,4), GAM(1)
      DIMENSION BETA(1)
      DIMENSION XTXIX2(2,2),TX2(2,2),ALPHA(1,2)
      DIMENSION IR(100,4)
      DIMENSION STORE(5000),USE(5000)
      DIMENSION BHAT(4),ALPHAT(2)
      DIMENSION BMEAN(4),ALPHAM(2),BP(4),ALPHAP(2)
      DIMENSION TEMP(4)
      DIMENSION Z(1), U(1)

C DATA(I,1)=Y
C DATA(I,2)=X1=BINARY VARIABLE
C DATA(I,3)=X2=CONTINUOUS VARIABLE
C DATA(I,4)=X1*X2
C R = MATRIX OF RESPONSE PATTERNS
C XTXIY = 4 BY 4 SUMSQ-CROSSPROD MATRIX FOR PREDICTING Y
C        FROM X0 X1 X2 X1*X2.
C XTXIX2 2 BY 2 FOR PREDICTING X2 FROM X1.
C TY=CHOLESKY FACTOR FOR GENERATING 4 BY 1 VECTOR OF B WEIGHTS.
C TX2=CHOLESKY FACTOR FOR GENERATING 2 BY 1 VECTOR OF ALPHA WEIGHTS.
C BHAT = LS REG WEIGHTS PREDICTING Y FROM ALL X'S BASED
C ON FILLED-IN DATA SeT.

C ALPHAT=LS REG WEIGHTS PREDICTING X2 FROM X1.
C BP, ALPHAP,V1P ETC. ARE PARAMETER VALUES READ IN TO
C    GENERATE THE DATA SET WHICH IS THEN MADE PARTIALLY
C MISSING.
C
       READ*,ISEED
       CALL RNSET (ISEED)
C
       READ*,PROBM
       PRINT 444,PROBM
444    FORMAT(1X,'PROBABILITY OF DATA MISSING = ',F6.4)
C
C READING IN NUMBER OF SUBJECTS, MISSING DATA INDICATOR, NUMBER OF
```

```
C  REPETITIONS AND PARAMETER VALUES FROM FIRST LINE OF
C DATA
      READ*,NSUBJ,XMISS,NREP,(BP(J),J=1,4),V1P,(ALPHAP(J),J=1,2),
     &   V2P,PIP
      PRINT*,'NUMBER OF SUBJECTS = ',NSUBJ
      PRINT 21,XMISS
21    FORMAT(1X,'MISSING DATA INDICATOR = ',F3.0)
      PRINT 22,(BP(J),J=1,4)
      PRINT*,'NUMBER OF REPETITIONS = ',NREP
22    FORMAT(1X,'PARAMETER B WEIGHTS ',4F6.3)
      PRINT 23,V1P
23    FORMAT(1X,'PARAMETER VALUE FOR V1 ',F8.3)
      PRINT 24,(ALPHAP(J),J=1,2)
24    FORMAT(1X,'PARAMETER VALUE FOR ALPHA ',2F6.3)
      PRINT 26,V2P
26    FORMAT(1X,'PARAMETER VALUE FOR V2 ',F8.3)
      PRINT 27,PIP
27    FORMAT(1X,'PARAMETER VALUE FOR PI ',F6.3)
C
C COMPUTING RSQUARED FROM THE PARAMETER VALUES
C PRSQ = 1 - VAR(Y/X1X2X3)/VAR(Y), USING PARAMETER VALUES.
C VAR(Y|X1X2X3) IS V1P.
C WE CALCULATE VAR(Y) CALLED VARYP IN THESE CALCULATIONS.
C PIP REFERS TO P(X1=1).
C
C
      BP32 = BP(3)**2
      P1 = (1-PIP)*(V1P + BP32*V2P)
      B322 = (BP(3) + BP(4))**2
      P2 = PIP * (V1P + B322*V2P)
      B33 = BP(2) + BP(3)*ALPHAP(2)
     &  + BP(4)*(ALPHAP(1)+ALPHAP(2))
      P3 = PIP * (1-PIP) * (B33**2)
      VARYP = P1 + P2 + P3
C     PRINT*,'VARYP = ',VARYP
C
      R2P = 1 - (V1P/VARYP)
C
      PRINT 313,R2P
313   FORMAT(1X,'PARAMETER R-SQUARED = ',F6.4/)
C
C COMPUTING DATA VALUES
      DO 1 I=1,NSUBJ
      II = I
      CALL GENDAT (TEMP,BP,ALPHAP,V1P,V2P,PIP,XMISS,II,PROBM)
```

```
      DO 213 J=1,4
213   DATA(I,J) = TEMP(J)
C DISPLAY THE MISSING PATTERN
      DO 555 J=1,4
555   IR(I,J)=1.
      DO 557 J=1,4
557   IF (TEMP(J) .EQ. XMISS)  IR(I,J)=0.
C     PRINT 78,(IR(I,J),J=1,4),(TEMP(J),J=1,4)
78    FORMAT(1X,'THE MISSING INDICATORS ',4I2,4F6.3)
1     CONTINUE
C
C SET NEW START VALUES
      DO 214 J=1,4
214   B(1,J) = 2.0
      DO 215 J = 1,2
215   ALPHA(1,J) = 2.0
      V1 = 4
      V2 = 12
      PI = .4
C
C  INITIALIZING VARIABLES HOLDING PARAMETER MEANS
      DO 2222 I=1,4
2222  BMEAN(I) = 0.
      DO 2223 I=1,2
2223  ALPHAM(I) = 0.
      V1M = 0.
      V2M = 0.
      PIM=0.
      R2M = 0.
C
C
C IBURN TO DISCARD FIRST 1000 ITERATIONS (BURN-IN PERIOD)
      IBURN = 1000
      KOUNT=0
C  HERE BEGINS THE BIG LOOP!!!!
      DO 2000 L = 1, NREP
1000  CONTINUE
      KOUNT = KOUNT +1

C  FILL IN THE MISSING DATA
C
        CALL GEN (B,V1,ALPHA,V2,PI,IR,DATA,NSUBJ)
C
C  PERFORM ANALYSES ON FILLED-IN DATA
C  SUBROUTINE REGY PREDICTS Y FROM X1, X2 , AND X1*X2
```

```
C  THE PROGRAM RETURNS THE PARAMETERS OF THE POSTERIOR
C  DISTRIBUTION OF BETA GIVEN V1 WHICH IS
C  N4(BHAT,XTXIY * V1)
C
      CALL REGY(NSUBJ,DATA,V1,XTXIY,BHAT)

C  GENERATE NEW VALUES FOR B(1,J),J=1,4), USING BHAT(4)
C  RETURNED FROM REGY
C
C  FIRST OBTAIN CHOLESKY FACTOR
      TOL = 100.0 * DMACH(4)
      CALL DCHFAC(4,XTXIY,4,TOL,IRANK,TY,4)
      CALL DRNMVN(1,4,TY,4,B,1)
      DO 1234 J=1,4
1234  B(1,J)=B(1,J)+BHAT(J)

C  SUBROUTINE REGX2 PREDICTS X2 FROM X1.
C  THE SUBROUTINE RETURNS ALPHA HAT AND XTYX2-1 * V2.
      CALL REGX2(NSUBJ,DATA,V2,XTXIX2,ALPHAT)
C
C  GENERATE NEW VALUES FOR ALPHA(1,J),J=1,2),USING ALPHAT
C
C  FIRST OBTAIN CHOLESKY FACTOR
      TOL = 100.0 * DMACH(4)
      CALL DCHFAC(2,XTXIX2,2,TOL,IRANK,TX2,2)
      CALL DRNMVN(1,2,TX2,2,ALPHA,1)
      DO 1235 J=1,2
1235  ALPHA(1,J)=ALPHA(1,J)+ALPHAT(J)

C  GENERATE A VALUE FOR V1 GIVEN B, M 0
C  COMPUTE Q1= 1/2 * SUM(Y-B'X)**2
      Q1=0.0
      DO 10 I2 = 1,NSUBJ
      PRED1=0.0
      DO 345 J=1,3
345   PRED1=PRED1 + DATA(I2,J+1)*B(1,J+1)
      PRED1=PRED1+B(1,1)
      ERROR1=DATA(I2,1)-PRED1
C     PRINT 366,DATA(I2,1),PRED1,ERROR1
366   FORMAT(1X,'DATA(I2,1),PRED1,ERROR1',1X,3F10.4)
        Q1 = Q1 + ERROR1**2
10    CONTINUE
       Q1 = .5*Q1
       DF=NSUBJ
       A=DF/2.0
```

```
       CALL DRNGAM (1,A,GAM)

       V1 = GAM(1)/Q1
       V1 = 1/V1
C      PRINT*,'V1 = ',V1

C  1/V1 OR PRECISION HAS A GAMMA DISTRIBUTION
C
C
C  GENERATE A VALUE FOR V2 GIVEN ALPHA,M,O
C  COMPUTE Q2= 1/2 * SUM(X2-ALPHA'X0X1)**2
      Q2=0.0
      DO 11 I2 = 1,NSUBJ
      PRED2=0.0
      PRED2=PRED2 + DATA(I2,2)*ALPHA(1,2)
      PRED2=PRED2+ALPHA(1,1)
      ERROR2=DATA(I2,3)-PRED2
C     PRINT 337,DATA(I2,3),PRED2,ERROR2
337   FORMAT(1X,'X2,PRED2,ERROR2 ',1X,3F10.4)
        Q2 = Q2 + ERROR2**2
11    CONTINUE

       Q2 = .5*Q2
      DF=NSUBJ
       A=DF/2.0
       CALL DRNGAM (1,A,GAM)
       V2 = GAM(1)/Q2
       V2 = 1/V2
C      PRINT*,'V2 = ',V2
C  1/V2 OR PRECISION HAS A GAMMA DISTRIBUTION

C  GENERATE NEW VALUE FOR PI
C     PI = BETA(SUMX1+1, NSUBJ-SUMX1+1)
C  CALCULATE THE SUM OF X1
      SUMX1 = 0.0
      DO 20 I3 = 1,NSUBJ
C     IF (DATA(I3,2) .EQ. XMISS) GO TO 19
C     PRINT*,'X1 = ',DATA(I3,2)
      SUMX1 = SUMX1 + DATA(I3,2)
19    CONTINUE
20    CONTINUE
C     PRINT*,'SUMX1 = ',SUMX1
      PIN = SUMX1 + 1
      QIN = NSUBJ - SUMX1 + 1
      CALL DRNBET(1,PIN,QIN,BETA(1))
```

```
      PI = BETA(1)
      IF (L .EQ. 1 .AND. KOUNT .LT. IBURN) GO TO 1000
C
C  PRINT OUT FILLED-IN DATA
C     PRINT*,'NEW FILLED-IN DATA SET'
      DO 335 I=1,NSUBJ
C     PRINT 336,(DATA(I,J),J=1,4)
336   FORMAT(1X,'Y X1 X2 X3 ',4F8.4)
335   CONTINUE
C
C     PRINT 1236,(B(1,J),J=1,4)
1236  FORMAT(1X,'NEW B-VALUES = ',4F8.4)
C     PRINT*,'NEW V1 = ',V1
C     PRINT*,'NEW V2 = ',V2
C
C     PRINT*,'NEW PI VALUE = ',PI
C     PRINT*
C
C  CALLING SUBROUTINE TO CALCULATE R-SQUARED
      CALL  RSQ(PI,ALPHA,B,V1,V2,R2)
C
C COPY R-SQUARED VALUE INTO STORE
      STORE(L) = R2
C
C     PRINT 1237,(ALPHA(1,J),J=1,2)
1237  FORMAT(1X,'NEW ALPHA-VALUES = ',2F8.4)
C     PRINT 800, (B(1,J),J=1,4),(ALPHA(1,J),J=1,2),V1,V2,R2
800   FORMAT(1X,4F10.4/1X,2F10.4/1X,F10.4/1X,F10.4,1X,F6.4//)
C
C ACCUMULATE SUMS OF RSQ,B,ALPHA,V1,V2,PI
      DO 522 I = 1,4
522   BMEAN(I)= BMEAN(I) + B(1,I)
      DO 523 I=1,2
523   ALPHAM(I) = ALPHAM(I)+ALPHA(1,I)
      V1M = V1M + V1
      V2M = V2M + V2
      PIM = PIM + PI
      R2M = R2M + R2
2000   CONTINUE
C
      DO 524 I=1,4
524   BMEAN(I) = BMEAN(I)/NREP
      DO 525 I=1,2
525   ALPHAM(I) = ALPHAM(I)/NREP
      V1M = V1M/NREP
```

```
      V2M = V2M/NREP
      PIM = PIM/NREP
      R2M = R2M/NREP
      PRINT*,'PARAMETER MEANS'
      PRINT*,'SEED = ',ISEED
      PRINT 526,(BMEAN(I),I=1,4)
526   FORMAT(1X,'B-MEANS = ',4F8.4)
      PRINT 527,(ALPHAM(I),I=1,2)
527   FORMAT(1X,'ALPHA-MEANS = ',2F8.4)
      PRINT 528,V1M,V2M
528   FORMAT (1X,'V1 AND V2 MEANS = ',2F10.4)
      PRINT 529,PIM
529   FORMAT(1X,'PI MEAN = ',F8.4)
      PRINT 531,R2M
531   FORMAT(1X,'MEAN OF R-SQUARED = ',F8.4)
      CALL SORT(STORE,NREP)
      CALL HDR (STORE,.90,NREP)
      STOP
      END

C
*************************************************************************
C  THE SUBROUTINE TO CREATE THE DATA SET AND MAKE SOME
C  MISSINC (MCAR)
C
*************************************************************************
C
      SUBROUTINE GENDAT (TEMP,BP,ALPHAP,V1P,V2P,PIP,XMISS,II,PROBM)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION BP(4),ALPHAP(2),Z(1),U(1)
      DIMENSION DATA(100,4),XTXIY(4,4),TY(4,4),B(1,4),GAM(1),BETA(1)
      DIMENSION XTXIX2(2,2),TX2(2,2),ALPHA(1,2)
      DIMENSION IR(100,4)
      DIMENSION BHAT(4),ALPHAT(2)
      DIMENSION TEMP(4)
      DIMENSION SAVE(4)
C
C  GENERATE X1 = TEMP(2)
      TEMP(2) = 0.0
      CALL DRNUN(1,U(1))
      IF (U(1) .LE. PIP) TEMP(2) = 1.0
C     PRINT*,'U(1) = ',U(1),'TEMP(2) = ',TEMP(2)
C  GENERATE X2 = TEMP(3) AND X1X2 = TEMP(4)
      CALL DRNNOR(1,Z(1))
      X2MEAN = ALPHAP(1) + ALPHAP(2)*TEMP(2)
```

```
      TEMP(3) = X2MEAN +DSQRT(V2P)*Z(1)

      TEMP(4) = TEMP(2)*TEMP(3)
C  GENERATE Y = TEMP(1)
      YMEAN=BP(1)+BP(2)*TEMP(2)+BP(3)*TEMP(3)+BP(4)*TEMP(4)
      CALL DRNNOR(1,Z(1))
      TEMP(1) = YMEAN + DSQRT(V1P)*Z(1)
C     PRINT*,'COMPLETE DATA ',TEMP
      DO 400 I=1,3
400   SAVE(I) = TEMP(I)
C  GENERATE MISSING DATA (MCAR; PROB. MISSING = PROBM)
      DO 3 I2 = 1,3
       CALL DRNUN(1,U(1))
       IF (U(1) .LE. PROBM) TEMP(I2) = XMISS
C  CHECK THAT NMISS LT 3 (SO NOT ALL 4 VARIABLES MISSING!)
3     CONTINUE
      NMISS = 0
      DO 500 I=1,3
      IF (TEMP(I) .EQ. XMISS) NMISS = NMISS + 1
500   CONTINUE
      IF (NMISS - 3) 599,600,600
600   TEMP(2) = SAVE(2)
599   CONTINUE
      IF (TEMP(2) .EQ. XMISS .OR. TEMP(3) .EQ. XMISS)
     & TEMP(4) = XMISS
C     PRINT 45,TEMP(1),TEMP(2),TEMP(3),TEMP(4)
45    FORMAT(1X,'Y X1 X2 X1X2 ',4F6.2)
C  WRITE GENERATED DATA (SOME MISSING) TO FILE
      WRITE (10,450) TEMP
450   FORMAT(1X,4F10.4)

C
      RETURN
      END
C
C
**********************************************************************
C  THE SUBROUTINE TO IDENTIFY WHICH VARIABLES ARE MISSING
C     AND CALL THE APPROPRIATE SUBROUTINE TO FILL THEM IN
C
**********************************************************************
C
      SUBROUTINE GEN(B,V1,ALPHA,V2,PI,IR,DATA,NSUBJ)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION DATA(100,4),XTXIY(4,4),TY(4,4),B(1,4),GAM(1),BETA(1)
```

```
      DIMENSION XTXIX2(2,2),TX2(2,2),ALPHA(1,2)
      DIMENSION IR(100,4)
      DIMENSION BHAT(4),ALPHAT(2)
      DIMENSION TEMP(4)
C
C     PRINT*,'SUBROUTINE GEN CALLED'
C   REMOVING SUBJECTS WITH COMPLETE DATA
      DO 1 III = 1,NSUBJ
C       SUM = 0.0
C      DO 2 J = 1,4
C         SUM = SUM + IR(J)
C         IF (SUM .EQ. 4.0) GO TO 1
C  COPYING SUBJECT'S DATA VALUES INTO TEMP VECTOR
      DO 3 J = 1,4
3     TEMP(J) = DATA(III,J)
C  IDENTIFYING EACH MISSING DATA PATTERN AND CALLING
C   THE APPROPRIATE SUBROUTINE
C **** MISSING Y
      IF (IR(III,1) .EQ. 0 .AND. IR(III,2) .EQ. 1 .AND.
     &  IR(III,3) .EQ. 1)
     & CALL GEN011(TEMP,B,V1,ALPHA,V2,PI)
C **** MISSING X1
      IF (IR(III,1) .EQ. 1 .AND. IR(III,2) .EQ. 0 .AND.
     &  IR(III,3) .EQ. 1)
     &  CALL GEN101(TEMP,B,V1,ALPHA,V2,PI)
C **** MISSING X2
      IF (IR(III,1) .EQ. 1 .AND. IR(III,2) .EQ. 1 .AND.
     &  IR(III,3) .EQ. 0)
     &  CALL GEN110(TEMP,B,V1,ALPHA,V2,PI)
C **** MISSING Y AND X1
      IF (IR(III,1) .EQ. 0 .AND. IR(III,2) .EQ. 0 .AND.
     &  IR(III,3) .EQ. 1)
     &  CALL GEN001(TEMP,B,V1,ALPHA,V2,PI)
C **** MISSING Y AND X2
      IF (IR(III,1) .EQ. 0 .AND. IR(III,2) .EQ. 1 .AND.
     &  IR(III,3) .EQ. 0)
     &  CALL GEN010(TEMP,B,V1,ALPHA,V2,PI)
C **** MISSING X1 AND X2
      IF (IR(III,1) .EQ. 1 .AND. IR(III,2) .EQ. 0 .AND.
     &  IR(III,3) .EQ. 0)
     &  CALL GEN100(TEMP,B,V1,ALPHA,V2,PI)


C   COPYING IN THE ESTIMATED VALUES FOR MISSING DATA
       DO 400 J = 1,4
```

```
400     DATA(III,J) = TEMP(J)

1       CONTINUE
        RETURN
        END
C
C ***************************************************************
C  THE SUBROUTINE TO FILL IN MISSING Y
C ***************************************************************
C
        SUBROUTINE GEN011(TEMP,B,V1,ALPHA,V2,PI)
        IMPLICIT REAL*8 (A-H,O-Z)
        DIMENSION DATA(100,4),XTXIY(4,4),TY(4,4),B(1,4),GAM(1),BETA(1)
        DIMENSION XTXIX2(2,2),TX2(2,2),ALPHA(1,2)
        DIMENSION IR(100,4)
        DIMENSION BHAT(4),ALPHAT(2)
        DIMENSION TEMP(4)
        DIMENSION Z(1)

C       PRINT*,'GEN011 CALLED'

C SAMPLING Y FROM NORMAL DISTRIBUTION WITH MEAN=B0+B1X1 ETC.
C  AND VARIANCE V1
        CALL DRNNOR(1,Z(1))
        SDEV = DSQRT(V1)
        YMEAN = B(1,1) + B(1,2)*TEMP(2) +B(1,3)*TEMP(3)
       &  +B(1,4)*TEMP(4)
        TEMP(1) = Z(1)*SDEV + YMEAN
C       PRINT 20, TEMP(1)
20      FORMAT(1X,'FROM GEN011,NEW Y VALUE=',F6.3)
        RETURN
        END
C
C ***************************************************************
C  THE SUBROUTINE TO FILL IN MISSING X1
C ***************************************************************
C
        SUBROUTINE GEN101(TEMP,B,V1,ALPHA,V2,PI)
        IMPLICIT REAL*8 (A-H,O-Z)
        DIMENSION DATA(100,4),XTXIY(4,4),TY(4,4),B(1,4),GAM(1),BETA(1)
        DIMENSION XTXIX2(2,2),TX2(2,2),ALPHA(1,2)
        DIMENSION U(1)
        DIMENSION IR(100,4)
        DIMENSION BHAT(4),ALPHAT(2)
        DIMENSION TEMP(4)
```

```
C     PRINT*,'GEN101 CALLED'

C COMPUTING P(X2|X1=1) CALLED P05
      TEMP(2) = 1.0
      TEMP(4) = TEMP(2) * TEMP(3)
      P01 = DSQRT(1/V2)
      P02 = -1/(2*V2)
      P03 = (TEMP(3) - ALPHA(1,1) - ALPHA(1,2)*TEMP(2))**2
      P04 = DEXP(P02*P03)
      P05 = P01*P04
C COMPUTING THE CONDITIONAL PROBABILITY (Y|X1=1,X2) CALLED P10
      P06 = DSQRT(1/V1)
      P07 = -1/(2*V1)
      P08 = (TEMP(1)-B(1,1)-B(1,2)*TEMP(2)-B(1,3)
     &    *TEMP(3)-B(1,4)*TEMP(4))**2
      P09 = DEXP(P07*P08)
      P10 = P06 * P09
C  COMPUTING THE NUMERATOR
      XNUM = P05 * P10 * PI
C  THE SAME FOR X1 = 0 (P01, P02, P06 AND P07 DON'T CHANGE)
C  CALCULATING P(X2|X1=0) CALLED P050
      TEMP(2) = 0.0
      TEMP(4) = 0
      P030 = (TEMP(3) - ALPHA(1,1))**2
      P040 = DEXP(P02*P030)
      P050 = P01*P040
C CALCULATING (Y|X1=0,X2) CALLED P100
      P080 = (TEMP(1)-B(1,1)-B(1,3)*TEMP(3))**2
      P090 = DEXP(P07*P080)
      P100 = P06 * P090
C COMPUTE DENOMINATOR
C FOR X1 = 0
      DENOM1 = XNUM
      DENOM0 = P050 * P100 * PI
      DENOM = DENOM1 + DENOM0
C GETTING THE FINAL PROBABILITY
      PFINAL = XNUM/DENOM
C GENERATING THE X1 VALUE
      CALL DRNUN (1,U(1))
      IF (U(1) .LE. PFINAL) TEMP(2) = 1
      IF (U(1) .GT. PFINAL) TEMP(2) = 0
C  RESET X1X2 VALUE
      TEMP(4) = TEMP(2) * TEMP(3)
```

```
C  PRINT VALUES AND RETURN
C      PRINT 33,TEMP(2)
33     FORMAT (1X,'FROM GEN101,NEW X1 VALUE=',F4.2)
       RETURN
       END
C
C ****************************************************************
C THE SUBROUTINE TO FILL IN MISSING X2
C ****************************************************************
C
       SUBROUTINE GEN110(TEMP,B,V1,ALPHA,V2,PI)
       IMPLICIT REAL*8 (A-H,O-Z)
       DIMENSION DATA(100,4),XTXIY(4,4),TY(4,4),B(1,4),GAM(1),BETA(1)
       DIMENSION XTXIX2(2,2),TX2(2,2),ALPHA(1,2)
       DIMENSION U(1)
       DIMENSION IR(100,4)
       DIMENSION BHAT(4),ALPHAT(2)
       DIMENSION TEMP(4)
       DIMENSION Z(1)

C      PRINT*,'GEN110 CALLED'

C  FIRST GET THE COMPONENT PARTS
C  COMPUTE A1 I.E. E(X2|X1)
       A1 = ALPHA(1,1) + ALPHA(1,2) * TEMP(2)
C  COMPUTE B1 I.E. E(Y|X1)
       B1 = B(1,1) + B(1,2)*TEMP(2) + A1*(B(1,3)+B(1,4)*TEMP(2))
C  C1 IS DEFINED AS VAR(X2|X1)
       C1 = V2
C  COMPUTE D1 = COV(Y,X2|X1)
       D11 = B(1,4)*TEMP(2)
       D12 = B(1,3)+D11
       D1 =  D12*V2
C  COMPUTE E1 = VAR(Y|X1)
       E11 = B(1,3)+B(1,4)*TEMP(2)
       E12 = E11**2
       E1 = V1 + V2 * E12
C  NEXT GENERATE THE MISSING X2 AND WRITE IT TO TEMP(3)
       CALL DRNNOR(1,Z(1))
       VAR1 = D1**2/(C1*E1)
       VAR2 =  1 - VAR1
       VAR3 = C1 * VAR2
       SDEV = DSQRT(VAR3)
       X2MEAN = A1 - (D1/E1)*B1 + (D1/E1)*TEMP(1)
       TEMP(3) = Z(1)*SDEV + X2MEAN
```

```
      TEMP(4) = TEMP(3) * TEMP(2)
C     PRINT 25,TEMP(3)
25    FORMAT(1X,'FROM GEN110,NEW X2 VALUE=',1X,F6.3)
      RETURN
      END
C
C *********************************************************************
C THE SUBROUTINE TO FILL IN MISSING Y AND X2
C *********************************************************************
C
      SUBROUTINE GEN010(TEMP,B,V1,ALPHA,V2,PI)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION DATA(100,4),XTXIY(4,4),TY(4,4),B(1,4),GAM(1),BETA(1)
      DIMENSION XTXIX2(2,2),TX2(2,2),ALPHA(1,2)
      DIMENSION U(1)
      DIMENSION IR(100,4)
      DIMENSION BHAT(4),ALPHAT(2)
      DIMENSION TEMP(4)
      DIMENSION Z(1)
C
C     PRINT*,'GEN010 CALLED'

C  FIRST GENERATE THE X2 GIVEN X1
      CALL DRNNOR(1,Z(1))
      SDEV = DSQRT(V2)
      X2MEAN = ALPHA(1,1) + ALPHA(1,2) * TEMP(2)
      TEMP(3) = Z(1)*SDEV + X2MEAN
      TEMP(4) = TEMP(2) * TEMP(3)

C
C NOW GENERATE Y GIVEN X1 AND X2
C SAMPLING Y FROM NORMAL DISTRIBUTION WITH MEAN=B0+B1X1 ETC.
C  AND VARIANCE V1
      CALL DRNNOR(1,Z(1))
      SDEV = DSQRT(V1)
      YMEAN = B(1,1) + B(1,2)*TEMP(2) +B(1,3)*TEMP(3)
     &  +B(1,4)*TEMP(4)
      TEMP(1) = Z(1)*SDEV + YMEAN
C     PRINT 23,TEMP(1),TEMP(3)
23    FORMAT(1X,'FROM GEN010,NEW Y = ',F6.2,1X,'NEW X2 = ',F6.2)
300   CONTINUE
      RETURN
      END
C
C *********************************************************************
```

```
C THE SUBROUTINE TO FILL IN MISSING X1 AND X2
C ***************************************************************************
C
      SUBROUTINE GEN100(TEMP,B,V1,ALPHA,V2,PI)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION DATA(100,4),XTXIY(4,4),TY(4,4),B(1,4),GAM(1),BETA(1)
      DIMENSION XTXIX2(2,2),TX2(2,2),ALPHA(1,2)
      DIMENSION U(1)
      DIMENSION IR(100,4)
      DIMENSION BHAT(4),ALPHAT(2)
      DIMENSION TEMP(4)
      DIMENSION Z(1)
C
C     PRINT*,'GEN100 CALLED'
C FIRST FILL IN X1 FROM P(X1|Y)
C GET B1 = E(Y|X1) AND E1 = VAR(Y|X1)
C LEAVE OUT CONSTANT (1/2PI)**1/2  WHERE PI=3.14.....
C
C FOR X1 = 1
      TEMP(2) = 1.
      B11 = ALPHA(1,1) + ALPHA(1,2)*TEMP(2)
      B12 = B(1,3) + B(1,4)*TEMP(2)
      B13 = B11*B12
      B14 = B(1,1) + B(1,2)*TEMP(2)
      B1 = B13 +B14
C
      E11 = B(1,3)+B(1,4)*TEMP(2)
      E12 = E11**2
      E1 = V1 + V2 * E12
C
      XNUM11 = 1/DSQRT(E1)
      XNUM12 = -1/(2*E1)
      XNUM13 = (TEMP(1) - B1)**2
      XNUM14 = DEXP(XNUM12*XNUM13)
      XNUM15 = XNUM11 * XNUM14
C
      XNUM1 = XNUM15 * PI
C
C  FOR X1 = 0
      TEMP(2) = 0
      B0 = B(1,1) + (ALPHA(1,1) *B(1,3))
C
      E0 = V1 + V2*((B(1,3)**2))
C
      XNUM01 = 1/DSQRT(E0)
```

```
      XNUM02 = -1/(2*E0)
      XNUM03 = (TEMP(1) - B0)**2
      XNUM04 = DEXP(XNUM02*XNUM03)
      XNUM05 = XNUM01 * XNUM04
      XNUM0 =  XNUM05 * (1-PI)
C
      DENOM = XNUM1 + XNUM0
      PFINAL = XNUM1/DENOM
C
C GENERATING THE X1 VALUE
      CALL DRNUN (1,U(1))
      IF (U(1) .LE. PFINAL) TEMP(2) = 1
      IF (U(1) .GT. PFINAL) TEMP(2) = 0
C     PRINT 36, PFINAL,U(1),TEMP(2)
36    FORMAT(1X,'PFINAL= ',F6.4,1X,'U(1)= ',F6.4,1X,
     & 'TEMP(2)= ',F4.2)
C
C  NEXT FILL IN X2 GIVEN X1
C
      CALL DRNNOR(1,Z(1))
      SDEV = DSQRT(V2)
      X2MEAN = ALPHA(1,1) + ALPHA(1,2)*TEMP(2)
      TEMP(3) = Z(1)*SDEV + X2MEAN
      TEMP(4) = TEMP(2) * TEMP(3)
C     PRINT 32,TEMP(2),TEMP(3)
32    FORMAT(1X,'FROM GEN100, NEW X1 = ',F4.2,1X,'NEW X2= ',F6.2)
C
300   CONTINUE
      RETURN
      END
C
C
C ********************************************************************
C THE SUBROUTINE TO FILL IN MISSING X1 AND Y
C ********************************************************************
C
      SUBROUTINE GEN001(TEMP,B,V1,ALPHA,V2,PI)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION DATA(100,4),XTXIY(4,4),TY(4,4),B(1,4),GAM(1),BETA(1)
      DIMENSION XTXIX2(2,2),TX2(2,2),ALPHA(1,2)
      DIMENSION IR(100,4)
      DIMENSION BHAT(4),ALPHAT(2)
      DIMENSION TEMP(4)
      DIMENSION U(1)
      DIMENSION Z(1)
```

```
C
C
C     PRINT*,'GEN001 CALLED'
C FIRST FILL IN X1
C P(X1|X2) = P(X1,X2)/P(X2)
C P(X1,X2) = P(X2|X1)*P(X1)
C P(X2) IS GOTTEN BY ADDING TOGETHER P(X2|X1=1) AND P(X2|X1=0)
C P(X1) IS CANCELLED FROM BOTH NUMERATOR AND DENOMINATOR
C
C COMPUTING P(X2|X1=1) CALLED P05
      TEMP(2) = 1.0
      P01 = DSQRT(1/V2)
      P02 = -1/(2*V2)
      P03 = TEMP(3) - ALPHA(1,1) -ALPHA(1,2)*TEMP(2)
      P03 = P03**2
      P04 = DEXP(P02*P03)
      P05 = P01*P04
C COMPUTING P(X2|X1=0) CALLED P10
      TEMP(2) = 0.0
      P06 = DSQRT(1/V2)
      P07 = -1/(2*V2)
      P08 = TEMP(3) - ALPHA(1,1)
      P08 = P08**2
      P09 = DEXP(P07*P08)
      P10 = P06*P09
C   COMPUTE THE PROBABILITY (X2|X1), CANCELLING P(X1) FROM ALL TERMS
      XNUM = P05
      DENOM = P05 + P10
      PROB = XNUM/DENOM
C
C GENERATING THE X1 VALUE AND WRITE INTO TEMP(2)
      CALL DRNUN (1,U(1))
      IF (U(1) .LE. PROB) TEMP(2) = 1.
      IF (U(1) .GT. PROB) TEMP(2) = 0.
C
      TEMP(4) = TEMP(2) * TEMP(3)
C
C
C NEXT FILL IN Y USING X1 AND X2
C SAMPLING Y FROM NORMAL DISTRIBUTION WITH MEAN=B0+B1X1 ETC.
C  AND VARIANCE V1
      CALL DRNNOR(1,Z(1))
      SDEV = DSQRT(V1)
      YMEAN = B(1,1) + B(1,2)*TEMP(2) +B(1,3)*TEMP(3)
     &  +B(1,4)*TEMP(4)
```

```
      TEMP(1) = Z(1)*SDEV + YMEAN

C     PRINT 36,TEMP(1),TEMP(2)
36    FORMAT(1X,'FROM GEN001,NEW Y VALUE = ',F6.2,1X,
     & 'NEW X1 VALUE = ',F4.2)
      RETURN
      END
C
C
*************************************************************************
C THE SUBROUTINE TO PREDICT Y FROM X1 AND X2 USING THE FILLED-IN
C DATA, AND RETURNS BHAT(4) IS USED IN GENERATING THE NEW BETA
C PARAMETERS
C
*************************************************************************
C
       SUBROUTINE REGY(NSUBJ,DATA,V1,XTXIY,BHAT)
       IMPLICIT REAL*8(A-H,O-Z)
       DIMENSION DATA(100,4), XTXIY(4,4), XTY(4)
       DIMENSION TEMP(4), BHAT(4)

C      PRINT*,'GEN REGY CALLED'

C   INITIALIZE XTY AND XTXIY MATRICES WITH ZEROES
       DO 100 I = 1,4
        XTY(I) = 0.0
100   CONTINUE
       DO 200 I = 1,4
       DO 210 J = 1,4
       XTXIY(I,J) = 0.0
210    CONTINUE
200    CONTINUE
C
C  CREATE VECTOR TEMP CONTAINING: 1 X1 X2 X1*X2
       TEMP(1)  = 1.0
       DO 1000 IJK = 1,NSUBJ
       DO 20 J=2,4
20    TEMP(J) = DATA(IJK,J)
C
       DO 30 J = 1,4
30    XTY(J) = XTY(J) + TEMP(J) * DATA(IJK,1)
       DO 40 J = 1,4
       DO 45 K = 1,4
       XTXIY(J,K) = XTXIY (J,K) + TEMP(J) * TEMP(K)
45     CONTINUE
```

```
40      CONTINUE
1000    CONTINUE
C
C       DO 8888 II=1,4
C       PRINT 8889,(XTXIY(II,K),K=1,4)
8889    FORMAT(1X,4F10.4)
8888    CONTINUE
C      INVERT XTXIY
        CALL DLINDS(4,XTXIY,4,XTXIY,4)
C
C    COMPUTE BHAT
        DO 50 I=1,4
C    INITIALIZE BHAT AS ZEROES
        BHAT(I) = 0.0
        DO 50 J=1,4
50     BHAT(I) = BHAT(I) + XTXIY(I,J)*XTY(J)
C    COMPUTE THE VARIANCE/COVAR MATRIX FOR BHAT
        DO 60 I=1,4
        DO 70 J = 1,4
        XTXIY(I,J) = XTXIY(I,J)*V1
70      CONTINUE
60      CONTINUE
        RETURN
        END
C
C
*************************************************************************
C THE SUBROUTINE TO PREDICT X2 FROM X1 USING THE FILLED-IN
C DATA, AND COMPUTE ALPHAT WHICH IS USED IN OBTAINING THE
C NEW ALPHA PARAMETERS
C
*************************************************************************
C
        SUBROUTINE REGX2(NSUBJ,DATA,V2,XTXIX2,ALPHAT)
        IMPLICIT REAL*8(A-H,O-Z)
        DIMENSION DATA(100,4), XTXIX2(2,2), XTX2(2)
        DIMENSION TEMP(4), ALPHAT(2)

C       PRINT*,'REGX2 CALLED'

C   INITIALIZE EVERYTHING
        DO 90 II= 1,2
90      XTX2(II) = 0.0
        DO 100 I=1,2
        DO 100 J=1,2
```

```
100      XTXIX2(I,J) = 0.0
C  CREATE A VECTOR TEMP WITH VALUES 1 X1 X2
       TEMP(1) = 1.0
       DO 1000 IJK = 1,NSUBJ
       DO 20 J = 2,3
20     TEMP(J) = DATA(IJK,J)
C CREATE XTX2 VECTOR
       DO 25 J = 1,2
25     XTX2(J) = XTX2(J) + TEMP(J) * TEMP(3)
       DO 30 J = 1,2
       DO 30 K = 1,2
30     XTXIX2(J,K) = XTXIX2(J,K) + TEMP(J) * TEMP(K)
1000   CONTINUE
C      DO 8888 II=1,2
C      PRINT 8889, (XTXIX2(II,K),K=1,2)
8889   FORMAT(1X,2F10.4)
8888   CONTINUE
C INVERT XTXIX2
       CALL DLINDS(2,XTXIX2,2,XTXIX2,2)
C  COMPUTE ALPHAT (FIRST INITIALIZE AS ZEROES)
       DO 50 I = 1,2
50       ALPHAT(I) = 0.0
       DO 60 I= 1,2
       DO 60 J = 1,2
60     ALPHAT(I) = ALPHAT(I) + XTXIX2(I,J) * XTX2(J)
C   COMPUTE VAR/COVAR MATRIX FOR ALPHAT
      DO 64 I = 1,2
      DO 65 J=I,2
       XTXIX2(I,J) = XTXIX2(I,J) * V2
65     CONTINUE
64     CONTINUE
       RETURN
       END
C
C
*************************************************************************
C  THE SUBROUTINE TO CALCULATE R-SQUARED ON THE FILLED-IN DATA SET
C
*************************************************************************
C
      SUBROUTINE RSQ(PI,ALPHA,B,V1,V2,R2)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION DATA(100,4),XTXIY(4,4),B(1,4)
      DIMENSION XTXIX2(2,2),TX2(2,2),ALPHA(1,2)
      DIMENSION BHAT(4),ALPHAT(2)
```

```
      DIMENSION TEMP(4)
C
C     PRINT*,'RSQ CALLED'
C
C RSQARED = 1 - VAR(Y/X1X2X3)/VAR(Y).
C VAR(Y|X1X2X3) IS V1.
C WE CALCULATE VAR(Y) CALLED VARY IN THIS SUBROUTINE.
C PI REFERS TO P(X1=1).
C
C
      B32 = B(1,3)**2
      P1 = (1-PI)*(V1 + B32*V2)
      B322 = (B(1,3) + B(1,4))**2
      P2 = PI * (V1 + B322*V2)
      B33 = B(1,2) + B(1,3)*ALPHA(1,2)
     &    + B(1,4)*(ALPHA(1,1)+ALPHA(1,2))
      P3 = PI * (1-PI) * (B33**2)
      VARY = P1 + P2 + P3
C
      R2 = 1 - (V1/VARY)
C     WRITE (10,323) R2
323   FORMAT(1X,F10.4)
C
      RETURN
      END
*****************************************************************
** THE SUBROUTINE TO SORT THE R-SQUARES
*****************************************************************
      SUBROUTINE SORT (STORE,MM)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION STORE (5000)
C
      ISTOP = MM-1
      DO 30 I=1,ISTOP
      JSTOP = MM-I
      DO 20 J=1,JSTOP
      IF (STORE(J) .LT. STORE(J+1)) GO TO 20
      TEMP = STORE(J)
      STORE(J) = STORE(J+1)
      STORE(J+1) = TEMP
20    CONTINUE
30    CONTINUE
      RETURN
      END
```

```
*************************************************************************

** THE SUBROUTINE TO FIND THE HIGHEST DENSITY REGION
*************************************************************************
*
      SUBROUTINE HDR (STORE,XLEVEL,MM)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION STORE (5000),USE(5000)
C
      ILOW = (1. - XLEVEL)*MM - 1
      IUP = XLEVEL*MM - 1
C
      RHOL = STORE(1)
      RHOU = STORE(IUP+1)
      XMIN = RHOU-RHOL
      DO 2 I=2,ILOW
      J= I + IUP
      DIFF = STORE(J) -STORE(I)
      IF (DIFF .GT. XMIN) GO TO 2
      XMIN = DIFF
      RHOL = STORE(I)
      RHOU = STORE(J)
2     CONTINUE
      PRINT 3333, MM,RHOL,RHOU
3333  FORMAT(1X,'MM =',I4,1X,'RHOL =',F6.4,1X,'RHOU = ',F6.4)
      RETURN
      END
//GO.SYSIN DD *
9591787
.10
 30  -9   5000 1. 1.  1.  1.  16.5 1. 1. 5. .5
//GO.FT10F001 DD UNIT=DISK,VOL=SER=SCR002,
// SPACE=(TRK,(5,1),RLSE),DCB=(LRECL=41,BLKSIZE=4100,RECFM=FB),
// DISP=(NEW,CATLG,DELETE),DSN=WYL.GC.SAR.DATA
```

Appendix 3: FORTRAN Program for MAR Data

```
// JOB TIME=5
// EXEC FORTVCLG,IMSLIF='SYS2.IMSL.SPIF',IMSL='SYS2.IMSL'
//*MAIN LINES=8
//SYSIN DD *

C THIS IS THE MAR PROGRAM.

      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION DATA(100,4),XTXIY(4,4),TY(4,4),B(1,4),GAM(1),BETA(1)
      DIMENSION XTXIX2(2,2),TX2(2,2),ALPHA(1,2)
      DIMENSION IR(100,4)
      DIMENSION STORE(5000),USE(5000)
      DIMENSION BHAT(4),ALPHAT(2)
      DIMENSION BMEAN(4),ALPHAM(2),BP(4),ALPHAP(2)
      DIMENSION TEMP(4)
      DIMENSION Z(1), U(1)

C DATA(I,1)=Y
C DATA(I,2)=X1=BINARY VARIABLE
C DATA(I,3)=X2=CONTINUOUS VARIABLE
C DATA(I,4)=X1*X2
C R = MATRIX OF RESPONSE PATTERNS
C XTXIY = 4 BY 4 SUMSQ-CROSSPROD MATRIX FOR PREDICTING Y
C        FROM X0 X1 X2 X1*X2.
C XTXIX2 2 BY 2 FOR PREDICTING X2 FROM X1.
C TY=CHOLESKY FACTOR FOR GENERATING 4 BY 1 VECTOR OF B WEIGHTS.
C TX2=CHOLESKY FACTOR FOR GENERATING 2 BY 1 VECTOR OF ALPHA WEIGHTS.
C BHAT = LS REG WEIGHTS PREDICTING Y FROM ALL X'S BASED ON
C        FILLED IN DATA SET.
C ALPHAT=LS REG WEIGHTS PREDICTING X2 FROM X1.
C BP, ALPHAP,V1P ETC. ARE PARAMETER VALUES READ IN TO
C    GENERATE THE DATA SET WHICH IS THEN MADE PARTIALLY MISSING.
C
       PRINT*,'Y MAR, X2 MCAR, X1 ALWAYS PRESENT'

       READ*,ISEED
       CALL RNSET (ISEED)
C
       READ*,PY1M
```

```
      PRINT 444,PY1M
444   FORMAT(1X,'PROB.Y MISSING, GIVEN X1=1 =',F6.4)

      READ*,PY0M
      PRINT 434,PY0M
434   FORMAT(1X,'PROB.Y MISSING, GIVEN X1=0 =',F6.4)

      READ*,PROBX2M
      PRINT 446,PROBX2M
446   FORMAT(1X,'PROB. X2 MISSING =',F6.4)
C
C READING IN NUMBER OF SUBJECTS, MISSING DATA INDICATOR, NUMBER OF
C  REPETITIONS AND PARAMETER VALUES FROM FIRST LINE OF DATA
      READ*,NSUBJ,XMISS,NREP,(BP(J),J=1,4),V1P,(ALPHAP(J),J=1,2),
     &   V2P,PIP
      PRINT*,'NUMBER OF SUBJECTS = ',NSUBJ
      PRINT 21,XMISS
21    FORMAT(1X,'MISSING DATA INDICATOR = ',F3.0)
      PRINT 22,(BP(J),J=1,4)
      PRINT*,'NUMBER OF REPETITIONS = ',NREP
22    FORMAT(1X,'PARAMETER B WEIGHTS ',4F6.3)
      PRINT 23,V1P
23    FORMAT(1X,'PARAMETER VALUE FOR V1 ',F8.3)
      PRINT 24,(ALPHAP(J),J=1,2)
24    FORMAT(1X,'PARAMETER VALUE FOR ALPHA ',2F6.3)
      PRINT 26,V2P
26    FORMAT(1X,'PARAMETER VALUE FOR V2 ',F8.3)
      PRINT 27,PIP
27    FORMAT(1X,'PARAMETER VALUE FOR PI ',F6.3)
C
C COMPUTING RSQUARED FROM THE PARAMETER VALUES
C PRSQ = 1 - VAR(Y/X1X2X3)/VAR(Y), USING PARAMETER VALUES.
C VAR(Y|X1X2X3) IS V1P.
C WE CALCULATE VAR(Y) CALLED VARYP IN THESE CALCULATIONS.
C PIP REFERS TO P(X1=1).
C
C
      BP32 = BP(3)**2
      P1 = (1-PIP)*(V1P + BP32*V2P)
      B322 = (BP(3) + BP(4))**2
      P2 = PIP * (V1P + B322*V2P)
      B33 = BP(2) + BP(3)*ALPHAP(2)
     &   + BP(4)*(ALPHAP(1)+ALPHAP(2))
      P3 = PIP * (1-PIP) * (B33**2)
      VARYP = P1 + P2 + P3
```

```
C     PRINT*,'VARYP = ',VARYP
C
      R2P = 1 - (V1P/VARYP)
C
      PRINT 313,R2P
313   FORMAT(1X,'PARAMETER R-SQUARED = ',F6.4/)
C
C COMPUTING DATA VALUES
      DO 1 I=1,NSUBJ
      II = I
      CALL GENDAT (TEMP,BP,ALPHAP,V1P,V2P,PIP,XMISS,II,PY0M,PY1M,
     & PROBX2M)
      DO 213 J=1,4
213   DATA(I,J) = TEMP(J)
C DISPLAY THE MISSING PATTERN
      DO 555 J=1,4
555   IR(I,J)=1.
      DO 557 J=1,4
557   IF (TEMP(J) .EQ. XMISS)  IR(I,J)=0.
C     PRINT 78,(IR(I,J),J=1,4),(TEMP(J),J=1,4)
78    FORMAT(1X,'THE MISSING INDICATORS ',4I2,4F6.3)
1     CONTINUE
C
C SET NEW START VALUES
      DO 214 J=1,4
214   B(1,J) = 2.0
      DO 215 J = 1,2
215   ALPHA(1,J) = 2.0
      V1 = 4
      V2 = 12
      PI = .4
C
C  INITIALIZING VARIABLES HOLDING PARAMETER MEANS
      DO 2222 I=1,4
2222  BMEAN(I) = 0.
      DO 2223 I=1,2
2223  ALPHAM(I) = 0.
      V1M = 0.
      V2M = 0.
      PIM=0.
      R2M = 0.
C
C
C IBURN TO DISCARD FIRST 1000 ITERATIONS (BURN-IN PERIOD)
      IBURN = 1000
```

```
      KOUNT=0
C  HERE BEGINS THE BIG LOOP!!!!
      DO 2000 L = 1, NREP
1000  CONTINUE
      KOUNT = KOUNT +1

C  FILL IN THE MISSING DATA
C
        CALL GEN (B,V1,ALPHA,V2,PI,IR,DATA,NSUBJ)
C
C  PERFORM ANALYSES ON FILLED-IN DATA
C  SUBROUTINE REGY PREDICTS Y FROM X1, X2 , AND X1*X2
C  THE PROGRAM RETURNS THE PARAMETERS OF THE POSTERIOR
C  DISTRIBUTION OF BETA GIVEN V1 WHICH IS
C  N4(BHAT,XTXIY * V1)
C
      CALL REGY(NSUBJ,DATA,V1,XTXIY,BHAT)

C  GENERATE NEW VALUES FOR B(1,J),J=1,4), USING BHAT(4)
C  RETURNED FROM REGY
C
C  FIRST OBTAIN CHOLESKY FACTOR
      TOL = 100.0 * DMACH(4)
      CALL DCHFAC(4,XTXIY,4,TOL,IRANK,TY,4)
      CALL DRNMVN(1,4,TY,4,B,1)
      DO 1234 J=1,4
1234  B(1,J)=B(1,J)+BHAT(J)

C  SUBROUTINE REGX2 PREDICTS X2 FROM X1.
C  THE SUBROUTINE RETURNS ALPHA HAT AND XTYX2-1 * V2.
      CALL REGX2(NSUBJ,DATA,V2,XTXIX2,ALPHAT)
C
C  GENERATE NEW VALUES FOR ALPHA(1,J),J=1,2),USING ALPHAT
C
C  FIRST OBTAIN CHOLESKY FACTOR
      TOL = 100.0 * DMACH(4)
      CALL DCHFAC(2,XTXIX2,2,TOL,IRANK,TX2,2)
      CALL DRNMVN(1,2,TX2,2,ALPHA,1)
      DO 1235 J=1,2
1235  ALPHA(1,J)=ALPHA(1,J)+ALPHAT(J)

C  GENERATE A VALUE FOR V1 GIVEN B, M 0
C  COMPUTE Q1= 1/2 * SUM(Y-B'X)**2
      Q1=0.0
      DO 10 I2 = 1,NSUBJ
```

```
      PRED1=0.0
      DO 345 J=1,3
345   PRED1=PRED1 + DATA(I2,J+1)*B(1,J+1)
      PRED1=PRED1+B(1,1)
      ERROR1=DATA(I2,1)-PRED1
C     PRINT 366,DATA(I2,1),PRED1,ERROR1
366   FORMAT(1X,'DATA(I2,1),PRED1,ERROR1',1X,3F10.4)
        Q1 = Q1 + ERROR1**2
10    CONTINUE

       Q1 = .5*Q1
       DF=NSUBJ
       A=DF/2.0
       CALL DRNGAM (1,A,GAM)
       V1 = GAM(1)/Q1
       V1 = 1/V1
C      PRINT*,'V1 = ',V1

C  1/V1 OR PRECISION HAS A GAMMA DISTRIBUTION
C
C
C  GENERATE A VALUE FOR V2 GIVEN ALPHA,M,O
C  COMPUTE Q2= 1/2 * SUM(X2-ALPHA'X0X1)**2
      Q2=0.0
      DO 11 I2 = 1,NSUBJ
      PRED2=0.0
      PRED2=PRED2 + DATA(I2,2)*ALPHA(1,2)
      PRED2=PRED2+ALPHA(1,1)
      ERROR2=DATA(I2,3)-PRED2
C     PRINT 337,DATA(I2,3),PRED2,ERROR2
337   FORMAT(1X,'X2,PRED2,ERROR2 ',1X,3F10.4)
        Q2 = Q2 + ERROR2**2
11    CONTINUE

       Q2 = .5*Q2
       DF=NSUBJ
       A=DF/2.0
       CALL DRNGAM (1,A,GAM)
       V2 = GAM(1)/Q2
       V2 = 1/V2
C      PRINT*,'V2 = ',V2
C  1/V2 OR PRECISION HAS A GAMMA DISTRIBUTION

C  GENERATE NEW VALUE FOR PI
C     PI = BETA(SUMX1+1, NSUBJ-SUMX1+1)
```

```
C  CALCULATE THE SUM OF X1
      SUMX1 = 0.0
      DO 20 I3 = 1,NSUBJ
C     IF (DATA(I3,2) .EQ. XMISS) GO TO 19
C     PRINT*,'X1 = ',DATA(I3,2)
      SUMX1 = SUMX1 + DATA(I3,2)
19    CONTINUE
20    CONTINUE
C     PRINT*,'SUMX1 = ',SUMX1
      PIN = SUMX1 + 1
      QIN = NSUBJ - SUMX1 + 1
      CALL DRNBET(1,PIN,QIN,BETA(1))
      PI = BETA(1)
      IF (L .EQ. 1 .AND. KOUNT .LT. IBURN) GO TO 1000
C
C  PRINT OUT FILLED-IN DATA
C     PRINT*,'NEW FILLED-IN DATA SET'
      DO 335 I=1,NSUBJ
C     PRINT 336,(DATA(I,J),J=1,4)
336   FORMAT(1X,'Y X1 X2 X3 ',4F8.4)
335   CONTINUE
C
C     PRINT 1236,(B(1,J),J=1,4)
1236  FORMAT(1X,'NEW B-VALUES = ',4F8.4)
C      PRINT*,'NEW V1 = ',V1
C     PRINT*,'NEW V2 = ',V2
C
C     PRINT*,'NEW PI VALUE = ',PI
C     PRINT*
C
C  CALLING SUBROUTINE TO CALCULATE R-SQUARED
      CALL  RSQ(PI,ALPHA,B,V1,V2,R2)
C
C COPY R-SQUARED VALUE INTO STORE
      STORE(L) = R2
C
C     PRINT 1237,(ALPHA(1,J),J=1,2)
1237  FORMAT(1X,'NEW ALPHA-VALUES = ',2F8.4)
C     PRINT 800, (B(1,J),J=1,4),(ALPHA(1,J),J=1,2),V1,V2,R2
800   FORMAT(1X,4F10.4/1X,2F10.4/1X,F10.4/1X,F10.4,1X,F6.4//)
C
C ACCUMULATE SUMS OF RSQ,B,ALPHA,V1,V2,PI
      DO 522 I = 1,4
522   BMEAN(I)= BMEAN(I) + B(1,I)
      DO 523 I=1,2
```

```
523   ALPHAM(I) = ALPHAM(I)+ALPHA(1,I)

      V1M = V1M + V1
      V2M = V2M + V2
      PIM = PIM + PI
      R2M = R2M + R2
2000  CONTINUE
C
      DO 524 I=1,4
524   BMEAN(I) = BMEAN(I)/NREP
      DO 525 I=1,2
525   ALPHAM(I) = ALPHAM(I)/NREP
      V1M = V1M/NREP
      V2M = V2M/NREP
      PIM = PIM/NREP
      R2M = R2M/NREP
      PRINT*,'PARAMETER MEANS'
      PRINT*,'SEED = ',ISEED
      PRINT 526,(BMEAN(I),I=1,4)
526   FORMAT(1X,'B-MEANS = ',4F8.4)
      PRINT 527,(ALPHAM(I),I=1,2)
527   FORMAT(1X,'ALPHA-MEANS = ',2F8.4)
      PRINT 528,V1M,V2M
528   FORMAT (1X,'V1 AND V2 MEANS = ',2F10.4)
      PRINT 529,PIM
529   FORMAT(1X,'PI MEAN = ',F8.4)
      PRINT 531,R2M
531   FORMAT(1X,'MEAN OF R-SQUARED = ',F8.4)
      CALL SORT(STORE,NREP)
      CALL HDR (STORE,.90,NREP)
      STOP
      END

C*********************************************************************
C  THE SUBROUTINE TO CREATE THE DATA SET AND MAKE SOME
C  MISSINC (MAR)
C  X2 IS ALWAYS PRESENT, Y MISSING DEPENDS ON THE VALUE OF
C  X1, AND X1 IS MCAR
C*********************************************************************
C
      SUBROUTINE GENDAT(TEMP,BP,ALPHAP,V1P,V2P,PIP,XMISS,II,PY0M,PY1M,
     &  PROBX2M)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION BP(4),ALPHAP(2),Z(1),U(1)
      DIMENSION DATA(100,4),XTXIY(4,4),TY(4,4),B(1,4),GAM(1),BETA(1)
```

```
      DIMENSION XTXIX2(2,2),TX2(2,2),ALPHA(1,2)
      DIMENSION IR(100,4)
      DIMENSION BHAT(4),ALPHAT(2)
      DIMENSION TEMP(4)
      DIMENSION SAVE(4)
C
C  GENERATE X1 = TEMP(2)
      TEMP(2) = 0.0
      CALL DRNUN(1,U(1))
      IF (U(1) .LE. PIP) TEMP(2) = 1.0
C     PRINT*,'U(1) = ',U(1),'TEMP(2) = ',TEMP(2)
C  GENERATE X2 = TEMP(3) AND X1X2 = TEMP(4)
      CALL DRNNOR(1,Z(1))
      X2MEAN = ALPHAP(1) + ALPHAP(2)*TEMP(2)
      TEMP(3) = X2MEAN +DSQRT(V2P)*Z(1)
      TEMP(4) = TEMP(2)*TEMP(3)
C  GENERATE Y = TEMP(1)
      YMEAN=BP(1)+BP(2)*TEMP(2)+BP(3)*TEMP(3)+BP(4)*TEMP(4)
      CALL DRNNOR(1,Z(1))
      TEMP(1) = YMEAN + DSQRT(V1P)*Z(1)
C     PRINT*,'COMPLETE DATA ',TEMP
C
C  GENERATE MISSING DATA (Y MAR, X2 MCAR, X1 ALWAYS PRESENT)
C FIRST DEAL WITH Y
        CALL DRNUN(1,U(1))
        IF  (TEMP(2) .EQ. 1. .AND. U(1) .LE. PY1M) TEMP(1) = XMISS
        CALL DRNUN(1,U(1))
        IF  (TEMP(2) .EQ. 0. .AND. U(1) .LE. PY0M) TEMP(1) = XMISS

C THEN DEAL WITH X2
       CALL DRNUN(1,U(1))
       IF (U(1) .LE. PROBX2M) TEMP(3) = XMISS
C      PRINT*,V(1),PROBX2M,TEMP(2)

C THEN DEAL WITH X3
      IF (TEMP(3) .EQ. XMISS) TEMP(4) = XMISS
C     PRINT 45,TEMP(1),TEMP(2),TEMP(3),TEMP(4)
45    FORMAT(1X,'Y X1 X2 X1X2 ',4F6.2)
C  WRITE GENERATED DATA (SOME MISSING) TO FILE
      WRITE (10,450) TEMP
450   FORMAT(1X,4F10.4)

C
      RETURN
      END
```

```
C
C*********************************************************************
C  THE SUBROUTINE TO IDENTIFY WHICH VARIABLES ARE MISSING
C     AND CALL THE APPROPRIATE SUBROUTINE TO FILL THEM IN
C*********************************************************************
C
      SUBROUTINE GEN(B,V1,ALPHA,V2,PI,IR,DATA,NSUBJ)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION DATA(100,4),XTXIY(4,4),TY(4,4),B(1,4),GAM(1),BETA(1)
      DIMENSION XTXIX2(2,2),TX2(2,2),ALPHA(1,2)
      DIMENSION IR(100,4)
      DIMENSION BHAT(4),ALPHAT(2)
      DIMENSION TEMP(4)
C
C     PRINT*,'SUBROUTINE GEN CALLED'
C   REMOVING SUBJECTS WITH COMPLETE DATA
      DO 1 III = 1,NSUBJ
C       SUM = 0.0
C      DO 2 J = 1,4
C          SUM = SUM + IR(J)
C          IF (SUM .EQ. 4.0) GO TO 1
C  COPYING SUBJECT'S DATA VALUES INTO TEMP VECTOR
      DO 3 J = 1,4
3     TEMP(J) = DATA(III,J)
C  IDENTIFYING EACH MISSING DATA PATTERN AND CALLING
C   THE APPROPRIATE SUBROUTINE
C **** MISSING Y
      IF (IR(III,1) .EQ. 0 .AND. IR(III,2) .EQ. 1 .AND.
     &  IR(III,3) .EQ. 1)
     &  CALL GEN011(TEMP,B,V1,ALPHA,V2,PI)
C **** MISSING X1
      IF (IR(III,1) .EQ. 1 .AND. IR(III,2) .EQ. 0 .AND.
     &  IR(III,3) .EQ. 1)
     &  CALL GEN101(TEMP,B,V1,ALPHA,V2,PI)
C **** MISSING X2
      IF (IR(III,1) .EQ. 1 .AND. IR(III,2) .EQ. 1 .AND.
     &  IR(III,3) .EQ. 0)
     &  CALL GEN110(TEMP,B,V1,ALPHA,V2,PI)
C **** MISSING Y AND X1
      IF (IR(III,1) .EQ. 0 .AND. IR(III,2) .EQ. 0 .AND.
     &  IR(III,3) .EQ. 1)
     &  CALL GEN001(TEMP,B,V1,ALPHA,V2,PI)
C **** MISSING Y AND X2
      IF (IR(III,1) .EQ. 0 .AND. IR(III,2) .EQ. 1 .AND.
     &  IR(III,3) .EQ. 0)
```

```
     &  CALL GEN010(TEMP,B,V1,ALPHA,V2,PI)

C **** MISSING X1 AND X2
      IF (IR(III,1) .EQ. 1 .AND. IR(III,2) .EQ. 0 .AND.
     &  IR(III,3) .EQ. 0)
     &  CALL GEN100(TEMP,B,V1,ALPHA,V2,PI)


C   COPYING IN THE ESTIMATED VALUES FOR MISSING DATA
       DO 400 J = 1,4
400    DATA(III,J) = TEMP(J)
1     CONTINUE
      RETURN
      END
C
C ****************************************************************
C  THE SUBROUTINE TO FILL IN MISSING Y
C ****************************************************************
C
      SUBROUTINE GEN011(TEMP,B,V1,ALPHA,V2,PI)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION DATA(100,4),XTXIY(4,4),TY(4,4),B(1,4),GAM(1),BETA(1)
      DIMENSION XTXIX2(2,2),TX2(2,2),ALPHA(1,2)
      DIMENSION IR(100,4)
      DIMENSION BHAT(4),ALPHAT(2)
      DIMENSION TEMP(4)
      DIMENSION Z(1)

C     PRINT*,'GEN011 CALLED'

C SAMPLING Y FROM NORMAL DISTRIBUTION WITH MEAN=B0+B1X1 ETC.
C  AND VARIANCE V1
      CALL DRNNOR(1,Z(1))
      SDEV = DSQRT(V1)
      YMEAN = B(1,1) + B(1,2)*TEMP(2) +B(1,3)*TEMP(3)
     &  +B(1,4)*TEMP(4)
      TEMP(1) = Z(1)*SDEV + YMEAN
C     PRINT 20, TEMP(1)
20    FORMAT(1X,'FROM GEN011,NEW Y VALUE=',F6.3)
      RETURN
      END
C
C ****************************************************************
C  THE SUBROUTINE TO FILL IN MISSING X1
C ****************************************************************
```

```
C
      SUBROUTINE GEN101(TEMP,B,V1,ALPHA,V2,PI)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION DATA(100,4),XTXIY(4,4),TY(4,4),B(1,4),GAM(1),BETA(1)
      DIMENSION XTXIX2(2,2),TX2(2,2),ALPHA(1,2)
      DIMENSION U(1)
      DIMENSION IR(100,4)
      DIMENSION BHAT(4),ALPHAT(2)
      DIMENSION TEMP(4)

C     PRINT*,'GEN101 CALLED'

C COMPUTING P(X2|X1=1) CALLED P05
      TEMP(2) = 1.0
      TEMP(4) = TEMP(2) * TEMP(3)
      P01 = DSQRT(1/V2)
      P02 = -1/(2*V2)
      P03 = (TEMP(3) - ALPHA(1,1) - ALPHA(1,2)*TEMP(2))**2
      P04 = DEXP(P02*P03)
      P05 = P01*P04
C COMPUTING THE CONDITIONAL PROBABILITY (Y|X1=1,X2) CALLED P10
      P06 = DSQRT(1/V1)
      P07 = -1/(2*V1)
      P08 = (TEMP(1)-B(1,1)-B(1,2)*TEMP(2)-B(1,3)
     &    *TEMP(3)-B(1,4)*TEMP(4))**2
      P09 = DEXP(P07*P08)
      P10 = P06 * P09
C  COMPUTING THE NUMERATOR
      XNUM = P05 * P10 * PI
C  THE SAME FOR X1 = 0 (P01, P02, P06 AND P07 DON'T CHANGE)
C  CALCULATING P(X2|X1=0) CALLED P050
      TEMP(2) = 0.0
      TEMP(4) = 0
      P030 = (TEMP(3) - ALPHA(1,1))**2
      P040 = DEXP(P02*P030)
      P050 = P01*P040
C CALCULATING (Y|X1=0,X2) CALLED P100
      P080 = (TEMP(1)-B(1,1)-B(1,3)*TEMP(3))**2
      P090 = DEXP(P07*P080)
      P100 = P06 * P090
C COMPUTE DENOMINATOR
C FOR X1 = 0
      DENOM1 = XNUM
      DENOM0 = P050 * P100 * PI
```

```
      DENOM = DENOM1 + DENOM0
C GETTING THE FINAL PROBABILITY
      PFINAL = XNUM/DENOM
C GENERATING THE X1 VALUE
      CALL DRNUN (1,U(1))
      IF (U(1) .LE. PFINAL) TEMP(2) = 1
      IF (U(1) .GT. PFINAL) TEMP(2) = 0
C  RESET X1X2 VALUE
      TEMP(4) = TEMP(2) * TEMP(3)
C  PRINT VALUES AND RETURN
C     PRINT 33,TEMP(2)
33    FORMAT (1X,'FROM GEN101,NEW X1 VALUE=',F4.2)
      RETURN
      END
C
C ****************************************************************
C THE SUBROUTINE TO FILL IN MISSING X2
C ****************************************************************
C
      SUBROUTINE GEN110(TEMP,B,V1,ALPHA,V2,PI)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION DATA(100,4),XTXIY(4,4),TY(4,4),B(1,4),GAM(1),BETA(1)
      DIMENSION XTXIX2(2,2),TX2(2,2),ALPHA(1,2)
      DIMENSION U(1)
      DIMENSION IR(100,4)
      DIMENSION BHAT(4),ALPHAT(2)
      DIMENSION TEMP(4)
      DIMENSION Z(1)

C     PRINT*,'GEN110 CALLED'

C  FIRST GET THE COMPONENT PARTS
C  COMPUTE A1 I.E. E(X2|X1)
      A1 = ALPHA(1,1) + ALPHA(1,2) * TEMP(2)
C  COMPUTE B1 I.E. E(Y|X1)
      B1 = B(1,1) + B(1,2)*TEMP(2) + A1*(B(1,3)+B(1,4)*TEMP(2))
C  C1 IS DEFINED AS VAR(X2|X1)
      C1 = V2
C  COMPUTE D1 = COV(Y,X2|X1)
      D11 = B(1,4)*TEMP(2)
      D12 = B(1,3)+D11
      D1 =  D12*V2
C  COMPUTE E1 = VAR(Y|X1)
      E11 = B(1,3)+B(1,4)*TEMP(2)
      E12 = E11**2
```

```
      E1 = V1 + V2 * E12
C  NEXT GENERATE THE MISSING X2 AND WRITE IT TO TEMP(3)
      CALL DRNNOR(1,Z(1))
      VAR1 = D1**2/(C1*E1)
      VAR2 =  1 - VAR1
      VAR3 = C1 * VAR2
      SDEV = DSQRT(VAR3)
      X2MEAN = A1 - (D1/E1)*B1 + (D1/E1)*TEMP(1)
      TEMP(3) = Z(1)*SDEV + X2MEAN
      TEMP(4) = TEMP(3) * TEMP(2)
C     PRINT 25,TEMP(3)
25    FORMAT(1X,'FROM GEN110,NEW X2 VALUE=',1X,F6.3)
      RETURN
      END
C
C ***************************************************************************
C THE SUBROUTINE TO FILL IN MISSING Y AND X2
C ***************************************************************************
C
      SUBROUTINE GEN010(TEMP,B,V1,ALPHA,V2,PI)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION DATA(100,4),XTXIY(4,4),TY(4,4),B(1,4),GAM(1),BETA(1)
      DIMENSION XTXIX2(2,2),TX2(2,2),ALPHA(1,2)
      DIMENSION U(1)
      DIMENSION IR(100,4)
      DIMENSION BHAT(4),ALPHAT(2)
      DIMENSION TEMP(4)
      DIMENSION Z(1)
C
C     PRINT*,'GEN010 CALLED'

C  FIRST GENERATE THE X2 GIVEN X1
      CALL DRNNOR(1,Z(1))
      SDEV = DSQRT(V2)
      X2MEAN = ALPHA(1,1) + ALPHA(1,2) * TEMP(2)
      TEMP(3) = Z(1)*SDEV + X2MEAN
      TEMP(4) = TEMP(2) * TEMP(3)

C
C NOW GENERATE Y GIVEN X1 AND X2
C SAMPLING Y FROM NORMAL DISTRIBUTION WITH MEAN=B0+B1X1 ETC.
C  AND VARIANCE V1
      CALL DRNNOR(1,Z(1))
      SDEV = DSQRT(V1)
      YMEAN = B(1,1) + B(1,2)*TEMP(2) +B(1,3)*TEMP(3)
```

```
     &  +B(1,4)*TEMP(4)

      TEMP(1) = Z(1)*SDEV + YMEAN
C     PRINT 23,TEMP(1),TEMP(3)
23    FORMAT(1X,'FROM GEN010,NEW Y = ',F6.2,1X,'NEW X2 = ',F6.2)
300   CONTINUE
      RETURN
      END
C
C *************************************************************************
C THE SUBROUTINE TO FILL IN MISSING X1 AND X2
C *************************************************************************
C
      SUBROUTINE GEN100(TEMP,B,V1,ALPHA,V2,PI)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION DATA(100,4),XTXIY(4,4),TY(4,4),B(1,4),GAM(1),BETA(1)
      DIMENSION XTXIX2(2,2),TX2(2,2),ALPHA(1,2)
      DIMENSION U(1)
      DIMENSION IR(100,4)
      DIMENSION BHAT(4),ALPHAT(2)
      DIMENSION TEMP(4)
      DIMENSION Z(1)
C
C     PRINT*,'GEN100 CALLED'
C FIRST FILL IN X1 FROM P(X1|Y)
C GET B1 = E(Y|X1) AND E1 = VAR(Y|X1)
C LEAVE OUT CONSTANT (1/2PI)**1/2  WHERE PI=3.14.....
C
C FOR X1 = 1
      TEMP(2) = 1.
      B11 = ALPHA(1,1) + ALPHA(1,2)*TEMP(2)
      B12 = B(1,3) + B(1,4)*TEMP(2)
      B13 = B11*B12
      B14 = B(1,1) + B(1,2)*TEMP(2)
      B1 = B13 +B14
C
      E11 = B(1,3)+B(1,4)*TEMP(2)
      E12 = E11**2
      E1 = V1 + V2 * E12
C
      XNUM11 = 1/DSQRT(E1)
      XNUM12 = -1/(2*E1)
      XNUM13 = (TEMP(1) - B1)**2
      XNUM14 = DEXP(XNUM12*XNUM13)
      XNUM15 = XNUM11 * XNUM14
```

```
C
      XNUM1 = XNUM15 * PI
C
C  FOR X1 = 0
      TEMP(2) = 0
      B0 = B(1,1) + (ALPHA(1,1) *B(1,3))
C
      E0 = V1 + V2*((B(1,3)**2))
C
      XNUM01 = 1/DSQRT(E0)
      XNUM02 = -1/(2*E0)
      XNUM03 = (TEMP(1) - B0)**2
      XNUM04 = DEXP(XNUM02*XNUM03)
      XNUM05 = XNUM01 * XNUM04
      XNUM0 =  XNUM05 * (1-PI)
C
      DENOM = XNUM1 + XNUM0
      PFINAL = XNUM1/DENOM
C
C GENERATING THE X1 VALUE
      CALL DRNUN (1,U(1))
      IF (U(1) .LE. PFINAL) TEMP(2) = 1
      IF (U(1) .GT. PFINAL) TEMP(2) = 0
C     PRINT 36, PFINAL,U(1),TEMP(2)
36    FORMAT(1X,'PFINAL= ',F6.4,1X,'U(1)= ',F6.4,1X,
     & 'TEMP(2)= ',F4.2)
C
C  NEXT FILL IN X2 GIVEN X1
C
      CALL DRNNOR(1,Z(1))
      SDEV = DSQRT(V2)
      X2MEAN = ALPHA(1,1) + ALPHA(1,2)*TEMP(2)
      TEMP(3) = Z(1)*SDEV + X2MEAN
      TEMP(4) = TEMP(2) * TEMP(3)
C     PRINT 32,TEMP(2),TEMP(3)
32    FORMAT(1X,'FROM GEN100, NEW X1 = ',F4.2,1X,'NEW X2= ',F6.2)
C
300   CONTINUE
      RETURN
      END
C
C
C ********************************************************************
C THE SUBROUTINE TO FILL IN MISSING X1 AND Y
```

```
C ***********************************************************************

C
      SUBROUTINE GEN001(TEMP,B,V1,ALPHA,V2,PI)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION DATA(100,4),XTXIY(4,4),TY(4,4),B(1,4),GAM(1),BETA(1)
      DIMENSION XTXIX2(2,2),TX2(2,2),ALPHA(1,2)
      DIMENSION IR(100,4)
      DIMENSION BHAT(4),ALPHAT(2)
      DIMENSION TEMP(4)
      DIMENSION U(1)
      DIMENSION Z(1)
C
C
C     PRINT*,'GEN001 CALLED'
C FIRST FILL IN X1
C P(X1|X2) = P(X1,X2)/P(X2)
C P(X1,X2) = P(X2|X1)*P(X1)
C P(X2) IS GOTTEN BY ADDING TOGETHER P(X2|X1=1) AND P(X2|X1=0)
C P(X1) IS CANCELLED FROM BOTH NUMERATOR AND DENOMINATOR
C
C COMPUTING P(X2|X1=1) CALLED P05
      TEMP(2) = 1.0
      P01 = DSQRT(1/V2)
      P02 = -1/(2*V2)
      P03 = TEMP(3) - ALPHA(1,1) -ALPHA(1,2)*TEMP(2)
      P03 = P03**2
      P04 = DEXP(P02*P03)
      P05 = P01*P04
C COMPUTING P(X2|X1=0) CALLED P10
      TEMP(2) = 0.0
      P06 = DSQRT(1/V2)
      P07 = -1/(2*V2)
      P08 = TEMP(3) - ALPHA(1,1)
      P08 = P08**2
      P09 = DEXP(P07*P08)
      P10 = P06*P09
C   COMPUTE THE PROBABILITY (X2|X1), CANCELLING P(X1) FROM ALL TERMS
      XNUM = P05
      DENOM = P05 + P10
      PROB = XNUM/DENOM
C
C GENERATING THE X1 VALUE AND WRITE INTO TEMP(2)
      CALL DRNUN (1,U(1))
      IF (U(1) .LE. PROB) TEMP(2) = 1.
```

```
      IF (U(1) .GT. PROB) TEMP(2) = 0.
C
      TEMP(4) = TEMP(2) * TEMP(3)
C
C
C NEXT FILL IN Y USING X1 AND X2
C SAMPLING Y FROM NORMAL DISTRIBUTION WITH MEAN=B0+B1X1 ETC.
C  AND VARIANCE V1
      CALL DRNNOR(1,Z(1))
      SDEV = DSQRT(V1)
      YMEAN = B(1,1) + B(1,2)*TEMP(2) +B(1,3)*TEMP(3)
     &  +B(1,4)*TEMP(4)
      TEMP(1) = Z(1)*SDEV + YMEAN
C     PRINT 36,TEMP(1),TEMP(2)
36    FORMAT(1X,'FROM GEN001,NEW Y VALUE = ',F6.2,1X,
     & 'NEW X1 VALUE = ',F4.2)
      RETURN
      END
C
C
*************************************************************************
C THE SUBROUTINE TO PREDICT Y FROM X1 AND X2 USING THE FILLED-IN
C DATA, AND RETURNS BHAT(4) IS USED IN GENERATING THE NEW BETA
C PARAMETERS
C
*************************************************************************
C
       SUBROUTINE REGY(NSUBJ,DATA,V1,XTXIY,BHAT)
       IMPLICIT REAL*8(A-H,O-Z)
       DIMENSION DATA(100,4), XTXIY(4,4), XTY(4)
       DIMENSION TEMP(4), BHAT(4)

C      PRINT*,'GEN REGY CALLED'

C   INITIALIZE XTY AND XTXIY MATRICES WITH ZEROES
       DO 100 I = 1,4
        XTY(I) = 0.0
100   CONTINUE
       DO 200 I = 1,4
       DO 210 J = 1,4
       XTXIY(I,J) = 0.0
210    CONTINUE
200    CONTINUE
C
C  CREATE VECTOR TEMP CONTAINING: 1 X1 X2 X1*X2
```

```
       TEMP(1)  = 1.0
       DO 1000 IJK = 1,NSUBJ
       DO 20 J=2,4
20     TEMP(J) = DATA(IJK,J)
C
       DO 30 J = 1,4
30     XTY(J) = XTY(J) + TEMP(J) * DATA(IJK,1)
       DO 40 J = 1,4
       DO 45 K = 1,4
       XTXIY(J,K) = XTXIY (J,K) + TEMP(J) * TEMP(K)
45     CONTINUE
40     CONTINUE
1000   CONTINUE
C
C      DO 8888 II=1,4
C      PRINT 8889,(XTXIY(II,K),K=1,4)
8889   FORMAT(1X,4F10.4)
8888   CONTINUE
C     INVERT XTXIY
       CALL DLINDS(4,XTXIY,4,XTXIY,4)
C
C   COMPUTE BHAT
       DO 50 I=1,4
C   INITIALIZE BHAT AS ZEROES
       BHAT(I) = 0.0
       DO 50 J=1,4
50     BHAT(I) = BHAT(I) + XTXIY(I,J)*XTY(J)
C   COMPUTE THE VARIANCE/COVAR MATRIX FOR BHAT
       DO 60 I=1,4
       DO 70 J = 1,4
       XTXIY(I,J) = XTXIY(I,J)*V1
70     CONTINUE
60     CONTINUE
       RETURN
       END
C
C*********************************************************************
C THE SUBROUTINE TO PREDICT X2 FROM X1 USING THE FILLED-IN
C DATA, AND COMPUTE ALPHAT WHICH IS USED IN OBTAINING THE
C NEW ALPHA PARAMETERS
C*********************************************************************
C
       SUBROUTINE REGX2(NSUBJ,DATA,V2,XTXIX2,ALPHAT)
       IMPLICIT REAL*8(A-H,O-Z)
       DIMENSION DATA(100,4), XTXIX2(2,2), XTX2(2)
```

```
      DIMENSION TEMP(4), ALPHAT(2)

C     PRINT*,'REGX2 CALLED'

C  INITIALIZE EVERYTHING
      DO 90 II= 1,2
90    XTX2(II) = 0.0
      DO 100 I=1,2
      DO 100 J=1,2
100     XTXIX2(I,J) = 0.0
C  CREATE A VECTOR TEMP WITH VALUES 1 X1 X2
      TEMP(1) = 1.0
      DO 1000 IJK = 1,NSUBJ
      DO 20 J = 2,3
20    TEMP(J) = DATA(IJK,J)
C CREATE XTX2 VECTOR
      DO 25 J = 1,2
25    XTX2(J) = XTX2(J) + TEMP(J) * TEMP(3)
      DO 30 J = 1,2
      DO 30 K = 1,2
30    XTXIX2(J,K) = XTXIX2(J,K) + TEMP(J) * TEMP(K)
1000  CONTINUE
C     DO 8888 II=1,2
C     PRINT 8889, (XTXIX2(II,K),K=1,2)
8889  FORMAT(1X,2F10.4)
8888  CONTINUE
C INVERT XTXIX2
      CALL DLINDS(2,XTXIX2,2,XTXIX2,2)
C  COMPUTE ALPHAT (FIRST INITIALIZE AS ZEROES)
      DO 50 I = 1,2
50      ALPHAT(I) = 0.0
      DO 60 I= 1,2
      DO 60 J = 1,2
60    ALPHAT(I) = ALPHAT(I) + XTXIX2(I,J) * XTX2(J)
C   COMPUTE VAR/COVAR MATRIX FOR ALPHAT
     DO 64 I = 1,2
     DO 65 J=I,2
      XTXIX2(I,J) = XTXIX2(I,J) * V2
65    CONTINUE
64    CONTINUE
      RETURN
      END
C
C*********************************************************************
C  THE SUBROUTINE TO CALCULATE R-SQUARED ON THE FILLED-IN DATA SET
```

```
C*********************************************************************
C
      SUBROUTINE RSQ(PI,ALPHA,B,V1,V2,R2)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION DATA(100,4),XTXIY(4,4),B(1,4)
      DIMENSION XTXIX2(2,2),TX2(2,2),ALPHA(1,2)
      DIMENSION BHAT(4),ALPHAT(2)
      DIMENSION TEMP(4)
C
C     PRINT*,'RSQ CALLED'
C
C RSQARED = 1 - VAR(Y/X1X2X3)/VAR(Y).
C VAR(Y|X1X2X3) IS V1.
C WE CALCULATE VAR(Y) CALLED VARY IN THIS SUBROUTINE.
C PI REFERS TO P(X1=1).
C
C
      B32 = B(1,3)**2
      P1 = (1-PI)*(V1 + B32*V2)
      B322 = (B(1,3) + B(1,4))**2
      P2 = PI * (V1 + B322*V2)
      B33 = B(1,2) + B(1,3)*ALPHA(1,2)
     &   + B(1,4)*(ALPHA(1,1)+ALPHA(1,2))
      P3 = PI * (1-PI) * (B33**2)
      VARY = P1 + P2 + P3
C
      R2 = 1 - (V1/VARY)
C     WRITE (10,323) R2
323   FORMAT(1X,F10.4)
C
      RETURN
      END
*********************************************************************
** THE SUBROUTINE TO SORT THE R-SQUARES
*********************************************************************
      SUBROUTINE SORT (STORE,MM)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION STORE (5000)
C
      ISTOP = MM-1
      DO 30 I=1,ISTOP
      JSTOP = MM-I
      DO 20 J=1,JSTOP
      IF (STORE(J) .LT. STORE(J+1)) GO TO 20
      TEMP = STORE(J)
```

```
      STORE(J) = STORE(J+1)
      STORE(J+1) = TEMP
20    CONTINUE
30    CONTINUE
      RETURN
      END

*********************************************************************
** THE SUBROUTINE TO FIND THE HIGHEST DENSITY REGION
*********************************************************************
      SUBROUTINE HDR (STORE,XLEVEL,MM)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION STORE (5000),USE(5000)
C
      ILOW = (1. - XLEVEL)*MM - 1
      IUP = XLEVEL*MM - 1
C
      RHOL = STORE(1)
      RHOU = STORE(IUP+1)
      XMIN = RHOU-RHOL
      DO 2 I=2,ILOW
      J= I + IUP
      DIFF = STORE(J) -STORE(I)
      IF (DIFF .GT. XMIN) GO TO 2
      XMIN = DIFF
      RHOL = STORE(I)
      RHOU = STORE(J)
2     CONTINUE
      PRINT 3333, MM,RHOL,RHOU
3333  FORMAT(1X,'MM =',I4,1X,'RHOL =',F6.4,1X,'RHOU = ',F6.4)
      RETURN
      END
//GO.SYSIN DD *
5721571
.1
.33
.05
 50  -9   5000 1. 1.  1.  1.  16.5 1. 1. 5. .5
//GO.FT10F001 DD UNIT=DISK,VOL=SER=SCR002,
// SPACE=(TRK,(5,1),RLSE),DCB=(LRECL=41,BLKSIZE=4100,RECFM=FB),
// DISP=(NEW,CATLG,DELETE),DSN=WYL.GC.SAR.DATA
```

References

Box, George E. P., and Tiao, George C. (1973). Bayesian inference in statistical analysis. Reading, MA: Addison-Wesley.

Dellaportas, P., & Smith, A. F. J. (1993). Bayesian inference for generalized linear and proportional hazards models via Gibbs sampling. Applied Statistics, 42 (3), 443-459.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood estimation from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, B, 39, 1-38.

Dixon, W. J. (Ed.). (1990). BMDP Statistical Software Manual (Vols.1-2). Berkeley: University of California Press.

Gelfand, Alan E., & Smith, Adrian F. M. (1990). Sampling-based approaches to calculating marginal densities. Journal of the American Statistical Association, 85, 398-409.

IMSL user's manual: Stat/Library FORTRAN subroutines for statistical analysis. (Version 1.0). (1987). Houston, TX: IMSL.

Lee, Peter M. (1989). Bayesian statistics: An introduction. New York: Oxford University Press.

Little, Roderick J. A., & Rubin, Donald R. (1987). Statistical analysis with missing data. New York: Wiley.

Little, Roderick J. A., & Schluchter, Mark D. (1985). Maximum likelihood estimation for mixed continuous and categorical data with missing values. Biometrika, 72 (3), 497-512.

MacEachern, Steven N., & Berliner, L. Mark. (1994). Subsampling the Gibbs Sampler. American Statistician, 48 (3), 188-190.

Mood, Alexander M., Graybill, Franklin A., & Boes, Duane C. (1974). Introduction to the theory of statistics (3rd ed.). New York: McGraw-Hill.

Ross, Sheldon M. (1974). A first course in probability. New York: Macmillan.

SAS procedures guide, version 6. (1990). Third edition. Cary, NC: SAS Institute, Inc.

Smith, A. F. M., & Roberts, G. O. (1993). Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods. Journal of the Royal Statistical Society B, 55 (1), 3-23.

SPSS-X user's guide (3rd ed.). (1988) Chicago: SPSS, Inc.

Winkler, Robert L. (1972). An introduction to Bayesian inference and decision. New York: Holt, Rinehart & Winston.