

Paper 111-30

Maximum Likelihood Parameter Estimation with Incomplete Data

Catherine Truxillo, Ph.D., SAS Institute Inc, Cary, NC

ABSTRACT

Incomplete data presents a problem in both inferential and predictive modeling applications. Traditional methods such as complete case analysis and simple imputation tend to produce results that inadequately estimate standard errors and/or parameter estimates, particularly where missingness is not completely at random (MCAR). Modern methods for handling incomplete data, including maximum likelihood parameter estimation and multiple imputation methods, enable researchers to derive appropriate parameter estimates and inference from incomplete data when data are missing at random (MAR). However, for a variety of reasons, multiple imputation is not always practical, particularly with very large data sets or in applications where it is important to have only one working data set (such as public sector data).

This paper illustrates the use of PROC MI, which provides maximum likelihood (ML) estimates of the covariance matrix and mean vector using the expectation-maximization (EM) algorithm. Using simulations, a number of incomplete data handling methods and resulting parameter estimates, standard errors, and error rates in classification of a validation data set are compared and evaluated in the context of discriminant analysis. Additionally, the proper specification of an appropriate nominal sample size for the analysis when the ML covariance matrix is used as input to the analysis is discussed. Results suggest that the covariance estimates from the EM algorithm produce results superior to other methods evaluated, particularly with an appropriately specified nominal sample size. Suggestions for handling missingness in the context of predictive and inferential modeling applications are presented.

INTRODUCTION

Most data analysis situations involve the estimation of unknown parameters from data that consist of observations that represent each distinct case in the study, and variables that represent the characteristics that have been either measured or fixed by the experimenter. Each observation provides information for estimating the parameters corresponding to the variables used in the analysis.

In spite of the best efforts, it is typical for some data to be incompletely observed. Missingness can be caused by a variety of mechanisms, some of which are fairly benign and others that can be quite problematic. Data might be miscoded and subsequently deleted. Participants in longitudinal studies might drop out. People might refuse to respond to sensitive questions.

Traditionally, researchers have relied on a variety of methods to perform statistical analyses that have missing data. Statistical procedures in SAS have methods in place for dealing with missing data, which typically imply certain assumptions about the mechanisms leading to missingness in your data. These methods work well under a variety of circumstances when missingness is small.

One approach to handling incomplete data is *complete case analysis* (CCA). In complete case analysis, a case (a subject, a participant, or an observation) is deleted from the analysis if any of the variables used in the analysis are missing for that case. This is one of the methods commonly implemented in statistical software. An alternative to CCA is available case analysis (ACA) in which all pairwise jointly available observations are used, and only jointly missing data are deleted from the analysis. ACA is less commonly used than CCA mainly because of the potential for covariance matrices that are less than full-rank and other statistical problems (Haitovsky, 1968).

When missingness is moderate to large, imputation can be helpful. *Imputation* involves replacing an incomplete observation with complete information based on an estimate of the true value of the unobserved variable. Obviously, the value used for imputing can have a substantial impact on the resulting usefulness of the resulting parameter estimates in the fitted model.

Some of the less desirable properties of imputation and CCA are well known (see Little & Rubin, 1987; Allison, 2002). CCA and some imputation methods assume that the data are missing completely at random. If a variable is *missing completely at random* (MCAR), then the variable's missingness does not depend on the true value of the unobserved variable, and it also does not depend on the values of other variables in the data set. MCAR is

characteristic of miscoded values or accidental loss of data. Many missing data handling methods assume MCAR, although it is not a realistic assumption in many real-data situations.

Violation of the MCAR assumption can result in biased parameter estimates and incorrect inference. A more general pattern of missingness, of which MCAR is a subset, is *missing at random* (MAR). If a variable is missing at random, missingness does not depend on the true value of the missing variable, but it might depend on the value of other variables that are observed.

This paper focuses on methods that assume, at a minimum, that data are MAR. Notice that if a variable is MCAR, then it is also MAR. However, if a variable is MAR, it does not necessarily follow that it is also MCAR.

More complex approaches to incomplete data handling have been discussed in statistical literature for several decades (Little & Rubin, 1983; Rubin, 1976; Little & Rubin, 1987) and, in recent years, such methods have been implemented in SAS/STAT. Two such procedures include the MI and MIANALYZE procedures. The MI procedure performs multiple imputation of missing values via a number of methods such as MCMC, regression, propensity score method, and others. The MIANALYZE procedure takes covariance and parameter estimates from analyses on the multiple data sets and re-combines them, producing parameter estimates and inferences that account for the uncertainty of imputation.

Multiple imputation methods demonstrated superior properties when compared to simple imputation methods and CCA. By imputing more than one value for each missing observation, uncertainty due to imputation is introduced into the analysis. Furthermore, because in most cases, imputed values are estimated by using information about other variables in the data, it is assumed that data are MAR, rather than the more restrictive MCAR. Re-combining estimates of parameters and covariance matrices results in efficient and unbiased estimates and correct inference (see Schafer, 1997 for detailed coverage of multiple imputation methods).

USING PROC MI WHEN MULTIPLE IMPUTATION IS NOT FEASIBLE

In many cases, multiple imputation is a reasonable and, often preferable, approach to handling incomplete data. However, there are circumstances when multiple imputation is not feasible. For example, with very large data sets, imputation can be slow and computationally intensive. Particularly in predictive modeling, where a large number of models must be fit and the best candidate selected, it might not be practical to work with several versions of a large data set, perform analyses on all versions, and re-combine the estimates to make inferences. Multiple imputation might not always produce models with better predictive power than some simple imputation methods, and the time cost of multiple imputation might prohibit its use in these sorts of applications.

Sometimes the researcher cannot use multiple data sets for other reasons. For instance, in large organizations, it might be necessary to have one house file that is shared by every analyst in the enterprise. In some government organizations, public sector data sets must be made widely available through, for example, an internet site. For some researchers, a summary data set such as a mean vector and covariance matrix is the simplest way of sharing information, as is often the case in academic research.

It is not the purpose of this paper to discuss or compare multiple imputation methods. In most cases, it will be assumed that multiple imputation, performed correctly, would result in estimates for the models at least as good as the methods discussed here. This paper focuses instead on methods that do not involve creating multiple data sets to handle missing data. Methods in PROC MI will be compared to other traditional methods such as CCA and single imputation.

THE METHODS EXAMINED IN THIS STUDY

COMPLETE CASE ANALYSIS (CCA)

CCA assumes that variables are MCAR. This is because any information about missingness on X_{ki} that is contained in X_{pi} ($k \neq p$) is lost with CCA, where k is the k^{th} variable and p is the p^{th} variable and i is the i^{th} observation. From the perspective of statistical inference and parameter estimation, CCA is a remarkably robust method for dealing with incomplete data. When observations are MCAR, parameter estimates are unbiased. Furthermore, even if data are MAR, when X_{mis} depends on the independent variables X_{obs} but not on the response Y , estimates are still unbiased with CCA (Little, 1992).

The primary disadvantage of CCA is loss of information and, hence, statistical power. Even a smattering of missing values can cause an enormous loss of data in high dimensions. For instance, a 1% probability of missing for 100 variables would leave only 37% of the data for analysis. If the missingness was increased to 5%, then <1% of the data would be available with 100 variables.

IMPUTATION

A very common method for performing single imputation is to fill in the marginal mean of the variable for missing values. Mean imputation assumes MCAR, and if this assumption is not met, parameter estimates are likely to be biased. Furthermore, if predictors in a model are associated with one another (even weakly associated) parameter estimates are typically biased with mean imputation. This is true even when data are MCAR. Furthermore, even if parameters are correctly estimated, inferences might still be incorrect because the variances are underestimated (Little & Rubin, 1987).

A variant on mean imputation, but which assumes MAR rather than MCAR, is to impute the mean of a variable conditional on another variable or variables in the analysis. For example, if observations are segmented, then the mean of the segment is imputed. Cluster imputation and Buck's method (Buck 1960) are both forms of conditional mean imputation. If observations are classified into groups, then the mean of the group can be used as the conditional mean.

In general, single imputation methods share the problem that, by imputing one value for each missing value, the variances of the variables are underestimated. This can result in incorrect inference because analyses on singly-imputed data are performed as though the data were completely observed.

MAXIMUM LIKELIHOOD ESTIMATION OF COVARIANCE AND MEAN PARAMETERS

Thus far we have discussed methods in which a subspace of the observations are used (such as CCA and the less commonly used ACA) as well as imputation methods, including single imputation (such as mean or conditional mean) and multiple imputation methods.

An alternative approach to handling incomplete data entails obtaining maximum likelihood estimates of the mean vector and covariance matrix for a set of variables. These estimates are obtained in PROC MI by using an iterative expectation-maximization algorithm that will be described later.

The resulting estimates of μ and Σ can be used as input for a variety of multivariate analyses. They can also be used as starting values for MCMC multiple imputation. Finally, they can be used to obtain a single imputed data set based on the estimates. The EM algorithm estimates parameters in such a way as to account for dependencies in missingness among the variables (MAR), and provides realistic variance and covariance estimates where data are not completely observed.

THE EXPECTATION MAXIMIZATION (EM) ALGORITHM ¹

In general, EM iterates through two steps to obtain estimates. The first step is an Expectation (E) step, in which missing values are filled-in with a guess, that is, an estimate of the missing value, given the observed values in the data. The second step is a Maximization (M) step, in which the completed data from the E step are processed using ML estimation as though they were complete data, and the mean and covariance estimates are updated. Using the newly updated mean and covariance matrix, the E step is repeated to find new estimates of the missing values. These two steps (E and M) are repeated until the maximum change in the estimates from one iteration to the next does not exceed a convergence criterion. The result of this process is a mean vector and covariance matrix that uses all available information.

The EM algorithm assumes that data are multivariate normal and that missingness is MAR. The EM estimates of the mean vector and covariance matrix can then be used in multivariate analyses to obtain estimates of model parameters and standard errors, to test hypotheses, and to score or predict values for observations using the model selected.

USING EM COVARIANCE ESTIMATES

One problem emerges in using the EM covariance estimates as input to subsequent analyses. Although the estimates account for the incomplete data in estimating μ and Σ , model standard error estimates might still be

biased when the covariance matrix is treated as though it came from complete data. One solution is to use the EM estimates of μ and Σ and specify the nominal sample size for the analysis $n < N$.²

SPECIFYING A NOMINAL N FOR USE WITH EM $\hat{\Sigma}$ AND $\hat{\mu}$

Specifying an appropriate nominal sample size is not a trivial issue. For example, if the complete data sample size is used, then the analysis fails to fully account for uncertainty due to missingness. At the other extreme, one can choose to use the complete-case count as the sample size. However, this is unlikely to be helpful and might produce drastically overestimated standard errors.

Specifying an appropriate sample size would account for uncertainty due to missingness while also accounting for the observed data. There are a number of ways you can define a nominal sample size that might be appropriate. One purpose of this paper is to compare several different sample size specifications to determine which specifications consistently produce better results. By "better," it is meant that the results come as close as possible to the true, complete-data values.

Three sample size specifications are considered in this paper. In order to illustrate the three methods of selecting a nominal sample size, consider the data pattern in Table 1.

In Table 1, (x) represents observed data and (.) represents missing data. There are 15 observations, and only 2 complete observations. X1 is observed for 13 cases, X2 is observed for 9 cases, X3 is observed for 11 cases, and X4 is observed for 10 cases.

X1	X2	X3	X4
x	x	x	x
x	.	x	x
.	x	x	x
.	x	x	.
x	x	.	x
x	x	x	x
x	x	x	.
x	x	x	.
x	x	.	x
x	.	.	.
x	.	.	x
x	.	x	x
x	.	x	x
x	.	x	x
x	x	x	.

Table 1. Sample Data Pattern

Column-wise minimum n

First, the minimum number of complete observations in a variable can be used. For example, in the preceding data set, X2 has the most missingness (with only 9 cases observed). This could be the nominal sample size for the analysis. This is referred to as column-wise minimum n .

Column-wise average n

You could also use the arithmetic mean of the number of cases observed for each variable. In the preceding example, this would result in

$$\frac{13+9+11+10}{4} = 10.75$$

In the analysis, you could use the integer part and specify a nominal sample size of 10 cases. This is referred to as column-wise average n .

Pairwise minimum n

Finally, the minimum number of pairwise complete cases could be used. In the example above, construct a table of pairwise complete data indicators, as shown in Table 2.

In the first observation, all values are observed so all indicators = 1. In the second observation, X2 is missing, so all pairwise indicators involve X2=0, for example, x12, x23, x24. The sum of each indicator column provides the number of pairwise cases that are observed in the data.

8	10	9	7	5	7
---	----	---	---	---	---

As a conservative nominal sample size estimate, you could select the worst-case scenario from these values, and use the minimum number of pairwise observed cases in the data. This is equivalent to using the smallest sample size from a correlation matrix of the variables using available case analysis. This is referred to as pairwise minimum n .

EM IMPUTED DATA

A simpler option is to use the EM estimates to produce a single data set with imputed values based on the EM estimates. In other words, the point estimate for each missing value, based on the EM estimates of the means and covariances, are filled in. This is a single imputation method and, therefore, does not account for uncertainty due to missingness, but it can produce good estimates of the true missing values. This method can be performed with PROC MI using the EM statement.

	x12	x13	x14	x23	x24	x34
1	1	1	1	1	1	1
0	1	1	0	0	1	
0	0	0	1	1	1	
1	1	0	1	0	0	
1	0	1	0	1	0	
1	1	1	1	1	1	
1	1	0	1	0	0	
1	1	0	1	0	0	
1	0	1	0	1	0	
0	0	0	0	0	0	
0	0	1	0	0	0	
0	1	1	0	0	1	
0	1	1	0	0	1	
0	1	1	0	0	1	
1	1	0	1	0	0	

Table 2. Pairwise Complete Data Indicators

THE MISSING DATA SIMULATION ³

The population parameters were defined as mean vectors and covariance matrices computed from 150 complete cases of iris measurements published by Fisher (1936). The original data consist of 50 observations from each of three species of iris (Setosa, Virginica, Versicolor) and four measurements taken on each flower (Petal length, Petal width, Sepal length, Sepal width). The measurements can be used to predict differences among the species. The three resulting covariance matrices and mean vectors were used to simulate data values for use in the study.

A series of SAS macros were used to perform the simulations. Data were generated from the three sets of mean and covariance parameters described above. At each trial, two multivariate normal data sets, each with 3000 observations (1000 per species) were generated using a specified seed.

The primary macro for the simulation includes the macro parameter NTRIALS, which specifies the number of times the simulation is performed using a new seed to generate the data each time. For this study, ntrials = 100, which indicates that 100 trials were performed. In other words, two data sets (N=3000 each) were generated (a calibration data set and a validation data set), and the analyses were performed on them 100 times.

DETAILS

A pattern of missingness was imposed (MAR) on the calibration data. The validation data were complete data for scoring and error count at the validation step of each trial.

Eight versions of the calibration data set were created and compared, as described in Table 3.

Data set #	Description	Label
1	Complete data	Complete
2	Incomplete data	CCA
3	Mean imputed	Mean Imp
4	Conditional mean imputed	CondMean
5	EM imputed	EM Impute
	EM Covariance matrix and mean vector:	
6	Column-wise minimum sample size	EMCovMinN
7	Column-wise average sample size	EMCovPwiseN
8	Pairwise minimum sample size	EMCovAvgN

Table 3. Comparison of the Calibration Data Set

Discriminant function analysis, evaluating differences among the three species of iris as a function of the four measurements, was chosen for the present study. PROC DISCRIM makes it simple to evaluate inferential as well as predictive accuracy using calibration and validation data sets. The results of canonical and linear discriminant analyses were compared for the eight data sets.

Statistics chosen for comparison in canonical discriminant analysis included the two canonical correlation coefficients and their approximate standard errors. These statistics evaluate the extent to which various methods over- or under-estimate strength of association (in this case, group differences) and uncertainty in estimating the parameters. The results of these comparisons are shown in Figures 1 and 2 and Tables 4 and 5.

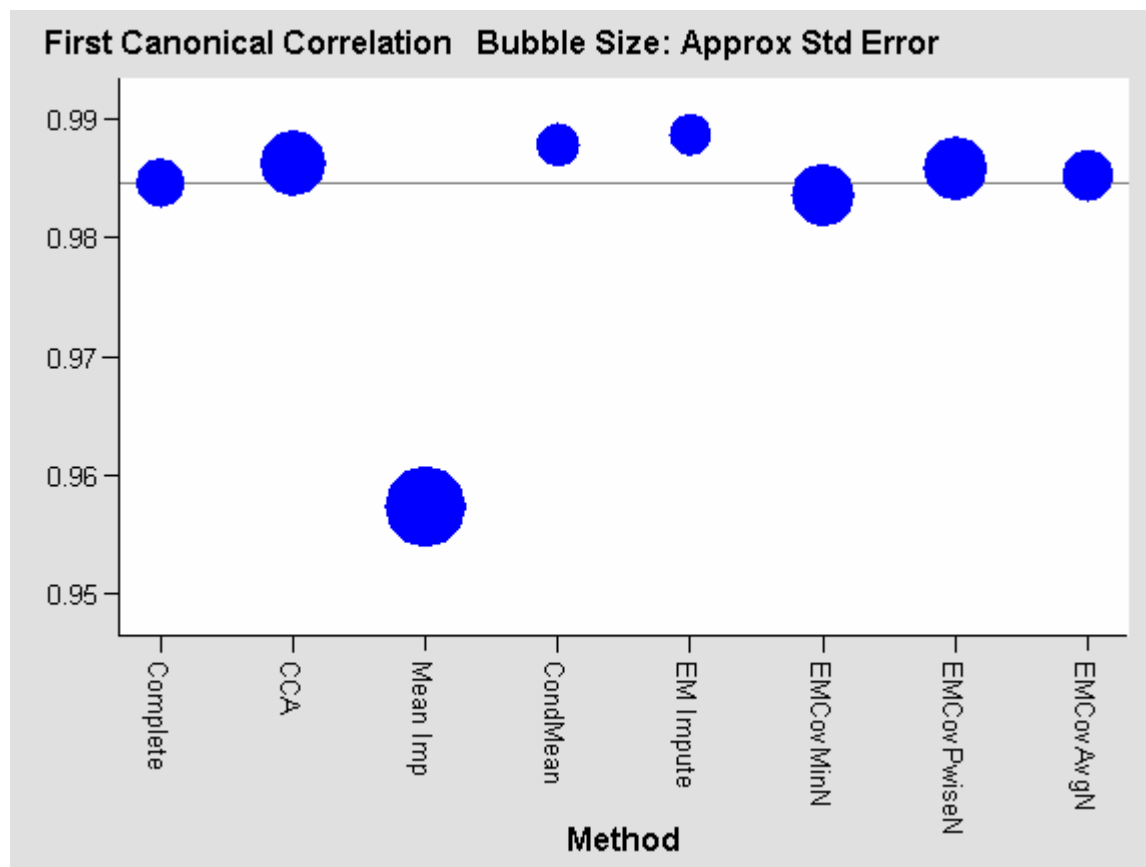


Figure 1. Bubble Plot—First Canonical Correlation and Approximate Standard Errors

The bubble plot in Figure 1 shows the first canonical correlation and the approximate standard errors. The vertical axis shows the association between the first canonical variate (an optimally weighted linear combination of the four measurement variables that maximally separates the group centroids) and the largest dimension of inter-group distances, which measures the strength of the first canonical discriminant function. The size of the bubble represents the approximate standard error of the canonical correlation.

The bubble on the left shows the results for complete data, which we will consider the “true” value. All other methods are compared to this value, which is marked by the reference line on the plot. The worst result of the remaining seven methods is clearly mean imputation. Mean imputation consistently underestimated the first canonical correlation. In other words, the mean imputation did not perform well in finding differences among the three groups.

Recall that mean imputation assumes that data are MCAR, which was not the case with the data used in this simulation. Furthermore, the bubble for mean imputation is considerably larger than for complete data, which demonstrates the inflated standard error estimate for the first discriminant function.

Other methods performed reasonably well, although conditional mean imputation and single imputation using the EM results (EM Impute) tended to slightly overestimate the first canonical correlation. In other words, these methods tended to exaggerate differences among the groups. These methods also underestimated the standard errors as shown by the relatively small bubbles on the plot.

CONCLUSIONS BASED ON THE FIRST CANONICAL CORRELATION

CCA and the three methods using the EM covariance matrix gave results closest to the true values. Of these, the EM covariance matrix with average column-wise n gave estimates closest to the true values with the closest standard error estimate. The EM covariance matrix approach produced results superior to traditional methods.

For the first canonical discriminant function, the EM covariances using average column-wise n produced results that were the closest to the true values.

Method	N Obs	Variable	Label	Mean	Range
Complete	100	CanCorr StdErr	Canonical Correlation Approximate Standard Error	0.9846417 0.000556589	0.0020641 0.000074223
CCA	100	CanCorr StdErr	Canonical Correlation Approximate Standard Error	0.9863056 0.0010091	0.0038089 0.000296005
Mean Imp	100	CanCorr StdErr	Canonical Correlation Approximate Standard Error	0.9573462 0.0015245	0.0068328 0.000238853
CondMean	100	CanCorr StdErr	Canonical Correlation Approximate Standard Error	0.9878325 0.000441661	0.0028462 0.000102712
EM Impute	100	CanCorr StdErr	Canonical Correlation Approximate Standard Error	0.9886902 0.000410700	0.0036658 0.000132374
EMCovMinN	100	CanCorr StdErr	Canonical Correlation Approximate Standard Error	0.9835979 0.000909969	0.0040432 0.000210391
EMCovPwiseN	100	CanCorr StdErr	Canonical Correlation Approximate Standard Error	0.9858507 0.000940563	0.0031488 0.000213132
EMCovAvgN	100	CanCorr StdErr	Canonical Correlation Approximate Standard Error	0.9852469 0.000617928	0.0032350 0.000132907

Table 4. Statistics on First Canonical Correlation

The means shown on the bubble plot are reproduced in Table 4 for ease of comparison. In addition, the range of each statistic for the 100 trials is displayed in Table 4. Notice that mean imputation showed the most sampling variability in estimating the first canonical correlation. CCA and mean imputation showed the largest variability for estimating the approximate standard errors.

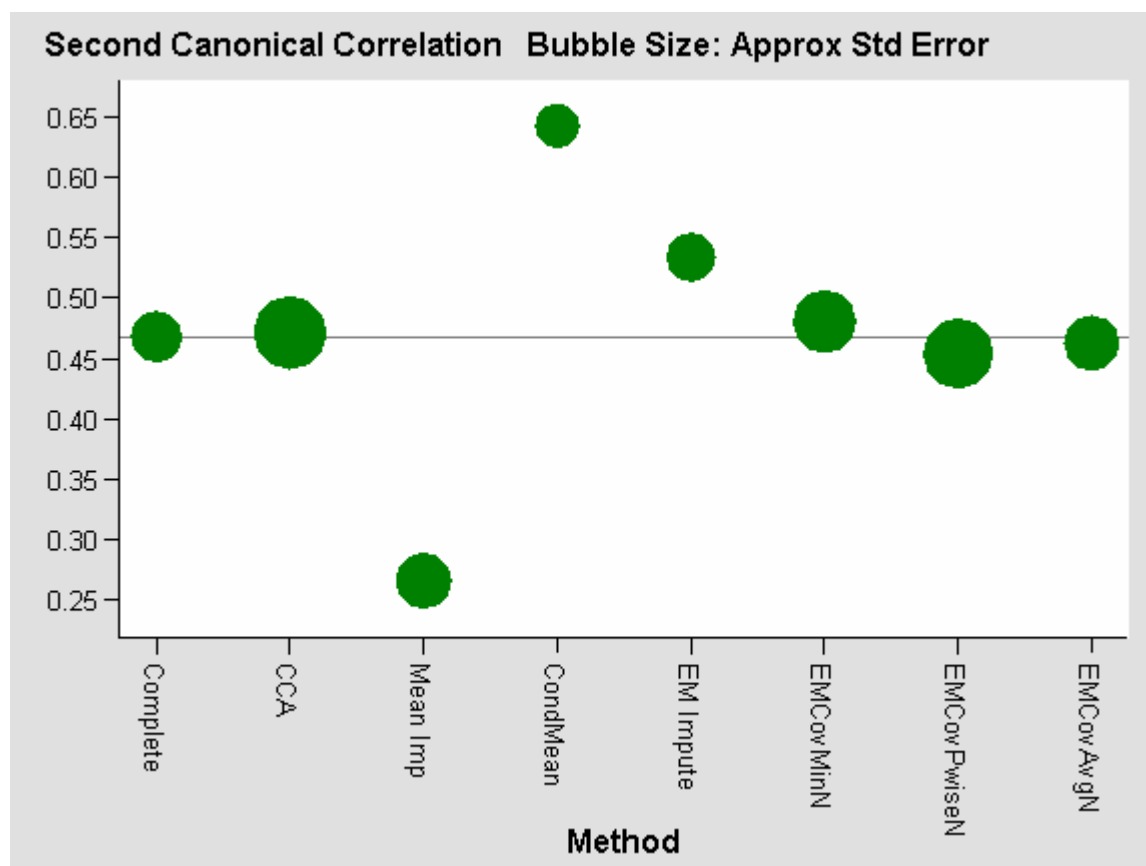


Figure 2. Bubble Plot— Second Canonical Correlation and Approximate Standard Errors

The second canonical correlation, showing the strength of the second discriminant function in differentiating among group centroids, is compared in Figure 2. The reference line shows the value for the complete data. CCA and the three EM covariance methods show estimates that are reasonably close to the true values.

The two worst performers were mean imputation, which consistently underestimated the second canonical correlation, and conditional mean imputation, which consistently overestimated the second canonical correlation. EM imputation also overestimated the second canonical correlation.

Again, the size of the bubble reflects the approximate standard error of the second canonical correlation. CCA, EM Covariance with column-wise minimum n , and EM covariance with pairwise minimum n overestimated the standard errors. Conditional mean underestimated the standard errors. Although mean imputation and EM imputation produced standard errors that were reasonably close to the true values, neither method produced accurate estimates of the second canonical correlation.

CONCLUSIONS BASED ON THE SECOND CANONICAL CORRELATION

The EM covariance with average column-wise n produced reasonably close standard error estimates, and also produced accurate estimates of the second canonical correlation.

Therefore, the EM covariances with average column-wise n produced the best results of the seven comparison methods.

Method	N Obs	Variable	Label	Mean	Range
Complete	100	CanCorr	Canonical Correlation	0.4680049	0.0460550
		StdErr	Approximate Standard Error	0.0142597	0.000779815
CCA	100	CanCorr	Canonical Correlation	0.4713221	0.1042737
		StdErr	Approximate Standard Error	0.0288409	0.0047465
Mean Imp	100	CanCorr	Canonical Correlation	0.2657742	0.0616709
		StdErr	Approximate Standard Error	0.0169683	0.000603475
CondMean	100	CanCorr	Canonical Correlation	0.6427157	0.0643952
		StdErr	Approximate Standard Error	0.0107142	0.0015098
EM Impute	100	CanCorr	Canonical Correlation	0.5338702	0.0734526
		StdErr	Approximate Standard Error	0.0130519	0.0014227
EMCovMinN	100	CanCorr	Canonical Correlation	0.4804976	0.0627530
		StdErr	Approximate Standard Error	0.0215091	0.0019650
EMCovPwiseN	100	CanCorr	Canonical Correlation	0.4540960	0.0609207
		StdErr	Approximate Standard Error	0.0265697	0.0023213
EMCovAvgN	100	CanCorr	Canonical Correlation	0.4625850	0.0621928
		StdErr	Approximate Standard Error	0.0165808	0.0012217

Table 5. Statistics on Second Canonical Correlation

Statistics summarizing the previous graph are displayed in Table 5 for ease of comparison. Trial-to-trial variability was larger for CCA than for complete data or for the three EM Covariance methods.

From the perspective of statistical inference and parameter estimation, it appears that EM covariances with average column-wise n outperformed other methods. Mean imputation produced the worst estimates of the canonical correlations. CCA and mean imputation performed worst in estimating standard errors.

It is interesting and important to note that simply changing the nominal sample size specification method for the EM covariance matrix had an impact on the statistics that were reported.

PREDICTING A NEW SAMPLE

In some cases, the purpose of a statistical analysis is to find a model that maximizes predictive accuracy, for example, determining the best marketing strategy to use or finding the best way to identify high-risk customers. In predictive modeling, the researcher is interested in predicting responses outside of a training data set, and requires adequate model fitting capabilities to obtain good estimates of parameters.

Canonical discriminant analysis provides good information about the extent to which the methods lead to incorrect inference or biased estimates of the strength of association. However, in many data analysis situations, such as predictive modeling, it is equally (perhaps more) important that models have the ability to accurately predict responses in the population independently of the sample used to fit the model.

In order to evaluate the predictive usefulness of the various incomplete data methods, linear discriminant analysis was performed, and a second sample (the *validation data*) of $N=3000$ complete observations, independently sampled from the population, were scored using the model from the original (*calibration*) sample. Misclassifications were counted, and using an uninformative (equal) prior, expected error rates in classification were summarized. Comparisons are shown in Figure 3 and Table 6.

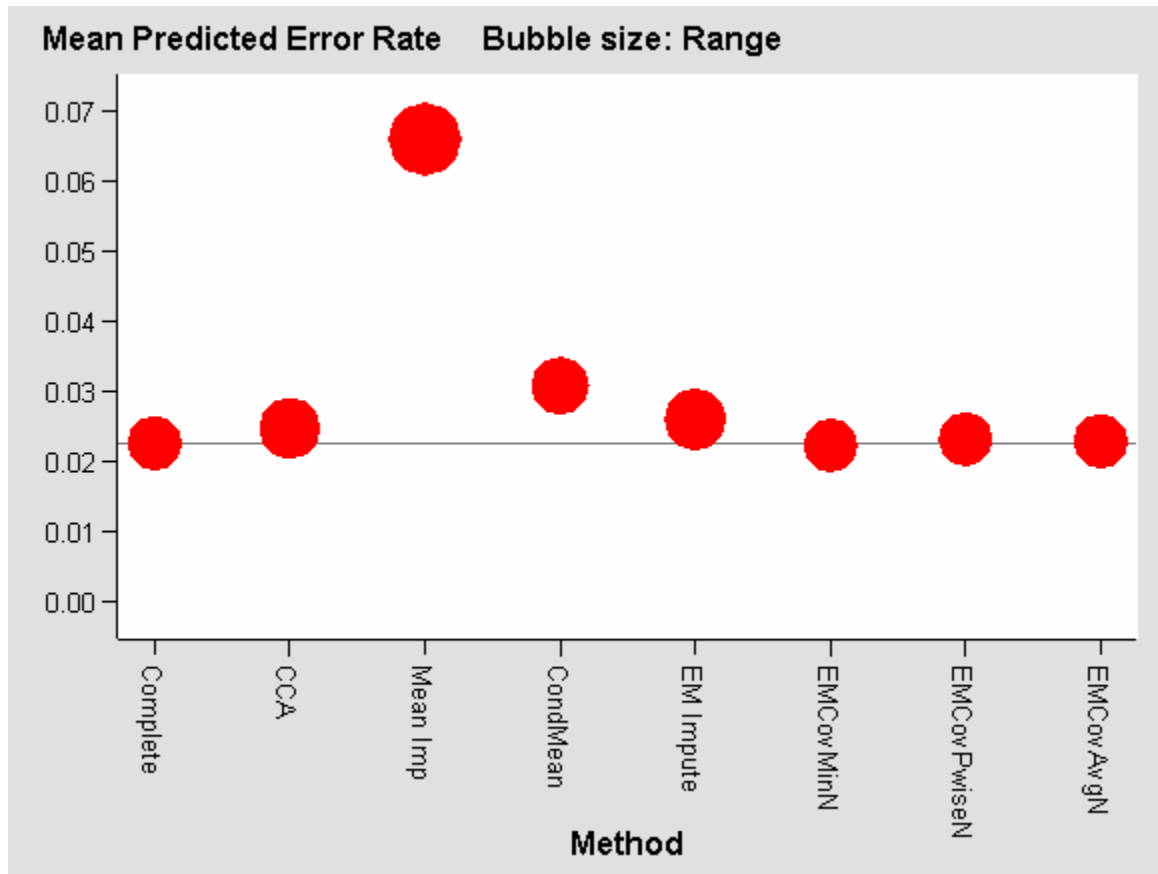


Figure 3. Bubble Plot—Mean Predicted Error

The calibration model from complete data showed just over 2% of the validation data misclassified. Mean imputation performed consistently worse than all other methods, with over 6% of the validation data misclassified. The complete data and the three EM covariance methods produced the models with the lowest error rates in classification.

Analysis Variable : Total			
Method	N Obs	Mean	Range
Complete	100	0.0225933	0.0140000
CCA	100	0.0247133	0.0176667
Mean Imp	100	0.0660233	0.0246667
CondMean	100	0.0307967	0.0156667
EM Impute	100	0.0259933	0.0180000
EMCovMinN	100	0.0222833	0.0136667
EMCovPwiseN	100	0.0231700	0.0136667
EMCovAvgN	100	0.0228700	0.0140000

Table 6. Statistics on Mean Predicted Error for Validation Data

It is interesting to note that two methods investigated here, which are commonly used in predictive modeling (mean imputation and conditional mean imputation), performed worse than all other methods in predicting data from a new sample. It might be beneficial for predictive modelers to consider the alternative methods described in this paper when devising plans to handle incomplete data in their own work.

GENERAL CONCLUSIONS

Taken as a whole, using the ML covariance matrices and mean vectors, which were produced by PROC MI using the EM algorithm, outperformed many other methods in inferential and predictive modeling applications. An important lesson from this study is the importance of specifying an appropriate nominal sample size for an analysis. This is a task that the analyst should not take lightly. In the present study, using the arithmetic mean of the column-wise observed n outperformed two other candidate approaches to specifying n . However, it is reasonable that other specifications of n , for example, a geometric or harmonic mean, might outperform the arithmetic mean. Such is a topic for future research.

There are many other comparisons you might want to make using the paradigm presented in this paper. Such comparisons represent a trivial amount of work from a programming and computational perspective using SAS software and, with threaded technology, such simulations could be even faster. This study does not purport to provide definitive answers as to the best way to handle incomplete data or the best way to specify nominal sample size. Other methods and choices for specifying n should be investigated. Furthermore, the present study only evaluates the methods using clean, multivariate normal data with a strong predictive model. Different findings might result from non-normal data or poorly specified models.

Nonetheless, whether the goal is inferential statistics or predictive modeling, the modern tools that are available in PROC MI can provide solutions for incomplete data that consistently outperform simpler methods such as conditional mean imputation and mean imputation. Even if multiple imputation is not a viable option in a particular data analytic situation, statisticians and predictive modelers might want to consider the tools available from PROC MI when choosing a strategy to handle incomplete data.

APPENDIX—FOUR PROGRAMS TO PERFORM THE SIMULATION

PROGRAM 1: FIND MINIMUM COLUMN-WISE *N*

```
%macro nmin;
proc means data=irismiss1 n noprint;
    var SepalLength SepalWidth PetalLength PetalWidth;
    by Species;
    output out=n(drop = _TYPE_ _FREQ_) n=;
run;
proc transpose data = n out = nmin;
id species;
run;

proc sql noprint;
select min(setosa) into :n1_1 from nmin;
select min(versicolor) into :n2_2 from nmin;
select min(virginica) into :n3_3 from nmin;
quit;

data nminimum;
_TYPE_ = 'N';
format SepalLength SepalWidth PetalLength PetalWidth 4.0;
do species = 1 to 3;
    if species = 1 then do;
        SepalLength = &n1_1;
        SepalWidth = &n1_1;
        PetalLength = &n1_1;
        PetalWidth = &n1_1;
    end;
    else if species = 2 then do;
        SepalLength = &n2_2;
        SepalWidth = &n2_2;
        PetalLength = &n2_2;
        PetalWidth = &n2_2;
    end;
    else do;
        SepalLength = &n3_3;
        SepalWidth = &n3_3;
        PetalLength = &n3_3;
        PetalWidth = &n3_3;
    end;
    output;
end;
run;
%mend nmin;
```

PROGRAM 2: FIND AVERAGE COLUMN-WISE *N*

```
%macro navg;
proc means data=irismiss1 n noprint;
    var SepalLength SepalWidth PetalLength PetalWidth;
    by Species;
    output out=n(drop = _TYPE_ _FREQ_) n=;
run;
proc transpose data = n out = navg;
id species;
run;

proc sql noprint;
select mean(setosa) into :n1_1 from navg;
select mean(versicolor) into :n2_2 from navg;
select mean(virginica) into :n3_3 from navg;
quit;
```

```

data naverage;
  _TYPE_ = 'N';
  format SepalLength SepalWidth PetalLength PetalWidth 4.0;
  do species = 1 to 3;
    if species = 1 then do;
      SepalLength = &n1_1;
      SepalWidth = &n1_1;
      PetalLength = &n1_1;
      PetalWidth = &n1_1;
    end;
    else if species = 2 then do;
      SepalLength = &n2_2;
      SepalWidth = &n2_2;
      PetalLength = &n2_2;
      PetalWidth = &n2_2;
    end;
    else do;
      SepalLength = &n3_3;
      SepalWidth = &n3_3;
      PetalLength = &n3_3;
      PetalWidth = &n3_3;
    end;
    output;
  end;
run;
%mend navg;

```

PROGRAM 3: FIND MINIMUM PAIRWISE *N*

```

%macro npair;
data new; set irismiss1;
a = (SepalLength ne .);
b = (SepalWidth ne .);
c = (PetalLength ne .);
d = (PetalWidth ne .);
x1 = a*b;
x2 = a*c;
x3 = a*d;
x4 = b*c;
x5 = b*d;
x6 = c*d;
run;

ods output summary = npair;
ods listing exclude ALL;
proc means data = new sum;
var x1 - x6;
by species;
run;

proc transpose data = npair out = np;
id species;
run;

proc sql noprint;
select min(setosa) into :n1 from np;
select min(versicolor) into :n2 from np;
select min(virginica) into :n3 from np;
quit;

data npairwise;
  _TYPE_ = 'N';
  do species = 1 to 3;
    if species = 1 then do;
      SepalLength = &n1;

```

```

    SepalWidth = &n1;
    PetalLength = &n1;
    PetalWidth = &n1;
end;
else if species = 2 then do;
    SepalLength = &n2;
    SepalWidth = &n2;
    PetalLength = &n2;
    PetalWidth = &n2;
end;
else do;
    SepalLength = &n3;
    SepalWidth = &n3;
    PetalLength = &n3;
    PetalWidth = &n3;
end;
output;
end;
run;
%mend npair;

```

PROGRAM 4: PRIMARY MACRO FOR THE STUDY (USES 3 COVARIANCE MATRICES FROM FISHER'S (1936) IRIS DATA SET, ONE FOR EACH SPECIES)

```

%macro doit (ntrials = 1, s1 = 100, s2 = 1000);

/* generate MVN data */
%do i = 1 %to &ntrials;
%let seed1 = &i + &s1;
%let seed2 = &i + &s2;
%mvn(varcov=sugi30.iriscov1, means=sugi30.imean1,
n=&num, seed=&seed1, sample=setosa);
%mvn(varcov=sugi30.iriscov2, means=sugi30.imean2,
n=&num, seed=&seed1, sample=versicolor);
%mvn(varcov=sugi30.iriscov3, means=sugi30.imean3,
n=&num, seed=&seed1, sample=virginica);

/*generate validation data*/
%mvn(varcov=sugi30.iriscov1, means=sugi30.imean1,
n=&num, seed=&seed2, sample=setosa_v);
%mvn(varcov=sugi30.iriscov2, means=sugi30.imean2,
n=&num, seed=&seed2, sample=versicolor_v);
%mvn(varcov=sugi30.iriscov3, means=sugi30.imean3,
n=&num, seed=&seed2, sample=virginica_v);

proc format;
value specname
1='Setosa '
2='Versicolor'
3='Virginica ';
run;

data iris1 (rename= (col1=SepalLength col2=SepalWidth col3=PetalLength
col4=PetalWidth));
set setosa versicolor virginica;
if _N_ le &num then species = 1;
else if _N_ le 2 * &num then species = 2;
else species = 3;
format species specname.;
label SepalLength='Sepal Length in mm.'
SepalWidth='Sepal Width in mm.'
PetalLength='Petal Length in mm.'
PetalWidth='Petal Width in mm.';
symbol = put(species, specname10.);

```

```

run;

data iris2 (rename= (coll=SepalLength col2=SepalWidth col3=PetalLength
col4=PetalWidth));
  set setosa_v versicolor_v virginica_v;
  if _N_ le &num then species = 1;
  else if _N_ le 2 * &num then species = 2;
  else species = 3;
  format species specname.;
  label SepalLength='Sepal Length in mm.'
        SepalWidth='Sepal Width in mm.'
        PetalLength='Petal Length in mm.'
        PetalWidth='Petal Width in mm.';
  symbol = put(species, specname10.);
run;

/* generate missingness */
%let seed /*for random cutting*/= &i*50;

data irismiss1;
  set iris1;
  cut1=ranuni(&seed);
  cut2=ranuni(&seed);
  cut3=ranuni(&seed);
  if cut1 lt .6 and sepallength lt 65 then sepalwidth = .;
  if cut2 lt .6 and petallength gt 40 then petalwidth = .;
  if cut3 lt .15 then sepallength = .;
  if cut1 gt .95 then petallength = .;
  drop cut1 cut2 cut3;
run;

/*impute the unconditional mean for each variable*/
proc stdize data = irismiss1 out = meanimpiris1 method = mean reponly;
  var SepalLength SepalWidth PetalLength PetalWidth;
run;

/*impute the conditional mean for each variable*/
proc stdize data = irismiss1 out = cmeanimpiris1 method = mean reponly;
  var SepalLength SepalWidth PetalLength PetalWidth;
  by species;
run;

proc mi data = irismiss1 noprint;
  var SepalLength SepalWidth PetalLength PetalWidth;
  em out = emiris1 outem = emcoviris1;
  mcmc /*these options are here to reduce computation
time because I am not using MCMC*/ nbiter = 0 niter = 0;
  by Species;
run;

%npair;
data emcovpair;
  set emcoviris1 npairwise;
run;

%navg;
data emcovmean;
  set emcoviris1 naverage;
run;

%nmin;
data emcovmin;
  set emcoviris1 nminimum;
run;

ods listing exclude ALL;

```

```

ods output cancorr(persist = proc)= cancorr_&i;
ods output ErrorTestClass(persist = proc)= prederror_&i;

/* discriminant analysis on the data sets */
proc discrim data = iris1 can testdata = iris2;
title '1: Complete data';
  var SepalLength SepalWidth PetalLength PetalWidth;
  class Species;
run;

proc discrim data = irismiss1 can testdata = iris2;
title '2: Incomplete data (CCA)';
  var SepalLength SepalWidth PetalLength PetalWidth;
  class Species;
run;

proc discrim data = meanimpiris1 can testdata = iris2;
title '3: Mean imputed data';
  var SepalLength SepalWidth PetalLength PetalWidth;
  class Species;
run;

proc discrim data = cmeanimpiris1 can testdata = iris2;
title '4: Conditional mean imputed';
  var SepalLength SepalWidth PetalLength PetalWidth;
  class Species;
run;

proc discrim data = emiris1 can testdata = iris2;
title '5: EM imputed data';
  var SepalLength SepalWidth PetalLength PetalWidth;
  class Species;
run;

/*this is the minimum number of observations in
a column in the analysis (column-wise minimum n)*/
proc discrim data = emcovmin(TYPE=COV) can testdata = iris2;
title '6: ML COV from EM: Min column N';
  var SepalLength SepalWidth PetalLength PetalWidth;
  class Species;
run;

/*this is the minimum number of pairwise observations
(pairwise minimum n)*/
proc discrim data = emcovpair(TYPE=COV) can testdata = iris2;
title '7: ML COV from EM: Min pairwise columns';
  var SepalLength SepalWidth PetalLength PetalWidth;
  class Species;
run;

/* average number of observations in a column in the analysis
(column-wise average n)*/
proc discrim data = emcovmean(TYPE=COV) can testdata = iris2;
title '8: ML COV from EM: Avg N per column';
  var SepalLength SepalWidth PetalLength PetalWidth;
  class Species;
run;

ods output clear;
ods listing select ALL;
title;

/*combine cancorr results from all runs*/
data can1_&i can2_&i;
  set cancorr_&i;
  if number = 1 then output can1_&i;

```



```

    if number = 2 then output can2_&i;
run;
proc format;
    value method
        1='Complete  '
        2='CCA      '
        3='Mean Imp   '
        4='CondMean  '
        5='EM Impute  '
        6='EMCovMinN '
        7='EMCovPwiseN'
        8='EMCovAvgN ' ;
run;

data can1_&i;
set can1_&i;
Method = _N_;
format Method method.;
run;
data can2_&i;
set can2_&i;
Method = _N_;
format method method.;
run;

data prederror_&i;
set prederror_&i;
where type = 'Rate';
Method = _N_;
format Method method.;
run;

%if &i gt 1 %then %do;
    proc append base = can1_1 data = can1_&i;
    run;
    proc append base = can2_1 data = can2_&i;
    run;
    proc append base = prederror_1 data = prederror_&i;
    run;
    proc sql;
        drop table can1_&i, can2_&i, cancorr_&i, prederror_&i;
    quit;
%end;

%end;
%mend doit;

%doit(ntrials = 100, s1 = 500, s2 = 6000);
%let extradata = npairwise, new, npair, np, iris1,
                 irismiss1, meanimpiris1, cmeanimpiris1,
                 n, emiris1, emcoviris1, emcovmin,
                 emcovpair, emcovmean, naverage,
                 navg, setosa, versicolor, virginica;

goptions dev = activex;
ods listing close;
ods html;
proc means data = can1_1 mean range nway;
var cancorr stderr;
class Method;
output out = meanc mean=;
run;
proc means data = can2_1 mean range nway;
var cancorr stderr;
class Method;
output out = meanc2 mean=;
run;

```

```

proc means data = prederror_1 mean range nway;
var total;
class Method;
output out = meanc3 mean=total range = range;
run;

proc sql;
create table sugi30.simcombined as
  select meanc.Method format = method.,
         meanc.cancorr as cancorr1,
         meanc.stderr as stderr1,
         meanc2.cancorr as cancorr2,
         meanc2.stderr as stderr2,
         meanc3.total as total,
         meanc3.range as range
  FROM meanc AS meanc
        INNER JOIN meanc2 AS meanc2 ON (meanc.method = meanc2.method)
        INNER JOIN meanc3 AS meanc3 ON (meanc.method = meanc3.method);
quit;

Axis1 MINOR=NONE
      LABEL=(HEIGHT=12pt JUSTIFY=Left
            'First Can Corr      Bubble Size: Approx Std Error ');
Axis2 MINOR=NONE VALUE = (A = -90)
      LABEL=(HEIGHT=12pt JUSTIFY=Center 'Method');
Axis3 MINOR=NONE
      LABEL=(HEIGHT=12pt JUSTIFY=Right
            'Second Can Corr');

PROC GPLOT DATA = sugi30.simcombined;
BUBBLE CanCorr1      * method      = StdErr1
  / bcolor = blue VAXIS=AXIS1 HAXIS=AXIS2 BSIZE=20
  NOFRAME vref = 0.9846417;
BUBBLE CanCorr2 * method = StdErr2
  / bcolor = green VAXIS=AXIS3 BSIZE=20
  NOFRAME vref = 0.4680049;

RUN; QUIT;
title;

Axis1 MINOR=NONE
      LABEL=(HEIGHT=12pt JUSTIFY=Left
            'Mean Predicted Error Rate      Bubble size: Range');
TITLE;
symbol v = plus c = red;
PROC GPLOT DATA = meanc3;
BUBBLE total * method      = RANGE
  / bcolor = red VAXIS=AXIS1 HAXIS=AXIS2 BSIZE=20
  NOFRAME VZERO VREF=.0225993 ;
RUN; QUIT;

goptions reset = all;
title;
ods html close;
ods listing;

proc sql;
  drop table &extradata;
quit;

```

ENDNOTES

1. The reader can find details about the EM algorithm and maximum likelihood estimation in the literature (Cox & Hinkley, 1974; Rubin & Little, 1987; Schafer, 1997).
2. Another solution is to use a method known as Direct ML (see Allison, 2002), which is beyond the scope of this paper.
3. The programs used to perform the simulation are shown in the Appendix. The MVN macro, not shown here, is available from the SAS technical support website, <http://support.sas.com/>

REFERENCES

Allison, P.D. (2002). *Missing Data*. (Sage University Papers Series on Quantitative Applications in the Social Sciences, series no. 07-136). Thousand Oaks, CA: Sage.

Buck, S.F. (1960). "A method of estimation of missing values in multivariate data suitable for use with an electronic computer". *Journal of the Royal Statistical Society, B22*, 302-306.

Cox, D.R. & Hinkley, D.V. (1974). *Theoretical Statistics*. New York: Wiley.

Fisher, R.A. (1936). "The use of multiple measurements in taxonomic problems". *Annals of Eugenics, 7*, 179 - 188.

Haitovsky, Y. (1968). "Missing data in regression analysis". *Journal of the Royal Statistical Society, B30*, 67-82.

Little, R.J.A. (1992). "Regression with missing Xs: A review" *Journal of the American Statistical Association, 88*, 125-134.

Little, R.J.A. & Rubin, D.B. (1983). On jointly estimating parameters and missing data by maximizing the complete-data likelihood. *The American Statistician, 37*, 218-220.

Little, R.J.A. & Rubin, D.B. (1987). *Statistical Analysis with Missing Data* (Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics). New York: Wiley.

Rubin, D.B. (1976). "Inference and missing data". *Biometrika, 63*, 581-592.

Schafer, J.L. (1997). *Analysis of Incomplete Multivariate Data* (Monographs on Statistics and Applied Probability 72). Boca Raton, FL: Chapman & Hall/CRC.

ACKNOWLEDGEMENTS

The author would like to thank Rob Agnelli, Anthony Mancuso, and Mike Patetta for their suggestions on an earlier draft of this paper and on the simulation programs.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author:

Catherine Truxillo, Ph.D.
 SAS Institute Inc.
 Cary, NC 27513
 919-531-4641
Catherine.Truxillo@sas.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.