This paper is organized into eight different sections. First the mechanisms of missing data, i.e. the types of missingness, and patterns of missing data are introduced, followed by a brief introduction to simple methods for handling missing data. A simulation is done to compare the methods for a Missing Completely at Random dataset. The Expectation Maximization algorithm is then discussed followed by an introduction to Multiple Imputation and imputation algorithms for Missing at Random data with a Monotone Missing Pattern. A tennis dataset with a Monotone Missing Pattern was found to demonstrate the Expectation Maximization algorithm and the Multiple Imputation algorithms previously discussed. Statistics unique to Multiple Imputation are also discussed, in the results and discussion section for this application. Finally the programming scripts used in this final project are revealed. Excel VBA, R and SAS were all used to obtain results.

**Types of Missingness**

For any method of handling missing data, it’s important to identify or make assumptions about the way the data is missing. In general, there are three ways that a dataset can have missing values. The values may be missing completely at random, missing at random, or missing not at random.

Data is missing completely at random (MCAR) if the missingness of a variable does not depend upon any variables, observed or unobserved in the dataset. This definition can be made more precise with an example. Suppose that a dataset has only one missing variable , which has a total of missing and observed values. Define to be an indicator variable for the missingness of, where if the observation is missing and otherwise for . Let indicate the completely observed variables from the dataset. The data is missing completely at random if and only if .

Continuing this example, whenever , the data is missing at random (MAR). In other words, data is missing at random whenever the missingness of a variable does not depend on the variable with missing values. With this definition, missing completely at random is just a special case of missing at random.

If we allow the missing data to depend on the variable with missing data, then the data is called missing not random (MNAR). Continuing the example from before, the data is missing not at random if we cannot simplify . While there are some special situations MNAR data is desirable, many methods of imputation rely on a MAR assumption. It is unfortunate that, currently, there is no way to verify the MAR assumption for a dataset , but this assumption is more believable as the size of the dataset and number of variables increases (Wikipedia, “Missing Data”).

Complementing the types of missingness are the various types of missing patterns that may be encountered. The missing data patterns should not be confused with the mechanism for missing data e.g. MCAR. Missing data patterns are not probabilistic statements, just simple ways to describe a dataset.

If there is only one variable missing in the dataset, then the data has a Univariate Missing Pattern. When only one group of one or more variables are always simultaneously missing, then the dataset has a Unit Non-Response Pattern. In other words, rather than having a single missing variable, a group of variables, i.e. a unit, is acting as the one missing variable. *Figure 1* shows pictures to illustrate the idea.

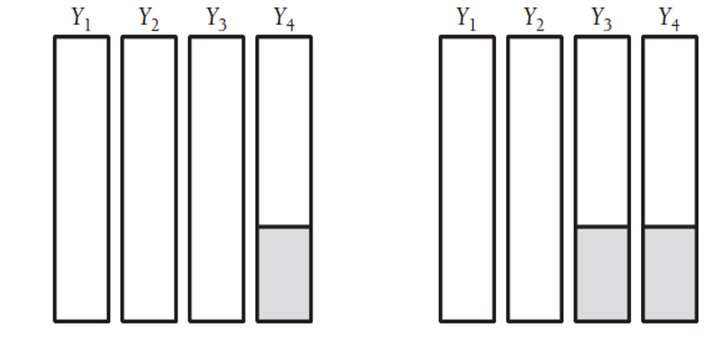


Figure : Univariate & Unit Non-Response Patterns

A Monotone Missing Pattern and Arbitrary Missing Pattern are shown in *Figure 2*. A Monotone Missing Pattern occurs whenever subsequent variables are missing, following a missing variable. The Monotone Missing Pattern is special, because it allows simpler imputation methods than an Arbitrary Pattern. As the name implies, an Arbitrary Missing Pattern occurs when the data appears missing in a random fashion. Like the image in *Figure 2*, you can think of an Arbitrary Missing Pattern as resembling Swiss cheese.

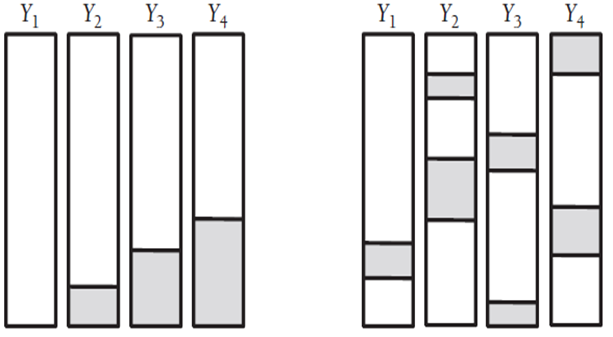
****

Figure : Monotone & Arbitrary Patterns

Two other missing data patterns that can occur are the Planned Pattern and Latent Variable Pattern. These patterns are described by *Figure 3*. Both of these patterns occur intentionally, so they won’t be emphasized in this report.

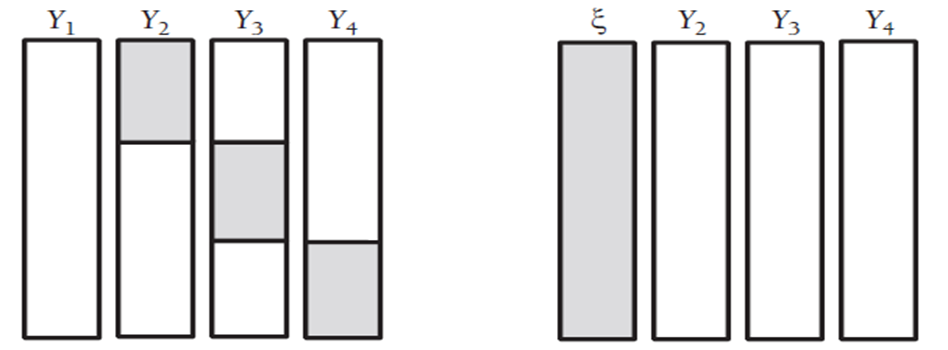


Figure : Planned & Latent Variable Pattern

**Simple Ways for Handling Missing Values**

There are a variety of simple ways for handling missing data. While these methods are not recommended, they have been used in the past.

One of the first things that comes to mind when viewing missing data, is to ignore the problem. List-wise deletion essentially does this. In list-wise deletion, observations which contain missing values are removed from the dataset so analysis can continue. One problem with List-wise deletion, is that it reduces the sample size and therefore the power of a statistical test. In addition, parameter estimates can be biased when the data is not MCAR (Enders, 2010).

Another alternative is to fill in the missing values with estimates. Filling in missing values is called imputation, and will be referred to as imputation throughout the rest of this paper. The simplest methods of imputation are called single imputation, because the missing values are imputed only once. Mean Imputation, Regression Mean Imputation and Stochastic Regression Mean Imputation are all single imputation methods.

Mean Imputation imputes the missing values with the mean of the observed variables. In regression where a design matrix is constructed, the imputed values for a variable would be the column means. While this method is simple, it adversely affects measures of association between covariates. For example, the sample covariance is given by the formula . Replacing a missing value with its mean results in a value of 0 in the summation. Repeatedly replacing missing values with their corresponding means will drastically reduce the sample covariance. In addition biases are present under the assumption of any missing data mechanism (Enders, 2010).

Regression Mean and Stochastic Regression Mean Imputation impute the missing values by regressing the missing variable with completely observed covariates. Regression Mean Imputation imputes the missing variable with the regression mean. For example, if the variable is missing, and is a matrix of completely observed random variables, then regression coefficients, , are estimated from the model , i.e. the multiple least squares model with standard assumptions, using the observed values of . Using the estimates, the missing values for are imputed in using the equation , i.e. the regression mean. Stochastic Regression Mean Imputation would follow the same procedure except the missing values would be imputed with , where is a vector or independent, random draws from a standard normal distribution. The problem with Regression Mean Imputation, which Stochastic Regression Mean Imputation addresses, is that the regression model assumes a relationships between covariates, and imputes the missing values with values that are highly correlated with the completely observed covariates. Stochastic Regression Mean Imputation reduces the artificially high correlation by introducing a random error term.

While literature on the subject has plenty of examples showing the problems with simple imputation methods, experience is the best teacher. A simulation was made to calculate the bias, variance and mean square error for regression estimates after using these four simple methods.

For the simulation, the response variable is the a linear combination of independent covariates and plus an error term, where , and , , and . 250 observations were generated using R functions, to create the dataset.

Next ’s observations were deleted from the complete dataset, in a random way for 10,000 iterations to create 10,000 datasets with missing values. The probability of missingness for , was a Bernoulli random variable with probability of missingness, . Because the probability of missingness does not depend on any variable from the dataset, the mechanism for missing data is MCAR.

For each of the 10,000 MCAR datasets, regression estimates for the multiple linear regression model , were estimated, and collected into a matrix of ’s for each method. Altogether there were four matrices full of regression coefficients for the four simple methods: List-wise deletion, Mean Imputation, Regression Mean Imputation, and Stochastic Mean Regression Imputation.

The advantage of a simulation, is that the true values are known. For this simulation, . With the true values known, the bias, variance and mean square error for the regression estimates can be calculated. The bias for a regression coefficient is , for , where . The variance for a regression coefficient is the sample variance for the coefficient, , for . Likewise the standard error is the square root of the sample variance. Finally, the estimate for the means square error is . The table below lists these three statistics for each regression coefficient for each method.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **Method** | | | |
| **Statistics** | | **List-Wise Deletion** | **Mean Imputation** | **Regression Mean Imputation** | **Stochastic Regression Mean Imputation** |
| **bias** | intercept | -2.2806992 | -2.4247228 | -12.49792946 | -12.49828979 |
| X1 | 0.01337073 | 0.03805876 | 0.08809639 | 0.08810158 |
| X2 | -0.0332110 | -0.0273975 | -0.01093928 | -0.01095588 |
| X3 | -0.0230649 | -0.0227010 | -0.98631352 | -0.98637205 |
| **standard error** | intercept | 0.48718556 | 0.48521607 | 0.110522073 | 0.110314516 |
| X1 | 0.04398384 | 0.02809106 | 0.009371896 | 0.009353041 |
| X2 | 0.03757761 | 0.02534665 | 0.008419968 | 0.008411341 |
| X3 | 0.04215475 | 0.04251008 | 0.012602017 | 0.012574046 |
| **MSE** | intercept | 5.438938765 | 6.114715059 | 1.56E+02 | 1.56E+02 |
| X1 | 0.002113354 | 0.002237577 | 7.85E-03 | 7.85E-03 |
| X2 | 0.002515046 | 0.001393075 | 1.91E-04 | 1.91E-04 |
| X3 | 0.002309012 | 0.002322441 | 9.73E-01 | 9.73E-01 |

From the table some interesting results appear for this MCAR simulation. List-wise deletion and Mean Imputation produce less biased estimates for all regression coefficients except for the coefficient corresponding to . On the other hand, Regression Mean Imputation and Stochastic Regression Mean Imputation estimates have a smaller standard error than List-wise deletion and Mean Imputation. Combining these two statistics is the estimated mean square error. For the Regression Mean and Stochastic Regression Mean Imputation, the is smaller for the regression coefficient, but Mean Imputation and List-wise deletion have a smaller for every other regression coefficient. Probably the most condemning piece of information is the for the intercept. The for the Regression Mean Imputation and Stochastic Regression Mean Imputation is approximately 156! This is far too large to warrant use in the MCAR situation. Since List-wise deletion has the smallest for the majority of the regression coefficients, this method should be used whenever the MCAR assumption is true and the statistical power of a test is not too adversely affected.

While this simulation has compared single imputation methods under MCAR conditions, imputation methods are also needed for the MAR mechanism. While it is true that if data is MCAR then it is MAR, the converse is not true. More sophisticated methods have been developed for imputation under the MAR assumption and certain missing data patterns.

**E-M Algorithm**

**Ganga**

**Multiple Imputation**

**Sara**

**Imputation Algorithms**

Under the MAR assumption, a dataset with a monotone missing pattern may have multiply impute datasets generated using the Monotone Regression Method, Monotone Regression Predictive Mean Matching Method or the Propensity Score Method (SAS/STAT(R) 9.3 User's Guide, 2009). While the Propensity Score Method is a non-parametric imputation method, the Monotone Regression Method and Monotone Regression Predictive Mean Matching Method both assume a multivariate normal distribution for the independent variables with missing data. As the name suggests, both the Monotone Regression Method and Monotone Regression Predictive Mean Matching Method involve a regression equation.

For the Monotone Regression Method algorithm begin by fitting a regression equation for the variable with the least number of missing values. In *figure 2*, this would be the variable . For this variable, fit obtain regression coefficients using the observed values of the missing variable and completely observed covariates, i.e. find the values for the equation , where is a vector of the observed values and is a matrix containing the corresponding observations for the completely observed covariates.

Using the coefficient estimates the next step is to draw new values from the posterior predictive distribution, . Draw from the distribution , where is the number of non-missing observations for , and is the squared residual standard error from the previous regression. Next draw from , where is from the choleski decomposition , is a vector of independent random normal variates, and is the vector of parameter estimates from the previous regression.

The final step is use the new regression coefficients to impute the missing values. That is for observation for which is missing, impute the missing values by using the stochastic regression equation , where is a draw from an independent standard normal variate (SAS/STAT(R) 9.3 User's Guide, 2009).

Now that one of missing variables has been imputed, you can repeat the process for the next variable with missing values. Referring to *figure 2*, once we’ve imputed values for , we can consider to be one of the completely observed variables, and repeat the process for . This process is repeated for each missing variable, so that all monotonically missing variables have been imputed; and once all values have been imputed, the entire process is repeated times.

The Monotone Regression Predictive Mean Matching Method is very similar to the previous imputation method. In fact, the algorithm is identical, with the exception of a few additional steps. As before, the first step is to fit a regression equation to the observed values of the variable with missing values , i.e. the response variable is the observed values of , and the explanatory variables are the corresponding, completely observed values. Once the regression coefficients have been estimated, draw new coefficients from the posterior predictive distribution, . For each missing value, find a predicted value with the regression equation , where are the draws from . For each predicted value, create a list of observed values of , for which is closest to the predicted value, then from the list randomly choose one of the values to impute the missing value (SAS/STAT(R) 9.3 User's Guide, 2009).

Just as before, once all missing values for a variable have been imputed, the process would be repeated for the next monotone missing variable, until all missing values have been imputed. This entire process is repeated times to form different multiply imputed datasets.

.

**Application**

**SARA**

Using these variables, there is a mononte missing pattern for the data. *Figure 4* presents a table which illustrates the pattern. In *Figure 4*, the 2nd through 15th column names represent variables and row names represent various groups of missing data. We see in Group 1, a pattern , where an indicates that the variable is completely observed. This means Group 1 is the subset of observations which were completely observed. The frequency and relative frequency for Group 1 was 21 and 11.29%, with the interpretation that only 21 observations, or 11.29% of the data, from the tennis dataset were completely observed. For Group 2, there is a sequence of ’s followed by a period, . , where the period indicates a missing value for a variables. Group 2 is the subset of observations where all variables, except for ST51 are missing. Examining all groups, the pattern here is that, if a variable is missing, then all subsequent variables are missing as well, so our dataset with the following variables chosen, has a monotone missing pattern.

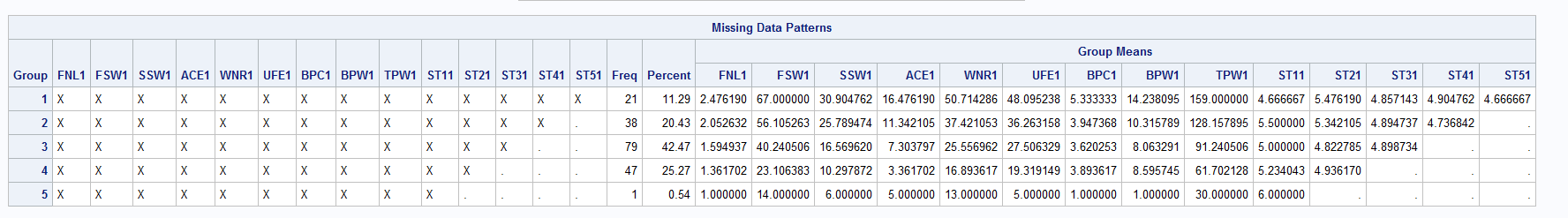


Figure :The Pattern of Missing Values

Because the dataset has a monotone missing pattern, the Monotone Regression Method and Monotone Regression Predictive Mean Matching Methods was used to multiply impute the dataset and obtain confidence intervals regression coefficients. In addition, the regression coefficients from the E-M algorithm were also calculated and are discussed in the next section.

**Results and Discussion**

Using the Monotone Regression Method first to impute the dataset times, the regression coefficients were estimated for the model . The sets of estimates were combined using Rubin’s Rules.

*Figure 5* shows SAS output which displays information about the variance as well as a few other statistics. We see the between, within and total variance which are calculated using Rubin’s Rules, for all 13 regression coefficients. We also see the degrees of freedom for each regression coefficient. There are three additional statistics displayed in *Figure 5*: the Relative Increase in Variance, the Fraction of Missing Information and the Relative Efficiency. These statistics are important enough for their own explanation and discussion.

Define the Relative Increase in Variance due to Non-Response to be , where is the between imputation variance and is the within imputation variance. The yields the proportional increase in the sampling variance that is due to missing data (Enders 2010, pp. 225). Define the Fraction of Missing Informationto be , where is the degrees of freedom for the *t*-test statistic. The gives the proportion of the total sampling variance that is due to missing data (Enders 2010, pp. 226). Define the Relative Efficiency to be . quantifies the magnitude of a multiple imputation standard error relative the hypothetical minimum, achieved by an infinite number of imputations! For example, if and , then , meaning the hypothetical minimum standard error is 96% as large as the standard error given 5 imputations (Enders 2010, pp. 213).

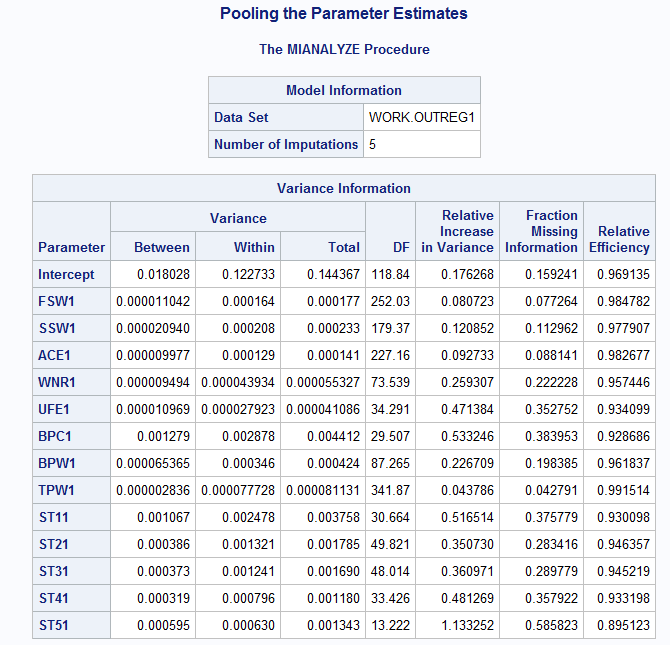


Figure : Monotone Regression Method Variance Info

Returning to *Figure 5*, we can now accurately interpret the last three statistics. We see that the regression coefficient for the variable has the largest Fraction of Missing Information, which is not surprising, since this was the most missing variable. We also see the Relative Efficiency for this parameter estimate is 89.5%, meaning the hypothetical minimum variance is 89.5% as large as variance with 5 imputations. For the other coefficients the Relative Efficiency is larger than 92%.

*Figure 6* is another table output from SAS. *Figure 6* displays the estimates, standard error, 95% confidence intervals and *p*-values for the regression coefficients. Examining the *p*-values, regression coefficients corresponding to the intercept and variables and are significant at the 5% significance level. It is also important to note that these results are valid under the MAR and multivariate normality assumption, and may not be robust to violations of these assumptions.

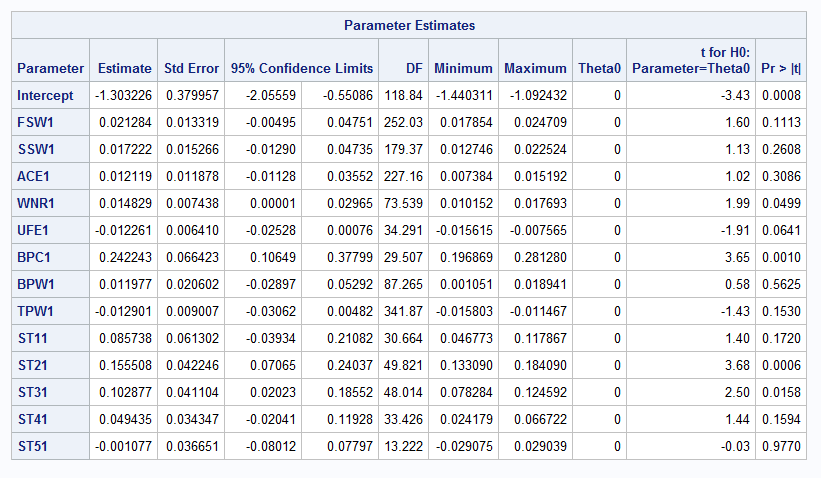


Figure : Monotone Regression Parameter Estimates

In addition to the Monotone Regression Method, analysis was carried out using the Monotone Regression Predictive Mean Matching Method. Using this method 5 multiply impute datasets were generated, then the regression coefficients were estimated for the model . The sets of estimates were combined using Rubin’s Rules.

The variance information for the parameter estimates is given in *Figure 7*. We see that this time the Relative Efficiencies for the parameters estimates are not quite as good. The regression coefficients corresponding to the variables and have approximately 88% Relative Efficiency. This suggests that more than 5 imputations would lower the total variance for the parameter estimates. If we compare the total variance for and in *Figure 7* with those in *Figure 5*, we see that the Monotone Regression Predictive Mean Matching Method achieves smaller total variances than the Monotone Regression Method. For this reason, and to make a better comparison between methods, 5 imputations is sufficient.

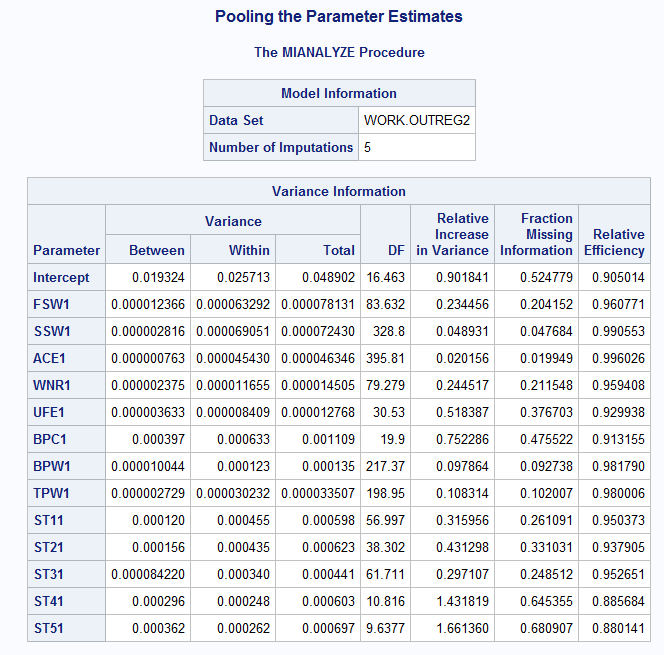


Figure : Predictive Mean Matching Variance Info

*Figure 8* displays more SAS output. The parameter estimates, standard errors, 95% confidence intervals and *p*-values for the regression coefficients, using the Monotone Regression Predictive Mean Matching Method are shown. Examining the *p*-values for the individual regression coefficients we see that the intercept and coefficients corresponding to the variables and are all significant at the 5% significance level. This contrasts with the results from the Monotone Regression Method where coefficients corresponding to the and variables were not significant. This difference can be attributed to the smaller standard errors achieved using the Monotone Regression Predictive Mean Matching Method.

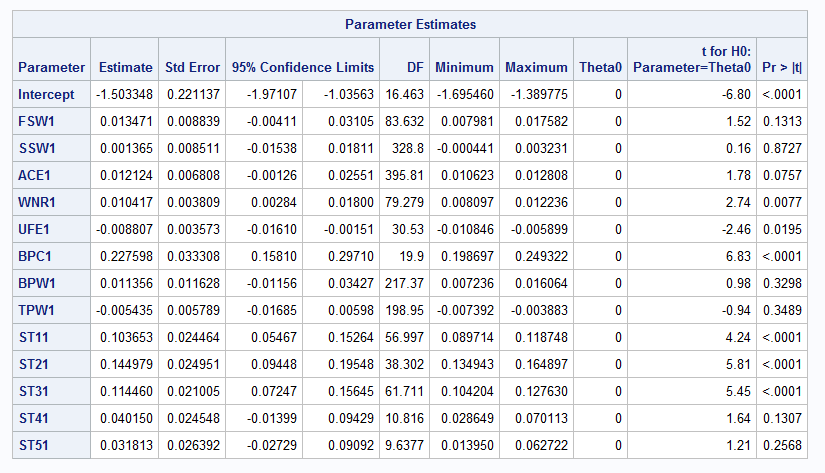


Figure : Predictive Mean Matching Parameter Estimates

Thus far only the results from multiple imputation methods have been used to analyze the dataset. Expectation Maximization can also be used to obtain good estimates, in fact, the estimates from the E-M algorithm approximate the maximum likelihood estimates. The default convergence criterion in SAS is , and the maximum number of iterations is 200. For this dataset the maximum number of iterations was reached before the convergence criterion was satisfied. *Figure 9* shows SAS output using the E-M algorithm. Many of the parameter estimates are radically different than those using multiple imputation methods. From *Figures 6* and *8* the intercept estimate was -1.30 and -1.50 respectively, while using the E-M algorithm the intercept estimate was 0.06. In addition to different estimates, all regression coefficients are significant in *Figure 9*, but this is an error. Using the E-M algorithm alone provides no way of quantifying variability for the parameter estimates, thus the standard errors and *p*-values displayed in *Figure 9* are artifacts from an invalid covariance matrix which SAS uses.



Figure : E-M Algorithm Estimates

*Figure 10* shows a message from the SAS log, which warns about the *p*-values from *Figure 9*. In particular the messages warns that the *p*-values and other statistics that depend upon the number of observations should not be used in interpretations.

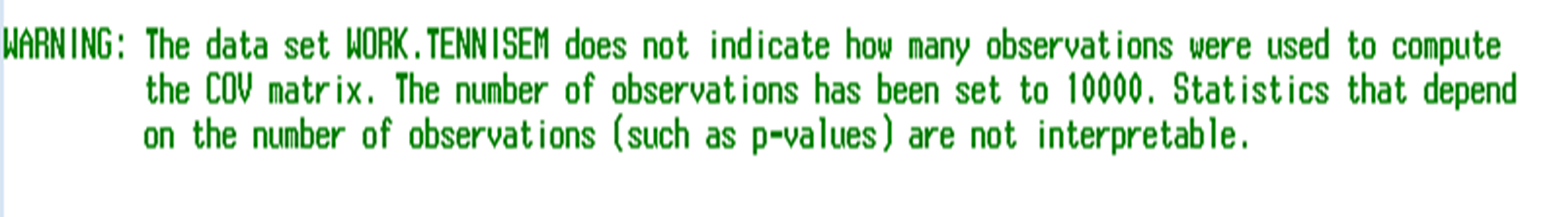


Figure : E-M Warning Message

Because Multiple Imputation yields accurate standard errors for imputed values and allows for the interpretation of *p*-values and confidence intervals, either Multiple Imputation method is preferred over the Expectation Maximization algorithm, under the assumption of MAR and multivariate normality. To keep this analysis simple, parametric methods were used, although the Propensity Score method is a valid Multiple Imputation method for MAR data with a Monotone Missing Pattern.

**Code Used**

To produce the results for the simulation and application, R, Excel VBA and SAS programming was used. Beginning with the simulation, the following lines of R code, create MCAR random, and estimates the bias, variance and mean square error.

library('HotDeckImputation')

#Creating the dataset

N <- 250

x1<-rnorm(N,6,1)

x2<-rnorm(N,0,1)

x3<-rnorm(N,-10,1)

y<--1+1\*x1-1\*x2+1\*x3+rnorm(N,0,1)

#Making x3 MCAR

alpha=.3

r.x3.mcar<-rbinom(N,1,prob=alpha)

x3.mcar<-x3\*(1-r.x3.mcar)+r.x3.mcar\*99999

x3.mcar[x3.mcar==99999]=NA

#The complete case analysis

mm1 <- lm(y~x1+x2+x3.mcar, na.action=na.omit)

beta<-mm1$coeff

beta<- t(beta)

beta<- t(beta)

beta1 <- beta

std<-summary(mm1)$sigma

#The mean imputation analysis

design<- cbind(y,x1,x2,x3.mcar)

design<-data.frame(design)

x1.1<-design$x1[complete.cases(design)]

x2.1<-design$x2[complete.cases(design)]

x3.1<-design$x3.mcar[complete.cases(design)]

y.1<-design$y[complete.cases(design)]

x1.2<-design$x1[!complete.cases(design)]

x2.2<-design$x2[!complete.cases(design)]

y.2<-design$y[!complete.cases(design)]

x0.2<-rep(1,length(x2.2))

design<-as.matrix(design)

design<- impute.mean(design)

design<-data.frame(design)

mm1 <- lm(data=design,X1~X2+X3+X4)

beta<-mm1$coeff

beta<- t(beta)

beta<- t(beta)

beta2<- beta

#This plot illustrates reduction in variance of x3

#plot(design$X4, design$X1, xlab='Variable with Imputed Values', ylab='Response')

#The Regresion Mean Analysis

matrix <- cbind(x0.2,x1.2,x2.2)

beta<-beta1[-4,]

beta <- t(beta)

beta <- t(beta)

x3.2 <- matrix %\*% beta

matrix <- cbind(y.2,x1.2,x2.2,x3.2)

matrix0 <- cbind(y.1,x1.1,x2.1,x3.1)

matrix<- rbind(matrix0,matrix)

design<-matrix

design<-data.frame(design)

mm1<-lm(data=design,y.1~x1.1+x2.1+x3.1)

beta<-mm1$coeff

beta3<- beta

beta3<- t(beta3)

beta3<- t(beta3)

#This plot illustrates problems with Regression Mean Imputation

#plot(design$x3.1, design$x2.1, xlab='Imputed Variable', ylab='Covariate')

#The stocahstic regression mean imputation

matrix <- cbind(x0.2,x1.2,x2.2)

z <- rnorm(length(x2.2),0,std)

beta<-beta1[-4,]

beta <- t(beta)

beta <- t(beta)

x3.2 <- matrix %\*% beta + z

matrix <- cbind(y.2,x1.2,x2.2,x3.2)

matrix0 <- cbind(y.1,x1.1,x2.1,x3.1)

matrix<- rbind(matrix0,matrix)

design<-matrix

design<-data.frame(design)

mm1<-lm(data=design,y.1~x1.1+x2.1+x3.1)

beta<-mm1$coeff

beta4<- beta

beta4<- t(beta4)

beta4<- t(beta4)

for(i in 1:9999){

r.x3.mcar<-rbinom(N,1,prob=alpha)

x3.mcar<-x3\*(1-r.x3.mcar)+r.x3.mcar\*99999

x3.mcar[x3.mcar==99999]=NA

mm1 <- lm(y~x1+x2+x3.mcar, na.action=na.omit)

std<-summary(mm1)$sigma

beta <- mm1$coeff

beta <- t(beta)

beta <- t(beta)

beta1.loop <- beta

beta1 <- cbind(beta1,beta)

design<- cbind(y,x1,x2,x3.mcar)

design<-data.frame(design)

x1.1<-design$x1[complete.cases(design)]

x2.1<-design$x2[complete.cases(design)]

x3.1<-design$x3.mcar[complete.cases(design)]

y.1<-design$y[complete.cases(design)]

x1.2<-design$x1[!complete.cases(design)]

x2.2<-design$x2[!complete.cases(design)]

y.2<-design$y[!complete.cases(design)]

x0.2<-rep(1,length(x2.2))

design<-as.matrix(design)

design<- impute.mean(design)

design<-data.frame(design)

mm1 <- lm(data=design,X1~X2+X3+X4)

beta<-mm1$coeff

beta<- t(beta)

beta<- t(beta)

beta2<- cbind(beta2,beta)

matrix <- cbind(x0.2,x1.2,x2.2)

beta<-beta1.loop[-4,]

beta <- t(beta)

beta <- t(beta)

x3.2 <- matrix %\*% beta

matrix <- cbind(y.2,x1.2,x2.2,x3.2)

matrix0 <- cbind(y.1,x1.1,x2.1,x3.1)

matrix<- rbind(matrix0,matrix)

design<-matrix

design<-data.frame(design)

mm1<-lm(data=design,y.1~x1.1+x2.1+x3.1)

beta<-mm1$coeff

beta<- t(beta)

beta<- t(beta)

beta3<- cbind(beta3,beta)

matrix <- cbind(x0.2,x1.2,x2.2)

z <- rnorm(length(x2.2),0,std)

beta<-beta1.loop[-4,]

beta <- t(beta)

beta <- t(beta)

x3.2 <- matrix %\*% beta + z

matrix <- cbind(y.2,x1.2,x2.2,x3.2)

matrix0 <- cbind(y.1,x1.1,x2.1,x3.1)

matrix<- rbind(matrix0,matrix)

design<-matrix

design<-data.frame(design)

mm1<-lm(data=design,y.1~x1.1+x2.1+x3.1)

beta<-mm1$coeff

beta<-t(beta)

beta<-t(beta)

beta4<-cbind(beta4,beta)}

betahat01 <- mean(beta1[1,])

betahat11 <- mean(beta1[2,])

betahat21 <- mean(beta1[3,])

betahat31 <- mean(beta1[4,])

betahat02 <- mean(beta2[1,])

betahat12 <- mean(beta2[2,])

betahat22 <- mean(beta2[3,])

betahat32 <- mean(beta2[4,])

betahat03 <- mean(beta3[1,])

betahat13 <- mean(beta3[2,])

betahat23 <- mean(beta3[3,])

betahat33 <- mean(beta3[4,])

betahat04 <- mean(beta4[1,])

betahat14 <- mean(beta4[2,])

betahat24 <- mean(beta4[3,])

betahat34 <- mean(beta4[4,])

bias1 <- c(betahat01-1, betahat11 - 1, betahat21 +1, betahat31-1)

bias2 <- c(betahat02-1, betahat12 - 1, betahat22 +1, betahat32-1)

bias3 <- c(betahat03-1, betahat13 - 1, betahat23 +1, betahat33-1)

bias4 <- c(betahat04-1, betahat14 - 1, betahat24 +1, betahat34-1)

var1 <- c(sd(beta1[1,]),sd(beta1[2,]),sd(beta1[3,]),sd(beta1[4,]))

var2 <- c(sd(beta2[1,]),sd(beta2[2,]),sd(beta2[3,]),sd(beta2[4,]))

var3 <- c(sd(beta3[1,]),sd(beta3[2,]),sd(beta3[3,]),sd(beta3[4,]))

var4 <- c(sd(beta4[1,]),sd(beta4[2,]),sd(beta4[3,]),sd(beta4[4,]))

mse1 <- bias1^2+var1^2

mse2 <- bias2^2+var2^2

mse3 <- bias3^2+var3^2

mse4 <- bias4^2+var4^2

To analyze the tennis dataset in SAS the *.csv* files needed to be modified. Originally missing values were indicated by the characters NA. In order to read the data set in SAS, the NA characters needed to be deleted. A simple macro in Excel VBA can accomplish this task for each *.csv* file. The following line of code are the Excel VBA macro.

Sub DataCleanUp()

Dim rng As Range, cell As Range

Set rng = Range("A1:AP127")

For Each cell In rng

If cell.Value = "NA" Then

cell.Select

Selection.ClearContents

End If

Next cell

End Sub

After the *.csv* files were modified, the files were read into a SAS dataset. The Multiple Imputation methods used and E-M algorithm were all options for the SAS procedure, PROC MI. The regression coefficients for each multiply impute dataset were found using PROC REG, and the coefficients were pooled together using PROC MIANALYZE. *Figures 5, 6, 7, 8* are output from PROC MIANALYZE, while *Figure 9* is output from PROC REG. The following lines of code were used to accomplish this.

filename tennis1 'E:\Tennis\AusOpen-men-2013.csv';

data test26;

infile tennis1 DSD;

input Player1 Player2 Round Result FNL1 FNL2

FSP1 FSW1 SSP1 SSW1 ACE1 DBF1 WNR1

UFE1 BPC1 BPW1 NPA1 NPW1 TPW1 ST11

ST21 ST31 ST41 ST51 FSP2 FSW2 SSP2

SSW2 ACE2 DBF2 WNR2 UFE2 BPC2 BPW2

NPA2 NPW2 TPW2 ST12 ST22 ST32 ST42

ST52;

data tennis1;

set work.test26;

keep FNL1 FSW1 SSW1 ACE1 WNR1 UFE1 BPC1 BPW1

TPW1 ST11 ST21 ST31 ST41 ST51;

run;

filename tennis1 'E:\Tennis\AusOpen-women-2013.csv';

data test27;

infile tennis1 DSD;

input Player1 Player2 Round Result FNL1 FNL2

FSP1 FSW1 SSP1 SSW1 ACE1 DBF1 WNR1

UFE1 BPC1 BPW1 NPA1 NPW1 TPW1 ST11

ST21 ST31 ST41 ST51 FSP2 FSW2 SSP2

SSW2 ACE2 DBF2 WNR2 UFE2 BPC2 BPW2

NPA2 NPW2 TPW2 ST12 ST22 ST32 ST42

ST52;

data tennis2;

set work.test27;

keep FNL1 FSW1 SSW1 ACE1 WNR1 UFE1 BPC1 BPW1

TPW1 ST11 ST21 ST31 ST41 ST51;

run;

filename tennis3 'E:\Tennis\FrenchOpen-men-2013.csv';

data test28;

infile tennis3 DSD;

input Player1 Player2 Round Result FNL1 FNL2

FSP1 FSW1 SSP1 SSW1 ACE1 DBF1 WNR1

UFE1 BPC1 BPW1 NPA1 NPW1 TPW1 ST11

ST21 ST31 ST41 ST51 FSP2 FSW2 SSP2

SSW2 ACE2 DBF2 WNR2 UFE2 BPC2 BPW2

NPA2 NPW2 TPW2 ST12 ST22 ST32 ST42

ST52;

data tennis3;

set work.test28;

keep FNL1 FSW1 SSW1 ACE1 WNR1 UFE1 BPC1 BPW1

TPW1 ST11 ST21 ST31 ST41 ST51;

run;

filename tennis4 'E:\Tennis\FrenchOpen-men-2013.csv';

data test29;

infile tennis4 DSD;

input Player1 Player2 Round Result FNL1 FNL2

FSP1 FSW1 SSP1 SSW1 ACE1 DBF1 WNR1

UFE1 BPC1 BPW1 NPA1 NPW1 TPW1 ST11

ST21 ST31 ST41 ST51 FSP2 FSW2 SSP2

SSW2 ACE2 DBF2 WNR2 UFE2 BPC2 BPW2

NPA2 NPW2 TPW2 ST12 ST22 ST32 ST42

ST52;

data tennis4;

set work.test29;

keep FNL1 FSW1 SSW1 ACE1 WNR1 UFE1 BPC1 BPW1

TPW1 ST11 ST21 ST31 ST41 ST51;

run;

proc append base=work.tennis1

data=work.tennis2;

run;

proc append base=work.tennis1

data=work.tennis3;

run;

proc append base=work.tennis1

data=work.tennis4;

run;

data tennis0;

set tennis1;

if FNL1=. then delete;

if ACE1=. then delete;

run;

title'The Dataset';

proc print data=tennis0;

run;

title'Imputation via Monotone Regression Method';

proc mi data=work.tennis0

seed=123456 out=outex1;

monotone reg(ST21 = FSW1 SSW1 ACE1 WNR1 UFE1 BPC1 BPW1 TPW1 ST11);

monotone reg(ST31 = FSW1 SSW1 ACE1 WNR1 UFE1 BPC1 BPW1 TPW1 ST11 ST21);

monotone reg(ST41 = FSW1 SSW1 ACE1 WNR1 UFE1 BPC1 BPW1 TPW1 ST11 ST21 ST31);

monotone reg(ST51 = FSW1 SSW1 ACE1 WNR1 UFE1 BPC1 BPW1 TPW1 ST11 ST21 ST31 ST41);

var FNL1 FSW1 SSW1 ACE1 WNR1 UFE1 BPC1 BPW1

TPW1 ST11 ST21 ST31 ST41 ST51;

run;

proc reg data=outex1 outest=outreg1 covout noprint;

model FNL1= FSW1 SSW1 ACE1 WNR1 UFE1 BPC1 BPW1

TPW1 ST11 ST21 ST31 ST41 ST51;

by \_Imputation\_;

run;

title'Pooling the Parameter Estimates';

proc mianalyze data=outreg1 edf=422;

modeleffects Intercept FSW1 SSW1 ACE1 WNR1 UFE1 BPC1 BPW1

TPW1 ST11 ST21 ST31 ST41 ST51;

run;

title'Imputation via Monotone Predictive Mean Matching Method';

proc mi data=work.tennis0

seed=123456 out=outex2;

monotone regpmm(ST21 = FSW1 SSW1 ACE1 WNR1 UFE1 BPC1 BPW1 TPW1 ST11);

monotone regpmm(ST31 = FSW1 SSW1 ACE1 WNR1 UFE1 BPC1 BPW1 TPW1 ST11 ST21);

monotone regpmm(ST41 = FSW1 SSW1 ACE1 WNR1 UFE1 BPC1 BPW1 TPW1 ST11 ST21 ST31);

monotone regpmm(ST51 = FSW1 SSW1 ACE1 WNR1 UFE1 BPC1 BPW1 TPW1 ST11 ST21 ST31 ST41);

var FNL1 FSW1 SSW1 ACE1 WNR1 UFE1 BPC1 BPW1

TPW1 ST11 ST21 ST31 ST41 ST51;

run;

proc reg data=outex2 outest=outreg2 covout noprint;

model FNL1= FSW1 SSW1 ACE1 WNR1 UFE1 BPC1 BPW1

TPW1 ST11 ST21 ST31 ST41 ST51;

by \_Imputation\_;

run;

title'Pooling the Parameter Estimates';

proc mianalyze data=outreg2 edf=422;

modeleffects Intercept FSW1 SSW1 ACE1 WNR1 UFE1 BPC1 BPW1

TPW1 ST11 ST21 ST31 ST41 ST51;

run;

title'Using the E-M Alg';

proc mi data=tennis0 nimpute=0;

em outem=tennisem;

var FNL1 FSW1 SSW1 ACE1 WNR1 UFE1 BPC1 BPW1

TPW1 ST11 ST21 ST31 ST41 ST51;

run;

proc reg data=tennisem outest=outreg2 covout;

model FNL1= FSW1 SSW1 ACE1 WNR1 UFE1 BPC1 BPW1

TPW1 ST11 ST21 ST31 ST41 ST51;

run;

**References**

Enders, C. (2010). Applied missing data analysis. New York: Guilford Press.

Imputation (statistics). (2015, October 31). Retrieved December 4, 2015, from <https://en.wikipedia.org/wiki/Imputation_(statistics)>

Missing data. (2015, October 30). Retrieved December 4, 2015, from <https://en.wikipedia.org/wiki/Missing_data>

SAS/STAT(R) 9.3 User's Guide. (2009, September 1). Retrieved December 4, 2015, from <http://support.sas.com/documentation/cdl/en/statug/63962/HTML/default/viewer.htm#mi_toc.htm>

UCI Machine Learning Repository: Data Set. (2014, June 1). Retrieved December 4, 2015, from <https://archive.ics.uci.edu/ml/datasets/Tennis+Major+Tournament+Match+Statistics>