# Simulation of Imputation Effects Under Different Assumptions

A Senior Project

presented to

the Faculty of the Statistics Department

California Polytechnic State University, San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Science in Statistics

by

Danny Rithy

March, 2016

Abstract:

Missing data is something that we cannot prevent when data become missing while in the process of data collection. There are many reasons why data can be missing due to respondent refusing to answer a sensitive question or in fear of embarrassment. Researchers often assume their data are "missing completely at random" or "missing at random". Unfortunately, we cannot test whether the mechanism condition is satisfied because missing values cannot be calculated. For my senior project, I will run simulation studies in SAS to observe the behaviors of missing data under different assumptions: missing completely at random, missing at random and ignorability. I will also compare the effects from imputation methods when a set of variables of interest are set to missing. The objective of this simulation study of imputation methods is to see how efficient substituted values in a dataset affect further studies. This will let readers decide which imputation method(s) would be best to approach a dataset when it comes to missing data.

## I. Opening:

While I was working on a dataset during my first quarter at Cal Poly, I encountered something that I've never seen before: missing data. Since that day, I have been curious about the subject. Why do missing data exist anyway? What can we do about it? Is imputation always the best answer? I wanted to explore this topic, so I dedicated my senior project to learning about missing data. My senior project focuses on simulation under missing mechanism assumptions and measures mean square error (MSE) under different imputation methods. Note I am not completely fluent with the rigorous mathematical proofs on missing data techniques and advanced statistical techniques such as Bayesian analysis and structural equation modeling, so there are some limitations that I am not capable of until then.

There are goals that I want to get out of this senior project. First, I want to know what happens if we impute missing data and whether the imputation method is optimal. Beforehand, we must understand why data are missing. In Section III, I will explain the three common missing data mechanisms. Second, I want to present my senior project in professional conferences. Lastly, I want to learn to be able to deal with missing data in real world scenarios. In case I work in a research firm that does survey research in the future, what I learn from my senior project will contribute to my professional interests.

## II. Introduction:

Missing data can be problematic cases for all researchers and statisticians. They occur when respondents participate in a survey but do not answer a certain question. If one is familiar in survey sampling and methodology, this is known as item nonresponse. Because of it, there are missing data recorded in some variables in a dataset. Prior to data analysis, researchers must decide what to do with missing data. The common solutions are either removing observations with missing responses or replacing missing values with average values on the variable of interest. But removing these observations can decreases sample size, and thus decrease statistical power. This wastes a lot of efforts for those who go out and collect data.

In general, statistics has two goals: inference and prediction. We want to take a sample of the population and discover findings that apply to the entire population. In order to do that, we must ensure that our samples are random and representative of the population. However, in missing data, samples cannot be representative because we do not have information across the entire sample due to missing responses. Thus, it is impossible to compute statistical models because we have incomplete data. However, the analysis can be biased due to missing data. One solution deals with missing responses by replacing missing values with reasonable estimates,

which is known as imputation.

The next section will introduce three popular missing data mechanisms.

## III. Missing Data Mechanisms:

To understand missing data, we must consider reasons why they exist in the first place.

When data are **Missing Completely at Random (MCAR)** the missing value of the variable of interest does not depend on the variable of interest or any other variables, which are observed in the dataset. In other words, there is a missing value on the variable of interest due to chance alone. For example, respondents forget to answer some questions on the survey. Missing data is very rarely MCAR (Rubin, 1976).

When data are **Missing at Random (MAR)** the missing value of the variable of interest does not depend on the variable of interest but conditionally depends on other observed variables' values. In other words, there is a missing value on the variable of interest depending on some other observed variables' values. For example, high school students are more likely than college students to not answer some questionnaires in a health survey due to the fact that their parents are still their guardians and keep track of their needs. This is an unfortunate statistical term because the mechanism is not exactly "missing at random".

When data are **Not Missing at Random (NMAR)** the missing value of the variable of interest depends on the variable of interest itself and its observed value. Therefore, if respondents do not respond to certain questions their responses on that question are directly related to what their response would have been, had they answered the question. For example, during in-person interview, respondents are most likely to not respond to a question asking the number of illegal

drugs they currently possess, if they are currently in possession of illegal drugs. NMAR is a serious missing data mechanism and most difficult to model because the model must be precise because of sensitivity.

## IV. Type of Imputations:

Imputation is a method that attempts to replace missing values based on the available data. There are several approaches to deal with missing data. However, depending on the missing data mechanism, a selected imputation method may or may not be the best choice.

### Likewise Deletion:

Although Likewise Deletion is not an imputation method, it is worth mentioning it. Likewise Deletion is one of the most common method where each observation is removed if it contains one or more missing values. If the missing data mechanism is MCAR, then the parameter estimates from Likewise Deletion are shown to be unbiased. Although this method is easy and simple, the disadvantages are decreasing sample size and statistical power.

### Mean Imputation:

Mean imputation takes an average of the available data values for the variable of interest and replaces missing values with that average. It's one of the common techniques researchers use to deal with missing data. This imputation method is useful if the missing mechanism is assumed to be MCAR.

Consider this small dataset:

Table A:

| Subject | Height (inches) |
|---|---|
| 1 | . |
| 2 | 59 |
| 3 | . |
| 4 | 56 |
| 5 | 58 |

Heights are missing for Subject 1 and 3. To solve this, we simply take the average of the given heights.

$\bar{x} = \frac{59+56+58}{3} = 57.67$

So the imputed data looks like this:

| Subject | Height (inches) |
|---|---|
| **1** | **57.67** |
| 2 | 59 |
| **3** | **57.67** |
| 4 | 56 |
| 5 | 58 |

(Linear/Multiple) Regression Imputation:

Regression imputation estimates a regression model based on one or more explanatory variables to predict and replace a missing value given by the criteria of the data. This imputation method is useful if the missing mechanism is assumed to be MAR.

Consider another small dataset:

Table B:

| Subject | Handspan (cm) | Height (inches) |
|---|---|---|
| 1 | 26 | 59 |
| 2 | 30 | 71 |
| 3 | 28 | 68 |
| 4 | . | 70 |
| 5 | 32 | 65 |

Subject 4's handspan is missing. Given that Subject 4's height is given, let's predict the Subject's handspan using regression. Suppose we have estimated regression model that predicts handspan with known height. Assuming all the linear regression assumptions requirements are met, our model is:

$$\hat{y} = 11.47 + .2667(Height)$$

Where $\hat{y}$ represents predicted handspan. If we predict the person's handspan given height, we can replace the missing value, which becomes 30.14cm.

We can use multiple linear regression to predict a response variable given by the number of explanatory variables.

Hot Deck Imputation:

In general, hot deck imputation is a method for dealing with item nonresponses where every missing value is replaced with a value from a similar donor. For example, a respondent whose questions are missing can be replaced by another respondent based on similar physical characteristics.

Consider another small dataset:

Table C:

| Subject | Sex | Possess Driver License | BMI |
|---|---|---|---|
| 1 | Male | Y | 24.76 |
| 2 | Female | N | 23.81 |
| 3 | Male | Y | 28.12 |
| 4 | Male | Y | . |
| 5 | Male | Y | 23.52 |
| 6 | Female | N | . |

By using sequential hot-deck approach, we would replace the missing value by the last value read by the computer in the dataset. In this case, we are using Sex and Subject 2 and 5 will be the donor to Subject 6 and 4, respectively. Because there are many possible categorical variables that can be used, it's difficult to implement them because it will require complex algorithm in a statistical software. As a result, the imputed data looks like this:

| Subject | Sex | Possess Driver License | BMI |
|---|---|---|---|
| 1 | Male | Y | 24.76 |
| 2 | Female | N | 23.81 |
| 3 | Male | Y | 28.12 |
| **4** | **Male** | **Y** | **23.52** |
| 5 | Male | Y | 23.52 |
| **6** | **Female** | **N** | **23.81** |

Logistic Regression Imputation:

Logistic regression imputation is a method that uses a generalized linear model to predict the probability of a categorical response variable, given by the number of independent variables. Imputing categorical outcome based on the probability may vary from researcher to researcher. Many researchers recommend rounding the imputed values greater than or equal to .5 are set to 1 and anything else less than .5 is set to 0. However, rounding imputed values are inferior than non-rounding imputed values because it introduces bias (Allison, 2005). The proportion of success for a categorical response variable can be used as a criteria.

Consider the same dataset except it contains 1000 subjects.

Table D:

| Subject | Sex | Possess Driver License | BMI |
|---|---|---|---|
| 1 | Male | Y | 24.76 |
| 2 | Female | N | 23.81 |
| 3 | Male | . | 28.12 |

| … | … | … | … |
|---|---|---|---|
| 999 | Male | Y | 23.52 |
| 1000 | Female | . | 21.84 |

Some subjects' answers to whether they possess a driver's license are missing. The only difference between Table C and D is the categorical response variable where subjects 3 and 4's response to whether they possess a driver's license is missing. To solve this issue, we would use the logistic regression approach to predict the categorical variable (whether or not have a driver's license) using Sex and BMI as predictors where Sex is a factor.

Since the response variable is categorical, a logistic regression would be appropriate. Using Sex and BMI as predictors, the logistic regression model is estimated as:

$$logit(\hat{\pi}) = -0.557 + 1.28(BMI) + 1.8(Sex)$$

The estimated probabilities is calculated as:

$\hat{\pi} = \frac{e^{\hat{\alpha}+\hat{\beta}1+\hat{\beta}2}}{1+e^{\hat{\alpha}+\hat{\beta}1+\hat{\beta}2}}$ where $\hat{\alpha}\ and\ \hat{\beta}\imath$ are estimated intercept and predictors, respectively.

After estimating $\hat{\pi}$, it's up to the researchers whether they want to round the imputed values.

Logistic regression can be complicated depending on the dataset. If the proportion of success is very high, then quasi-completion will be most likely to occur. Quasi-completion becomes problematic when the estimated logistic regression model fits the probabilities numerically to be either 0 or 1.

Multiple Imputation:

Multiple imputation (MI) creates several iterations of imputed datasets and condenses all datasets into one. For each imputed dataset, we take into account that random error will be introduced per iteration so all imputed datasets are unique. Most of the imputation methods

assume the model has the deterministic component, however, without the stochastic component, the results can be biased. Therefore, Multiple Imputation has one of the main advantages over any single imputation. Instead of replacing a single missing value, MI estimates and replaces each missing value per observation with randomness added. In most situation, MI is the best choice of imputation because it minimizes bias for NMAR.

The MI procedure below works for SAS. The procedure and code can be found in the appendix.

1. Impute using PROC MI. PROC MI will output *m* datasets
2. Use PROC REG or PROC LOGISTIC and add the statement _imputation_
3. Combine *m* unique imputation results using PROC MIANALYZE. This will combine and estimate the parameters from all estimated results into one estimated regression model
4. Use the estimated model from PROC MIANALYZE to predict missing values

The next section will explain the setup of the simulation and how imputation method plays its role.

## V. Simulation Study:

The purpose of the simulation study is to show an audience the "what-if analysis" if we assume the missing data mechanism to impute missing values. In the real world, we cannot test the assumption because of incomplete cases. Instead, we will take a complete dataset and simulate under missing data mechanisms and compare imputation methods. The mean square error (MSE) will be calculated to show the mean squared error between the original values and imputed values.

Before the simulation occurs, we must assign some variable of interests from selected observations to missing. This can be done with the three missing data mechanisms. MCAR is induced by taking a simple random sample of the observations of the variable of interest and assigning them to missing. MAR takes a stratified sample of the observations of the variable of interest and assign them to missing. For example, divide the population based on gender and take a simple random sample of the group. NMAR uses a condition to assign the variable of interest to missing. For example, select respondents whose income is higher than $100,000 into a group and take a simple random sample of the group and assign income to missing. In SAS, I used PROC SURVEYSELECT to randomly select observations into a new dataset and assign the variable of interest to missing. Afterward, I merged the original dataset and the missing dataset together. From there, a selected imputation method is implemented to replace missing values so the MSE is calculated.

The following steps are the general procedures of how I used SAS and SAS MACRO to simulate missing data mechanism, and then execute imputation methods. SAS MACRO is very useful in simulation because it substitutes text strings in DATA steps and PROC steps as well as alternate logic flows based on user's input in a macro. Because SAS reads each observation sequentially, I had to think carefully how I implement the simulation. The full code can be found in the appendix.

1. Select a dataset and a response variable. Create a variable called original and assign the response variable to it. Get the number of rows of the dataset before using PROC SURVEYSELECT to randomly select sample rows into an output dataset, which we will call dataset A. The selected rows are the rows that assign variable of interest to missing.

2. Merge the original dataset and dataset A so that the merged dataset contains missing values, so we call this dataset B.

3. Create another dataset and subset observations with complete cases and call it C. By complete cases, it means all observations must contain no missing data. The imputation method will use this dataset to replace missing values.

4. Use dataset C to build and estimate an imputation method. Afterward, use the estimated imputation model to replace the missing values in dataset B. To compare the original values and imputed values, create a variable called imputed and assign the average value of the response variable to it. If we want to use mean imputation, we would use PROC MEANS or PROC SUMMARY to estimates an average value in dataset C. Afterward, substitute the missing values in dataset B using the average value.

5. Finally, calculate MSE where MSE = $\sum \frac{(original-imputed)^2}{N-1}$. N is the sample size based on the number of observations in a dataset with or without missing values.

The next section will explain the analysis and results of the simulation.

## IV. Analysis and Results:

For the simulation, I will be using a dataset called "MathAchieve", which is a dataset from the nlme package in R. The dataset contains 7185 rows and 6 columns where MathAch represents mathematics achievement scores and is the response variable. Note that the data frame is not a real dataset and used for simulation purposes.

***MathAchieve Dataset from 'nlme' package in R***

| School | Minority | Sex | SES | MathAch | MEANSES |
|---|---|---|---|---|---|
| **2629** | No | Male | -0.148 | 11.437 | -0.132 |
| **4931** | No | Male | 0.932 | 12.201 | 0.371 |
| **7364** | No | Female | -0.168 | 22.678 | -0.083 |
| **7919** | No | Female | 0.552 | 13.800 | 0.464 |
| **9225** | No | Female | -0.568 | 15.265 | 0.259 |
| **9508** | No | Male | -0.238 | 11.481 | -0.132 |

Table 1: Randomly selected six rows are output in the table.

The following tables (Tables 2 – 4) are the results of the simulation showing the estimated mean and standard deviation based on the percentage of missingness and imputation methods per missing data mechanism. Likewise Deletion is included for comparison.

| Percent Missing | Likewise Deletion Mean | Likewise Deletion SD | Mean Imputation Mean | Mean Imputation SD | Regression Imputation Mean | Regression Imputation SD | Multiple Imputation Mean | Multiple Imputation SD | Hot Deck Imputation Mean | Hot Deck Imputation SD |
|---|---|---|---|---|---|---|---|---|---|---|
| 0% | 12.75 | 6.88 | 12.75 | 6.88 | 12.75 | 6.88 | 12.75 | 6.88 | 12.75 | 6.88 |
| 10% | 12.72 | 6.87 | 12.72 | 6.53 | 12.70 | 6.56 | 12.67 | 6.58 | 12.64 | 6.94 |
| 20% | 12.73 | 6.87 | 12.71 | 6.12 | 12.82 | 6.24 | 12.53 | 6.26 | 12.93 | 7.09 |
| 30% | 12.72 | 6.86 | 12.68 | 5.82 | 12.72 | 5.97 | 12.50 | 5.93 | 12.80 | 7.02 |
| 40% | 12.72 | 6.84 | 12.72 | 5.33 | 12.79 | 5.62 | 12.41 | 5.55 | 12.82 | 6.96 |
| 50% | 12.80 | 6.91 | 12.74 | 4.91 | 12.66 | 5.25 | 12.17 | 5.26 | 12.86 | 6.96 |

**Table 2: Table of Mean and SD Per Imputation Method When Data are MCAR**

According to Table 2, when the data is MCAR, most of the imputation methods' mean estimates are unbiased except multiple imputation and hot deck imputation whereas multiple

imputation tends to underestimate and hot deck imputation tends to slightly overestimate. Not surprisingly, Likewise Deletion's mean and standard deviation are unbiased. In most cases, the standard deviation tends to underestimate as the percentage of missingness increases. This makes sense because the variability would decrease as the number of data are replaced by imputed values. This can be confirmed by taking a histogram of the imputed data.

| Percent Missing | Likewise Deletion Mean | Likewise Deletion SD | Mean Imputation Mean | Mean Imputation SD | Regression Imputation Mean | Regression Imputation SD | Multiple Imputation Mean | Multiple Imputation SD | Hot Deck Imputation Mean | Hot Deck Imputation SD |
|---|---|---|---|---|---|---|---|---|---|---|
| 0% | 12.75 | 6.88 | 12.75 | 6.88 | 12.75 | 6.88 | 12.75 | 6.88 | 12.75 | 6.88 |
| 10% | 12.84 | 6.86 | 12.83 | 6.78 | 12.75 | 6.80 | 12.80 | 6.79 | 11.98 | 7.37 |
| 20% | 12.91 | 6.86 | 12.92 | 6.65 | 12.72 | 6.73 | 12.88 | 6.70 | 12.05 | 7.23 |
| 30% | 13.04 | 6.82 | 12.99 | 6.58 | 12.78 | 6.63 | 12.89 | 6.61 | 11.32 | 7.21 |
| 40% | 13.11 | 6.82 | 13.14 | 6.44 | 12.78 | 6.56 | 12.99 | 6.50 | 11.70 | 7.15 |
| 50% | 13.24 | 6.81 | 13.23 | 6.33 | 12.78 | 6.46 | 13.03 | 6.42 | 11.84 | 7.08 |

**Table 3: Table of Mean and SD Per Imputation Method When Data are MAR**

According to Table 3, when the data are MAR, the results are different than when the data are MCAR. Interestingly, as the percentage of missingness increases, Likewise Deletion's mean tends to overestimate the mean for complete data but its standard deviation tends to be unbiased estimate of the standard deviation of the complete data. Mean imputation tends to overestimate the mean for complete data but the standard deviation tends to underestimate the standard deviation of the complete data. Regression imputation's mean shown to be unbiased estimate of the mean for complete data but its standard deviation tends to underestimate the standard deviation of the complete data. Multiple imputation's mean tends to slightly overestimate the mean for complete data but its standard deviation tends to underestimate standard deviation of the complete data. Hot Deck imputation's mean tends to slightly underestimate of the mean for complete data but its standard deviation tends to slightly overestimate the standard deviation of the complete data.

| Percent Missing | Likewise Deletion Mean | Likewise Deletion SD | Mean Imputation Mean | Mean Imputation SD | Regression Imputation Mean | Regression Imputation SD | Multiple Imputation Mean | Multiple Imputation SD | Hot Deck Imputation Mean | Hot Deck Imputation SD |
|---|---|---|---|---|---|---|---|---|---|---|
| 0% | 12.75 | 6.88 | 12.75 | 6.88 | 12.75 | 6.88 | 12.75 | 6.88 | 12.75 | 6.88 |
| 10% | 13.04 | 6.87 | 13.04 | 6.70 | 13.01 | 6.72 | 12.99 | 6.71 | 11.64 | 7.01 |
| 20% | 13.36 | 6.87 | 13.39 | 6.48 | 13.30 | 6.53 | 13.23 | 6.53 | 11.62 | 7.08 |
| 30% | 13.77 | 6.77 | 13.79 | 6.22 | 13.61 | 6.36 | 13.52 | 6.32 | 11.73 | 7.03 |
| 40% | 14.20 | 6.68 | 14.21 | 5.96 | 13.94 | 6.13 | 13.80 | 6.16 | 11.75 | 7.12 |
| 50% | 14.69 | 6.52 | 14.67 | 5.67 | 14.40 | 5.84 | 14.19 | 5.88 | 11.76 | 7.04 |

**Table 4: Table of Mean and SD Per Imputation Method When Data are NMAR**

According to Table 4, when the data is NMAR, the results show that most imputation methods tend to overestimate the mean for complete data while the standard deviation tends to decrease as the percent of missingness increases. The hot deck imputation is somewhat unusual because it tends to slightly underestimate the mean for complete data but its standard deviation tends to slightly overestimate of the standard derivation of the complete data.

The lesson from the simulation results is imputation will always affect the variance estimates. Variance estimates are essential to confidence intervals and statistical inferences. If the variance estimates decrease, then the standard error will be lower as well as the margin of error. This would influence the inferences and future studies depending on the percent of missingness.

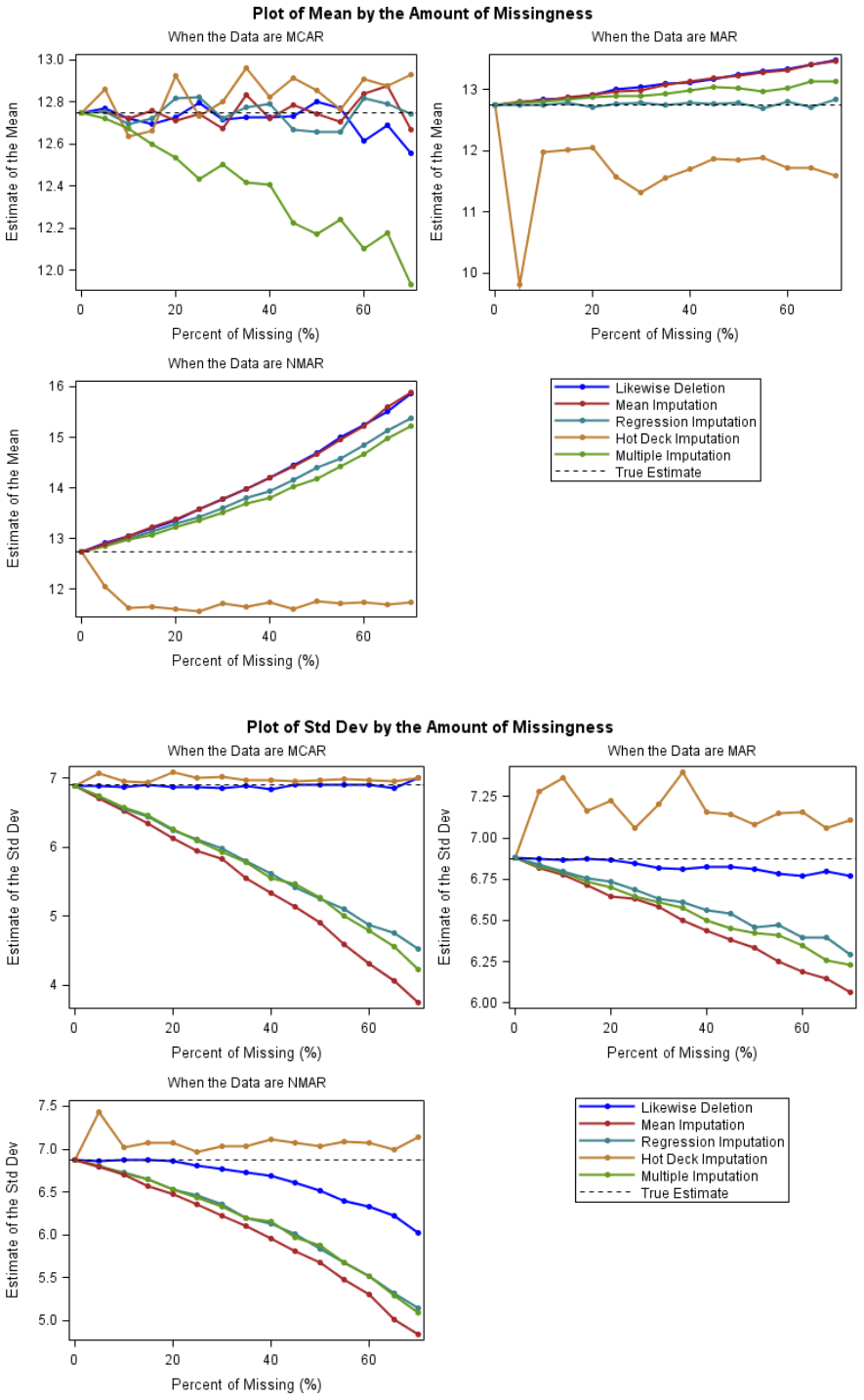


Figure 1: A graphical representation of the information presented in Tables 2 – 4.

If we use imputation to fill in the missing values, suppose we are curious how far is our prediction from the original values based on the percent of missingness. For the simulation, MSE will be calculated.

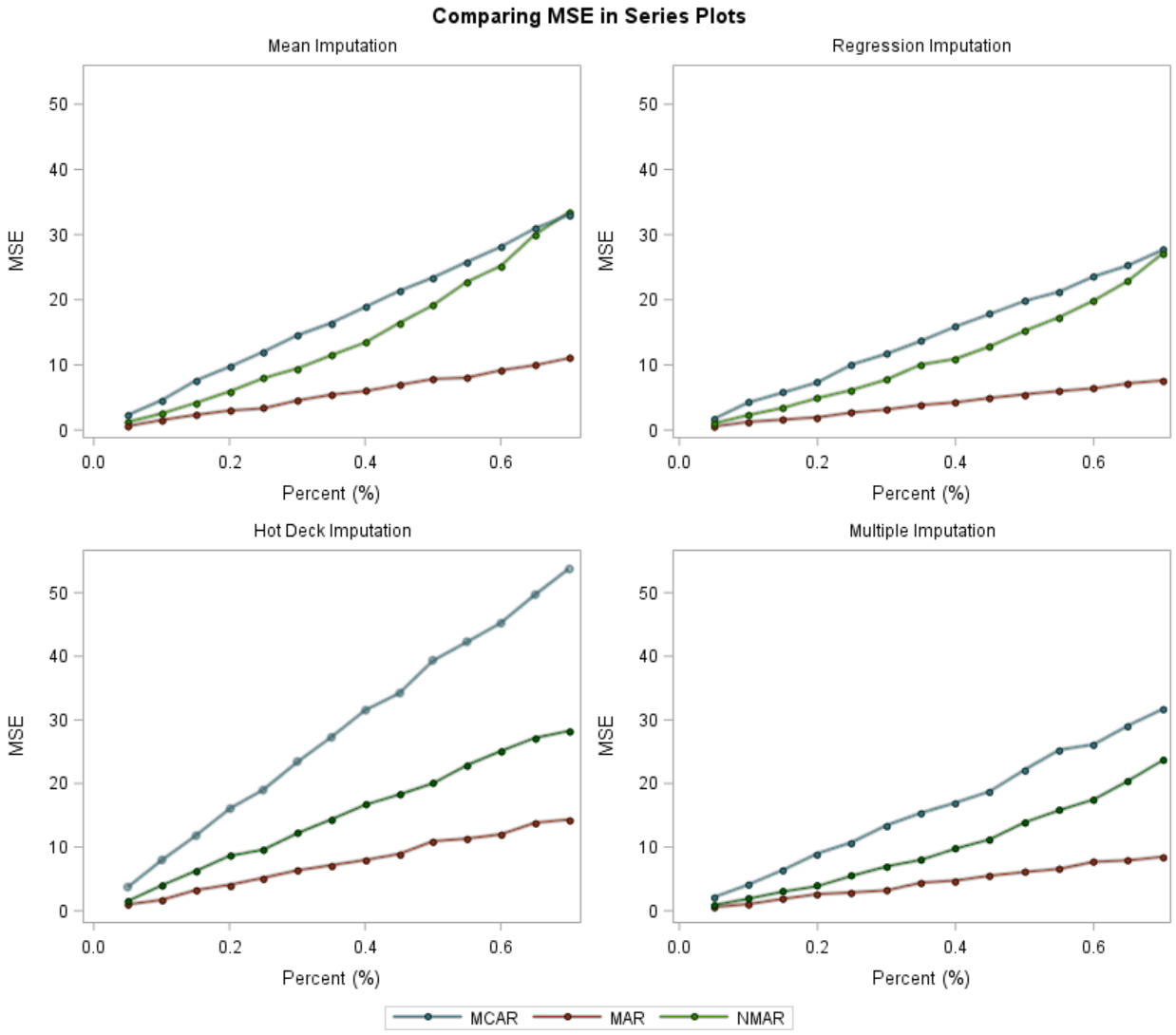


Figure 2: The relationship between MSE and % missing data, for the three missing data mechanisms, separated by imputation method

The graphs above represent the simulation results using four imputation methods: Mean Imputation, Regression Imputation, Hot Deck Imputation and Multiple Imputation. There are several ways to interpret the graph. As the percent of missingness increases, Mean Square Error tends to increase. When the data are MCAR, MSE is the largest whereas MAR's MSE is the smallest. Hot Deck Imputation's MCAR MSE is the largest compared to the other three imputation methods. This makes sense because Hot Deck Imputation replaces all missing values with all values from a similar donor, thus most of the imputed values can be very far from original values. Regression Imputation has the least MCAR MSE. MAR has MSE values that are lowest. Thus, MAR is preferable to MCAR and NMAR. In addition, Mean Imputation has the least MAR MSE. Multiple Imputation has the least NMAR MSE than the three imputations methods. Therefore Multiple Imputation is the best imputation method because multiple imputation minimizes MSE when the data is NMAR.

In general, it is difficult to identify the true missing data mechanism when it comes to real data because of unknown circumstance and uncertainty. There is no optimal answer to determine which imputation methods would be the best. As a rule of thumb, imputation would be useful if the percent of missingness is below 30%. In some cases, some imputation methods are useful. Mean Imputation is the best imputation if the data is MCAR and the percent of missingness is below 30%. However, Mean Imputation is not useful if the percent of missingness is above 30%. In extreme cases if the percent of missingess is above 50% and the data are NMAR, Multiple Imputation is the best choice. This choice is entirely subjective so everyone can consider their own percent of missingess.

The next section is an application to a real dataset and how imputation affects the dataset.

## V. Application to Real Data

The purpose of this section is to apply imputation to replace some missing values in some variables and compare the effects in the dataset. There are reasons why missing data cannot be eliminated by imputation. In this scenario, we must come up with feasible hypothesis on why the data are missing in the first place prior to imputation.

For the application to real data, Survey dataset will be used and can be found from the MASS package in R. The dataset will be used to compare the effects of imputation. The survey takes place in University of Adelaide in South Australia. The survey contains respondents of 237 Statistics I students and they were being asked to answer 12 questions. Most of the variables are related to the students' physical characteristics and health status.

| Sex | Writing Hand (cm) | Non-Writing Hand (cm) | Writing Hand | Fold | Pulse (bpm) | Clap | Exercise | Smoke | Height (cm) | Height in Metric/ Imperial | Age (yr) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Female | 18.5 | 18.0 | Right | R on L | 92 | Left | Sometimes | Never | 173.00 | Metric | 18.250 |
| Male | 19.5 | 20.5 | Left | R on L | 104 | Left | None | Regular | 177.80 | Imperial | 17.583 |
| Male | 18.0 | 13.3 | Right | L on R | 87 | Neither | None | Occasional | . | . | 16.917 |
| Male | 18.8 | 18.9 | Right | R on L | . | Neither | None | Never | 160.00 | Metric | 20.333 |
| Male | 20.0 | 20.0 | Right | Neither | 35 | Right | Sometimes | Never | 165.00 | Metric | 23.667 |
| Female | 18.0 | 17.7 | Right | L on R | 64 | Right | Sometimes | Never | 172.72 | Imperial | 21.000 |

Table 5: Randomly selected six rows are output in the table.

From Table 5, we can see that there is at least one respondents contain missing values on some variables. In Table 6, Pulse has the most missing values where there are 45 missing values.

In addition, Height and MI have 28 missing values. However, Height and MI are directly related to each other because if height is missing so is MI.

**PROC FREQ for WHnd**

The FREQ Procedure

| Writing Hand | |
|---|---|
| WHnd | Frequency |
| . | 1 |
| Left | 18 |
| Right | 218 |

**PROC FREQ for MI**

The FREQ Procedure

| Height in Metric/Imperial | |
|---|---|
| MI | Frequency |
| . | 28 |
| Imperial | 68 |
| Metric | 141 |

**PROC FREQ for Clap**

The FREQ Procedure

| Clap your hands! Which hand is on top? | |
|---|---|
| Clap | Frequency |
| . | 1 |
| Left | 39 |
| Neither | 50 |
| Right | 147 |

**PROC FREQ for Smoke**

The FREQ Procedure

| How much the student smokes | |
|---|---|
| Smoke | Frequency |
| . | 1 |
| Heavy | 11 |
| Never | 189 |
| Occasional | 19 |
| Regular | 17 |

Table 6. Frequency tables for Writing Hands, Clap, Smoke and Measurement.

In Table 7, one respondent's writing hands are missing as well as the response to "Clap". Although we do not know why the responses are missing, it's possible that this person does not have hands. Under extreme circumstance, this person cannot be imputed. Otherwise, if we assume the data is NMAR, the best choice is to leave it alone because the parameter estimates can be very sensitive. Another respondent do not answer the question regarding how frequent the student's smoke. Because the respondent do not exercise, it's possible the respondent refuses to answer the question, which makes the data NMAR.

| Sex | Writing Hand (cm) | Non-Writing Hand (cm) | Writing Hand | Fold | Pulse (bpm) | Clap | Excercise | Smoke | Height (cm) | Height in Metric/ Imperial | Age (yr) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Male | . | . | Right | R on L | 60 | . | Sometimes | Never | 172 | Metric | 28.583 |
| Male | 21 | 19.5 | Right | L on R | 80 | Left | None | . | . | . | 18.333 |

Table 7: Respondent's missing values for writing hand, non-writing hand and clap and another Respondent's missing values for Smoke, Height and MI.

Suppose we want to impute Pulse and Height/MI. Based on the patterns in the dataset, we can assume the data are MCAR for Pulse and Height/MI. For a single variable imputation, we will impute pulse and height separately and then compare each imputation method by variance estimates. If we want to impute pulse and height in one step, then multiple imputation would be the optimal choice. The percent of missingness for Pulse and Height/MI is 19% and 12%, respectively.

Based on the results, in Table 8, each imputation method impacts the variance and standard deviation estimates. Mean imputation's mean estimates is approximately equal to the complete dataset, except the variance decreases. Since we assume the data are MCAR, mean imputation would be appropriate. Hot Deck Imputation has the least influence of the variability, however, the mean estimate tends to be higher. The general procedures for imputation using SAS can be found in the appendix.

**Imputation for Height**

| Imputation | Mean | Standard Deviation | Variance |
|---|---|---|---|
| Complete Dataset | 172.381 | 9.84753 | 96.9738 |
| Mean Imputation | 172.381 | 9.24491 | 85.4684 |
| Regression Imputation | 172.203 | 9.48279 | 89.9233 |
| Hot Deck Imputation | 172.657 | 9.59933 | 92.1471 |
| Multiple Imputation | 172.143 | 9.41160 | 88.5782 |

Table 8: Imputation results for Height.

Based on the results, in Table 9, the four imputation methods estimate mean, standard deviation, and variance. The mean imputation tends to slightly underestimate the mean for imputed data and the standard derivation tends to underestimate the standard derivation of the imputed data. Hot Deck Imputation tends to overestimate the mean for imputed data and the standard derivation tends to overestimate the standard derivation of the imputed data. Multiple Imputation has the closest mean estimation of the complete dataset compared to the three imputation methods, however, the standard derivation decreases. In this case, Multiple Imputation would be preferable.

**Imputation for Pulse**

| Imputation | Mean | Standard Deviation | Variance |
|---|---|---|---|
| Complete Dataset | 74.1510 | 11.6872 | 136.590 |
| Mean Imputation | 73.9459 | 10.6465 | 113.349 |
| Regression Imputation | 74.1154 | 10.5462 | 111.223 |
| Hot Deck Imputation | 76.6751 | 11.7913 | 139.034 |
| Multiple Imputation | 74.1499 | 10.6870 | 114.212 |

Table 9: Imputation results for Pulse.

Imputation impacts the variance depending on the percent of missingness. For a single variable imputation, most of the imputation methods don't take into account for variability. Multiple Imputation would be the best choice for imputation because it takes into account for random error for each imputation model. Thus, each imputed values vary.

## VI. Shiny App

The purpose of this section is to briefly explain my integration between missing data and Shiny together. For those who are not familiar with Shiny, Shiny is a special tool in RStudio that lets a developer creates a web application framework in R. Shiny lets users interact with the user-interface to show R outputs such as graphics and tables based on inputs the user enters.

I originally wasn't planning to create a Shiny app. However, after attending and presenting my senior project work at the Conference on Statistical Practice 2016, I decided to create a ShinyApp to help students visually understand how imputation affects our inference. The app will be available under the Shiny page under the Cal Poly Statistics Department after this submission on my senior project.

The goal of this app is to provide intuitive understanding of missing data by comparing original and imputed data using data visualization. Data visualization is an easy way to grasp concepts in a universal manner. In this app, users will have four imputation methods to choose: mean imputation, regression imputation, hot-deck imputation and multiple imputation. The users will have a chance to input some data and the app will process through the code based on the inputs. On each page, we are comparing two graphics and descriptive summaries from both original and imputed data. Afterward, the app will output three things: graphics, descriptive summaries and MSE. Please note that Figure 3 shows a rough draft of the app and not a final version.

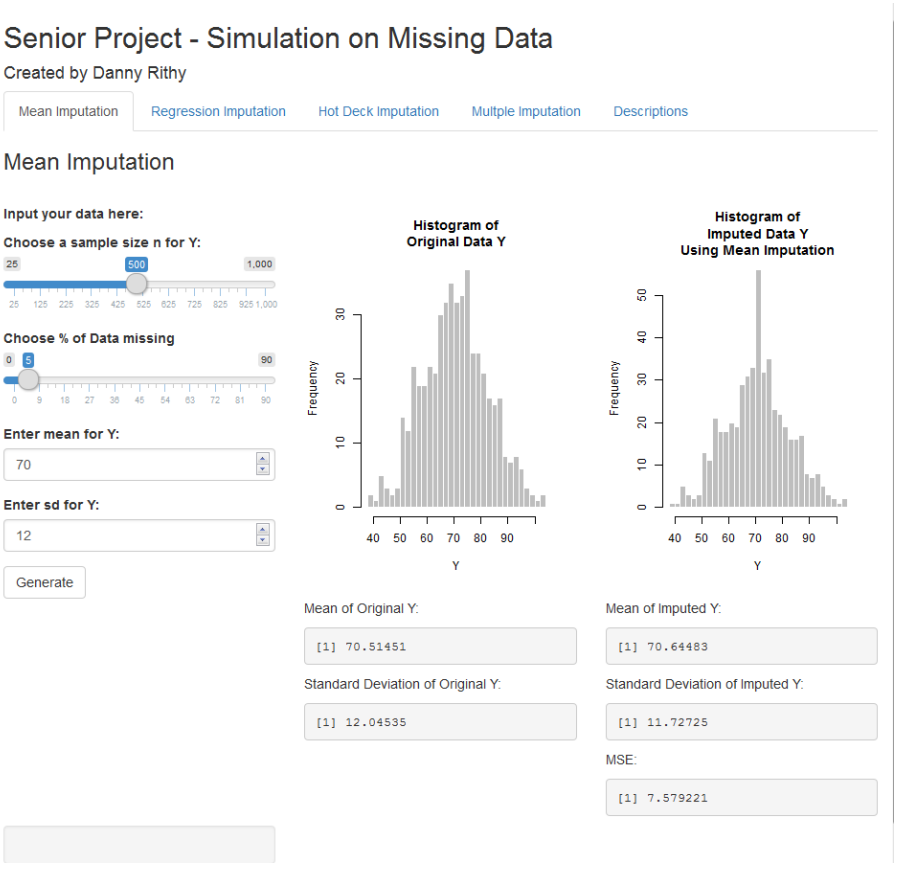


Figure 3: A screenshot of mean imputation showing the outputs based on the user's inputs.

## VII. Imputation Effects: Pros and Cons

Imputation has both advantages and disadvantages to fill in the missing values prior to data analysis. Imputation, by definition, attempts to fill in the missing values by using a selected statistical model. It also saves the incomplete samples from being removed from the dataset, thus increase the statistical power and minimize bias. However, imputed data does not represent real data. We attempt to substitute each respondent's incomplete answers without giving a penalty to the researchers themselves. However, each imputed value does not truly represent the respondent's answer. Depending on the percent of missingness, the results can be biased. Imputation reduces variability in parameter estimates, which also underestimates standard errors. Therefore, in missing data, there is no optimistic answer to our questions.

## VIII. Conclusion

Missing data are very common in research that researchers and statisticians come across. In practice, we do not directly interact with respondents who did not answers certain questions. Therefore, we have three missing data mechanisms that we need to come up to give reasons why missing data occur. According to the simulation results, MCAR is the safest assumption because the mean and standard deviation estimates are unbiased whereas when data are MAR and NMAR parameter estimates tend to be underestimated or overestimated. In imputation, MAR is preferable to MCAR and NMAR because the MSEs are the smallest. Multiple Imputation has the least bias when we assume the mechanism is NMAR. I had difficulty working on the real data due to the fact it's my first time working with actual missing data. After graduation, I hope I adopt the practices in missing data in an industry.

There are future studies that I would explore more into this topic. These include:

- Be more proficient in mathematical proofs and notation, when I read Rubin's book on missing data for the first time, I was easily lost due to the fact I wasn't aware that missing data contains theorems and proofs.
- Find out how to impute covariates instead of a response variable. While I was presenting my e-Poster during the conference, one statistician came up to me and asked me this question. Unfortunately, I did not know how to answer it but I told him that this is something I should look up.
- Explore further topics on missing data and imputation methods such as Markov chain Monte Carlo, Structural Equations Models, Time Series, and Maximum Likelihood Estimation.

## IX. Acknowledgements

I would like to thank Dr.Roy for being my senior project advisor and letting me choose my senior project on missing data.

I would also like to thank Dr.Doi for his helpful feedbacks and suggestions for my first Shiny app.

## X. References

Allison, Paul David. "Imputation of Categorical Variables with PROC MI." *113-30: Imputation of Categorical Variables with PROC MI* (2005): n. pag. Apr. 2005. Web. <http://www2.sas.com/proceedings/sugi30/113-30.pdf>.

Allison, Paul David. *Missing Data*. Thousand Oaks, CA: Sage Publications, 2001. Print.

Little, Roderick J. A., and Donald B. Rubin. *Statistical Analysis with Missing Data*. New York: Wiley, 1987. Print.

Rubin, Donald B. *Inference and Missing Data*. 3rd ed. Vol. 63. N.p.: Biometrika Trust, 1975. Biometrika. Web. <http://qwone.com/~jason/trg/papers/rubin-missing-76.pdf>.

Scheffer, Judi. "Dealing with Missing Data." *Research Letters in the Information and Mathematical Sciences* (2002): 153-60. Web. <https://www.researchgate.net/publication/2522396_Dealing_with_Missing_Data>

## XI. Appendix

```
/* By Danny Rithy */

* Used for Macro Debugging ;
OPTIONS MLOGIC; * Print message that indicates macro actions ;
OPTIONS MPRINT; * Passes to complier during the macro execution ;
OPTIONS MLOGICNEST; * Show nesting information to the log ;
OPTIONS SYMBOLGEN; * Dissolve a macro variable in the log window ;

/*
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~

  DIRECTORY:
  1. Likewise Deletion (LIKEDEL)
  2. Mean Imputation [MEANIMP]
  3. Regression Imputation [REGIMP]
  4. Hot Deck Imputation [HOTDECK]
  5. Multiple Imputation [MULTIIMP]
  6. MathAchieve Data [MATDAT]
  7. Survey Data [SURDAT]


~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~ */

%let wd = H:\F-\Senior Project\Final Documents;

libname myloc "&wd";

* LIKEDEL ;
/* Macro: LDprocedure
   Definitions:
   dat: Dataset
   percent: % of data will be missing on the variable of interest (rv)
   mechanism: Mechanism will be assumed on the cause of missingess
              (MCAR, MAR, NMAR)
   rv: Response variable
   catevar: Categorical explantory variable. Used for assigning missing
values if mechanism is MAR
   grp1: Categorical outcome for group 1 from catevar
   grp2: Categorical outcome for group 2 from catevar
*/

%macro LDprocedure(dat=, percent=, mechanism=, rv=, catevar=, grp1=, grp2=);

  %if "&mechanism" = "MCAR" %then
    %do;
```

```
      /* Get the number of rows from the data */
      proc SQL noprint;
        SELECT ROUND(COUNT(*) * &percent)
        INTO :nrow
        FROM &dat;
      quit;

      /* Simple random sampling where every samples in the population
         has an equal chance of being selected */
      proc surveyselect data = &dat noprint
        method = SRS n = &nrow out = MCARdata;
      run;

      /* Output objects with mean and std from PROC SUMMARY */
      %let availdata = %sysevalf(&percent * 100);
      title "Descriptive Statistics (&availdata% Missing)";
      proc summary data = MCARdata mean std print;
        var &rv;
          ODS OUTPUT Summary = MCARsummary&availdata; * Create a new table
called MCARsummary&availdata ;
      run;
      title;

    %end;
  %else %if "&mechanism" = "MAR" %then
    %do;

      data &grp1.tab &grp2.tab;
        set &dat;
          if &catevar = "&grp1" then
            output &grp1.tab;
        else
          output &grp2.tab;
      run;

      /* Get the number of rows from the data */
      proc SQL noprint;
        SELECT ROUND(COUNT(*) * &percent)
          INTO :nrow /* Create a macro variable called nrow where nrow is the
number of rows */
        FROM &grp1.tab;
      quit;

      /* Simple random sampling where every samples in the population
         has an equal chance of being selected */
      proc surveyselect data = &grp1.tab noprint
        method = SRS n = &nrow out = new&grp1.tab;
      run;

      /* Concatenate datasets into a new dataset called MARdata */
      data MARdata;
        set new&grp1.tab &grp2.tab;
      run;

      /* Output objects with mean and std from PROC SUMMARY */
```

```
      %let availdata = %sysevalf(&percent * 100);
      title "Descriptive Statistics (&availdata% Missing)";
      proc summary data = MARdata mean std print;
        var &rv;
          ODS OUTPUT Summary = MARsummary&availdata;
      run;
      title;

      %end;
  %else %if "&mechanism" = "NMAR" %then
    %do;
      /* Output a median response variable object in the table called medtab
from PROC MEANS */
      proc means data = &dat median maxdec = 0 noprint;
        var &rv;
        output out = medtab median = &rv._median;
      run;

      /* Subset dataset into either lowtab or hightab */
      data lowtab hightab;
        merge &dat medtab (keep = &rv._median);
        if &rv._median = . then
            &rv._median = temp_median;
          else
            do;
            temp_median = &rv._median;   /* Assign a median response variable
to a temporary variable */
              retain temp_median;          /* Retain that value to the next
observation */
            end;

          /* If response variable is below the median response variable, then
assign that row to lowtab (below median response variable table)*/
          if &rv < &rv._median then
            output lowtab;
        else
          output hightab;

        /* Don't include these variables in the dataset */
        drop temp_median &rv._median;
      run;

      /* Get the number of rows from the data */
      proc SQL noprint;
        SELECT ROUND(COUNT(*) * &percent)
          INTO :nrow
        FROM lowtab;
      quit;

      /* Simple random sampling where every samples in the population has an
equal chance of being selected */
      proc surveyselect data = lowtab noprint
        method = SRS n = &nrow out = newlowtab;
      run;
```

```
      /* Concatenate datasets into a new dataset called NMARdata */
      data NMARdata;
        set newlowtab hightab;
      run;

      /* Output objects with mean and std from PROC SUMMARY */
      %let availdata = %sysevalf(&percent * 100);
      title "Descriptive Statistics (&availdata% Missing)";
      proc summary data = NMARdata mean std print;
        var &rv;
          ODS OUTPUT Summary = NMARsummary&availdata; /* Create a new table
called NMARsummary&availdata */
      run;
      title;

      %end;
  %else
    %put ERROR: Mechanism does not match.;

%mend;

* MEANIMP ;
/* Macro: Mean Imputation
   Definitions:
   dat: Dataset
   percent: % of data will be missing on the variable of interest (rv)
   mechanism: Mechanism will be assumed on the cause of missingess
              (MCAR, MAR, NMAR)
   rv: Response variable
   catevar: Categorical explantory variable. Used for assigning missing
values if mechanism is MAR
   grp1: Categorical outcome for group 1 from catevar
   grp2: Categorical outcome for group 2 from catevar
*/
%macro mean_imputation(dat=, percent=, mechanism=, rv=, catevar=, grp1=,
grp2=);

  * If the macro variable, percent, is greater than zero then go to this
block ;
  /* Set up the data set */
  data temp_data;
    set &dat;
      id + 1; * Create an id for merging purpose ;
      original = &rv;
  run;

  * Get the number of rows for MSE estimate ;
  proc SQL noprint;
    SELECT COUNT(*) INTO :allrow
      FROM temp_data;
  QUIT;

  * If the macro variable, mechanism, is MCAR then go to this block ;
  %if "&mechanism" = "MCAR" %then
    %do;
```

```
      /* Get the number of rows for SRS purpose */
        proc SQL noprint;
        SELECT ROUND(COUNT(*) * &percent) INTO :nrow
          FROM temp_data;
        QUIT;

        proc surveyselect data = temp_data
        method = SRS n = &nrow out = temp_data2 noprint;
        run;

        * Assign all response variable to missing ;
        data temp_data2;
          set temp_data2;
          &rv = .;
        run;

      %end;
  %else %if "&mechanism" = "MAR" %then
    %do;
      data &grp1.tab &grp2.tab;
        set temp_data;
          if &catevar = "&grp1" then
            output &grp1.tab;
        else
          output &grp2.tab;
      run;

      * Get the number of rows from the data ;
      proc SQL noprint;
        SELECT ROUND(COUNT(*) * &percent)
          INTO :nrow /* Create a macro variable called nrow where nrow is the
number of rows */
        FROM &grp1.tab;
      quit;

      /* Simple random sampling where every samples in the population
         has an equal chance of being selected */
      proc surveyselect data = &grp1.tab noprint
        method = SRS n = &nrow out = new&grp1.tab;
      run;

        * Assign all response variable to missing ;
        data new&grp1.tab;
            set new&grp1.tab;
            &rv = .;
        run;

        * Merge two datasets together based on ID ;
        data &grp1.tab;
            merge &grp1.tab new&grp1.tab;
            by id;
        run;

        * Merge the temporary data with two datasets ;
      data temp_data2;
```

```
        set &grp1.tab &grp2.tab;
      run;

        * Sort the data for merging purpose ;
        proc sort data = temp_data2;
          by id;
        run;

      %end;
  %else %if "&mechanism" = "NMAR" %then
    %do;
      /* Output a median response variable in the table called medtab from
PROC MEANS */
      proc means data = temp_data median maxdec = 0 noprint;
        var &rv;
        output out = medtab median = &rv._median;
      run;

      /* Subset dataset into either lowtab or hightab */
      data lowtab hightab;
        merge temp_data medtab (keep = &rv._median);
        if &rv._median = . then
            &rv._median = temp_median;
          else
            do;
            temp_median = &rv._median;
              retain temp_median;
            end;

          if &rv < &rv._median then
            output lowtab;
        else
          output hightab;

        drop temp_median &rv._median;
      run;

      * Get the number of rows from the data ;
      proc SQL noprint;
        SELECT ROUND(COUNT(*) * &percent)
          INTO :nrow /* Create a macro variable called nrow where nrow is the
number of rows */
        FROM lowtab;
      quit;

      /* Simple random sampling where every samples in the population
         has an equal chance of being selected */
      proc surveyselect data = lowtab noprint
        method = SRS n = &nrow out = newlowtab;
      run;

      * Assign all selected response variables to missing ;
        data newlowtab;
          set newlowtab;
            &rv = .;
```

```
        run;

        * Merge both datasets with complete and missing values ;
        data lowtab;
        merge lowtab newlowtab;
            by id;
        run;

        * Merge the temporary data with two datasets ;
      data temp_data2;
        set lowtab hightab;
      run;

        * Sort the data for merging purpose ;
        proc sort data = temp_data2;
          by id;
        run;

    %end;
  %else
    %put ERROR: Mechanism does not match.;
        %if "&mechanism" = "MCAR" or "&mechanism" = "MAR" or "&mechanism" =
"NMAR" %then
          %do;
          /* Combine dataset with complete and missing values */
          data temp_data3;
            merge temp_data temp_data2;
              by id;
          run;

        /* Subset the complete dataset */
          data sub_dat;
            set temp_data3;
              if not missing(&rv);
        run;

        /* Get the average value from PROC MEANS */
        proc means data = sub_dat mean maxdec = 2 noprint;
          var &rv;
              output out = stats_data mean = imputed_mean;
        run;

          %let percent2 = %sysevalf(&percent * 100);
        data imputed_dataMEAN&mechanism&percent2 (drop = temp_imputed_&rv
imputed_mean);
          merge temp_data3 stats_data (keep = imputed_mean);

            /* Assign the mean response variable from previous row */
          if imputed_mean = . then
                imputed_mean = temp_imputed_&rv;

            /* Impute missing response variable values with mean */
              if &rv = . then
                do;
                  imputed = temp_imputed_&rv;
```

```
                    &rv = imputed;
                  end;


              temp_imputed_&rv = imputed_mean; * Assign the mean response
variable to a temporary value ;
              retain temp_imputed_&rv; * Retain the temporary value to a next
observation ;

              * If the variable of interest is imputed, then estimate MSE ;
              if imputed = . then
                MSE = 0;
              else
                MSE = (original - imputed)**2 / (&allrow - 1);
          run;

            /* Create a table after the simulation for the result */
          %let percent2 = %sysevalf(&percent * 100);
        proc SQL noprint;
            CREATE TABLE &mechanism.Mean_Impute&percent2 AS
              (
            SELECT SUM(MSE) AS MSE
                FROM imputed_dataMEAN&mechanism&percent2
              );
            CREATE TABLE &mechanism.Mean_Impute_MEAN_STD&percent2 AS
              (
            SELECT MEAN(&rv) AS Mean, STD(&rv) AS STD
                FROM imputed_dataMEAN&mechanism&percent2
              );
          QUIT;
    %end;
  %else
    %do;
        %let percent2 = %sysevalf(&percent * 100);
      data &mechanism.Mean_Impute&percent2;
          MSE = 0;
        run;
      %end;

%mend;


* REGIMP ;
/* Macro: Regression Imputation
   Definitions:
   dat: Dataset
   percent: % of data will be missing on the variable of interest (rv)
   mechanism: Mechanism will be assumed on the cause of missingess
              (MCAR, MAR, NMAR)
   rv: Response variable
   catevar: Categorical explantory variable. Used for assigning missing
values if mechanism is MAR
   grp1: Categorical outcome for group 1 from catevar
   grp2: Categorical outcome for group 2 from catevar
   Notes: The variables in this macro may vary from a specified dataset.
```

```
   May have to write the code manually to match the variables.
   Create dummy variables for categorical variables prior to PROC REG
*/
%macro regression_imputation(dat=, percent=, mechanism=, rv=, catevar=,
grp1=, grp2=);

  /* Set up the data set */
  data temp_data;
    set &dat;
      id + 1;
    original = &rv;
  run;

  proc SQL noprint;
    SELECT COUNT(*) INTO :allrow
      FROM temp_data;
  QUIT;

  /* MCAR*/
  %if "&mechanism" = "MCAR" %then
    %do;
    /* Get the number of rows for SRS purpose */
        proc SQL noprint;
        SELECT ROUND(COUNT(*) * &percent) INTO :nrow
          FROM temp_data;
        QUIT;

        proc surveyselect data = temp_data
        method = SRS n = &nrow out = temp_data2 noprint;
        run;

        data temp_data2;
          set temp_data2;
          &rv = .;
        run;

    %end;

  %else %if "&mechanism" = "MAR" %then
    %do;
      data &grp1.tab &grp2.tab;
        set temp_data;
          if &catevar = "&grp1" then
            output &grp1.tab;
        else
          output &grp2.tab;
      run;

      * Get the number of rows from the data ;
      proc SQL noprint;
        SELECT ROUND(COUNT(*) * &percent)
          INTO :nrow /* Create a macro variable called nrow where nrow is the
number of rows */
        FROM &grp1.tab;
      quit;
```

```
    /* Simple random sampling where every samples in the population
       has an equal chance of being selected */
    proc surveyselect data = &grp1.tab noprint
      method = SRS n = &nrow out = new&grp1.tab;
    run;

    data new&grp1.tab;
          set new&grp1.tab;
        &rv = .;
    run;

    data &grp1.tab;
          merge &grp1.tab new&grp1.tab;
        by id;
    run;

    data temp_data2;
      set &grp1.tab &grp2.tab;
    run;

    proc sort data = temp_data2;
          by id;
    run;

    %end;
%else %if "&mechanism" = "NMAR" %then
  %do;

    proc means data = temp_data median maxdec = 0 noprint;
      var &rv;
      output out = medtab median = &rv._median;
    run;

    data lowtab hightab;
      merge temp_data medtab (keep = &rv._median);
      if &rv._median = . then
          &rv._median = temp_median;
        else
          do;
          temp_median = &rv._median;
            retain temp_median;
          end;

        if &rv < &rv._median then
          output lowtab;
      else
        output hightab;

      drop temp_median &rv._median;
    run;

    * Get the number of rows from the data ;
    proc SQL noprint;
      SELECT ROUND(COUNT(*) * &percent)
```

```
          INTO :nrow /* Create a macro variable called nrow where nrow is the
number of rows */
        FROM lowtab;
      quit;

      /* Simple random sampling where every samples in the population
         has an equal chance of being selected */
      proc surveyselect data = lowtab noprint
        method = SRS n = &nrow out = newlowtab;
      run;

      data newlowtab;
            set newlowtab;
          &rv = .;
        run;

        data lowtab;
            merge lowtab newlowtab;
          by id;
      run;

      data temp_data2;
        set lowtab hightab;
      run;

      proc sort data = temp_data2;
            by id;
        run;
      %end;
  %else
      %put ERROR: Mechanism does not exist.;

  %if "&mechanism" = "MCAR" or "&mechanism" = "MAR" or
    "&mechanism" = "NMAR" %then
      %do;
        /* Combine dataset with complete and missing values */
        data temp_data3;
          merge temp_data temp_data2;
            by id;
        run;

      /* Subset the complete dataset */
        data reg_dat;
          set temp_data3;
          if Minority = "Yes" then
            cate_pred1 = 1;
          else
            cate_pred1 = 0;

          if Sex = "Female" then
              cate_pred2 = 1;
          else
            cate_pred2 = 0;
            if not missing(&rv);
      run;
```

```
        /* Fit the regression model with complete dataset
             output the parameter estimates to a new dataset */
        ODS GRAPHICS OFF;
        ODS OUTPUT ParameterEstimates = pe;
      proc reg data = reg_dat;
          model &rv = cate_pred1 cate_pred2 SES ;
        run;
        quit;
        ODS GRAPHICS ON;

        * Tranpose the dataset so that the estimates become variables instead
of observations ;
        proc transpose data = pe out = mod;
          var Estimate;
        run;

        * Rename the column variables in the regression model ;
        data mod (rename = (col1 = yhat col2 = x1 col3 = x2 col4 = x3));
            set mod (drop = _NAME_ _LABEL_);
        run;

        %let percent2 = %sysevalf(&percent * 100);
        data imputed_dataREG&mechanism&percent2 (drop = yhat x1 x2 x3
temp_yhat
        temp_x1 temp_x2 temp_x3);
          merge temp_data2 mod;

          * Fill in the parameter estimates for every rows ;
          if yhat = . and x1 = . and x2 = . and x3 = . then
            do;
            yhat = temp_yhat;
            x1 = temp_x1;
            x2 = temp_x2;
            x3 = temp_x3;
            end;

          * Impute missing values under different criteria;
          if Minority = "Yes" and Sex = "Female" and &rv = . then
            imputed = yhat + x1 + x2 + (x3 * SES);
          else if Minority = "Yes" and Sex ~= "Female" and &rv = . then
              imputed = yhat + x1 + (x3 * SES);
          else if Minority ~= "Yes" and Sex = "Female" and &rv = . then
              imputed = yhat + x2 + (x3 * SES);
          else if Minority ~= "Yes" and Sex ~= "Female" and &rv = . then
              imputed = yhat + (x3 * SES);
          else
              imputed = .;

          temp_yhat = yhat;
        temp_x1 = x1;
        temp_x2 = x2;
        temp_x3 = x3;

        * Retain the temporary value to a next observation ;
```

```
          retain temp_yhat;
          retain temp_x1;
          retain temp_x2;
          retain temp_x3;

          if imputed = . then
            MSE = 0;
          else
              do;
              MSE = (original - imputed)**2 / (&allrow - 1);
                  &rv = imputed;
              end;
        run;

        * If the variable of interest is imputed, then estimate MSE ;
        %let percent2 = %sysevalf(&percent * 100);
      proc SQL noprint;
          CREATE TABLE &mechanism.Reg_Impute&percent2 AS
            (
          SELECT SUM(MSE) AS MSE
              FROM imputed_dataREG&mechanism&percent2
            );
          CREATE TABLE &mechanism.REG_Impute_MEAN_STD&percent2 AS
            (
          SELECT MEAN(&rv) AS Mean, STD(&rv) AS STD
              FROM imputed_dataREG&mechanism&percent2
            );
        QUIT;
    %end;
  %else
    %do;
        %let percent2 = %sysevalf(&percent * 100);
      data &mechanism.Reg_Impute&percent2;
          MSE = 0;
        run;
      %end;

%mend;

* HOTDECK ;
/* Macro: (Sequential) Hot Deck Imputation
   Definitions:
   dat: Dataset
   percent: % of data will be missing on the variable of interest (rv)
   mechanism: Mechanism will be assumed on the cause of missingess
              (MCAR, MAR, NMAR)
   rv: Response variable
   catevar: Categorical explantory variable. Used for assigning missing
values if mechanism is MAR
   grp1: Categorical outcome for group 1 from catevar
   grp2: Categorical outcome for group 2 from catevar
   Notes: The variables in this macro may vary from a specified dataset.
   May have to write the code manually to match the variables.
   Create dummy variables for categorical variables prior to PROC REG
```

```
   This is a Sequential Hot Deck Imputation. The donor will be subset based
on the last value read by
   on the categorical variable.
*/
%macro hot_deck_imputation(dat=, mechanism=, percent=, rv=,catevar=, grp1=,
grp2=);

  * If the macro variable, percent, is greater than zero then go to this
block ;
  /* Set up the data set */
  data temp_data;
    set &dat;
      id + 1; * Create an id for merging purpose ;
      original = &rv;
  run;

  * Get the number of rows for MSE estimate ;
  proc SQL noprint;
    SELECT COUNT(*) INTO :allrow
    FROM temp_data;
  QUIT;

  * If the macro variable, mechanism, is MCAR then go to this block ;
  %if "&mechanism" = "MCAR" %then
    %do;
      /* Get the number of rows for SRS purpose */
      proc SQL noprint;
        SELECT ROUND(COUNT(*) * &percent) INTO :nrow
          FROM temp_data;
        QUIT;

        proc surveyselect data = temp_data
        method = SRS n = &nrow out = temp_data2 noprint;
        run;

        /* This DATA step creates an array for donor */
      data temp_data2;
        set temp_data2;
        array aschool{1} $;
        array aMinority{1} $;
        array aSex{1} $;
        &rv = .;

        do i = 1 to dim(aschool);
          aschool(i) = School;
          aMinority(i) = Minority;
          aSex(i) = Sex;
        end;

        drop i;
      run;
      %end;
  %else %if "&mechanism" = "MAR" %then
    %do;
      data &grp1.tab &grp2.tab;
```

```
        set temp_data;
          if &catevar = "&grp1" then
            output &grp1.tab;
        else
          output &grp2.tab;
      run;

      * Get the number of rows from the data ;
      proc SQL noprint;
        SELECT ROUND(COUNT(*) * &percent)
          INTO :nrow /* Create a macro variable called nrow where nrow is the
number of rows */
        FROM &grp1.tab;
      quit;

      /* Simple random sampling where every samples in the population
         has an equal chance of being selected */
      proc surveyselect data = &grp1.tab noprint
        method = SRS n = &nrow out = new&grp1.tab;
      run;

        /* This DATA step creates an array for donor */
      data new&grp1.tab;
        set new&grp1.tab;
        array aschool{1} $;
        array aMinority{1} $;
        array aSex{1} $;
        &rv = .;

        do i = 1 to dim(aschool);
          aschool(i) = School;
          aMinority(i) = Minority;
          aSex(i) = Sex;
        end;
        drop i;
      run;

        * Merge the data with both observed and missing values ;
        data temp_data2;
            merge &grp1.tab new&grp1.tab;
            by id;
        run;

    %end;
  %else %if "&mechanism" = "NMAR" %then
    %do;
      /* Output a median sales object in the table called medtab from PROC
MEANS */
      proc means data = temp_data median maxdec = 0 noprint;
        var &rv;
        output out = medtab median = &rv._median;
      run;

      /* Subset dataset into either lowtab or hightab */
      data lowtab hightab;
```

```
        merge temp_data medtab (keep = &rv._median);
        if &rv._median = . then
            &rv._median = temp_median;
          else
            do;
            temp_median = &rv._median;
              retain temp_median;
            end;

          if &rv < &rv._median then
            output lowtab;
        else
          output hightab;

        drop temp_median &rv._median;
      run;

      /* Get the number of rows from the data */
        proc SQL noprint;
          SELECT ROUND(COUNT(*) * &percent)
            INTO :nrow /* Create a macro variable called nrow where nrow is
the number of rows */
          FROM lowtab;
        quit;

        /* Simple random sampling where every samples in the population
           has an equal chance of being selected */
        proc surveyselect data = lowtab noprint
          method = SRS n = &nrow out = newlowtab;
        run;

          /* This DATA step creates an array for donor */
        data newlowtab;
          set newlowtab;
          array aschool{1} $;
          array aMinority{1} $;
          array aSex{1} $;

          do i = 1 to dim(aschool);
            aschool(i) = School;
            aMinority(i) = Minority;
            aSex(i) = Sex;
          end;

          &rv = .;

          drop i;
        run;

            /* Merge both datasets with complete and missing values */
            data temp_data2;
              merge lowtab newlowtab;
        by id;
            run;
```

```
      %end;
    %else
      %put ERROR: Mechanism does not match.;

  %if "&mechanism" = "MCAR" or "&mechanism" = "MAR" or "&mechanism" = "NMAR"
%then
      %do;

        /* If the last value of the school is read then output */
      data donor;
        set &dat;
        by school;
            id + 1;
        if last.school then
          output;
        keep school minority sex MathAch id;
      run;

        /* This DATA step fills in the missing value */
      data step1;
        merge temp_data2 (drop = MathAch) donor (rename = (id = donorid));
        by school;
        /* If the missing's respondent matches the donor's observed values,
then output the row */
        if (aschool1 = school & minority = aminority1 & asex1 = sex) or
          (aschool1 = school) then
            output;
          minority = aMinority1;
          sex = aSex1;
          drop aschool1 aMinority1 aSex1;
        run;

            %let percent2 = %sysevalf(&percent * 100);
        data imputed_dataHD&mechanism&percent2;
          merge temp_data step1 (rename = (MathAch = imputed));
          by id;
          MathAch = imputed;

            if imputed = . then
                  MSE = 0;
              else
                do;
                  MSE = (original - imputed)**2 / (&allrow - 1);
                    &rv = imputed;
                  end;
        run;

            /* Create a table after the simulation for the result */
        proc SQL noprint;
            CREATE TABLE &mechanism.HD_Impute&percent2 AS
              (
            SELECT SUM(MSE) AS MSE
                FROM imputed_dataHD&mechanism&percent2
              );
            CREATE TABLE &mechanism.HD_Impute_MEAN_STD&percent2 AS
```

```
            (
           SELECT MEAN(&rv) AS Mean, STD(&rv) AS STD
               FROM imputed_dataHD&mechanism&percent2
            );
        QUIT;
    %end;
  %else
    %do;
        %let percent2 = %sysevalf(&percent * 100);
      data &mechanism.HD_Impute&percent2;
          MSE = 0;
        run;
      %end;

%mend;

* MULTIIMP ;
/* Macro: multiple_imputation
   Definitions:
   dat: Dataset
   percent: % of data will be missing on the variable of interest (rv)
   mechanism: Mechanism will be assumed on the cause of missingess
              (MCAR, MAR, NMAR)
   rv: Response variable
   catevar: Categorical explantory variable. Used for assigning missing
values if mechanism is MAR
   grp1: Categorical outcome for group 1 from catevar
   grp2: Categorical outcome for group 2 from catevar
   Notes: The variables in this macro may vary from a specified dataset.
   May have to write the code manually to match the variables.
   Create dummy variables for categorical variables prior to PROC REG, PROC
MI & PROC MIANALYZE
*/
%macro multiple_imputation(dat=, percent=, mechanism=, rv=, catevar=, grp1=,
grp2=);

  /* Set up the data set */
  data temp_data;
    set &dat;
      id + 1;
    original = &rv;
  run;

  proc SQL noprint;
    SELECT COUNT(*) INTO :allrow
      FROM temp_data;
  QUIT;

  /* MCAR*/
  %if "&mechanism" = "MCAR" %then
    %do;
    /* Get the number of rows for SRS purpose */
        proc SQL noprint;
        SELECT ROUND(COUNT(*) * &percent) INTO :nrow
          FROM temp_data;
```

```
        QUIT;

        proc surveyselect data = temp_data
        method = SRS n = &nrow out = temp_data2 noprint;
        run;

        data temp_data2;
          set temp_data2;
          &rv = .;
        run;

    %end;

  %else %if "&mechanism" = "MAR" %then
    %do;
      data &grp1.tab &grp2.tab;
        set temp_data;
          if &catevar = "&grp1" then
            output &grp1.tab;
        else
          output &grp2.tab;
      run;

      * Get the number of rows from the data ;
      proc SQL noprint;
        SELECT ROUND(COUNT(*) * &percent)
          INTO :nrow /* Create a macro variable called nrow where nrow is the
number of rows */
        FROM &grp1.tab;
      quit;

      /* Simple random sampling where every samples in the population
         has an equal chance of being selected */
      proc surveyselect data = &grp1.tab noprint
        method = SRS n = &nrow out = new&grp1.tab;
      run;

      data new&grp1.tab;
            set new&grp1.tab;
          &rv = .;
      run;

      data &grp1.tab;
            merge &grp1.tab new&grp1.tab;
          by id;
      run;

      data temp_data2;
        set &grp1.tab &grp2.tab;
      run;

      proc sort data = temp_data2;
            by id;
      run;
```

```
      %end;
  %else %if "&mechanism" = "NMAR" %then
    %do;

      proc means data = temp_data median maxdec = 0 noprint;
        var &rv;
        output out = medtab median = &rv._median;
      run;

      data lowtab hightab;
        merge temp_data medtab (keep = &rv._median);
        if &rv._median = . then
            &rv._median = temp_median;
          else
            do;
            temp_median = &rv._median;
              retain temp_median;
            end;

          if &rv < &rv._median then
            output lowtab;
        else
          output hightab;

        drop temp_median &rv._median;
      run;

      * Get the number of rows from the data ;
      proc SQL noprint;
        SELECT ROUND(COUNT(*) * &percent)
          INTO :nrow /* Create a macro variable called nrow where nrow is the
number of rows */
        FROM lowtab;
      quit;

      /* Simple random sampling where every samples in the population
         has an equal chance of being selected */
      proc surveyselect data = lowtab noprint
        method = SRS n = &nrow out = newlowtab;
      run;

      data newlowtab;
            set newlowtab;
          &rv = .;
        run;

        data lowtab;
            merge lowtab newlowtab;
          by id;
      run;

      data temp_data2;
        set lowtab hightab;
      run;
```

```
      proc sort data = temp_data2;
            by id;
        run;
      %end;
  %else
      %put ERROR: Mechanism does not exist.;

  %if "&mechanism" = "MCAR" or "&mechanism" = "MAR" or
    "&mechanism" = "NMAR" %then
      %do;
        /* Combine dataset with complete and missing values */
        data temp_data3;
          merge temp_data temp_data2;
            by id;
        run;

      /* Subset the complete dataset */
        data reg_dat;
          set temp_data3;
          if Minority = "Yes" then
            Minority_DV = 1;
          else
            Minority_DV = 0;

          if Sex = "Female" then
              Sex_DV = 1;
          else
            Sex_DV = 0;
      run;

      * Use PROC MI to implement multiple imputation ;
      proc mi data = reg_dat out = miout;
        var &rv Sex_DV Minority_DV SES;
      run;

      * Use PROC REG to estimate parameters from a regression model ;
      title "Multiple Imputation - Regression";
      proc reg data = miout plots = none;
        model &rv = Sex_DV Minority_DV SES;
        by _imputation_;
        ODS OUTPUT ParameterEstimates = param;
      run;

      * Combine parameter estimates into one using PROC MIANALYZE ;
      proc mianalyze parms = param;
        modeleffects intercept Sex_DV Minority_DV SES; * Take an average of
the individual coefficients from all models into one ;
        ODS OUTPUT ParameterEstimates = imp_param (keep = parm estimate);
      run;
      title;

        * Tranpose the dataset so that the estimates become variables instead
of observations ;
        proc transpose data = imp_param out = mod;
          var Estimate;
```

```
        run;

        * Rename the column variables in the regression model ;
        data mod (rename = (col1 = yhat col2 = x1 col3 = x2 col4 = x3));
            set mod (drop = _NAME_);
        run;

        %let percent2 = %sysevalf(&percent * 100);
        data imputed_dataMI&mechanism&percent2 (drop = yhat x1 x2 x3
temp_yhat
        temp_x1 temp_x2 temp_x3);
        merge temp_data3 mod;

            * Fill in the parameter estimates for every rows ;
            if yhat = . and x1 = . and x2 = . and x3 = . then
              do;
            yhat = temp_yhat;
            x1 = temp_x1;
            x2 = temp_x2;
            x3 = temp_x3;
              end;

            * Impute missing values under different criteria;
          if Minority = "Yes" and Sex = "Female" and &rv = . then
            imputed = yhat + x1 + x2 + (x3 * SES);
          else if Minority = "Yes" and Sex ~= "Female" and &rv = . then
              imputed = yhat + x1 + (x3 * SES);
          else if Minority ~= "Yes" and Sex = "Female" and &rv = . then
              imputed = yhat + x2 + (x3 * SES);
            else if Minority ~= "Yes" and Sex ~= "Female" and &rv = . then
              imputed = yhat + (x3 * SES);
          else
              imputed = .;

          temp_yhat = yhat;
        temp_x1 = x1;
        temp_x2 = x2;
        temp_x3 = x3;

          * Retain the temporary value to a next observation ;
          retain temp_yhat;
            retain temp_x1;
            retain temp_x2;
            retain temp_x3;

            if imputed = . then
            MSE = 0;
            else
              do;
                MSE = (original - imputed)**2 / (&allrow - 1);
                &rv = imputed;
              end;
        run;
```

```
        * If the variable of interest is imputed, then estimate MSE ;
        %let percent2 = %sysevalf(&percent * 100);
      proc SQL noprint;
          CREATE TABLE &mechanism.MI_Impute&percent2 AS
            (
          SELECT SUM(MSE) AS MSE
              FROM imputed_dataMI&mechanism&percent2
            );
          CREATE TABLE &mechanism.MI_Impute_MEAN_STD&percent2 AS
            (
          SELECT MEAN(&rv) AS Mean, STD(&rv) AS STD
              FROM imputed_dataMI&mechanism&percent2
            );
        QUIT;
    %end;
  %else
    %do;
        %let percent2 = %sysevalf(&percent * 100);
      data &mechanism.MI_Impute&percent2;
          MSE = 0;
        run;
      %end;

%mend;

* MATDAT ;
* Math Achieve Dataset ;
* Read in raw data before doing any procedure ;
data MathAchieve;
  infile "&wd\MathAchieve.csv" firstobs = 2 dsd truncover;
  input School Minority $ Sex $ SES MathAch MEANSES;
  if Minority = "No" then
    Minority_DV = 0;
  else
    Minority_DV = 1;

  if Sex = "Female" then
    Sex_DV = 0;
  else
    Sex_DV = 1;
run;

* Create a SAS Macro to start the simulation for mean and std under different
missing data mechanism ;
%macro LD_loop(mechanism=);

  %do i = 2 %to 15;
    %LDprocedure(dat= MathAchieve, percent = %sysevalf(-.05 + .05 * &i),
      mechanism = &mechanism, rv= MathAch, catevar= Minority, grp1=Yes,
grp2=No);
  %end;

%mend LD_loop;

* Call the SAS Macro pro ;
```

```
%LD_loop(mechanism = MCAR);
%LD_loop(mechanism = MAR);
%LD_loop(mechanism = NMAR);

* Create a SAS Macro to start the simulation for mean and std under different
missing data mechanism ;
%macro MEI_loop(mechanism=);

  %do i = 2 %to 15;
    %mean_imputation(dat= MathAchieve, percent = %sysevalf(-.05 + .05 * &i),
      mechanism = &mechanism, rv= MathAch, catevar= Minority, grp1=Yes,
grp2=No);
  %end;

%mend MEI_loop;

* Call the SAS Macro imputation ;
%MEI_loop(mechanism=MCAR);
%MEI_loop(mechanism=MAR);
%MEI_loop(mechanism=NMAR);

* Create a SAS Macro to start the simulation for mean and std under different
missing data mechanism ;
%macro RI_loop(mechanism=);

  %do i = 2 %to 15;
    %regression_imputation(dat= MathAchieve, percent = %sysevalf(-.05 + .05 *
&i),
      mechanism = &mechanism, rv= MathAch, catevar= Minority, grp1=Yes,
grp2=No);
  %end;

%mend RI_loop;

* Call the SAS Macro imputation ;
%RI_loop(mechanism=MCAR);
%RI_loop(mechanism=MAR);
%RI_loop(mechanism=NMAR);

* Create a SAS Macro to start the simulation for mean and std under different
missing data mechanism ;
%macro HDI_loop(mechanism=);

  %do i = 2 %to 15;
    %hot_deck_imputation(dat= MathAchieve, percent = %sysevalf(-.05 + .05 *
&i),
      mechanism = &mechanism, rv= MathAch, catevar= Minority, grp1=Yes,
grp2=No);
  %end;

%mend HDI_loop;

* Call the SAS Macro imputation ;
%HDI_loop(mechanism=MCAR);
%HDI_loop(mechanism=MAR);
```

```
%HDI_loop(mechanism=NMAR);

* Create a SAS Macro to start the simulation for mean and std under different
missing data mechanism ;
%macro MI_loop(mechanism=);

  %do i = 2 %to 15;
    %multiple_imputation(dat= MathAchieve, percent = %sysevalf(-.05 + .05 *
&i),
      mechanism = &mechanism, rv= MathAch, catevar= Minority, grp1=Yes,
grp2=No);
  %end;

%mend MI_loop;

* Call the SAS Macro imputation ;
%MI_loop(mechanism=MCAR);
%MI_loop(mechanism=MAR);
%MI_loop(mechanism=NMAR);

* Create a DATA step called percent where percent will be used to represents
the
  percent of missingness when merge into the respective imputation data set;
data percent;
  do Percent = 0 to 70 by 5;
    output;
  end;
run;

* Take the average of all observed dataset and store a data set called
complete_dat;
proc means data = MathAchieve std noprint;
  var MathAch;
  output out = complete_dat;
run;

* Transpose a data set so it can be used later ;
proc transpose data = complete_dat out = t_complete_dat;
run;

data t_complete_dat (drop = _NAME_);
  set t_complete_dat (rename = (COL4 = MEAN COL5 = STD));
  drop COL1 COL2 COL3;
  if _NAME_  = "MathAch";
run;

* Concatenate rows together into one dataset. Repeat this for three different
SAS Macro ;
%macro tab(string=);

  data &string.Table;
    set t_complete_dat (rename = (MEAN = MathAch_Mean STD = MathAch_StdDev))
        &string.SUMMARY5
        &string.SUMMARY10 &string.SUMMARY15
        &string.SUMMARY20 &string.SUMMARY25
```

```
        &string.SUMMARY30 &string.SUMMARY35
        &string.SUMMARY40 &string.SUMMARY45
        &string.SUMMARY50 &string.SUMMARY55
        &string.SUMMARY60 &string.SUMMARY65
        &string.SUMMARY70;
  run;

  data &string.Table;
    merge percent &string.Table;
  run;

%mend tab;

%tab(string = MCAR);
%tab(string = MAR);
%tab(string = NMAR);

* Get all MSE from simulation into one dataset ;
* Concatenate rows together into one dataset. Repeat this for four different
SAS Macro ;
%macro tab2(string=, imputation=);

  data &string.&imputation._Impute0;
    MSE = 0;
  run;

  data &string.&imputation._Imput_Tab;
    set &string.&imputation._Impute0 &string.&imputation._Impute5
        &string.&imputation._Impute10 &string.&imputation._Impute15
        &string.&imputation._Impute20 &string.&imputation._Impute25
        &string.&imputation._Impute30 &string.&imputation._Impute35
        &string.&imputation._Impute40 &string.&imputation._Impute45
        &string.&imputation._Impute50 &string.&imputation._Impute55
        &string.&imputation._Impute60 &string.&imputation._Impute65
        &string.&imputation._Impute70;
  run;

  data &string.&imputation._Imput_Tab;
    merge percent &string.&imputation._Imput_Tab;
  run;

%mend tab2;

%tab2(string = MCAR, imputation = Mean);
%tab2(string = MAR, imputation = Mean);
%tab2(string = NMAR, imputation = Mean);
%tab2(string = MCAR, imputation = Reg);
%tab2(string = MAR, imputation = Reg);
%tab2(string = NMAR, imputation = Reg);
%tab2(string = MCAR, imputation = HD);
%tab2(string = MAR, imputation = HD);
%tab2(string = NMAR, imputation = HD);
%tab2(string = MCAR, imputation = MI);
%tab2(string = MAR, imputation = MI);
%tab2(string = NMAR, imputation = MI);
```

```
/* Merge all MSE from each imputation method into one dataset */
%macro mech_MSE_data(mechanism=);
  data &mechanism._MSE_data;
    merge &mechanism.Mean_Imput_Tab (rename = (MSE = MEI_MSE))
    &mechanism.Reg_Imput_Tab (rename = (MSE = REG_MSE))
    &mechanism.HD_Imput_Tab (rename = (MSE = HD_MSE))
    &mechanism.MI_Imput_Tab (rename = (MSE = MI_MSE));
    by Percent;
  run;
%mend;

%mech_MSE_data(mechanism=MCAR);
%mech_MSE_data(mechanism=MAR);
%mech_MSE_data(mechanism=NMAR);

* Get all mean and std from the simulation into one dataset ;
* Concatenate rows of mean and std together into one dataset. Repeat this for
each different SAS Macro ;
%macro tab3(string=, imputation=);

  data &string.&imputation._MEAN_STD_Tab;
    set t_complete_dat
        &string.&imputation._Impute_MEAN_STD5
        &string.&imputation._Impute_MEAN_STD10
        &string.&imputation._Impute_MEAN_STD15
        &string.&imputation._Impute_MEAN_STD20
        &string.&imputation._Impute_MEAN_STD25
        &string.&imputation._Impute_MEAN_STD30
        &string.&imputation._Impute_MEAN_STD35
        &string.&imputation._Impute_MEAN_STD40
        &string.&imputation._Impute_MEAN_STD45
        &string.&imputation._Impute_MEAN_STD50
        &string.&imputation._Impute_MEAN_STD55
        &string.&imputation._Impute_MEAN_STD60
        &string.&imputation._Impute_MEAN_STD65
        &string.&imputation._Impute_MEAN_STD70;
  run;

  data &string.&imputation._MEAN_STD_Tab;
    merge percent &string.&imputation._MEAN_STD_Tab;
  run;

%mend tab3;

%tab3(string = MCAR, imputation = Mean);
%tab3(string = MAR, imputation = Mean);
%tab3(string = NMAR, imputation = Mean);
%tab3(string = MCAR, imputation = Reg);
%tab3(string = MAR, imputation = Reg);
%tab3(string = NMAR, imputation = Reg);
%tab3(string = MCAR, imputation = HD);
%tab3(string = MAR, imputation = HD);
%tab3(string = NMAR, imputation = HD);
%tab3(string = MCAR, imputation = MI);
```

```
%tab3(string = MAR, imputation = MI);
%tab3(string = NMAR, imputation = MI);

/* Merge all MEAN and STD from each imputation method and likewise deletion
into one dataset */
%macro mech_SUMMARY_data(string=);
  data &string._SUMMARY_data;
    merge &string.Table (rename = (MathAch_Mean = LD_Mean MathAch_StdDev =
LD_STD))
    &string.MEAN_MEAN_STD_Tab (rename = (Mean = MEI_Mean STD = MEI_STD))
    &string.REG_MEAN_STD_Tab (rename = (Mean = REG_Mean STD = REG_STD))
    &string.HD_MEAN_STD_Tab (rename = (Mean = HD_Mean STD = HD_STD))
    &string.MI_MEAN_STD_Tab (rename = (Mean = MI_Mean STD = MI_STD));
    by Percent;
  run;
%mend;

%mech_SUMMARY_data(string=MCAR);
%mech_SUMMARY_data(string=MAR);
%mech_SUMMARY_data(string=NMAR);

/* To print the graphics, I used ODS Graphics Designer to design SAS graphics
instead of traditional PROCs
   To access to the ODS Graphics Designer, [Tools > ODS Graphics Designer] */

* SURDAT ;
* University of Adelaide is located in South Australia ;
data survey;
  infile "&wd.\survey.csv" dsd firstobs = 2;
  input Sex $ WrHndt $ NWHndt $ WHnd $ Fold $ Pulset $ Clap $ Exer $ Smoke $
Heightt $ MI $ Age;

  * Convert all NA to . since this was pulled from R ;
  if WrHndt = "NA" then
    WrHndt = ".";
  if WHnd = "NA" then
    WHnd = ".";
  if NWHndt = "NA" then
    NWHndt = ".";
  if Pulset = "NA" then
    Pulset = ".";
  if Clap = "NA" then
    Clap = ".";
  if Smoke = "NA" then
    Smoke = ".";
  if Heightt = "NA" then
    Heightt = ".";
  if MI = "NA" then
    MI = ".";

  * Include dummy variables for regression and MI later ;
  if exer = "Never" then
    exer_DV = 0;
  else if exer = "Some" then
    exer_DV = 1;
```

```
  else if exer = "Freq" then
    exer_DV = 2;

  if sex = "Female" then
    sex_DV = 1;
  else
    sex_DV = 0;

  WrHnd = WrHndt * 1;
  Pulse = Pulset * 1;
  Height = Heightt * 1;
  NWHnd = NWHndt * 1;

  id + 1;

  drop WrHndt Pulset Heightt NWHndt;
run;

/* Imputation for Height */

/* Mean Imputation */

* Take the average of observed data and store in a data set called
mean_value;
proc means data = survey mean noprint;
  var height;
  output out = meandat mean = mean_value;
run;

* This DATA step fills in the missing values ;
data meanimputation_height;
  merge survey meandat (keep = mean_value);

  if mean_value = . then
    mean_value = temp_mean_value;

  if height = . then
    height = mean_value;

  temp_mean_value = mean_value;
  retain temp_mean_value;

  drop temp_mean_value mean_value;
run;

title "Mean Imputation for Survey dataset";
proc means data = meanimputation_height mean var std;
  var height;
run;
title;
footnote;

/* Regression Imputation */

* Estimate a regression model and store in a data set called regdat;
```

```
proc reg data = survey noprint;
  model Height = Age WrHnd NWHnd Pulse sex_DV / selection = stepwise;
  output out = regdat
               predicted = yhat;
run;
quit;

* This DATA step fills in the missing values ;
data regimputation;
  set regdat;
  if height = . then
    height = yhat;
run;

title "Regression Imputation for Survey dataset";
proc means data = regimputation mean var std;
  var height;
run;
title;
footnote;

/* Hot Deck Imputation */

* This DATA step creates an array for the donor ;
data missing;
  set survey;
  array aSex{1} $;
  array aWHnd{1} $;
  array AFold{1} $;
  array aExer(1) $;
  array aSmoke(1) $;

  do i = 1 to dim(aSex);
    aSex(i) = Sex;
    aWHnd(i) = WHnd;
    AFold(i) = Fold;
      aExer(i) = Exer;
    aSmoke(i) = Smoke;
  end;

  where height = .;
  drop i;
run;

proc sort data = missing out = sorted_missing;
  by sex;
run;

proc sort data = survey out = sorted_survey;
  by sex;
run;

* If the last value of Sex is read in the data set then output;
data donor;
  set sorted_survey;
```

```
  by sex;
  if last.sex & sex ~= "NA" then
    output;
run;

* This DATA step fills in the missing values ;
data step1;
  merge sorted_missing donor (rename = (id = donorid height = donorheight));
  by Sex;
  if (aSex1 = Sex) then
    output;

  Height = donorheight;
run;

proc sort data = step1;
  by id;
run;

data step2;
  merge missing step1;
  by id;
  Height = donorheight;

  Sex = aSex1;
  aWHnd = aWHnd1;
  Fold = AFold1;
  Exer = aExer1;
  Smoke = aSmoke1;
  drop aSex1 aWHnd1 AFold1 aExer1 aSmoke1 aWHnd;
run;

data hotdeckimputation;
  merge survey step2;
  by id;
run;

title "Hot Deck Imputation for Survey dataset";
proc means data = hotdeckimputation mean var std;
  var height;
run;

/* Multiple Imputation */

* PROC MI will generate multiple imputation based on the regression model ;
proc mi data = survey out = miout nimpute = 5;
  var Height Age WrHnd sex_DV;
  fcs regression(Height Age WrHnd sex_DV);
run;

 * Use PROC REG to estimate parameters from a regression model ;
title "Multiple Imputation - Regression";
proc reg data = miout plots = none;
  model Height = WrHnd;
  by _imputation_;
```

```
  ods output ParameterEstimates = param;
run;

* Combine parameter estimates into one using PROC MIANALYZE ;
 proc mianalyze parms = param;
   modeleffects intercept WrHnd; * Take an average of the individual
coefficients from all models into one ;
   ODS OUTPUT ParameterEstimates = imp_param (keep = parm estimate);
run;

* Tranpose a dataset ;
proc transpose data = IMP_PARAM out = t_imp_param;
run;

data multipleimputation (drop = yhat x1 temp_yhat temp_x1);
  merge survey t_imp_param (rename = (col1 = yhat col2 = x1) drop = _name_);

  * Fill in the parameter estimates for every rows ;
  if yhat = . and x1 = . then
    do;
      yhat = temp_yhat;
      x1 = temp_x1;
    end;

  * Assign the mean sales to a temporary value ;
  temp_yhat = yhat;
  temp_x1 = x1;

  * Retain the temporary value to a next observation ;
  retain temp_yhat;
  retain temp_x1;

  if Height = . then
      Height = yhat + (x1 * WrHnd);

run;

title "Multiple Imputation for Survey dataset";
proc means data = multipleimputation mean var std;
  var height;
run;
title;

data height_result;
  set height_result;
  if _n_ = 1 then
    Imputation = "Complete Dataset";
run;

title "Original Survey Descriptive Summary";
proc means data = survey mean var std noprint;
  var height;
  output out = original mean = o_mean std = o_std var = o_var;
run;
title;
```

```
title "Mean Imputation Descriptive Summary";
proc means data = meanimputation_height mean var std noprint;
  var height;
  output out = mei mean = me_mean std = me_std var = me_var;
run;
title;

title "Regression Imputation Descriptive Summary";
proc means data = regimputation mean var std noprint;
  var height;
  output out = reg  mean = reg_mean std = reg_std var = reg_var;
run;
title;

title "Hot Deck Imputation Descriptive Summary";
proc means data = hotdeckimputation mean var std noprint;
  var height;
  output out = hd mean = hd_mean std = hd_std  var = hd_var;
run;
title;

title "Multiple Imputation Descriptive Summary";
proc means data = multipleimputation mean var std noprint;
  var height;
  output out = mi mean = mi_mean std = mi_std var = mi_var;
run;
title;


data height_result (drop = _TYPE_ _FREQ_);
  length Imputation $22.;
  set original (rename = (o_mean = mean o_var = var o_std = std))
      mei (rename = (me_mean = mean me_var = var me_std = std))
      reg (rename = (reg_mean = mean reg_var = var reg_std = std))
      hd  (rename = (hd_mean = mean hd_var = var hd_std = std))
      mi (rename = (mi_mean = mean mi_var = var mi_std = std));
  if _n_ = 1 then
    Imputation = "Complete Dataset";
  else if _n_ = 2 then
    Imputation = "Mean Imputation";
  else if _n_ = 3 then
    Imputation = "Regression Imputation";
  else if _n_ = 4 then
    Imputation = "Hot Deck Imputation";
  else if _n_ = 5 then
    Imputation = "Multiple Imputation";
run;

title "Imputation for Height";
proc print data =  height_result noobs label split = "*"
  style(header) = {background = midnightblue color = white}
  style(table) = {rules = box frame = box cellspacing = 1 background = STB
                  borderwidth = 2 bordercolor = STB};
  label mean = "Mean"
```

```
        var = "Variance"
        std = "Standard*Deviation";
run;
title;

/* Imputation for Pulse */

/* Mean Imputation */

* Tahe the average of the observed data set prior to imputation ;
proc means data = survey mean var std noprint;
  var pulse;
  output out = meandat mean = mean_value;
run;

* This DATA step fills in the missing values ;
data meanimputation;
  merge survey meandat (keep = mean_value);

  if mean_value = . then
    mean_value = temp_mean_value;

  if height = . then
    Pulse = mean_value;

  temp_mean_value = mean_value;
  retain temp_mean_value;

  drop temp_mean_value mean_value;
run;

title "Mean Imputation for Survey dataset";
proc means data = survey mean var std;
  var Pulse;
run;
title;

/* Regression Imputation */

* Estimate a regression model prior to imputation ;
proc reg data = survey noprint ;
  model pulse = Age WrHnd NWHnd Height sex_DV / selection = stepwise;
  output out = regdat
               predicted = yhat;
run;
quit;

* This DATA step fills in the missing values ;
data regimputation;
  set regdat;
  if pulse = . then
    pulse = yhat;
run;

/* Hot Deck Imputation */
```

```
* Create a DATA step to create an array for the donor ;
data missing;
  set survey;
  array aSex{1} $;
  array aWHnd{1} $;
  array AFold{1} $;
  array aExer(1) $;
  array aSmoke(1) $;

  do i = 1 to dim(aSex);
    aSex(i) = Sex;
    aWHnd(i) = WHnd;
    AFold(i) = Fold;
      aExer(i) = Exer;
    aSmoke(i) = Smoke;
  end;

  where pulse = .;
  drop i;
run;

proc sort data = missing out = sorted_missing;
  by sex;
run;

proc sort data = survey out = sorted_survey;
  by sex;
run;

* If the last value of Sex is read then output ;
data donor;
  set sorted_survey;
  by sex;
  if last.sex & sex ~= "NA" then
    output;
run;

* This DATA step fills in the missing values ;
data step1;
  merge sorted_missing donor (rename = (id = donorid pulse = donorpulse));
  by Sex;
  if (aSex1 = Sex) then
    output;

  pulse = donorpulse;
run;

proc sort data = step1;
  by id;
run;

data step2;
  merge missing step1;
  by id;
```

```
  pulse = donorpulse;

  Sex = aSex1;
  aWHnd = aWHnd1;
  Fold = AFold1;
  Exer = aExer1;
  Smoke = aSmoke1;
  drop aSex1 aWHnd1 AFold1 aExer1 aSmoke1 aWHnd;
run;

data hotdeckimputation;
  merge survey step2;
  by id;
run;

title "Hot Deck Imputation for Survey dataset";
proc means data = hotdeckimputation mean var std;
  var Pulse;
run;

/* Multiple Imputation */

* PROC MI will generate multiple datasets using regression model ;
proc mi data = survey out = miout  nimpute = 5;;
  var Pulse Height Age WrHnd sex_DV;
  fcs regression(Pulse Age WrHnd sex_DV);
run;

 * Use PROC REG to estimate parameters from a regression model ;
title "Multiple Imputation - Regression";
proc reg data = miout plots = none;
  model Pulse = Height WrHnd;
  by _imputation_;
  ods output ParameterEstimates = param;
run;

* Combine parameter estimates into one using PROC MIANALYZE ;
 proc mianalyze parms = param;
   modeleffects intercept Height WrHnd; * Take an average of the individual
coefficients from all models into one ;
   ODS OUTPUT ParameterEstimates = imp_param (keep = parm estimate);
run;

* Tranpose a dataset ;
proc transpose data = IMP_PARAM out = t_imp_param;
run;

data multipleimputation (drop = yhat x1 temp_yhat temp_x1 x2 temp_x2);
  merge survey t_imp_param (rename = (col1 = yhat col2 = x1 col3 = x2) drop =
_name_);

  * Fill in the parameter estimates for every rows ;
  if yhat = . and x1 = . and x2 = . then
    do;
      yhat = temp_yhat;
```

```
      x1 = temp_x1;
        x2 = temp_x2;
    end;

  * Assign the mean sales to a temporary value ;
  temp_yhat = yhat;
  temp_x1 = x1;
  temp_x2 = x2;

  * Retain the temporary value to a next observation ;
  retain temp_yhat;
  retain temp_x1;
  retain temp_x2;

  if Pulse = . then
      Pulse = yhat + (x1 * Height) + (x2 * WrHnd);

run;

title "Original Survey Descriptive Summary";
proc means data = survey mean var std noprint;
  var Pulse;
  output out = original mean = o_mean std = o_std var = o_var;
run;
title;

title "Mean Imputation Descriptive Summary";
proc means data = meanimputation mean var std noprint;
  var Pulse;
  output out = mei mean = me_mean std = me_std var = me_var;
run;
title;

title "Regression Imputation Descriptive Summary";
proc means data = regimputation mean var std noprint;
  var Pulse;
  output out = reg  mean = reg_mean std = reg_std var = reg_var;
run;
title;

title "Hot Deck Imputation Descriptive Summary";
proc means data = hotdeckimputation mean var std noprint;
  var Pulse;
  output out = hd mean = hd_mean std = hd_std  var = hd_var;
run;
title;

title "Multiple Imputation Descriptive Summary";
proc means data = multipleimputation mean var std noprint;
  var Pulse;
  output out = mi mean = mi_mean std = mi_std var = mi_var;
run;
title;
```

```
data pulse_result (drop = _TYPE_ _FREQ_);
  length Imputation $22.;
  set original (rename = (o_mean = mean o_var = var o_std = std))
      mei (rename = (me_mean = mean me_var = var me_std = std))
      reg (rename = (reg_mean = mean reg_var = var reg_std = std))
      hd  (rename = (hd_mean = mean hd_var = var hd_std = std))
      mi (rename = (mi_mean = mean mi_var = var mi_std = std));
  if _n_ = 1 then
    Imputation = "Complete Dataset";
  else if _n_ = 2 then
    Imputation = "Mean Imputation";
  else if _n_ = 3 then
    Imputation = "Regression Imputation";
  else if _n_ = 4 then
    Imputation = "Hot Deck Imputation";
  else if _n_ = 5 then
    Imputation = "Multiple Imputation";
run;

title "Imputation for Pulse";
proc print data = pulse_result noobs label split = "*"
  style(header) = {background = midnightblue color = white}
  style(table) = {rules = box frame = box cellspacing = 1 background = STB
                  borderwidth = 2 bordercolor = STB};
  label mean = "Mean"
        var = "Variance"
        std = "Standard*Deviation";
run;
title;

/* END OF SAS CODE */
```