Comparison of Two Newly Developed Multiple Imputation Methods for MNAR Cross-Sectional Data

A Thesis submitted to the

College of Graduate and Postdoctoral Studies

in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy

in the Collaborative Biostatistics Program

within the School of Public Health

University of Saskatchewan

Saskatoon, Saskatchewan, Canada

By

April Xianxian Liu

PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other uses of materials in this thesis in whole or part should be addressed to:

Department Head
School of Public Health
College of Medicine
University of Saskatchewan
104 Clinic Place,
Saskatoon, SK, S7N 2Z4
Canada

OR

Dean
College of Graduate and Postdoctoral Studies
University of Saskatchewan
116 Thorvaldson Building, 110 Science Place
Saskatoon, SK, S7N 5C9
Canada

ACKNOWLEDGMENTS

I would like to first and foremostly express my sincere gratitude to the University of Saskatchewan for giving me the opportunity and financial support to pursue my biostatistics PhD. I would like to thank the Canadian Center of Health and Safety in Agriculture and the Rand Health Insurance Experiment for providing me with the datasets used for my study. I would like to thank the Collaborative Biostatistics Program for accepting me as a PhD student.

I would like to give my first thanks to my supervisor, Dr. Punam Pahwa, for all the many lessons she taught me for the last 6 years; in statistics, in research, and most importantly in being a human being. I will never forget them.

I would like to thank my research committee: Drs. Shahedul Khan, Bonnie Janzen, Cindy Feng, Andrei Volodin, and James Dosman for their encouragements and expert-level suggestions.

I would like to thank my mother Shelley for always lovingly guided me to the well-worned path yet always offering me her greatest support when I strayed to pursue my dreams and interests.

I would like to thank all my Saskatoon friends; some are busy pursuing their own academic careers, some have begun their post-academic careers. Thank you for your contributions in making my Saskatchewan experience a great one. We will meet again someday!

DEDICATION

This thesis is dedicated to two amazing women who left this world before I could complete my PhD: my grandmother Jinzhu Zhou and my best friend and mother-figure Sanja Avlijas. Although I don't know where to find you to share my accomplishments, you two were never far from my thoughts as I went through this difficult yet rewarding process. The love you gave me during your time in this world was my greatest source of strength and purpose for getting my PhD.

ABSTRACT

The problem of missing not at random (MNAR) data is a highly complex problem to the difficulty of joint modeling the outcome values and missing pattern while taking the variability of the missing data into consideration. In recent years, two methods by Galimard et. al (2016) and Ogundimu & Collins (2017) each developed their own multiple imputation (MI) methods for handling MNAR data. However, they have yet to be tested for their effectiveness in research sufficiently. This dissertation investigates the effectiveness of Galimard et. al and Ogundimu & Collins' MIs alongside complete case (CC) analysis and Rubin's MI when applied to two real-life datasets of different size ($n_1 = 4451$, $n_2 = 1607$) with induced missing data of MCAR, MAR, and MNAR mechanisms of 15%, 30%, and 50% missing data percentage. In addition, the methods will also be applied to simulated datasets with imputation and response models more complicated than in Galimard et. al and Ogundimu & Collins' studies to see how widely they can be applied in datasets with different missing mechanisms and data percentage. It was found in the application results that Galimard et. al's MI delivered the same results as CC in all missing mechanism and percentage combinations. For both datasets, Ogundimu & Collins' MI performed better than the other 3 methods for 50% MNAR, though overall, both Galimard et. al and Ogundimu & Collins' MIs performed better on MCAR and MAR data than MNAR. In simulation, Galimard et. al's MI also delivered results consistently identical to CC for all missing percentage and mechanism combinations. Ogundimu & Collins' MI consistently delivered superior results than the other 3 methods for 15% and 30% MNAR. However, Ogundimu & Collins' MI should be used with caution because it did not converge for 50% missing and only converged for approximately 100 – 400 datasets out of 1000 for 15% and 30%. It will be interesting if future studies can apply Galimard et. al and Ogundimu & Collins' MI methods other real-life datasets and easily-converge simulated datasets to see how well they can work when applied broadly in research and industry.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

MI:	Multiple Imputation
MCAR:	Missing Completely At Random
MAR:	Missing At Random
MNAR:	Missing Not At Random
SRHS:	Saskatchewan Rural Health Study
RANDHIE:	RAND Health Insurance Experiment
CC	Complete Case Analysis
MSE	Mean Square Error
CI	Confidence Interval

CHAPTER 1

INTRODUCTION

The issue of missing data is a frequently encountered problem in research and has numerous causes, including nonresponse from subjects and improper data entry. Without proper handling, missing data can significantly impact the results of analyses and lead to biased parameter estimates (Carpenter & Kenward, 2012). In order to choose the best method for handling missing data, the missing data mechanism (reason) needs to be taken into consideration (Kenward & Molenberghs, 2007). There are three principal missing data mechanisms: (i) Missing Completely At Random (MCAR); (ii) Missing At Random (MAR); and (iii) Missing Not At Random (MNAR) (Rubin, 1987; Little & Rubin, 2002). For MCAR, the missing data are neither missing due to values of the outcome variable (the variable with missing data in the proposed study) nor the covariates (variables that significantly influences the outcome variable in the proposed study), which allows unbiased parameters to be estimated through complete case (CC) analysis. However, this mechanism occurs rarely in research (Stef van Buuren, 2012). For MAR, the missing data are missing due to values of the covariate data, and therefore can be handled by using information given by the covariates. A widely used method originally designed to handle MAR data is multiple imputation (MI). MI involves replacing the missing data with multiple sets of possible values generated with an imputation model, model the multiple "complete" datasets, and then combine the results from each "complete" datasets to reach a final result (Rubin, 1987; Little & Rubin, 2002). The method is widely used due to its simplicity of implementation and its ability to account for the uncertainty of the missing data (Sterne et al., 2009). MNAR is the most difficult missing data mechanism to handle because the missing data are influenced by the values of the outcome variable itself (Kenward & Molenberghs, 2007). It occurs frequently in health and social science research where questions may occasionally be awkward to answer or answering the questions can lead to other detrimental effects on their lives (Galimard, Chevret, Protopopescu, & Resche-Rigon, 2016). Examples include questions related to money (earned and spent) and the participants' health conditions. To produce more accurate research results where the outcome is related to money or

health (in spite of missing data), newly developed methods designed to handle MNAR data in the outcome variables must go through vigorous testing using simulations and applications to more datasets. The missing data mechanisms and methods to handle them mentioned from the above sections are presented in Figure 1.1 below.

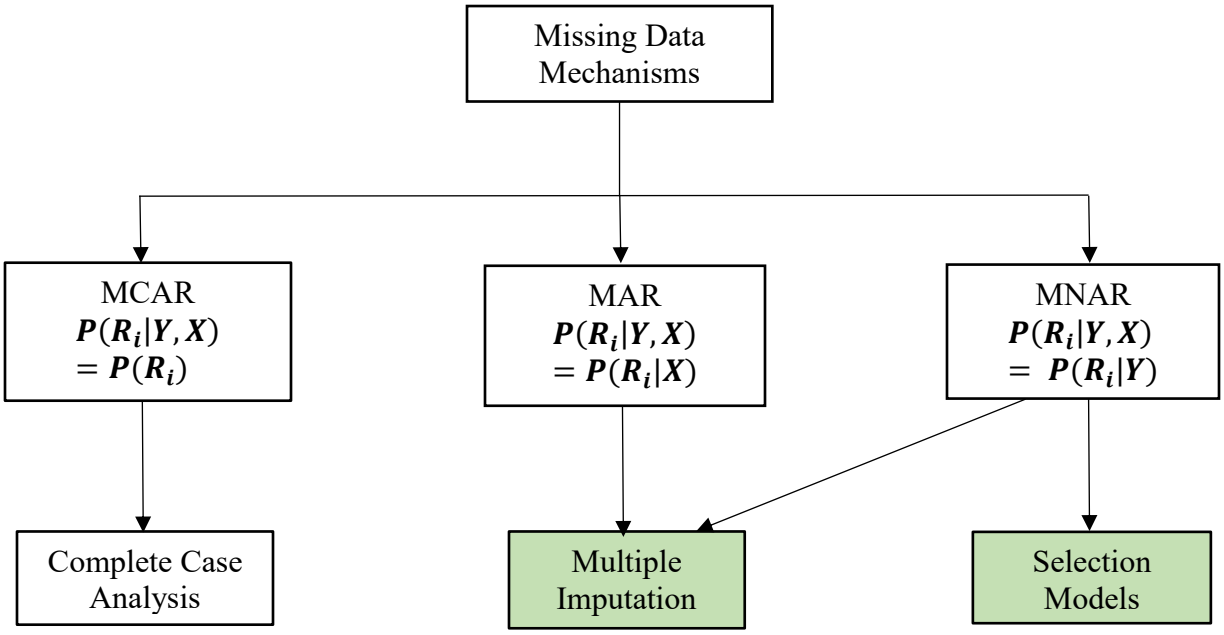

Figure 1.1 Methods for Handling Missing Data Under Each Missing Mechanism (only methods relevant to topic of this thesis are presented in this figure)

To determine how well methods used to handle MNAR data work, it is important to understand the thoughts and rationale behind handling MNAR data in cross-sectional datasets. Modeling a cross-sectional dataset with MNAR data involves taking a joint distribution of the observed outcome values ($\boldsymbol{Y}$) and the missing pattern ($\boldsymbol{R}$). Methods used to handle MNAR data depends on how this joint distribution is partitioned. A simple partition method known as selection models, involves partitioning the joint model into the hypothetical complete data outcome model and the response model which is dependent on the values of the outcomes (Kenward & Molenberghs, 2007):

$$f(\boldsymbol{Y}, \boldsymbol{R}|\, \boldsymbol{X}; \boldsymbol{\eta}) = f(\boldsymbol{Y} \,|\, \boldsymbol{X}; \boldsymbol{\theta_S}) f(\boldsymbol{R}|\boldsymbol{Y}, \boldsymbol{X}; \boldsymbol{\phi_S}) \quad (1.1)$$

The idea was first introduced in 1979 by James Heckman as a method to correct for selection bias (which MNAR data can introduce to the dataset) in economics studies (Heckman, 1979). The

$\boldsymbol{\theta_S}$ is the parameter vector of the outcome model and $\boldsymbol{\phi_S}$ is the parameter vector of the response model. **Figure 1.1** shows where selection models method fit among the methods for handling missing data, alongside complete case analysis for MCAR data and multiple imputation for MAR data.

The concept of multiple imputation (MI) was first introduced in 1978 by Donald Rubin (Rubin, 1978). Although MI is a method first designed for handling MAR data, the method itself is not limited to the mechanism. MI's convenience and ability to introduce uncertainty to the missing data motivates researchers to bring its benefits to the handling of MNAR data (Ogundimu & Collins, 2017). The handling of MNAR data has been an area of ongoing research where major discoveries occur once every few years (Heckman, 1979; Greenlees, Reece, & Ziexhang, 1982; Qin, Leung, & Shao, 2002; Durrant & Skinner, 2006; Kim, 2011; Riddles, Kim, & Im, 2016; Galimard, Chevret, Protopopescu, & Resche-Rigon, 2016; Im & Kim, 2017; Tseng & Chen, 2017). In recent years, there have been significant breakthroughs in extending the MI method to MNAR data (Durrant & Skinner, 2006; Kim, 2011; Galimard et al., 2016; Im & Kim, 2017). This thesis focuses on comparing the effectiveness of two recent methods developed to extend MI to MNAR cross-sectional data, one by Galimard et. al (2016) and the other by Ogundimu and Collins (2017). The comparison was done through simulation and applying them on two datasets the two methods have yet been applied to (one with a monetary outcome and the other with a health outcome). The objectives of the study were:

Objective 1. To compare the effectiveness of the two recently developed MI methods (Galimard et al and Ogundimu & Collins' MI methods) along side the complete case and Rubin's MI methods for handling missing not at random data via application to two real life datasets.

Objective 2. To investigate which statistical method mentioned in objective 1 is optimal for analyzing MNAR data through applying the methods on simulated datasets.

The main purpose of these objectives is to answer the following questions:

1) Which method (Galimard et. al and Ogundimu and Collins) is the better method for handling MNAR data?
2) Should Galimard et. al and Ogundimu and Collins' methods be used to the same extent as Rubin's MI in research and industry for MNAR data?

A more elaborate explanation of the history and progress of handling MNAR data through imputation is presented in Chapter 2. Chapter 3 gives a detailed description of the two real-life datasets where Galimard et. al's MI and Ogundimu & Collins' MI as well as CC and Rubin's MI were applied to. Chapter 4 explains how Galimard et. al and Ogundimu & Collins' MIs work and how they were applied to the real-life datasets and simulated datasets generated based on the real-life datasets. Chapter 5 presents the results of the application and simulation study explained in chapter 4. Chapter 6 discusses the results of chapter 5, how they fit into the big picture of handling missing data, and how future studies can help answer the additional questions that are uncovered by the results of this study.

CHAPTER 2

LITERATURE REVIEW

There are many aspects of missing data that makes it a difficult topic in statistics and the root of all of them is that it is impossible (or highly inconvenient) to obtain the actual values of the missing data. Therefore, the "truth" represented by the data must be deduced by other means. The difficulty of handling missing data depends largely on how much is missing and why the data were missing. Throughout the history of developing new methods for handling missing data, most methods developed have served roughly their intended purpose for the problem at the time. However, the problem of missing data is not a single dragon that can be slayed with one perfect method. The problem of missing data resembles a hydra where once you propose a method to solve one aspect of the problem it simultaneously adds new imperfections to the results, as well as bringing to the surface problems caused by missing data that were not previously known. There are three types of missing mechanisms: missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR) (Little & Rubin, 2002). The first two missing mechanisms (MCAR and MAR) are known to be ignorable mechanisms because the missingness of the data do not require explicit modeling to be consistently handled. The missing data mechanism which bares the most resemblance to a hydra would be the missing not at random (MNAR) mechanism because the missingness of data is due to the values of the variable containing the missing data. In this case, MNAR data is also called non-ignorable mechanism because the missingness of the data must be explicitly modeled inorder to handle it. Although missing data (ignorable and non-ignorable) can be handled using direct likelihood methods (Kenward & Molenberghs, 2007), in recent years, multiple imputation (MI) has been gaining momentum to becoming one of the most sought-after methods for handling missing data. This is mostly due to its easy implementation, its ability retains useable incomplete data and maintain the variability of the missing data compared to other methods. This review chapter gives a thorough account of the history of the imputation methods developed to handle MNAR data in cross-sectional datasets, as well as the direct likelihood methods which were used to develop their individual imputation

models. In addition, this review will discuss the "hydra-like" aspects of each imputation methods such as "what new imperfections were added to the problem by the methods?" and "what new problems were brought to the surface by the methods?"

2.1 Beginnings - Deletion

Missing data is a significant problem at the data analytics level in population health and socio-economic research. Improper handling of missing data (due to their missing patterns and reasons) can produce biased model parameter estimates, which will in turn lead to wrongful conclusions (Carpenter & Kenward, 2012). Until the early 1970s, missing values were handled mainly by deletion/elimination methods whereby the participants with missing data were excluded from the study. Today, case deletion, also known as listwise deletion and complete-case analysis, is used as the default in many commercially available statistical softwares (Schafer & Graham 2002). The procedure, however, is valid only for MCAR data, where neither the outcome variable (the variable with missing data for the scope of this thesis) nor the independent variable(s) has influence on whether the data were observed or missing (Schafer & Graham 2002). If a missing data problem can be resolved by discarding only a small portion of the sample, then the case deletion approach is quite effective and easy to implement (Schafer & Graham 2002). Detailed discussion on the properties of case deletion can be found in Little and Rubin (1987). Although there is no concrete evidence of how much missing data is enough to cause bias in a dataset, it has been shown that the larger the percentage of missing data, coupled with non-MCAR missing data, the more severe the bias will be if the missing individuals were deleted (Demissie, LaValley, Horton, Glynn, & Cupples, 2003; Knol et al., 2010; Masconi, Matsha, Erasmus, & Kengne, 2015). Also, MCAR is a missing mechanism that rarely occurs in research, which means handling missing data through deleting individuals with missing data will cause the study to lose a significant amount of information.

2.2 Elementary Imputation

Single and multiple imputation methods were developed for reducing bias and avoiding information loss when handling datasets that are not MCAR by creating complete dataset(s)(Rubin, 1996; Zhang, 2016). Mean, median, and mode imputations (replacing the missing values with the mean, median, and mode of the observed variable values) are examples of

single imputation (Zhang, 2016). After replacing the missing values with one of these measures of central tendency, the dataset is then modeled like a complete dataset to determine the significant predictors. Although mean, median, and mode imputations are easily obtainable, they decrease the variance of the variable and ignore the relationship between the outcome and independent variables, which can also introduce bias to the dataset through narrowing the range of the possibilities for the missing values (Rubin, 1976). The mean, median, and mode values can also be highly inaccurate when used as placeholders for the missing values. An attempt to improve the accuracy of the missing value placeholder were two methods: Hot Deck imputation (where the missing values are replaced by the same value as another individual in the dataset who has the most similar qualities as the individual with missing data) and Cold Deck imputation (where the similar individuals selected come from another dataset). The Cold Deck imputation was developed to increase the variability of the outcome variable because the variability for the Hot Deck imputation (like in mean, median, and mode imputations) would still be lowered because the placeholder is taken from the same dataset (Andridge, 2011). In order to further increase the range of the imputed values, thereby maintaining the variability of the imputed variable, imputations generated through a regression model were developed. The regression model is built using the observed values of the outcome variable and independent variables that influences the outcome values. The values of the missing data are then generated using the observed values of the independent variables for the individual. The method is better than mean, median, mode, Hot Deck and Cold Deck imputations because it considers the correlation among the outcome and independent variables, as well as ensuring that the missing values are replaced with a large range of values instead of a few. But nevertheless, the method also underestimates the variability of missing values because: 1) the values generated are still too similar to the other values; and 2) every missing value is only imputed one time, which means the uncertainty of the missing values is not taken into consideration (Zhang, 2016). Therefore, multiple imputation, where multiple placeholders were generated for each missing datum, was developed to compensate for these shortfalls.

2.2.1 Additional Methods for MCAR, MAR, and MNAR Data

MCAR is a missing mechanism which rarely occurs in research. In order to differentiate it from MAR and MNAR, Little's test can be used. Another method to test for MCAR is by recoding the variable containing missing data into a dummy variable of whether or not the data is missing, then run t-tests and chi-square tests between this variable and other variables in the dataset to see whether the values of other variables influence the missing pattern. However, this method along with Little's test cannot precisely determine whether the missing mechanism is MCAR, therefore, it is uncommon for MCAR mechanism to be assumed from testing.

MCAR is the easiest missing mechanism to handle. One of the most straightforward method is the complete case analysis method where individuals with missing data are deleted from the dataset and the complete individuals are analyzed like a complete dataset. In addition, MCAR data can also be handled using single value imputation where the missing values are replaced with estimates of central tendencies (e.g. Mean, median, mode) and multiple value imputation such as Rubin's MI.

For determining the difference between MAR and MNAR, it is difficult because it requires specialized knowledge in the field of study to deduce which variable(s)' values were responsible for the missing data. The maximum likelihood method involves analyzing the full, MAR incomplete data by computing the likelihood separately for variables with complete data and incomplete data, then maximize the combined log-likelihood function. This method, however, is limited to studies involving only linear models. Another method which can be used to analyze MAR data is the fractional imputation method. Unlike Rubin's MI and its alternatives, which involves generating multiple imputed values for each missing value, fractional imputation involves a more complicated process. The method involves generating not only the imputed values but also

fractional weights for each imputed value. The fractional weights represent the conditional probability of the imputed value given the observed data, which are computed by the iterative method EM algorithm.

For MNAR data, handling them becomes tricky because the method involves joint modeling of the observed outcome variable values as well as the missing patterns of the outcome variable. Other than selection model, there are two additional methods for handling MNAR data using the likelihood method: 1) Pattern-mixture method and 2) Shared parameter method. The main difference between these methods and selection models is the way the joint distribution is split into product of two conditional distributions for the outcome variable distribution and missing patterns of the outcome variable. Selection models involve separating the joint distribution into the marginal distribution of the outcome variable and the distribution of the missing pattern conditional to the value of the outcome variable. Both distributions are unknown and must be specified by the user. The pattern-mixture model involves separating the joint distribution into the marginal distribution of the missing pattern and the distribution of the outcome variable conditional on the missing distribution for both the observed and non-observed data. For the observed data, the distributions can easily be obtained by modeling after the data. However, for non-observed data, the distributions can be obtained by subtracting their observed data counterparts from 1.

2.3 Rubin's Missing Data Mechanisms

To better account for complex nature of the missing data, in 1976, Donald Rubin (Rubin, 1976) developed the MCAR, MAR, and MNAR framework for describing how missing data patterns are related to the observed and missing values in the dataset (Schafer & Graham, 2002).

Figure 2.1 is a graphic representation of this framework. This framework remains in use today and is the foundation of all methods developed to handle missing data. MCAR and MNAR are already discussed in the introduction of Chapter 2. MAR is when the missing mechanism of the outcome variable is dependent on the values of the independent variables only.

2.3.1 The Idea of Multiple Imputation (for Cross-Sectional Data)

Under Rubin's framework, MI was developed to handle MAR data. The method is more attractive than any previous imputation methods because it can compensate for the uncertainty of the missing data through generating multiple imputes for each missing datum. MI involves 3 steps:1) Imputation, 2) Modeling, and 3) Combination (Rubin, 1987).

STEP 1. Imputation Step

For the imputation step, an imputation model (built from the data of fully observed individuals) is used to ensure the imputed values are drawn correctly. The type of imputation model built is dependent on the style of data to be imputed. For cross-sectional data, it is in the form of

$$\boldsymbol{Y_{obs}} = f(\boldsymbol{X};\ \boldsymbol{\beta})\ \boldsymbol{or}\ f(\boldsymbol{Y_{obs}}|\boldsymbol{X};\boldsymbol{\beta}) \tag{2.1}$$

where $\boldsymbol{Y}$ is a vector of n outcome values with $\boldsymbol{Y_{mis}}$ as the missing part of the variable to be imputed and $\boldsymbol{Y_{obs}}$ as the observed. $\boldsymbol{X}$ is a $n_{\boldsymbol{Y_{obs}}} \times p$ matrix containingthe $n_{\boldsymbol{Y_{obs}}}$ (number of observed $\boldsymbol{Y}$ values) for each of the p independent variables that influences the $\boldsymbol{Y}$ values. $\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_p)$ is the $(p+1) \times 1$ column vector of the regression coefficients of the $\boldsymbol{X}$ for the imputation model. If we have a model where there is one outcome and 3 covariates, the imputation model will look like this:

$$\boldsymbol{Y_{obs}} = \beta_0 + \beta_1\boldsymbol{X_1} + \beta_2\boldsymbol{X_2} + \beta_3\boldsymbol{X_3} + \epsilon_i \tag{2.2}$$

where $\epsilon_i \sim \boldsymbol{N}(0, \sigma_{\boldsymbol{Y_{obs}}})$ with the same $n_{\boldsymbol{Y_{obs}}} \times 1$ dimension as $\boldsymbol{Y}$. For this model, the $\boldsymbol{X}$ has the matrix of dimension $n \times 4$ and the $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, \beta_3)$. If the covariates are completely observed with only the outcome variable containing missing data. To generate imputed values for one dataset, we first draw the $\boldsymbol{\epsilon_i}$ values using the $\boldsymbol{N}(0, \sigma^2_{\boldsymbol{Y_{obs}}})$ distribution. With this value, along with the corresponding $\boldsymbol{X}$ values from the individual and the $\widehat{\boldsymbol{\beta}}$ obtained from the observed data, the

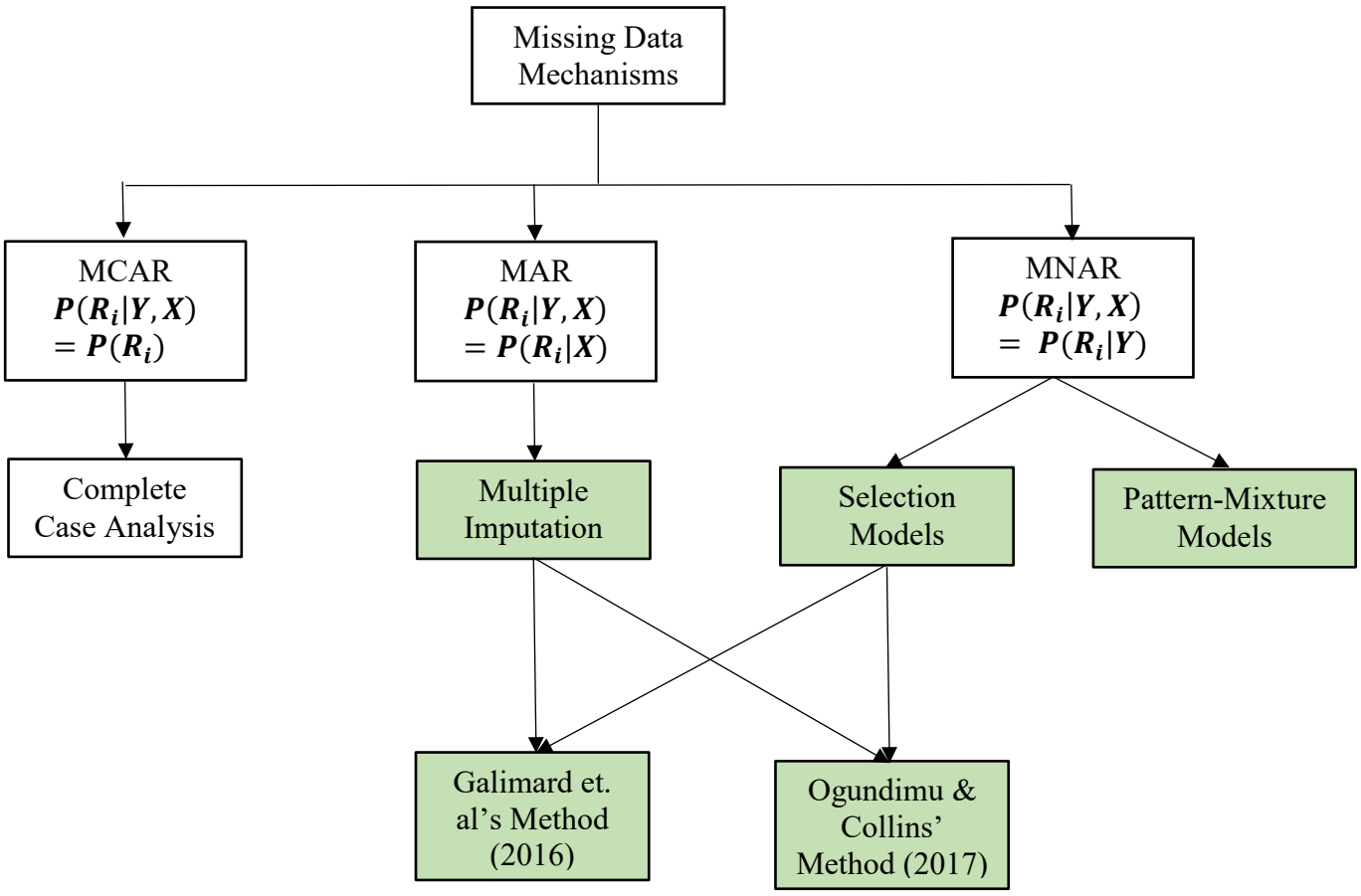

Figure 2.1 Methods for handling missing data under each missing mechanism (only methods relevant to topic of this thesis are presented in this figure)

imputed values $\boldsymbol{Y}^*$are generated. The imputed value sets $\boldsymbol{Y}^*$ are generated M times (which means M plausible versions of complete data are produced) and M (=1, …, m) can be a value 10 or above. According to Rubin (Rubin, 1987) the number M chosen can influence the efficiency [$=(1+r/m)^{-1}$] of an estimate, where r is the rate of missing information. For example, for 30% missing information, M=6 imputations is 100/(1+r/m) = 95% efficient (Schafer & Graham, 2002).

STEP 2. Modeling Step

The modeling step involves building M models in the form of model (2.2) using the M "complete" datasets.

STEP 3. Combination Step

The M models are combined into a final model to get the final estimate of $\boldsymbol{\beta}$ and its variance using Rubin's combination formula (Rubin, 1987). The MI estimate of the $\boldsymbol{\beta}$ is the average of the M estimates

$$\widehat{\boldsymbol{\beta}}^{*} = \frac{1}{\mathrm{M}} \sum_{m=1}^{M} \widehat{\boldsymbol{\beta}}^{\boldsymbol{m}} \tag{2.3}$$

where $\widehat{\boldsymbol{\beta}}^{\boldsymbol{m}}$ is the vector of the parameter estimates for the mth "complete" dataset.

The variance-covariance matrix of $\widehat{\boldsymbol{\beta}}^{*}$ has the dimension $p \times p$ derived by combining the within- and between-imputation variance-covariance matrices, both with the same dimensions as variance-covariance matrix of $\widehat{\boldsymbol{\beta}}^{*}$. The average within imputation variance-covariance matrix is defined as

$$\boldsymbol{W} = \frac{1}{M} \sum_{m=1}^{M} \boldsymbol{V}^{m} \tag{2.4}$$

where V^{m} is the variance-covariance matrix of the mth "complete" data. The between-imputation variance-covariance matrix of $\widehat{\boldsymbol{\beta}}^{*}$ is defined as

$$\boldsymbol{B} = \frac{1}{M-1} \sum_{m=1}^{M} (\widehat{\boldsymbol{\beta}}^{\boldsymbol{m}} - \widehat{\boldsymbol{\beta}}^{*})(\widehat{\boldsymbol{\beta}}^{\boldsymbol{m}} - \widehat{\boldsymbol{\beta}}^{*})' \tag{2.5}$$

The final variance-covariance matrix estimate of $\widehat{\boldsymbol{\beta}}^{*}$ is given by

$$\boldsymbol{V} = \boldsymbol{W} + \left(\frac{M+1}{M}\right)\boldsymbol{B} \tag{2.6}$$

If the independent and outcome variables both contain missing values, a MI alternative called multiple imputation with chained equations (MICE) can be used to handle missing data. The method was first invented by Stef van Buuren in 1999 (S. Van Buuren, Boshuizen, & Knook, 1999). The method involves creating imputation models for each variable with missing data using the rest of the variables involved in the imputation model. For example, if the independent variable $\boldsymbol{X_2}$ and $\boldsymbol{X_3}$ contained missing data, the missing values are first filled in by the means of the individual variables as placeholders, then the placeholders are replaced by imputed values that were generated using model (2.2) and the following models:

$$\boldsymbol{X_2} = \beta_0 + \beta_1 \boldsymbol{X_1} + \beta_Y \boldsymbol{Y} + \beta_3 \boldsymbol{X_3} + \epsilon_i \quad (2.7)$$

$$\boldsymbol{X_3} = \beta_0 + \beta_1 \boldsymbol{X_1} + \beta_Y \boldsymbol{Y} + \beta_2 \boldsymbol{X_2} + \epsilon_i \quad (2.8)$$

The placeholders are replaced for with generated imputes multiple times until the imputes converges. The final complete dataset is then modeled for the final result (Azur, Stuart, Frangakis, & Leaf, 2012).

Rubin's MI and MICE are great methods for MAR data (Azur et al., 2012; Bartlett, Seaman, White, & Carpenter, 2015). But they cannot be directly used for MNAR data because the methods rely on the idea that the missing outcome values were missing due to the values of the independent variables, which also means that the values can be accurately estimated using the independent variables through an imputation model. MI and MICE were designed with the idea of MAR data, but the methods can be used for any missing data mechanism if the imputation models can be properly specified.

2.4 Modeling MNAR Data (1977 – early 2000's)

The most important part of any type of MI is specifying the proper imputation model. From 1978 to the early 2000s, while MI was gaining momentum to becoming a popular method for handling MAR data, the development of more methods for unbiasedly modeling MNAR data was also progressing. Modeling MNAR data is a difficult task because the model must take both the observed and missing data (and how to properly model them) into consideration. Because for MNAR data, the missing pattern of the outcome variable is dependent on the values of the outcome variable itself, the values of the outcome variable and the missing pattern must be jointly modeled to obtain accurate results. The methods developed for modeling MNAR data relies on how this joint model is partitioned. Selection models is a method first introduced by James Heckman in 1979 and the idea of selection models is partitioning the joint model into 2 models (Heckman, 1979):

$$f(\boldsymbol{Y}, \boldsymbol{R} | \boldsymbol{X}; \boldsymbol{\eta}) = f(\boldsymbol{Y} \,|\, \boldsymbol{X}; \boldsymbol{\theta_S}) f(\boldsymbol{R} | \boldsymbol{Y}, \boldsymbol{X}; \boldsymbol{\phi_S}) \quad (2.9)$$

an outcome model (left) and a response model (right). $\boldsymbol{\theta_S}$ is the vector of the regression coefficients and the variance-covariance matrix for the outcome model. $\boldsymbol{\phi_S}$ is the vector of the regression

coefficients and the variance-covariance matrix for the response model. The outcome model is used to answer the research question, in the form of

$$\boldsymbol{Y_i} = \boldsymbol{X_i\beta} + \boldsymbol{\varepsilon_i} \tag{2.10}$$

where $\boldsymbol{Y_i}$ is the $n \times 1$ vector and is a continuous outcome, $\boldsymbol{X_{ij}}$ is the $n \times (p+1)$ design matrix of p covariates and a column of 1's for the intercepts for n individuals, $\boldsymbol{\beta}$ is the $(p+1) \times 1$ vector of the coefficients of covariates on the outcome, and $\boldsymbol{\varepsilon_i}$ is the $n \times 1$ error vector. When $\boldsymbol{Y_i}$ has MNAR data, the response model (also known as Heckman's model) is written as

$$\boldsymbol{P}(\boldsymbol{R_{y_i}} = 1 | \boldsymbol{X_i^s}) = \Phi(\boldsymbol{X_i^s\beta^s}) \tag{2.11}$$

where $\boldsymbol{X_i^s}$ is a $n \times q$ matrix containing $q(i = 1, \dots, q)$ covariates each with n $(j = 1, \dots, n)$ values that is potentially associated with the missingness of data. Φ represents the standard normal cumulative distribution function (CDF), and $\boldsymbol{\beta^s}$ is the $q \times 1$ vector of coefficients (effects) of covariates on the missingness of the outcome. Equation (2.11) can also be written as

$$\boldsymbol{R_{y_i}^*} = \boldsymbol{X_i^s\beta^s} + \boldsymbol{\varepsilon_i^s} \tag{2.12}$$

where $\boldsymbol{R_{y_i}^*}$ is the probability of $\boldsymbol{Y_i}$ being observed and the $\boldsymbol{\varepsilon_i^s} \sim N(0, 1)$.

The error terms of the 2 outcome and response models can be joined using bivariate normal distribution

$$\begin{pmatrix} \varepsilon^s \\ \varepsilon \end{pmatrix} \sim \mathrm{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_{\varepsilon^s}^2 & \rho\sigma_{\varepsilon^s}\sigma_\varepsilon \\ \rho\sigma_{\varepsilon^s}\sigma_\varepsilon & \sigma_\varepsilon^2 \end{pmatrix}\right) \tag{2.13}$$

which implies that the log-likelihood model for the final model for the MNAR dataset

$$\begin{aligned} l(\boldsymbol{\eta}) = \sum_{i=1}^{n} \boldsymbol{R_{y_i}}\left(\boldsymbol{ln} f\left(\boldsymbol{Y_i} | \boldsymbol{X_i}, \boldsymbol{R_{y_i}} = 1; \boldsymbol{\eta}\right)\right) + \sum_{i=1}^{n} \boldsymbol{R_{y_i}}\left(\boldsymbol{ln}\Phi(\boldsymbol{X_i^s\beta^s})\right) \\ + \sum_{i=1}^{n} (1 - \boldsymbol{R_{y_i}})\left(\boldsymbol{ln}\Phi(-\boldsymbol{X_i^s\beta^s})\right) \end{aligned} \tag{2.14}$$

where

$$f(\boldsymbol{Y_i}|\boldsymbol{X_i}, \boldsymbol{R_{y_i}} = 1;\ \boldsymbol{\eta})$$
$$= \frac{\frac{1}{\sigma}\phi\left(\frac{\boldsymbol{Y_i} - \boldsymbol{X_i\beta}}{\sigma}\right)\Phi\left(\frac{\boldsymbol{X_i^s\beta^s} + \ \rho\left(\frac{\boldsymbol{Y_i} - \boldsymbol{X_i\beta}}{\sigma}\right)}{\sqrt{1-\rho^2}}\right)}{\Phi(\boldsymbol{X_i^s\beta^s})} \tag{2.15}$$

This likelihood model function is very difficult to maximize, for this reason, in 1979 Heckman proposed another way to estimate the likelihood using the Heckman Two-Step method. For the individuals in the dataset who have observed outcome values, the conditional expectation of the outcome can be written as

$$\mathrm{E}[\boldsymbol{Y_i}\ |\boldsymbol{X_i}, \boldsymbol{R_{y_i}} > 0] = \boldsymbol{X_i}\beta + E[\boldsymbol{\varepsilon_i}|\boldsymbol{\varepsilon_i^s} > \ -\boldsymbol{X_i^s\beta^s}] \tag{2.16}$$

under the assumption that $\boldsymbol{\varepsilon_i}$ and $\boldsymbol{\varepsilon_i^s}$ have a bivariate normal distribution, the error term can be written as

$$E[\boldsymbol{\varepsilon_i}|\boldsymbol{\varepsilon_i^s} > \ -\boldsymbol{X_i^s\beta^s}] = \frac{\rho\sigma_{\varepsilon^s}\sigma_{\varepsilon}}{\sigma_{\varepsilon^s}}\frac{\phi(\boldsymbol{X_i^s\beta^s})}{\Phi\ (\boldsymbol{X_i^s\beta^s})} \tag{2.17}$$

where $\boldsymbol{\lambda_i}(\boldsymbol{X_i^s\beta^s}) = \frac{\phi(\boldsymbol{X_i^s\beta^s})}{\Phi\ (\boldsymbol{X_i^s\beta^s})}$ is known as the inverse Mills ratio. The top is the standard normal pdf (Probability Density Function) of $\boldsymbol{X_i^s\beta^s}$ and the bottom is the standard normal CDF of $\boldsymbol{X_i^s\beta^s}$ (Galimard et al., 2016). For the observed individuals,

$$\mathrm{E}[\boldsymbol{Y_i}\ |\boldsymbol{X_i}, \boldsymbol{X_i^s}, \boldsymbol{R_{y_i}} = 1] = \boldsymbol{X_i}\beta + \rho\sigma_{\varepsilon}\boldsymbol{\lambda_i} \tag{2.18}$$

The final model for the observed subjects is estimated by first estimating the inverse Mills ratio, then estimate the model

$$\boldsymbol{Y_i} = \boldsymbol{X_i}\widehat{\boldsymbol{\beta}} + \rho\frac{\sigma_{\varepsilon}}{\sigma_{\varepsilon^s}}\hat{\boldsymbol{\lambda}}_{\boldsymbol{i}}\left(\boldsymbol{X_i^s}\widehat{\boldsymbol{\beta}}^{\boldsymbol{s}}\right) + \ \boldsymbol{\eta_i}, \eta \sim N(0, \sigma_{\eta}^2\ ) \tag{2.19}$$

where ρ represents the correlation of the $\boldsymbol{R_{y_i}^*}$ and $\boldsymbol{Y_i}$ values (Heckman, 1979). When $\rho \neq 0$, it represents the MNAR mechanism in $\boldsymbol{Y_i}$.

Heckman's Two-Step method has been widely used since its conception, but the method has received some criticisms from the statistical community. In a 2000 article by Puhani, the criticisms were classified into 3 categories (Puhani, 2000):

1) It has been claimed that ordinary least squares (OLS) can perform just as well as Heckman's Two Steps (Duan et al., 1983; Duan et al., 1984; Duan et al., 1985). The reason for this is probably due to some inherent imperfections of Heckman's Two Step method elaborated in category 2.

2) The outcome and Heckman models are often identical or share a large portion of the same variables. When this happens in practice, model (2.19) can only identified when the inverse Mills ratio is nonlinear. However, the inverse Mills ratio takes on a linear form when $\boldsymbol{X_i^s}$ variables (when overlapping extensively with $\boldsymbol{X_i}$) can only produce $\boldsymbol{X_i^s\beta^s}/\sigma_{\varepsilon^s}$ higher than 2 when the probability $\Phi(\boldsymbol{X_i^s\beta^s}/\sigma_{\varepsilon^s})$ is higher than 97.5%. Since in most cases, the sample will not be this extreme, which means most examples will have inverse Mills' ratios for individuals that lie within a range which makes inverse Mills' ratios linear. This problem can be solved if the Heckman model contained variables that strongly affect the missingness of the outcome variable while not contain any variables in the outcome model (Little & Rubin, 2002). But it is almost impossible to find these unique variables in real life datasets.

3) The selection models method described above is also known as the parametric selection models because it makes strong assumptions for the distributions of the error terms of both outcome and Heckman models. To solve this issue, semi-parametric selection models and non-parametric selection models have been developed as alternatives to the parametric selection models method. Although these alternative methods have not been used as frequently as the parametric method by Heckman, there are evidence that shows the methods contain less evidence of selection bias in the analysis (Newey, Powell, & Walker, 1990; Lee, 1996). However, these methods will not be discussed in this review because thus far there has been no suggestions from literature for modifying semi-parametric and non-parametric selection models for imputation. This is probably due to:

a) MI, by theory, does not require a perfect imputation model to predict the values of missing outcome variables because its purpose is to use the many versions of possible values of the missing values to derive a final model. Therefore, an imperfect imputation model derived from a parametric selection model would still be appropriate.
b) The semi-parametric and non-parametric selection models are able to generate a model with similar results to the parametric selection models (Newey, Powell, & Walker, 1990; Lee, 1996). Although there is evidence that the semi- and non-parametric models might

contain less selection bias, the 3 methods have not been applied widely enough to show which one is superior.

To handle Heckman's Two-Step method's lack of robustness to deviation from the assumption of normality for the error terms, in 2012 Marchenko and Genton proposed assuming a Student's t-distribution for the error terms

$$\binom{\varepsilon^s}{\varepsilon} \sim t_2\left(\binom{0}{0}, \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}, \nu\right) \tag{2.20}$$

where t_2 is the probability density function of a bivariate t-distribution and ν is the degrees of freedom of the bivariate t distribution(Marchenko & Genton, 2012).

Another improvement of this method is that it uses a full maximum likelihood estimation of the actual likelihood model instead of approximating the final model like Heckman's Two-Step. The log likelihood is

$$\begin{aligned} l(\boldsymbol{\xi}) = \sum_{i=1}^{n} \boldsymbol{R}_{y_i}\left(\boldsymbol{ln} f\left(\boldsymbol{Y}_i \middle| \boldsymbol{X}_i, \boldsymbol{R}_{y_i}=1;\, \boldsymbol{\xi}\right)\right) + \sum_{i=1}^{n} \boldsymbol{R}_{y_i}\left(\boldsymbol{ln} T(\boldsymbol{X}_i^s \boldsymbol{\beta}^s; \nu)\right) \\ + \sum_{i=1}^{n} (1-\boldsymbol{R}_{y_i})\left(\boldsymbol{ln}\mathrm{T}(-\boldsymbol{X}_i^s \boldsymbol{\beta}^s;\, \nu)\right) \end{aligned} \tag{2.21}$$

where

$$f\left(\boldsymbol{Y}_i \middle| \boldsymbol{X}_i, \boldsymbol{R}_{y_i}=1;\, \boldsymbol{\xi}\right) = \frac{\frac{1}{\sigma} t\left(\frac{\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\beta}}{\sigma}; \nu\right) T\left\{\left(\frac{\boldsymbol{X}_i^s\boldsymbol{\beta}^s + \rho\left(\frac{\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\beta}}{\sigma}\right)}{\sqrt{1-\rho^2}}\right)\left(\frac{\nu+1}{\nu + \left(\frac{\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\beta}}{\sigma}\right)^2}\right)^{1/2}; \nu+1\right\}}{T(\boldsymbol{X}_i^s\boldsymbol{\beta}^s; \nu)} \tag{2.22}$$

with $\boldsymbol{\xi} = (\beta, \beta^s, \sigma, \rho, \nu)$ and t as the pdf of a univariate t-distribution (T as the CDF).

For the observed individuals,

$$\mathrm{E}[\boldsymbol{Y}_i \,|\boldsymbol{X}_i, \boldsymbol{X}_i^s, \boldsymbol{R}_{y_i}=1] = \boldsymbol{X}_i\beta + \rho\sigma\boldsymbol{\Lambda}_\nu(\boldsymbol{X}_i^s\boldsymbol{\beta}^s) \tag{2.23}$$

where $\boldsymbol{\Lambda_\nu}(k) = \frac{\nu+k^2}{\nu-1}\frac{t(k;\nu)}{T(k;\nu)}$ is the IMR. The k represents $\boldsymbol{X_i^s\beta^s}$ and the ρ, like in the normal distribution, is the correlation of the $\boldsymbol{Y}_i$ and the $\boldsymbol{R_{y_i}}$.

Marchenko and Genton's method is a relatively recent method and has not been applied widely enough in research to enable a detailed critique of its usefulness. But from their 2012 article, Marchenko and Genton's results showed their Heckman selection-t model performed better than Heckman selection normal model for heavier-tailed data when the selection model contained one more variable than the outcome model (Marchenko & Genton, 2012). The method also has more robustness against collinearity than the normal alternative. However, the method cannot accommodate various distribution as well as semi- and non-parametric selection models.

2.5 Recently Developed Imputation Methods for MNAR Data (2016 and 2017)

Galimard et. al. (2016) and Ogundimu & Collins (2017) came up with the idea of adapting Heckman's parametric normal selection models and Marchenko and Genton's parametric selection-t models to create two MI methods for MNAR cross-sectional data.

2.5.1 The Method of Galimard et. al. (2016)

In 2016, Galimard et. al published a MI approach using the idea of Heckman's model (Heckman, 1979). The model represents the expected value of $\boldsymbol{Y}$ when it is dependent on the covariates of the outcome and response models when the $\boldsymbol{Y_i}$ is observed. Galimard et. al proposed, based on equation (2.18), its counter part for the unobserved data, written as

$$\mathrm{E}[\boldsymbol{Y}_i \mid \boldsymbol{X_i}, \boldsymbol{X_i^s}, \boldsymbol{R_{y_i}} = 0] = \boldsymbol{X_i}\beta + \frac{-\phi(\boldsymbol{X_i^s\beta^s})}{1-\Phi\,(\boldsymbol{X_i^s\beta^s})}\rho\sigma_\varepsilon \qquad (2.24)$$

where the IMR $\boldsymbol{\lambda_i}$ is $\frac{-\phi(\boldsymbol{X_i^s\beta^s})}{1-\Phi\,(\boldsymbol{X_i^s\beta^s})}$ instead. This model can be modified into an imputation model:

$$\boldsymbol{Y_i} = \boldsymbol{X_i}\beta + \frac{-\phi(\boldsymbol{X_i^s\beta^s})}{1-\Phi\,(\boldsymbol{X_i^s\beta^s})}\boldsymbol{\beta_{\lambda_i}} + \boldsymbol{\eta_i}, \eta \sim N(0, \sigma_\eta^2) \qquad (2.25)$$

where $\boldsymbol{\beta_{\lambda_i}}$ is used to represent $\rho\sigma_\varepsilon$.The imputation process for Galimard's method is explained in **Chapter 4** (Methods chapter).

2.5.2 The Method of Ogundimu & Collins (2017)

Ogundimu and Collins' 2017 MI method is different from Galimard et. al's method in 3 main ways. The first is the bivariate t-distribution joining the distributions of the outcome and response models shown with distribution (2.20). Similar to Galimard et. al, expected value for the missing outcome for Ogundimu & Collins' method is

$$\mathrm{E}[\boldsymbol{Y}_i \mid \boldsymbol{X_i}, \boldsymbol{X_i^s}, \boldsymbol{R}_{\boldsymbol{y_i}} = 0] = \boldsymbol{X_i}\beta - \rho\sigma\boldsymbol{\Lambda}_\nu(\boldsymbol{X_i^s}\boldsymbol{\beta^s}) \tag{2.26}$$

where $\boldsymbol{\Lambda}_\nu(k) = \frac{\nu+k^2}{\nu-1}\frac{t(k;\nu)}{T(k;\nu)}$ is the IMR. The k represents $\boldsymbol{X_i^s}\boldsymbol{\beta^s}$ and the ρ is the correlation of the $\boldsymbol{Y}_i$ and the $\boldsymbol{R}_{\boldsymbol{y_i}}$. Therefore, the newly derived imputation model is

$$\boldsymbol{Y_i} = \boldsymbol{X_i}\boldsymbol{\beta} - \rho\sigma\boldsymbol{\Lambda}_\nu(\boldsymbol{X_i^s}\boldsymbol{\beta^s}) + \boldsymbol{\eta_i}, \eta \sim N(0, \sigma_\eta^2) \tag{2.27}$$

Second, the imputation process proposed by Ogundimu & Collins was different from Galimard et. al's method. Ogundimu & Collins used a MLE approach, which involves the missing value of the outcome be drawn from the posterior predictive distribution of the missing value:

$$Y_{i,mis}^{(k)} \sim p(Y_{i,mis} | Y_{i,obs}, \boldsymbol{X_i}, \boldsymbol{\Theta}) \tag{2.28}$$

where k = 1, …, M for the number of imputes generated, i = 1, …, n for the number of individuals in the dataset and $\boldsymbol{\Theta}$ is the parameters of the distribution. It is difficult to draw imputes from this distribution because the true value of $\boldsymbol{\Theta}$ is needed, but we cannot obtain it for MNAR data. Therefore, the posterior distribution (2.28) needs to be approximated. Ogundimu & Collins sampled the imputes from the approximated distribution

$$p\left(Y_{i,mis} \middle| Y_{i,obs}, \boldsymbol{X_i}, \boldsymbol{\Theta}\right) = \int p\left(Y_{i,mis} \middle| Y_{i,obs}, \boldsymbol{X_i}, \widehat{\boldsymbol{\Theta}}\right)\boldsymbol{\pi}\left(\widehat{\boldsymbol{\Theta}}\right)d\widehat{\boldsymbol{\Theta}} \tag{2.29}$$

with $\widehat{\boldsymbol{\Theta}}$ being the MLE of $\boldsymbol{\Theta}$ and $\boldsymbol{\pi}(\widehat{\boldsymbol{\Theta}})$ as its distribution. $\boldsymbol{\pi}(\widehat{\boldsymbol{\Theta}})$ needs to be approximated inorder to sample the possible values of $\widehat{\boldsymbol{\Theta}}$. We can draw possible parameter values $\widetilde{\boldsymbol{\Theta}}^{(\boldsymbol{k})} = (\widetilde{\boldsymbol{\beta}}^{(\boldsymbol{k})}, \widetilde{\boldsymbol{\sigma}}^{(\boldsymbol{k})}, \widetilde{\boldsymbol{X}}_{\boldsymbol{i}}^{\boldsymbol{s}(\boldsymbol{k})}, \widetilde{\boldsymbol{\beta}}^{\boldsymbol{s}(\boldsymbol{k})})$ from an asymptotic normal distribution of the $\widehat{\boldsymbol{\Theta}}$. Imputation process for Ogundimu & Collins' method is explained in section 4.2.2 of **Chapter 4**. After the imputation is done, unlike Galimard et. al's method (which required no combination step), Ogundimu & Collins' method follows the same idea of Rubin's MI for modeling and combining.

2.6 The Example Datasets of Galimard et. al and Ogundimu & Collins: Motivating, Simulation, and Application Studies

When the imputations were completed for Galimard et. al and Ogundimu & Collins' methods, the imputed datasets were modeled in the same way as their outcome models. The motivating example of Galimard's method was the Bivir (oseltamivir-zanamivir combination) study where the interest was in how did the initial body temperature, sick leave status, and tobacco-use status during study influenced the severity of flu symptoms (measured by a self-reported severity score) (Galimard et al., 2016)). Twenty-three percent (127 subjects) did not give a severity score at the time of study and 35% did not give a score for the initial body temperature. For simulation, Galimard et. al generated data for the three independent and identical normally distributed covariates of the selection model using normal distributions where

$$X_j \sim N(0, 0.3^2) \text{ where j = 1, …, 3} \tag{2.30}$$

X_1 and X_2 appeared as independents in both the outcome and response models while X_3 appears only in the selection model. The error terms of the outcome and selection models (ε and ε_s) were drawn from a bivariate normal distribution

$$\begin{pmatrix} \varepsilon^s \\ \varepsilon \end{pmatrix} \sim \mathrm{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right) \tag{2.31}$$

and the outcome (Y) values were generated through the model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon \tag{2.32}$$

where $(\beta_0, \beta_1, \beta_2)$ were fixed at (0, 1, 1). The missing data in the outcome variable were generated using the equations

$$\beta_0^s + \beta_1^s X_1 + \beta_2^s X_2 + \beta_3^s X_3 + \varepsilon_s < 0 \rightarrow missing$$

$$\beta_0^s + \beta_1^s X_1 + \beta_2^s X_2 + \beta_3^s X_3 + \varepsilon_s > 0 \rightarrow observed.$$

The ρ of (2.31) was set as 0, 0.3, and 0.6 for different degrees with MNAR in the outcome variable with 0 as missing at random; the higher the number, the higher the influence of the outcome variable itself on the missingness of data. The $(\beta_0^s, \beta_1^s, \beta_2^s, \beta_3^s)$ were fixed at (0.54, 1, -0.5, and 1) to ensure approximately 30% missing data in the outcome (similar yet slightly higher than the missing data in the Bivir data). In addition to these methods of generating MNAR outcome

data, Galimard et. al also generated MNAR outcome data in a "non-Heckman" fashion using a Bernoulli distribution with parameter $P(R_y = 1) = \Phi(0.75 + Y)$ for each observation to evaluate the performance of datasets generated from a different method. 1000 datasets of 2000 observations were generated for datasets with each degree of MNAR and the non-Heckman approach, with a total of 4000 simulated datasets. The results showed that complete case and Rubin's MI led to biased results for the data generated using Heckman approach with bias increasing with the degree of MNAR (ρ), while the Heckman 2-Step, and Heckman imputation methods gave unbiased results. The non-heckman generated data also showed higher bias for complete case and Rubin's MI methods while Heckman 2-step and Heckman imputation methods showed slight biases (with the Heckman imputation methods showing the lowest bias). The methods were also applied to the Bivir dataset. The Heckman imputation method yielded coefficient results closer to the Heckman 2-step while complete case and Rubin's MI had closer coefficient results to each other and farther from the Heckman methods.

For Ogundimu and Collins' simulation study, like Galimard et. al, they used a similar outcome (with 2 independent variables) and selection (3 independent models) models and sampled the error terms from the bivariate t-distribution with $\rho = 0.5$. The set up also ensured ~30% of each simulated dataset were missing. They generated 1000 simulated datasets. Ogundimu and Collins' compared the results of their imputation method based on bivariate t-distribution with other methods such as Galimard et. al's method (based on bivariate normal distribution) and found that the parameter estimations are more precise with their bivariate t-distribution imputation method. When the methods were applied to a real life dataset (on Ambulatory Expenditure) where ~15% of data were missing.

2.7 Strengths and Weaknesses of the Methods of Galimard et. al (2016) and Ogundimu & Collins (2017)

Galimard et. al and Ogundimu & Collins' methods reached a new stage of statistical innovation when they used parametric selection models methods to create multiple imputation models for imputing MNAR data. Their ideas successfully leveraged the strengths of parametric selection models while weakening the imperfections of Heckman and Marchenko and Genton's selection models because imputation methods do not require perfect models while modeling methods do. Although the idea of deriving imputation models for MNAR data using selection

models is a pragmatic choice, it remains unknown how well this method can be used to handle MNAR data in general. Galimard et. al and Ogundimu & Collins' methods are relatively new methods which have not been widely applied enough in research nor industry to determine their general usability.

The design of the two newly developed imputation methods also leaves rooms for enquiries of their general usability. Both methods, the simulated data were generated by first generate the independent variables of the outcome and selection models individually, generate the error terms of each model using bivariate normal and t-distributions, then use the generated variables and error terms to calculate the outcome values. Although this method ensures that the selection model will have a stronger influence in the missingness of the outcome variable and easiness of specifying the correct selection model, the method is different from real life dataset where the outcome variable is collected independently from the other variables. Based on their methods' design, Galimard et. al and Ogundimu & Collins' showed that their methods can work well if the selection models specified is correct in terms of predicting the missingness of outcome data, which is difficult to accomplish using real life datasets.

For applying their methods to real life datasets, Galimard et al and Ogundimu & Collins' used two very different datasets. Galimard et. al used a dataset with 541 individuals with 30% of the outcome variable missing. Ogundimu & Collins used a dataset with 3328 individuals with 15% of the outcome variable missing. In addition, the outcome variables of Galimard et. al and Ogundimu & Collins' are flu symptom scores (health variable) and ambulatory care spending (money variable). For MNAR data, it's possible that the missing pattern and missing percentage to differ based on the nature of the variable. This leave the question "Are the methods proposed by Galimard et. al and Ogundimu & Collins robust against these differences (that are quite subjective dataset-wise)?"

2.8 Thesis Contribution/Originality

The research presented in this thesis examines the generalizability of Galimard et. al (2016) and Ogundimu and Collins' (2017) MI to other health and monetary datasets (that are not the motivating examples of the two methods) and in more complicated circumstances than the ones presented in Galimard et. al and Ogundimu and Collins' papers.

1. In the simulation studies, Galimard et. al and Ogundimu & Collins' studies both simulated datasets were with ~30% missing data only. The research presented in this thesis involves a simulation study with 15%, 30%, and 50% missing data in order to examine how well Galimard et. al's MI and Ogundimu and Collins' MI methods along with Rubin's MI and CC analysis can perform under circumstances involving higher and lower percentages of missing data.
2. For Galimard et. al and Ogundimu & Collins' studies, the missing mechanisms involved were only MNAR and MAR. The research presented in this thesis involves the missing mechanism MCAR as well for the purpose of comparing the effectiveness of the 4 missing data methods across different missing mechanisms.
3. The simulated independent variables of the study in this thesis were simulated using different distributions whereas in Galimard et. al and Ogundimu and Collins' studies, the independent variables were all simulated with normal distributions. This was used to look at how well Galimard et. al's MI and Ogundimu and Collins' MI can work when the outcome model involved non-normally distributed covariates.
4. For the application part of the research presented in this thesis, the selection models used in the datasets (RANDHIE and SRHS) contained 3 more variables in the selection models than the outcome models for both datasets whereas in Galimard et. al and Ogundimu and Collins' studies, the selection models both contained only 1 more variable than the outcome models. The purpose of this set up is to look at whether Galimard et. al's MI and Ogundimu and Collins' MI can maintain their robustness in circumstances where more variables were expected to influence the missing pattern of the data.

CHAPTER 3

DATASETS

The two real-life data sets that are used for the application of Galimard et. al (2016) and Ogundimu & Collins (2017)' methods are described in this chapter. They are datasets similar to the motivating example datasets used for developing the 2 methods. These datasets are the Saskatchewan Rural Health Study (SRHS) and the RAND Health Insurance Experiment (RANDHIE) data. Although both datasets are longitudinal, only the baseline data were used for this thesis.

3.1 Saskatchewan Rural Health Study

The SRHS dataset was from a Canadian Institutes of Health Research-funded longitudinal study (2009 – 2015) collected in two phases in 2010 (baseline survey) and 2014 (follow-up survey) through questionnaires to evaluate the importance of individual and contextual factors on respiratory health of the farming and small-town rural communities in the Saskatchewan Rural Municipalities. The detailed methodology of SRHS has been described elsewhere (Pahwa et al., 2017). The baseline survey consisted of two separate components for adults and children (although this thesis only used adult data). Thirty-two (89%) out of 36 farming communities and 15 out of 16 small towns within rural municipalities selected agreed to participate. Completed questionnaires were obtained from 4264 households (8261 individuals). Lung function and allergy tests were performed on a sub-sample of the subjects who answered "Yes" in the last question on the baseline questionnaire: "We wish to find out more about respiratory health of rural people. Would you be willing to be contacted about having breathing and/or allergy tests at a nearby location?" Lung function measurements and allergy skin tests were obtained from 1607 and 1615 adults respectively. Both measurements were available for 1549 adults (Pahwa et al., 2012). The individuals whose data were used in the analysis were individuals with complete data for the variables listed in **Table 3.1**.

Table 3.1 Variables from SRHS Used for This Study

Variable in Dataset	Variable Name in Thesis	Variable Information	Variable Type
c_FEVOBSER	Lung Function	Forced expiratory volume for lungs	Continuous
i_AGE	Age	Age of individual at time of data collection	Continuous
i_BMI	BMI	BMI of individual	Continuous
i_SEX	Sex	Sex of individual	Binary
PACKYEARS	Packyears	Measurement of cigarette smoking calculated by multiplying the number of packs of cigarettes smoked per day by the years the person smoked	Continuous
ri_LIVESTOCK	Livestock	Exposure to livestock	Binary
ri_GRAINDUST	Grain_dust	Exposure to grain dust	Binary
h_HOMEPESTICIDE	Pesticide	Use of pesticide in the home	Binary
h_LOCATION	Home Location	The location (rural/urban) of home	Binary

3.2 The RAND Health Insurance Experiment Study

The RANDHIE dataset was collected from 1974 to 1982 by the Rand corporation for a comprehensive investigation of the effects of different health insurance plans on the health care cost, utilization, quality and outcomes of United States citizens. The study was conducted in the urban and rural areas of 6 sites across the US: Dayton (Ohio), Seattle (Washington), Fitchburg-Leominster and Franklin County (Massachusetts), Charleston and Georgetown County (South Carolina) (Newhouse, 2005). The study team (within the Rand corporation) established an insurance company using funding provided by the then-US Department of Health, Education, and Welfare which insured 5809 randomly selected individuals using insurance plans with various degrees of co-payment rates with a maximum annual payment of $1000 (Newhouse, 2005). The dataset had 45 variables including health insurance information, socio-economic information, and health information of the individuals. Four dimensions of health were assessed: physical, mental, social and physiological. The subset of this dataset (used in chapter 4) were adult (>18) individuals with complete data for the variables of interests (shown in **Table 3.2**) at baseline, which contained 4451 individuals (Newhouse, 2005).

Table 3.2 Variables from RANDHIE Used for This Study

Variable in Dataset	Variable Name in Thesis	Variable Information	Variable Type
lnmeddol	Medical expenses	log of medical expenses	Continuous
logc	Coinsurance rate	a transformation of the coinsurance rate taken as log(coinsurance rate +1)	Continuous
linc	Family income	log of family income	Continuous
lfam	Family size	log of family size	Continuous
xage	Age	Age of individual	Continuous
female	Sex	Sex of individual	Binary
child	Child	whether the individual has a minor child	Binary
disea	Disease	number of chronic diseases the individual has	Continuous
educdec	Education	the head of household's education level in years	Continuous
idp	Individual deductible	whether the person has an individual deductible	Binary

CHAPTER 4

METHODS

4.1 Introduction

Rubin's multiple imputation (MI) method and its variants have dominated the missing data handling area of statistics since its development in 1987 (Azur et al., 2012). The method had strong plus points compared to its predecessors in the sense of maintaining the accuracy and variability of the imputed data. However, there has been a consistent limitation of not being able to use this method to handle missing not at random (MNAR) data. The source of this limitation lies in the design of MI. Rubin's MI (Rubin, 1987; Little & Rubin, 2002) and its' variants all contain 3 main steps: 1) Imputation, 2) Modeling, and 3) Combination. For the imputation step, an imputation model is built using the complete data, and this model is used to draw imputed values for the missing data. Drawing values from this model can work well for missing at random (MAR) datasets due to the missing data in these scenarios are not dependent on the values of the missing data themselves (Azur et al., 2012; Lee & Carlin, 2010).

MNAR data, on the other hand, are more difficult to handle because the data are missing due to the values of the data itself (Kenward & Molenberghs, 2007). Hence, if an imputation model was built with the complete data, the imputed values drawn from this model has a high chance of not being close to the actual values, which can lead to selection bias (Heckman, 1979). Due to MI's inherent limitations in handling MNAR data, the handling of MNAR data took a different development path. The purpose of this chapter is to describe the theories of two recently developed MI methods (by Galimard et. al and Ogundimu & Collins) and how they were applied to two real-life datasets and simulated datasets along with two other methods for handling missing data (CC and Rubin's MI). The two methods were both developed following idea of Heckman's selection models where the outcome model of interest and the response model of missing data are jointly modeled (the idea is explained in detail in section 2.4). The theories of Galimard et. al and Ogundimu & Collins' MI methods are explained in sections 2.5 while their imputation processes are explained in section 4.2 of this chapter.

4.2 Two Recently Developed MI Methods for MNAR Data

Although Rubin's MI has been shown to lead to bias when used to handle MNAR data (Azur et al., 2012; Bartlett et al., 2015), the motivation to extend the MI method to MNAR data has not been extinguished. MI's good qualities (convenience, validity, and wide availability in statistical softwares) convinced many medical research projects to readily use it for missing data (Chen et. al, 2018; Enders, 2017; Mühlenbruch et al., 2017). Extending MI to MNAR data could mean having a more accurate method to handle missing data which has the same good qualities of MI on MAR data while at the same time avoid introducing selection bias to research results. In the past 2 years, two methods involving extending MI to MNAR data have been developed; one method by Galimard et. al in 2016, the other by Ogundimu and Collins in 2017. The two methods differ from Rubin's MI and from each other in terms of their imputation models and imputation process.

4.2.1 The Method of Galimard et. al. (2016)

For Galimard et. al's MI, the imputation model takes the form of equation (2.25) and the imputation process involves:

1) Obtain MLE of $\boldsymbol{\beta^s}$ using (2.14)
2) $\widehat{\boldsymbol{\beta}}^{\boldsymbol{s}}$ are used to construct IMR through $\hat{\boldsymbol{\lambda}}_{\boldsymbol{i}} = \frac{-\boldsymbol{\phi}(\boldsymbol{X_i^s}\widehat{\boldsymbol{\beta}}^{\boldsymbol{s}})}{\mathbf{1}-\boldsymbol{\Phi}\,(\boldsymbol{X_i^s}\widehat{\boldsymbol{\beta}}^{\boldsymbol{s}})}$ for each subject i
3) Draw $(\sigma_\eta^{2*}, \beta^*, \beta_\lambda^*)$ using imputation model presented as equation (2.25) with $\widehat{\boldsymbol{\beta}}^{\boldsymbol{s}}$ and $\hat{\boldsymbol{\lambda}}_{\boldsymbol{i}}$ substituted

   a) Draw a random variable g* from a Chi-square distribution with df equal to the df of imputation model

   b) $\sigma_\eta^{2*} = \frac{\widehat{\boldsymbol{\sigma}}_{\boldsymbol{\eta}}^{\mathbf{2}}}{\mathrm{g}*}$, but the real distribution of $\boldsymbol{\eta}$ is not homoscedastic because $Var(\boldsymbol{\eta_i}|\boldsymbol{X_i},\ \boldsymbol{R_{yi}} = \mathbf{1},\ \widehat{\boldsymbol{\beta}}_{\boldsymbol{i}}^{\boldsymbol{s}}) = \boldsymbol{\sigma}_{\boldsymbol{\varepsilon}}^{\mathbf{2}*}(\mathbf{1} - \boldsymbol{\rho}^{\mathbf{2}}\boldsymbol{\delta_i})$, so we calculate $\boldsymbol{\sigma}_{\boldsymbol{\varepsilon}}^{\mathbf{2}*} = mean\left(\frac{\sigma_\eta^{2*}}{\mathbf{1}-\boldsymbol{\rho}^{\mathbf{2}}\boldsymbol{\delta_i}}\right)$

   c) Draw $\boldsymbol{q}$ independent ***N(0, 1)*** variables in vector $\boldsymbol{z}^*$, where $\boldsymbol{q}$ is the dimension of $\boldsymbol{V_c}$

   d) Calculate $\boldsymbol{V_c^{1/2}}$

   e) Calculate $(\boldsymbol{\beta}^*, \boldsymbol{\beta}_{\boldsymbol{\lambda}}^*) = (\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\beta}}_{\boldsymbol{\lambda_i}}) + \mathrm{z}^*\boldsymbol{\sigma}_{\boldsymbol{\varepsilon}}^{\mathbf{2}*}\boldsymbol{V_c^{1/2}}$

4) Draw η^* from $N(0, \sigma_\eta^{2*})$

5) For each missing Y, impute Y^* using imputation model presented as equation (2.25)

4.2.2 The Method of Ogundimu & Collins (2017)

Ogundimu and Collins' 2017 MI method is different from Galimard et. al's method in two ways. The first is that it uses a bivariate t-distribution (also known as selection-t model) to join the outcome and response models (Marchenko and Genton, 2012). The second is that the imputation process proposed by Ogundimu and Collins used a maximum likelihood estimator (MLE) approach which involved the missing value of the outcome should be drawn from the posterior predictive distribution of the missing value (2.28). The imputation process involves:

1. Estimate the MLE of $\boldsymbol{\Theta}$, $\widehat{\boldsymbol{\Theta}}$, using likelihood function as given in equation (2.21)
2. Draw $\widetilde{\boldsymbol{\Theta}}^{(1)}$ from N($\widehat{\boldsymbol{\Theta}}, \boldsymbol{C}(\widehat{\boldsymbol{\Theta}})$) with $\boldsymbol{C}(\widehat{\boldsymbol{\Theta}})$ as the variance-covariance matrix of $\widehat{\boldsymbol{\Theta}}$ obtained from inversion of observed information matrix
3. Draw $Y_{i,mis}^{(1)}$ from $p(Y_{i,mis}|Y_{i,obs}, \boldsymbol{X_i}, \widetilde{\boldsymbol{\Theta}}^{(1)})$
4. Repeat steps 2 and 3 for k times.

4.3 Methods for Comparing Methods of Galimard et. al and Ogundimu & Collins

The main purpose of this thesis is to answer the following questions:

1. Which method (Galimard et. al and Ogundimu and Collins) is the better method for handling MNAR data?
2. Should Galimard et. al and Ogundimu and Collins' methods be used as widely in research and industry for MNAR data in the same extent as Rubin's MI?

To address the above questions, Galimard et. al and Ogundimu and Collins' methods were tested for their effectiveness through simulation and applying them to two datasets that were not the datasets which served as motivating examples in their individual studies. The two datasets were the Saskatchewan Rural Health Study (SRHS) dataset and the Rand Health Insurance Experiment (RANDHIE) dataset. The SRHS dataset was collected between 2010 – 2014 from a prospective cohort study of people age 6 years and over residing in farming and non-farming communities in Saskatchewan Canada to evaluate respiratory health determinants in rural areas (Pahwa et al.,

2012). The portion of the dataset used for this study was the adult (>18 years of age) individuals with complete data in the variables of interest at the baseline, which included 1607 people. The variables of interests were forced expiratory volume for lung function, Age, BMI, Sex, the packyear (calculated by multiplying the number of packs of cigarettes smoked per day by the years the person smoked), exposure to grain dust, the use of pesticide in the home, exposure to livestock, and the urban/ rural location of the home. Lung function was used as the outcome variable for SRHS. The selection model for the SRHS dataset used independent variables **Age**, **BMI**, **Sex**, **Packyears**, **Home Location**, **Livestock**, **Graindust**, and **Pesticide** while the outcome model excluded **Home Location**, **Graindust** and **Pesticide**.

The RANDHIE dataset was collected between 1974 and 1982 for conducting a comprehensive study of health care cost, utilization and outcome in the United States. The subset of individuals included in the study were adults (>18 years) with complete data for the variables of interest at baseline, which included 4452 people. The variables of interest are variables log of medical expenses (**lnmeddol**), transformation of the coinsurance rate log(coinsurance rate +1) (**logc**), log of family income (**linc**), log of family size (**lfam**) , age (**xage**), sex of the individual (**female**), whether the individual has a minor child (**child**), number of chronic diseases (**disea**), the head of household's education level in years (**educdec**), and whether the person has an individual deductible plan (**idp**). The outcome variable for this dataset was the **lnmeddol**; for selection model, the independent variables are the 9 variables of interest. For the outcome model, the independent variables are the first 6 variables of the 9 variables of interest.

For this study, more continuous and categorical variables were included in the selection and outcome models for both datasets than in the studies by Galimard et. al and Ogundimu & Collins'. The RANDHIE dataset (for selection and outcome models) include more independent models than the SRHS data; it is also larger than the SRHS data (4451 vs. 1607) for purpose of comparing how well the methods will work for datasets of different sizes. The outcome variable for SRHS dataset was a health outcome (lung function) while a medical expenses (money) outcome was used for RANDHIE. These outcomes are similar to the outcome Galimard et. al and Ogundimu & Collins used but are different enough to see how robust the two methods are on the two new datasets.

4.4 Data Preprocessing

Before applying the imputation methods, linear regressions of the outcome models for each dataset were performed using the complete datasets. The results of the two outcome models are shown in **Tables A1** and **A2** in the Appendix A. These results will serve as the standard (true) results. After the complete data results were obtained, 9 new datasets with missing data were created for both SRHS and RANDHIE datasets by introducing missing data (15%, 30%, and 50%) into the chosen outcome variables of each dataset (in the 3 missing mechanisms MCAR, MAR, and MNAR (R-code in **Appendix B.1**). The process is illustrated in **Figure 4.1.** The data preprocessing, modeling, simulation, etc. were performed using R 3.5.1 (Wickham, 2015).

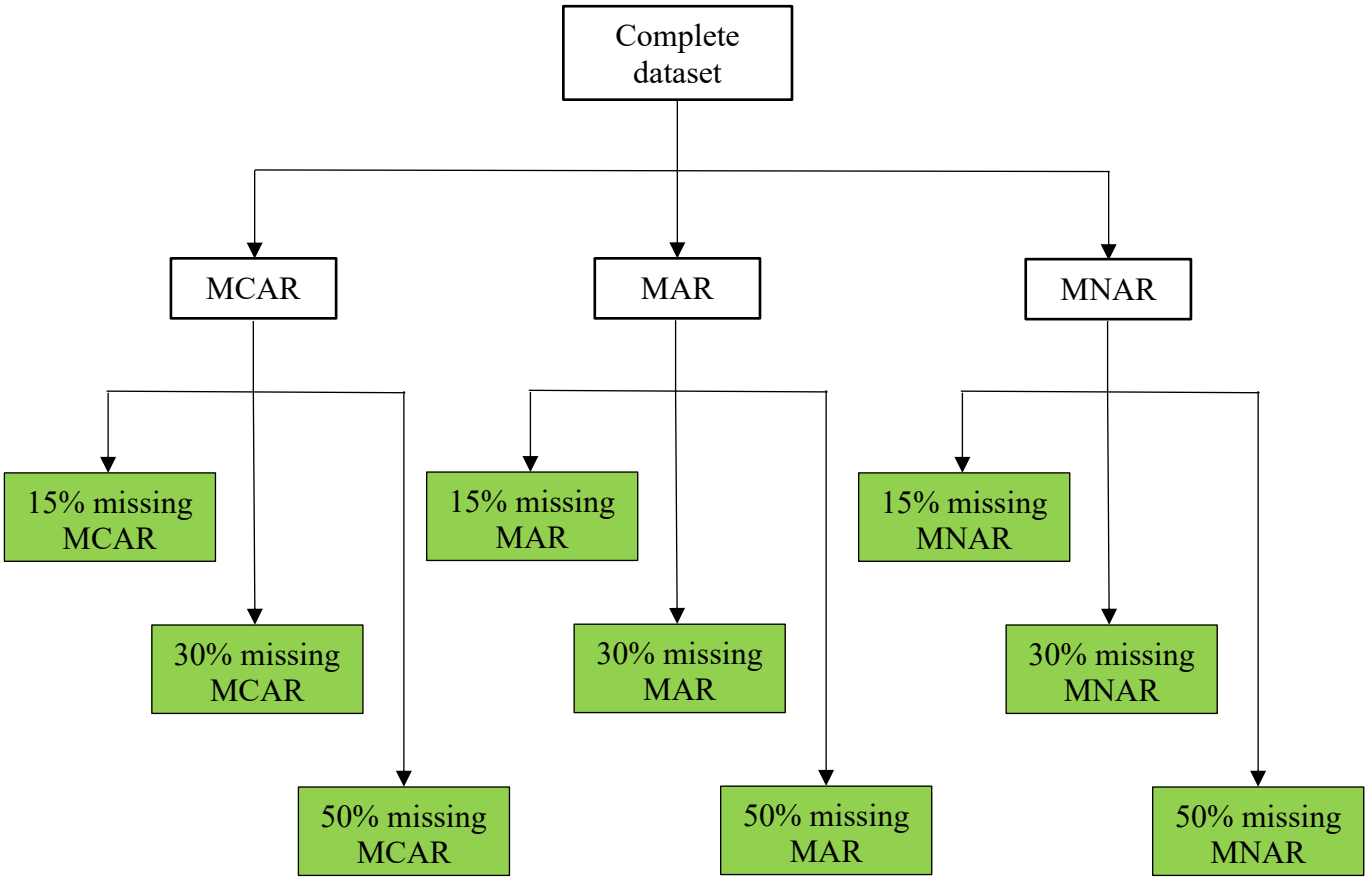

Figure 4.1 Process of generating datasets with missing data (15%, 30%, and 50%) for each mechanism (MCAR, MAR, and MNAR). For simulation, each combination in the green boxes are generated 1000 times.

The missing data introduced were accomplished by using a Bernoulli distribution where the probability of success (observed) was influenced by the outcome variable (MNAR), an independent variable (MAR) or no variable (MCAR). This was performed by specifying the probability of a datum being missing in the outcome using the rbinom R function (Wickham, 2015). For complete code, see **Appendix B**. The missing percentage, missing mechanism, and the probability specification for accomplishing the introduction of missing data into each dataset are illustrated in **Tables 4.1** and **4.2**. For the probability specification, the variables chosen were used to specify the missing mechanisms; their coefficients were chosen to ensure that the missing data introduced were approximately 15%, 30%, and 50%. These probability specifications are the values of the prob argument for the rbinom() function when it was used to introduce missing data into each dataset.

Table 4.1 Missing probability specification for each missing percentage and mechanism combination for RANDHIE dataset. The MCAR mechanism depends on no variables. The MAR mechanism depends on an independent variable which appeared in both selection and outcome models Age. The MNAR mechanism depends on the outcome variable Medical expenses.

Missing Percentage and Mechanism	Missing Probability Specification (prob =)
15% MCAR	1/(1+exp(1-2.75))
15% MAR	1/(1+exp(1-0.19*Age))
15% MNAR	1/(1+exp(1-0.75* Medical expenses)
30% MCAR	1/(1+exp(1-1.84))
30% MAR	1/(1+exp(1-0.088*Age))
30% MNAR	1/(1+exp(1-0.47* Medical expenses))
50% MCAR	1/(1+exp(1-0.99))
50% MAR	1/(1+exp(1-0.038*Age))
50% MNAR	1/(1+exp(1-0.25* Medical expenses))

For each of the of the 18 datasets with introduced missing data (9 for each RANDHIE and SRHS), 4 missing data methods were applied to each of them: complete case analysis (CC), Rubin's MI, Galimard et. al's MI, and Ogundimu and Collins' MI (R-code in **Appendix B.2**). How well the imputation methods worked on each missing mechanism and missing percentage combination for RANDHIE and SRHS were measured by the confidence interval lengths of the result model and the difference between the betas of the result models of the imputation methods

and the outcome model from directly modeling the complete dataset. The beta differences show how much do imputed datasets' betas are different from the "true" result of the dataset without the introduced missing data. The confidence interval lengths show the precision of the betas for the imputed datasets. The confidence interval lengths and beta differences are shown in Tables 5.1 – 5.16.

Table 4.2 Missing probability specification for each missing percentage and mechanism combination for SRHS dataset. The MCAR mechanism depends on no variables. The MAR mechanism depends on an independent variable which appeared in both selection and outcome models Age. The MNAR mechanism depends on the outcome variable Lung Function.

Missing percentage and Mechanism	Missing Probability Specification (prob =)
15% MCAR	1/(1+exp(1-2.73))
15% MAR	1/(1+exp(1-0.053* Age))
15% MNAR	1/(1+exp(1-0.93* Lung Function))
30% MCAR	1/(1+exp(1-1.9))
30% MAR	1/(1+exp(1-0.035* Age))
30% MNAR	1/(1+exp(1-0.6* Lung Function))
50% MCAR	1/(1+exp(1-0.99))
50% MAR	1/(1+exp(1-0.0185* Age))
50% MNAR	1/(1+exp(1-0.33* Lung Function))

The reason that 4 methods were applied to the dataset instead of just the 2 newer ones is to see how well the newer methods perform compared to the older methods. The methods were applied to all 3 missing data mechanisms to see how well they work for different missing data mechanisms. The methods were also applied to 3 different degrees of missing data (15%, 30%, and 50%) to see how well they work for different amount of missing data because it has been shown that imputation methods are subjected to limitations depending on how much data were missing (Sterne et al., 2009). Addressing these issues together will provide a comprehensive result of how well Galimard et. al and Ogundimu and Collins' methods can work if they are mass applied to population research in the future.

4.5 The Simulation Study

Other than real-life data, the 4 missing data methods were also applied to simulated datasets. For each missing mechanism and missing percentage combination, 1000 datasets were simulated. Simulating the datasets involved independently simulate 4 independent variables under different distributions (normal (Xnorm), uniform (Xunif), gamma (Xgamma), and binary categorical (Xcat)). To ensure a strong relationship of the outcome variable with the independents, the outcome variable (Y) was calculated using the values of the independents as well as an error term (generated using normal distribution) using the fixed coefficients (1, 1.5, 2, 1, -2.3). The missing data were introduced into each simulated dataset in a similar manner as real-life dataset (probability specifications shown in **Tables 4.3**). The performance of each method on each missing mechanism-missing percentage combination were assessed by computing 6 assessment statistics: the bias of regression coefficients, relative bias of regression coefficients, standardized bias of regression coefficient, mean square errors (MSE) of regression coefficients, average 95% confidence interval length among regression coefficients of simulated datasets, and percentage of successful coverage of the true regression coefficients (1, 1.5, 2, 1, -2.3) by the simulated 95% confidence intervals (Hossain & Pahwa, 2010). The R code for the simulation process is in **Appendix B.3**.

Table 4.3 Missing probability specification for each missing percentage and mechanism combination for Simulated datasets.

Missing Percentage and Mechanism	Missing Probability Specification (prob =)
15% MCAR	1/(1+exp(1-2.75))
15% MAR	1/(1+exp(1-0.95*XImpute))
15% MNAR	1/(1+exp(1-4.5*Y + 1.5*XImpute))
30% MCAR	1/(1+exp(1-1.84))
30% MAR	1/(1+exp(1-0.62*XImpute))
30% MNAR	1/(1+exp(1-3.25*Y + 2*XImpute))
50% MCAR	1/(1+exp(1-0.99))
50% MAR	1/(1+exp(1-0.3*XImpute))
50% MNAR	1/(1+exp(1-1.55*Y + 1.5*XImpute))

CHAPTER 5

RESULTS

This chapter presents the results from the application of the 4 missing data methods on the 18 datasets and 18000 simulated datasets described in chapter 4.

5.1 Application

For the application of 4 missing data handling methods (CC, Rubin's MI, Galimard et al's MI, Ogundimu & Collins' MI) to real life datasets, the results of each missing percentage-mechanism combination (for example; 15% MCAR, 15% MAR, and 15% MNAR, 30% MCAR, etc.) with each of the 4 missing data handling methods are compared with the results of the complete data to determine how well the missing data handling methods worked for each dataset with missing data. The missing percentage are chosen because 15%, 30%, and 50% are widely considered in the literature as low, medium, and high percentage of missing data (Kenward & Molenberghs, 2007). For the datasets with MCAR data, the CC method should produce best results; for MAR data, Rubin's MI should produce the best results; for MNAR data, Ogundimu & Collins' methods should produce the best results (over Galimard et. al's MI) because the method is better designed to accommodate for skewness of the outcome variables (which is slightly the case for both SRHS and RANDHIE datasets). For the different percentages of missing data, the datasets with the lowest percentage of missing data and the non-MNAR missing mechanisms are expected to perform better.

5.2 Comparison of Four methods for RANDHIE Data (Confidence Interval Lengths)

The confidence interval length results based on RANDHIE data for (i) CC analyses; (ii) Rubin's MI; (iii) Galimard et al's MI; and (iv) Ogundimu & Collins' MI are presented in **Tables 5.1 – 5.4.** The imputation and outcome models (for the 4 imputation methods) were fitted using

variables which had significant influence on the outcome variables and whether the values of the outcome variables were observed.

Interpretation of Results Based on Table 5.1 When CC analysis was used to analyze RANDHIE data, the confidence interval lengths of the coefficients followed a predictable pattern for the missing mechanisms. For all missing percentages, MCAR datasets had the lowest confidence interval length while MNAR data had the highest confidence interval length. This suggests that CC analysis delivers the most precise coefficients for MCAR data and the least precise for MNAR data, which is expected.

Interpretation of Results Based on Table 5.2 For Rubin's MI on RANDHIE data of all missing percentage-mechanism combinations, confidence interval lengths for the coefficients of the MNAR datasets were the longest while MCAR datasets had the most variables with the shortest confidence intervals for every missing percentage. The 15% MAR dataset had 2 variables with the shortest confidence interval lengths while the 50% MAR had only 1 and 30% had none. This shows that for RANDHIE dataset, Rubin's MI worked best on MCAR datasets for 15%, 30%, and 50% missing data compared to the other missing mechanisms.

Table 5.1 95% Confidence interval (CI) lengths of CC analyses on RANDHIE datasets with 15%, 30%, and 50% missing data

	15%			30%			50%		
	MCAR	MAR	MNAR	MCAR	MAR	MNAR	MCAR	MAR	MNAR
	CI Length	CI Length	CI Length	CI Length	CI Length	CI Length	CI Length	CI Length	CI Length
Intercept	0.812	0.88704	1.06149	0.83953	0.91324	1.17889	0.81201	0.90637	1.01392
logc	0.0425	0.04668	0.05489	0.04302	0.04718	0.05649	0.04245	0.04601	0.05471
linc	0.0856	0.09356	0.10983	0.08978	0.0974	0.12509	0.08561	0.09595	0.10516
lfam	0.1833	0.19948	0.24252	0.17849	0.19275	0.23284	0.18326	0.20018	0.23686
xage	0.0088	0.00959	0.01142	0.0085	0.00905	0.01101	0.00883	0.00932	0.01121
female	0.2261	0.24726	0.29614	0.21244	0.2251	0.27523	0.22614	0.24014	0.29007
child	0.3641	0.40093	0.46395	0.363	0.413	0.48485	0.36414	0.39497	0.47627

Green → lowest value among its own missing percentage
Yellow → medium value among its own missing percentage
Red → highest value among its own missing percentage

Table 5.2 95% Confidence interval (CI) lengths of Rubin's MI method on RANDHIE datasets with 15%, 30%, and 50% missing data

	15%			30%			50%		
	MCAR	MAR	MNAR	MCAR	MAR	MNAR	MCAR	MAR	MNAR
	CI Length	CI Length	CI Length	CI Length	CI Length	CI Length	CI Length	CI Length	CI Length
Intercept	0.92268	0.90901	1.25989	1.00354	1.13399	1.61347	0.89508	1.28657	1.24343
logc	0.05367	0.05269	0.06368	0.04905	0.05928	0.06147	0.04874	0.05495	0.06232
linc	0.09877	0.09846	0.12739	0.10551	0.11913	0.1694	0.0944	0.13367	0.14565
lfam	0.21611	0.24356	0.30296	0.21431	0.22452	0.2629	0.21161	0.22797	0.26284
xage	0.01088	0.01129	0.01701	0.01002	0.01008	0.01539	0.01034	0.01033	0.01539
female	0.27493	0.28744	0.33888	0.25021	0.27125	0.28489	0.25417	0.29263	0.33291
child	0.41679	0.48016	0.57352	0.41639	0.48001	0.54532	0.41161	0.48668	0.67247

Green → lowest value among its own missing percentage
Yellow → medium value among its own missing percentage
Red → highest value among its own missing percentage

Table 5.3 95% Confidence interval (CI) lengths of Galimard et. al's MI method for RANDHIE datasets with 15%, 30%, and 50% missing data

	15%			30%			50%		
	MCAR	MAR	MNAR	MCAR	MAR	MNAR	MCAR	MAR	MNAR
	CI Length	CI Length	CI Length	CI Length	CI Length	CI Length	CI Length	CI Length	CI Length
Intercept	0.95506	1.04325	1.2482	0.98744	1.07405	1.38626	0.95506	1.06597	1.1923
logc	0.04993	0.0549	0.06455	0.0506	0.05548	0.06642	0.04993	0.05412	0.06433
linc	0.1007	0.11003	0.12914	0.1056	0.11456	0.14709	0.1007	0.11284	0.12366
lfam	0.21555	0.23461	0.28518	0.20993	0.22669	0.2738	0.21555	0.23544	0.27853
xage	0.01038	0.01127	0.01343	0.01	0.01064	0.01294	0.01038	0.01096	0.01318
female	0.26598	0.2908	0.34823	0.24986	0.26474	0.32364	0.26598	0.28243	0.34111
child	0.42829	0.47153	0.54556	0.42695	0.48572	0.57013	0.42829	0.46453	0.56006

Green → lowest value among its own missing percentage
Yellow → medium value among its own missing percentage
Red → highest value among its own missing percentage

Table 5.4 95% Confidence interval (CI) lengths of Ogundimu & Collins' MI method for RANDHIE datasets with 15%, 30%, and 50% missing data

	15%			30%			50%		
	MCAR	MAR	MNAR	MCAR	MAR	MNAR	MCAR	MAR	MNAR
	CI Length	CI Length	CI Length	CI Length	CI Length	CI Length	CI Length	CI Length	CI Length
Intercept	1.03426	1.0102	1.04744	1.34189	1.2563	1.40086	1.40115	2.00373	1.95529
logc	0.05577	0.05042	0.05086	0.06379	0.05863	0.06223	0.0785	0.07524	0.07492
linc	0.10786	0.10558	0.10641	0.13693	0.11782	0.14264	0.15181	0.15563	0.18505
lfam	0.24022	0.20712	0.2397	0.26155	0.24168	0.27486	0.35789	0.30764	0.39414
xage	0.01099	0.0106	0.0111	0.01309	0.01404	0.01341	0.01442	0.01926	0.01516
female	0.28263	0.24914	0.27995	0.33001	0.29277	0.34121	0.42381	0.38094	0.40462
child	0.45249	0.46831	0.45265	0.56688	0.51041	0.51916	0.67084	0.64726	0.62152

Green → lowest value among its own missing percentage
Yellow → medium value among its own missing percentage
Red → highest value among its own missing percentage

Interpretation of Results Based on Table 5.3 The confidence interval lengths of Galimard et. al's MI method for RANDHIE data had the same pattern as the CC analysis, where for all missing percentages, MCAR datasets had the coefficients with the shortest confidence intervals while MNAR datasets had the widest. This shows that Galimard et. al's method, like the CC analysis, delivered the most precise results for MCAR data and least precise results for MNAR data.

Interpretation of Results Based on Table 5.4 For Ogundimu and Collins' MI method on RANDHIE data, the method delivered the most variables with shortest confidence intervals for MAR datasets. For 15% and 30% MAR, the method worked well with 5 coefficients each having the narrowest confidence interval lengths respectively. But for MNAR data, Ogundimu & Collins' method did not deliver shorter confidence interval lengths for the variable coefficients in all missing percentages. For 15% and 50% MNAR, Ogundimu & Collins' method had fewer variables with the widest confidence intervals than the MCAR, but 30% MNAR, had the most variables (3) with longest confidence intervals. This result suggest Ogundimu & Collins' method delivered the most precise coefficients for MAR datasets for all missing percentages and can handle MNAR data better for 15% missing compared to higher missing percentages.

5.3 Comparison of Four Methods for SRHS Data (Confidence Interval Lengths)

In this section, the confidence interval length results based on SRHS data for (i) CC analyses; (ii) Rubin's MI; (iii) Galimard et al's MI; and (iv) Ogundimu & Collins' MI are presented in **Tables 5.5 - 5.8.**

Table 5.5 95% Confidence Interval (CI) lengths of CC analysis method for SRHS datasets with 15%, 30%, and 50% missing data

	15%			30%			50%		
	MCAR	MAR	MNAR	MCAR	MAR	MNAR	MCAR	MAR	MNAR
	CI Length	CI Length	CI Length	CI Length	CI Length	CI Length	CI Length	CI Length	CI Length
Intercept	0.37615	0.40778	0.50056	0.39138	0.43322	0.49591	0.38226	0.42661	0.48049
Age	0.00457	0.00495	0.00602	0.00482	0.00524	0.00604	0.00452	0.00489	0.00596
BMI	0.00973	0.01039	0.0132	0.00975	0.0107	0.01233	0.01018	0.01147	0.01204
Sex	0.11217	0.12198	0.14789	0.11432	0.12406	0.14122	0.11092	0.12358	0.14949
Packyears	0.00452	0.00488	0.00551	0.00434	0.00466	0.00557	0.0043	0.0051	0.00556
Livestock	0.11207	0.12193	0.14832	0.1149	0.12427	0.14185	0.11113	0.12426	0.1503

Green → lowest value among its own missing percentage
Yellow → medium value among its own missing percentage
Red → highest value among its own missing percentage

Table 5.6 95% Confidence Interval (CI) lengths of Rubin's MI method for SRHS datasets with 15%, 30%, and 50% missing data

	15%			30%			50%		
	MCAR	MAR	MNAR	MCAR	MAR	MNAR	MCAR	MAR	MNAR
	CI Length	CI Length	CI Length	CI Length	CI Length	CI Length	CI Length	CI Length	CI Length
Intercept	0.44658	0.44084	0.68468	0.45674	0.45217	0.64369	0.47052	0.44596	0.5402
Age	0.00526	0.00547	0.00681	0.00562	0.00573	0.00703	0.00553	0.00586	0.00761
BMI	0.01162	0.01138	0.01867	0.01147	0.01225	0.0171	0.01167	0.01181	0.01385
Sex	0.12599	0.13754	0.15811	0.1315	0.16369	0.17181	0.13336	0.13681	0.19
Packyears	0.00578	0.00601	0.00735	0.00498	0.00489	0.00689	0.00486	0.00563	0.00601
Livestock	0.13386	0.13232	0.20083	0.12866	0.15282	0.17619	0.1295	0.15313	0.17556

Green → lowest value among its own missing percentage
Yellow → medium value among its own missing percentage
Red → highest value among its own missing percentage

Table 5.7 Confidence Interval (CI) lengths of Galimard et. al's MI method for SRHS datasets with 15%, 30%, and 50% missing data

	15%			30%			50%		
	MCAR	MAR	MNAR	MCAR	MAR	MNAR	MCAR	MAR	MNAR
	CI Length	CI Length	CI Length	CI Length	CI Length	CI Length	CI Length	CI Length	CI Length
Intercept	0.44213	0.47923	0.58796	0.46004	0.50912	0.5825	0.44932	0.50134	0.56443
Age	0.00538	0.00582	0.00707	0.00567	0.00616	0.00709	0.00531	0.00575	0.007
BMI	0.01144	0.0122	0.0155	0.01146	0.01257	0.01449	0.01197	0.01348	0.01414
Sex	0.13184	0.14335	0.17372	0.13438	0.1458	0.16588	0.13038	0.14523	0.17561
Packyears	0.00531	0.00573	0.00647	0.0051	0.00547	0.00654	0.00506	0.006	0.00653
Livestock	0.13173	0.1433	0.17421	0.13506	0.14604	0.16662	0.13062	0.14602	0.17656

Green → lowest value among its own missing percentage
Yellow → medium value among its own missing percentage
Red → highest value among its own missing percentage

Table 5.8 Confidence Interval (CI) lengths of Ogundimu & Collins' MI method for SRHS datasets with 15%, 30%, and 50% missing data

	15%			30%			50%		
	MCAR	MAR	MNAR	MCAR	MAR	MNAR	MCAR	MAR	MNAR
	CI Length	CI Length	CI Length	CI Length	CI Length	CI Length	CI Length	CI Length	CI Length
Intercept	0.42985	0.49302	0.45967	0.58462	0.58462	0.4758	0.65652	1.02851	0.55026
Age	0.00509	0.00584	0.00579	0.00558	0.00558	0.00907	0.00573	0.00948	0.00594
BMI	0.01077	0.01168	0.01242	0.01295	0.01295	0.01574	0.01533	0.01774	0.01138
Sex	0.12939	0.13622	0.12809	0.1517	0.1517	0.17865	0.19625	0.1558	0.15458
Packyears	0.00515	0.00487	0.00556	0.00518	0.00518	0.00571	0.00567	0.00648	0.00712
Livestock	0.12648	0.12691	0.13067	0.12397	0.12397	0.19127	0.14447	0.15824	0.14867

Green → lowest value among its own missing percentage
Yellow → medium value among its own missing percentage
Red → highest value among its own missing percentage

Interpretation of Results Based on Table 5.5 When CC analysis was used on SRHS data, the confidence interval lengths of the coefficients followed the same pattern as RANDHIE data where MCAR data in all missing percentages had the shortest confidence interval lengths and MNAR data had the longest. This confirms that CC analysis delivers the most precise coefficients for MCAR data and the least precise for MNAR data.

Interpretation of Results Based on Table 5.6 For Rubin's MI on SRHS data, like in RANDHIE data, the confidence interval lengths of the coefficients for the MNAR datasets were the longest while MCAR datasets had the most variables with the shortest confidence intervals. This further confirms that Rubin's MI works best on MCAR datasets for 15%, 30%, and 50% missing data compared to the other missing mechanisms.

Interpretation of Results Based on Table 5.7 The confidence interval lengths of Galimard et. al's method on SRHS data had the same pattern as RANDHIE data; MCAR data had the shortest confidence interval lengths and longest for MNAR data in all missing data percentages considered. This further confirms that Galimard et. al's method will delivered the most precise results for MCAR data and least precise results for MNAR data.

Interpretation of Results Based on Table 5.8 For Ogundimu and Collins' MI method on SRHS data, the result was slightly different from RANDHIE. For 15% missing, Ogundimu & Collins' MI delivered the most coefficients with the shortest confidence intervals for MCAR data while it worked slightly better for MAR than the MNAR data (with 1 less variable with the longest confidence interval for MAR). For 30% missing, the method delivered the same variables with the shortest confidence intervals for MCAR and MAR data while the variables for MNAR data had longer confidence intervals. For 50% missing, MAR had the most variables with the longest confidence intervals while MCAR had the most variables with the shortest confidence intervals. The results suggest for all missing percentages, Ogundimu & Collins' method on SRHS data worked best on MCAR data but the method becomes more effective on MNAR data when the missing data percentage is high (50%).

5.4 Comparison of Four Methods for RANDHIE Data (Beta Differences)

In this section, the results based on differences in beta estimates for RANDHIE data are presented for (i) CC analyses; (ii) Rubin's MI; (iii) Galimard et al's MI; and (iv) Ogundimu & Collins' MI are presented in **Tables 5.9 - 5.12**.

Table 5.9 Beta difference between CC analysis method for RANDHIE datasets with missing data and the complete dataset

	15%			30%			50%		
	MCAR	MAR	MNAR	MCAR	MAR	MNAR	MCAR	MAR	MNAR
	β diff	β diff	β diff	β diff	β diff	β diff	β diff	β diff	β diff
Intercept	-0.0367	-0.1182	0.01452	0.00152	-0.02	0.13779	-0.0367	0.38958	0.37639
logc	0.00303	-0.0006	0.00994	-0.0044	-0.0101	0.00148	0.00303	0.00455	0.02757
linc	0.01644	0.03865	-0.0168	-0.0079	0.0001	-0.0086	0.01644	-0.0187	-0.0166
lfam	-0.0097	-0.0274	0.03388	0.01678	0.02619	0.02134	-0.0097	0.04654	0.0761
xage	-0.0022	-0.0032	0.00068	0.00152	0.0009	-0.0002	-0.0022	-0.0021	-0.0013
female	-0.0218	-0.0821	0.04526	-0.0012	-0.0252	-0.126	-0.0218	0.03588	-0.0733
child	-0.0965	-0.1551	0.08459	0.01193	-0.0349	-0.106	-0.0965	-0.0461	-0.2198

Green → lowest value among its own missing percentage
Yellow → medium value among its own missing percentage
Red → highest value among its own missing percentage

Table 5.10 Beta difference between Rubin's MI method for RANDHIE datasets with missing data and the complete dataset

	15%			30%			50%		
	MCAR	MAR	MNAR	MCAR	MAR	MNAR	MCAR	MAR	MNAR
	β diff	β diff	β diff	β diff	β diff	β diff	β diff	β diff	β diff
Intercept	-0.0331	-0.1275	-0.02	0.03252	-0.0419	0.15165	-0.0565	0.33058	0.32677
logc	0.00036	-0.0022	0.00823	-0.0054	-0.0076	-0.0026	0.00365	0.00541	0.02933
linc	0.01818	0.03863	-0.014	-0.0093	0.00582	-0.0111	0.01739	-0.0178	-0.0154
lfam	-0.0203	-0.0398	0.0316	0.01722	0.00969	0.03667	-0.0167	0.05186	0.08573
xage	-0.0022	-0.0024	0.00079	0.00107	0.00025	8.56E-06	-0.0019	-0.0016	-0.0018
female	-0.0139	-0.0708	0.04662	0.00012	-0.0285	-0.1514	-0.0133	0.0502	-0.0373
child	-0.0913	-0.1149	0.11042	-0.0003	-0.0319	-0.0878	-0.0831	-0.0264	-0.1998

Green → lowest value among its own missing percentage
Yellow → medium value among its own missing percentage
Red → highest value among its own missing percentage

Table 5.11 Beta difference between Galimard et. al's MI method of RANDHIE datasets with missing data and the complete dataset

	15%			30%			50%		
	MCAR	MAR	MNAR	MCAR	MAR	MNAR	MCAR	MAR	MNAR
	β diff	β diff	β diff	β diff	β diff	β diff	β diff	β diff	β diff
Intercept	-0.0367	-0.1182	0.01452	0.00152	-0.02	0.13779	-0.0367	0.38958	0.37639
logc	0.00303	-0.0006	0.00994	-0.0044	-0.0101	0.00148	0.00303	0.00455	0.02757
linc	0.01644	0.03865	-0.0168	-0.0079	0.0001	-0.0086	0.01644	-0.0187	-0.0166
lfam	-0.0097	-0.0274	0.03388	0.01678	0.02619	0.02134	-0.0097	0.04654	0.0761
xage	-0.0022	-0.0032	0.00068	0.00152	0.0009	-0.0002	-0.0022	-0.0021	-0.0013
female	-0.0218	-0.0821	0.04526	-0.0012	-0.0252	-0.126	-0.0218	0.03588	-0.0733
child	-0.0965	-0.1551	0.08459	0.01193	-0.0349	-0.106	-0.0965	-0.0461	-0.2198

Green → lowest value among its own missing percentage
Yellow → medium value among its own missing percentage
Red → highest value among its own missing percentage

Table 5.12 Beta difference between Ogundimu & Collins' MI method of RANDHIE datasets with missing data and the complete dataset

	15%			30%			50%		
	MCAR	**MAR**	**MNAR**	**MCAR**	**MAR**	**MNAR**	**MCAR**	**MAR**	**MNAR**
	β diff	**β diff**	**β diff**	**β diff**	**β diff**	**β diff**	**β diff**	**β diff**	**β diff**
Intercept	-0.302	-0.2537	-0.0886	-0.5646	-0.745	-0.2253	-1.166	-1.2862	-0.9738
logc	0.00045	-0.012	-0.0126	-0.0011	-0.0144	-0.0151	-0.0058	-0.0067	-0.0082
linc	0.02112	0.00749	-0.006	0.03791	0.01364	-0.0123	0.0106	0.02361	-0.0034
lfam	-0.0252	0.02149	0.01039	-0.0645	0.02158	0.00526	0.04152	0.04566	0.05388
xage	-0.0034	0.00431	0.00138	-0.0037	0.01207	0.00052	0.00238	0.0128	0.00384
female	-0.0017	-0.0082	0.01353	-0.0667	-0.059	0.06897	0.0539	-0.1671	0.06869
child	-0.0907	-0.1763	-0.0438	-0.1938	-0.2952	-0.0802	0.22471	-0.0758	-0.0953

Green → lowest value among its own missing percentage
Yellow → medium value among its own missing percentage
Red → highest value among its own missing percentage

Interpretation of Results Based on Table 5.9 (Beta difference between CC analysis method for RANDHIE datasets with missing data and the complete dataset) For the beta difference of CC analysis and the complete dataset in the RANDHIE dataset, MCAR datasets have the most variables with the least amount of difference from the complete data for all 3 missing percentages. When the missing data percentage was low (15%), the MAR data had the most coefficients with highest difference from the complete data. But as the missing data increased (30% and 50%), MNAR data had the more coefficients that are most different from the complete data. The results make sense in the way that CC analysis performed best on MCAR data, with MAR coming in second while MNAR had the worst performance for each missing data percentage; it is also valid that the method's performance on MAR and MNAR data changes as the percentage of missing data increases because MNAR (as a non-ignorable missing mechanism) is more sensitive to biases in CC analysis.

Interpretations of Results Based on Table 5.10 (Beta difference between Rubin's MI method for RANDHIE datasets with missing data and the complete dataset) For Rubin's MI on RANDHIE, when missing data was 15%, MCAR data had the most variables (5) with the smallest beta differences while MAR had the most variables (6) with the largest beta differences. For 30% missing data, MAR and MCAR data had the most variables (2) with the smallest beta differences while MNAR had the most variables (4) with the largest beta differences. For 50% missing data, Rubin's MI performed best for MCAR data with 3 and 1 coefficients being the least and most different from complete data respectively. For MAR with 50% missing, 2 and 3 coefficients were the least and most different from complete data respectively. For MNAR with 50% missing, Rubin's MI performed the worst with 2 and 1 coefficients being the most and least different from the complete data. This result for 15% missing data was unexpected because Rubin's MI is expected to deliver results that are least different from the complete data for the MAR data and most different from complete data for MNAR data. The results for MI on 30% and 50% missing data show that Rubin's MI deliver results closer to the actual results for MCAR and MAR data than MNAR data.

Interpretation of Results Based on Table 5.11 (Beta difference between Galimard et. al's MI method for RANDHIE datasets with missing data and the complete dataset) The beta difference of the variables when Galimard et. al's method was applied to RANDHIE is identical to the CC analysis results. This suggests that like CC analysis, Galimard et. al's method will give

results least different from the actual results for MCAR data, MAR data second, and MNAR with results most different from the actual results.

Interpretation of Results Based on Table 5.12 (Beta difference between Ogundimu & Collins' MI method for RANDHIE datasets with missing data and the complete dataset) For Ogundimu & Collins' MI in 15% and 30% missing percentages, the MNAR datasets had the most variables with coefficients least different from the complete dataset. For 15% missing, the MNAR had 4 coefficients that were least different from the complete data while for 30% missing, the MNAR had 4. For 50% missing, Ogundimu & Collins' MI method delivered the most variables with coefficients that were most similar to the complete data in MCAR data while MNAR data had the most variables that were most different from the complete data. This suggests that Ogundimu & Collins' method works well on MNAR data when the missing data percentage is 30% or lower while for anything above that it works better on MCAR and MAR data.

5.5 Comparison of Four Methods for SRHS Data (Beta Differences)

In this section, the beta difference results based on RANDHIE data for (i) CC analyses; (ii) Rubin's MI; (iii) Galimard et al's MI; and (iv) Ogundimu & Collins' MI are presented in **Tables 5.13 - 5.16**.

Table 5.13 Estimates of Bias From Actual Estimates for CC Analysis Method on SRHS Datasets

	15%			30%			50%		
	MCAR	MAR	MNAR	MCAR	MAR	MNAR	MCAR	MAR	MNAR
	β diff	β diff	β diff	β diff	β diff	β diff	β diff	β diff	β diff
Intercept	-0.0347	-0.1023	0.018548	-0.0151	0.046675	0.07896	-0.0016	0.07151	0.00873
Age	0.00042	0.00082	0.000338	0.00015	-6.93E-05	-0.0015	0.00157	0.00054	-0.0016
BMI	0.00062	0.0026	-0.00112	1.14E-05	-0.00137	0.00024	-0.0013	-0.001	0.00311
Sex	-0.0073	-0.0159	-0.01031	0.00936	0.010474	0.0124	-0.0246	-0.04	0.0217
Packyears	0.00016	-0.0002	-5.76E-05	-0.0001	-0.0002	-0.0013	0.00026	0.00088	-0.0006
Livestock	-0.0013	0.00739	0.00056	-0.0039	-0.01693	-0.0071	-0.0108	-0.0257	0.03369

Green → lowest value among its own missing percentage
Yellow → medium value among its own missing percentage
Red → highest value among its own missing percentage

Table 5.14 Estimates of Bias From Actual Estimates for Rubin's MI Method on SRHS Datasets

	15%			30%			50%		
	MCAR	**MAR**	**MNAR**	**MCAR**	**MAR**	**MNAR**	**MCAR**	**MAR**	**MNAR**
	β diff	**β diff**	**β diff**	**β diff**	**β diff**	**β diff**	**β diff**	**β diff**	**β diff**
Intercept	-0.0522	-0.0803	0.05514	-0.00249	0.060587	0.07772	0.00434	0.09492	0.05637
Age	0.00051	0.00081	0.00044	-4.20E-05	-4.40E-05	-0.0015	0.00146	0.00013	-0.0018
BMI	0.00097	0.00241	-0.0027	-5.50E-05	-0.0014	0.00063	-0.0013	-0.0009	0.00181
Sex	-0.0067	-0.0287	-0.0094	0.00604	-0.00459	0.01037	-0.0276	-0.0463	0.01058
Packyears	9.53E-05	-0.001	-0.0004	-0.00011	-0.00043	-0.0015	0.00011	0.00036	-0.0003
Livestock	-0.0019	-0.0019	0.00482	-0.00319	-0.02755	-0.0267	-0.009	-0.0257	0.03128

Green → lowest value among its own missing percentage
Yellow → medium value among its own missing percentage
Red → highest value among its own missing percentage

Table 5.15 Estimates of Bias From Actual Estimates for Galimard et. al's MI method on SRHS datasets

	15%			30%			50%		
	MCAR	MAR	MNAR	MCAR	MAR	MNAR	MCAR	MAR	MNAR
	β diff	β diff	β diff	β diff	β diff	β diff	β diff	β diff	β diff
Intercept	-0.0347	-0.1023	0.018548	-0.0151	0.046675	0.07896	-0.0016	0.07151	0.00873
Age	0.00042	0.00082	0.000338	0.00015	-6.93E-05	-0.0015	0.00157	0.00054	-0.0016
BMI	0.00062	0.0026	-0.00112	1.14E-05	-0.00137	0.00024	-0.0013	-0.001	0.00311
Sex	-0.0073	-0.0159	-0.01031	0.00936	0.010474	0.0124	-0.0246	-0.04	0.0217
Packyears	0.00016	-0.0002	-5.76E-05	-0.0001	-0.0002	-0.0013	0.00026	0.00088	-0.0006
Livestock	-0.0013	0.00739	0.00056	-0.0039	-0.01693	-0.0071	-0.0108	-0.0257	0.03369

Green → lowest value among its own missing percentage
Yellow → medium value among its own missing percentage
Red → highest value among its own missing percentage

Table 5.16 Estimates of Bias From Actual Estimates for Ogundimu & Collins' MI Method of SRHS Datasets

	15%			30%			50%		
	MCAR	MAR	MNAR	MCAR	MAR	MNAR	MCAR	MAR	MNAR
	β diff	β diff	β diff	β diff	β diff	β diff	β diff	β diff	β diff
Intercept	0.08109	0.18572	0.03575	0.01207	0.01207	0.0466	0.05223	0.02529	-0.0121
Age	0.00085	-0.0012	0.00312	0.00123	0.00123	0.00205	0.00185	0.00014	-0.0009
BMI	-0.0011	-0.0016	-0.0028	-0.0005	-0.0005	-0.0011	-0.0013	-0.002	0.00155
Sex	-0.0436	-0.0286	-0.0215	-0.0506	-0.0506	-0.0439	-0.0677	-0.0255	0.01093
Packyears	-0.0001	8.18E-05	0.00107	-0.0005	-0.0005	0.00118	9.40E-05	-0.0013	-0.0011
Livestock	-0.0185	-0.0249	-0.0459	-0.019	-0.019	-0.0416	-0.023	-0.0322	0.03009

Green → lowest value among its own missing percentage
Yellow → medium value among its own missing percentage
Red → highest value among its own missing percentage

Interpretation of Results Based on Table 5.13 For the SRHS dataset, CC analysis delivered the most variables with the lowest beta differences for MNAR data when missing percentage is 15%. But for 30% and 50% missing, CC delivered the most variables with lowest beta differences for MCAR data while MNAR data had the most variables with the highest beta differences. This result shows that CC can work well for MCAR and MNAR data when missing percentage is low (15%), but it more noticeably delivered better results for MCAR data than MNAR when missing percentage is larger.

Interpretation of Results Based on Table 5.14 Rubin's MI method on SRHS data delivered most variables with lowest beta differences and least variables with highest beta differences for MCAR data in all missing percentages considered. For 15% missing, compared to higher missing percentages (30% and 50%), the MNAR datasets in the higher missing percentages contained more variables with highest beta differences than the 15%. The result suggests that Rubin's MI performs best for MCAR data, but it is likely to perform better for MNAR data when missing percentage is low.

Interpretation of Results Based on Table 5.15 When Galimard et. al's MI method was applied to the SRHS data, for 15% missing data the MNAR data had the most coefficients that were closest to the complete data while the MAR data had all variables with coefficients farthest from the complete data. For both 30% and 50% missing data, Galimard et. al's MI method delivered the best results for the MCAR datasets while MAR datasets each had 2 variables with coefficients farthest from the complete data and MNAR data each had 3 variables with coefficients farthest from the complete data. This result is similar to the RANDHIE data for Galimard et. al's MI method, which suggests that Galimard et. al's MI is likely to deliver coefficients closest to the complete data for 15% MNAR data or MCAR data of missing percentage below 50%, but it will deliver relatively inaccurate results for 30% and 50% MAR and MNAR datasets. A point worth noting is that Galimard et. al's MI, like in RANDHIE, gave the same result pattern as CC method for the SRHS datasets.

Interpretation of Results Based on Table 5.16 When Ogundimu & Collins' MI method was applied to SRHS data, for 15% missing data, MCAR had 3 coefficients closest to complete data while MNAR had 4 coefficients farthest from the complete data. For 30% missing data, the results for MCAR and MAR were the same, with 3 coefficients closest to the complete data while 2 were

farthest. For 50% missing data, MCAR had 3 variables coefficients closest to the complete data while MAR had 3 farthest from the complete data (with better performance in MNAR data than MAR). This suggests that Ogundimu & Collins' method is more likely to produce results similar to the true results for MCAR data in datasets with lower missing percentages (15% and 30%) while for higher missing percentages (50%) the results for MNAR data might be closer to the true results than other mechanisms. Another point worth noting is that Ogundimu & Collins' method, unlike CC, Rubin's MI, and Galimard et. al's MI, did not produce results in the SRHS dataset that is similar to the RANDHIE dataset.

5.6 Simulation Results

For the simulation study, the 4 missing data methods were applied to simulated data of different missing percentage and mechanism combinations to investigate their performances. For CC, Rubin's MI, and Galimard et. al's MI, the methods were successfully applied to 1000 simulated datasets of all 9 different missing percentage and mechanism combinations. Ogundimu & Collins' MI, however, was only successfully applied for few simulated datasets (~100 – 400 out of 1000) for missing percentages 15% and 30%; for 50% missing MCAR, MAR, and MNAR, Ogundimu and Collins' MI did not converge for imputing any of the 1000 simulated datasets. The issue was probably due to the simulated datasets' tendency to produce a Hessian matrix which produced a negative definite instead of a positive definite (Nocedal & Wright, 2000) . This in turn led to the lack of convergence for the optimization. Therefore, for 50% MCAR, MAR, and MNAR, the simulation results for Ogundimu and Collins' MI are not included in the tables. The 6 assessment statistics used to assess the effectiveness of the methods were: 1) Bias of regression coefficient, 2) Relative Bias of Regression Coefficient, 3) Standardized Bias of Regression Coefficient, 4) Mean Square Error of Regression Coefficient, 5) Average 95% Confidence Interval Length among simulated datasets, and 6) Percent (%) Coverage of True Regression Coefficients. Bias, Relative Bias, and Standardized Bias of Regression Coefficient were used to assess how much the regression coefficients of the simulated data differed from the real coefficients. The mean square error of regression coefficients is used to evaluate the quality of the coefficients. The average 95% CI length among simulated datasets was to look at how precise were the regression coefficients. The % coverage of true regression coefficients is to see the accuracy of the simulated regression coefficients. The method with the best performance will be the method with less bias,

smaller MSE, narrower average length of 95% confidence interval, and higher coverage of true coefficients by the 1000 simulated confidence intervals (Hossain & Pahwa, 2010). **Tables 5.17 – 5.25** present the assessment statistics for the final models of each coefficients for each missing percentage and mechanism combinations for the simulated datasets.

5.6.1 Simulations Results

5.6.1.1 15% MCAR Data

Table 5.17 Simulation results of CC, Rubin's MI, Galimard et. al's MI, and Ogundimu and Collins' MI on 15% MCAR Data

15% MCAR, simulated data					
Assessment Statisticcs	**Variables**	**CC**	**Rubin's MI**	**Galimard**	**Ogundimu**
Bias of regression coefficients	**Intercept**	0.001	0.001	0.001	-0.016
	Xnorm	-0.001	-0.001	-0.001	-0.002
	Xunif	0.001	0.001	0.002	0.002
	XGamma	0.001	0.001	0.002	0.004
	Xcat	0.000	0.000	0.000	-0.005
Relative bias of regression coefficients	**Intercept**	0.094	0.098	0.071	-1.640
	Xnorm	-0.048	-0.054	-0.047	-0.124
	Xunif	0.068	0.058	0.084	0.077
	XGamma	-0.147	-0.126	-0.158	-0.413
	Xcat	0.004	-0.016	-0.002	0.224
Standardized bias of regression coefficients	**Intercept**	25.7	18.6	19.4	-39.1
	Xnorm	-57.0	-43.5	-55.5	-61.3
	Xunif	38.3	19.9	46.8	18.8
	XGamma	44.6	33.2	48.2	80.0
	Xcat	-5.1	11.1	2.6	-96.1
Mean Square Error (MSE) of regression coefficients	**Intercept**	0.000	0.000	0.000	0.002
	Xnorm	0.000	0.000	0.000	0.000
	Xunif	0.000	0.000	0.000	0.000
	XGamma	0.000	0.000	0.000	0.000
	Xcat	0.000	0.000	0.000	0.000

Table 5.17 (cont'd) Simulation results of CC, Rubin's MI, Galimard et. al's MI, and Ogundimu and Collins' MI on 15% MCAR Data

15% MCAR, simulated data					
Assessment Statisticcs	**Variables**	**CC**	**Rubin's MI**	**Galimard**	**Ogundimu**
Coverage of true regression coefficients (%)	**Intercept**	0.936	0.939	0.936	0.765
	Xnorm	0.955	0.955	0.955	0.982
	Xunif	0.946	0.950	0.946	0.964
	XGamma	0.950	0.953	0.950	0.964
	Xcat	0.944	0.939	0.947	0.946
Average 95% confidence interval length among simulated datasets	**Intercept**	0.407	0.410	0.407	0.564
	Xnorm	0.135	0.136	0.135	0.151
	Xunif	0.466	0.469	0.466	0.523
	XGamma	0.192	0.193	0.192	0.217
	Xcat	0.275	0.276	0.275	0.309

For 15% MCAR, Ogundimu & Collins' MI had the most variables with the highest bias overall while Rubin's MI had the most variables with the lowest bias overall. Galimard et. al's MI had higher biases for Xunif and XGamma. For MSE, all 4 methods had very low values for all variables. For Coverage of True Regression Coefficients, for all 4 methods, all variable coefficients had more than 90% coverage. Ogundimu and Collins' MI had the most variables with the highest coverage, but it also had the lowest coverage for the intercept (77%). For the average 95% CI length, Ogundimu and Collins' MI had the most variables with the longest CI lengths while CC and Galimard et. al's MI had the shortest CI for most variables.

5.6.1.2 15% MAR Data

Table 5.18 Simulation results of CC, Rubin's MI, Galimard et. al's MI, and Ogundimu and Collins' MI on 15% MAR Data

15% MAR, simulated data					
Assessment Statisticcs	**Variables**	**CC**	**Rubin's MI**	**Galimard**	**Ogundimu**
Bias of regression coefficients	**Intercept**	0.003	0.003	0.003	0.009
	Xnorm	-0.001	-0.001	-0.001	-0.005
	Xunif	0.001	0.000	0.001	-0.001
	XGamma	0.001	0.001	0.001	-0.001
	Xcat	0.000	0.000	0.000	0.006

Table 5.18 (cont'd) Simulation results of CC, Rubin's MI, Galimard et. al's MI, and Ogundimu and Collins' MI on 15% MAR Data

15% MAR, simulated data					
Assessment Statisticcs	**Variables**	**CC**	**Rubin's MI**	**Galimard**	**Ogundimu**
Relative bias of regression coefficients	**Intercept**	0.275	0.255	0.275	0.851
	Xnorm	-0.093	-0.096	-0.093	-0.315
	Xunif	0.027	-0.005	0.027	-0.042
	XGamma	-0.098	-0.142	-0.098	0.080
	Xcat	0.012	0.001	0.012	-0.246
Standardized bias of regression coefficients	**Intercept**	73.3	44.5	73.3	59.7
	Xnorm	-106.9	-71.4	-106.9	-200.0
	Xunif	14.4	-1.5	14.4	-10.5
	XGamma	28.4	34.4	28.4	-19.9
	Xcat	-14.0	-0.4	-14.0	121.0
Mean Square Error (MSE) of regression coefficients	**Intercept**	0.000	0.000	0.000	0.000
	Xnorm	0.000	0.000	0.000	0.000
	Xunif	0.000	0.000	0.000	0.000
	XGamma	0.000	0.000	0.000	0.000
	Xcat	0.000	0.000	0.000	0.000
Coverage of true regression coefficients (%)	**Intercept**	0.932	0.935	0.932	0.960
	Xnorm	0.948	0.943	0.948	0.936
	Xunif	0.942	0.946	0.942	0.952
	XGamma	0.951	0.949	0.951	0.952
	Xcat	0.940	0.942	0.940	0.936
Average 95% confidence interval length among simulated datasets	**Intercept**	0.413	0.417	0.413	0.486
	Xnorm	0.136	0.138	0.136	0.142
	Xunif	0.472	0.475	0.472	0.490
	XGamma	0.194	0.196	0.194	0.200
	Xcat	0.278	0.281	0.278	0.290

For 15% MAR, Ogundimu & Collins' MI had the most variables with the highest bias overall while Rubin's MI had the most variables with the lowest bias overall. Galimard et. al's MI and CC had the exact same bias results as well as MSE, Coverage of True Regression Coefficients, and average 95% CI length. For MSE, all 4 methods had very low values for all variables. For Coverage of True Regression Coefficients, for all 4 methods, all variable coefficients had more than 90% coverage without significant difference between methods. For the average 95% CI length, Ogundimu and Collins' MI had the most variables with the longest CI lengths while CC and Galimard et. al's MI had the shortest CI for most variables.

5.6.1.3 15% MNAR Data

Table 5.19 Simulation results of CC, Rubin's MI, Galimard et. al's MI, and Ogundimu and Collins' MI on 15% MNAR Data

15% MNAR, simulated data					
Assessment Statisticcs	**Variables**	**CC**	**Rubin's MI**	**Galimard**	**Ogundimu**
Bias of regression coefficients	**Intercept**	0.379	0.383	0.379	-0.020
	Xnorm	-0.135	-0.137	-0.135	0.007
	Xunif	-0.178	-0.180	-0.178	0.014
	XGamma	0.114	0.116	0.114	-0.008
	Xcat	0.243	0.246	0.243	-0.012
Relative bias of regression coefficients	**Intercept**	37.9	38.3	37.9	-2.0
	Xnorm	-9.0	-9.1	-9.0	0.5
	Xunif	-8.9	-9.0	-8.9	0.7
	XGamma	-11.4	-11.6	-11.4	0.8
	Xcat	-10.6	-10.7	-10.6	0.5
Standardized bias of regression coefficients	**Intercept**	9755	4534	9755	-201
	Xnorm	-9927	-4368	-9927	203
	Xunif	-5109	-2925	-5109	221
	XGamma	3382	1555	3382	-107
	Xcat	11221	4722	11221	-216
Mean Square Error (MSE) of regression coefficients	**Intercept**	0.144	0.147	0.144	0.000
	Xnorm	0.018	0.019	0.018	0.000
	Xunif	0.032	0.032	0.032	0.000
	XGamma	0.013	0.013	0.013	0.000
	Xcat	0.059	0.060	0.059	0.000
Coverage of true regression coefficients (%)	**Intercept**	0.049	0.061	0.049	0.966
	Xnorm	0.034	0.032	0.034	0.971
	Xunif	0.658	0.650	0.658	0.964
	XGamma	0.499	0.494	0.499	0.976
	Xcat	0.073	0.070	0.073	0.966
Average 95% confidence interval length among simulated datasets	**Intercept**	0.423	0.429	0.423	0.469
	Xnorm	0.142	0.144	0.142	0.159
	Xunif	0.453	0.456	0.453	0.497
	XGamma	0.228	0.231	0.228	0.246
	Xcat	0.280	0.284	0.280	0.311

For 15% MNAR, the biases, MSE, average 95% CI length were higher for all 4 methods than the other two missing mechanisms for 15% missing data. The percent coverage of the true

regression coefficients were also significantly lower for CC, Rubin's MI, and Galimard et. al's MI. CC and Galimard et. al's MI had identical results for all 6 assessment statistics for all variables. Ogundimu and Collins' MI had the lowest biases and MSE for all variables while Rubin's MI had the highest. For coverage of true regression coefficient percentage, Ogundimu and Collins' MI had all variables with 95%+ coverage while the other 3 methods had significantly lower coverage for their variables with Rubin's MI having the most variables with the lowest coverage. For 95% CI length, Ogundimu and Collins' MI had the longest length while CC and Galimard et. al's MI had the shortest.

5.6.1.4 30% MCAR Data

Table 5.20 Simulation results of CC, Rubin's MI, Galimard et. al's MI, and Ogundimu and Collins' MI on 30% MCAR Data

30% MCAR, simulated data					
Assessment Statisticcs	**Variables**	**CC**	**Rubin's MI**	**Galimard**	**Ogundimu**
Bias of regression coefficients	**Intercept**	0.001	0.002	0.001	0.019
	Xnorm	-0.001	-0.001	-0.001	-0.002
	Xunif	0.002	0.001	0.002	-0.016
	XGamma	0.001	0.001	0.001	-0.003
	Xcat	-0.001	-0.001	0.000	0.016
Relative bias of regression coefficients	**Intercept**	0.144	0.223	0.144	1.867
	Xnorm	-0.082	-0.082	-0.088	-0.126
	Xunif	0.106	0.049	0.097	-0.824
	XGamma	-0.139	-0.137	-0.150	0.319
	Xcat	0.024	0.028	0.016	-0.712
Standardized bias of regression coefficients	**Intercept**	32.0	22.6	32.0	36.0
	Xnorm	-78.5	-36.2	-83.7	-43.3
	Xunif	48.0	8.8	43.8	-104.9
	XGamma	34.0	23.3	36.5	-39.7
	Xcat	-22.9	-9.9	-15.4	184.8
Mean Square Error (MSE) of regression coefficients	**Intercept**	0.000	0.000	0.000	0.003
	Xnorm	0.000	0.000	0.000	0.000
	Xunif	0.000	0.000	0.000	0.001
	XGamma	0.000	0.000	0.000	0.000
	Xcat	0.000	0.000	0.000	0.000

Table 5.20 (cont'd) Simulation results of CC, Rubin's MI, Galimard et. al's MI, and Ogundimu and Collins' MI on 30% MCAR Data

30% MCAR, simulated data					
Assessment Statisticcs	**Variables**	**CC**	**Rubin's MI**	**Galimard**	**Ogundimu**
Coverage of true regression coefficients (%)	**Intercept**	0.943	0.943	0.943	0.409
	Xnorm	0.950	0.955	0.950	0.939
	Xunif	0.952	0.953	0.952	0.948
	XGamma	0.947	0.947	0.947	0.939
	Xcat	0.948	0.941	0.948	0.965
Average 95% confidence interval length among simulated datasets	**Intercept**	0.450	0.456	0.450	0.732
	Xnorm	0.149	0.151	0.149	0.178
	Xunif	0.515	0.523	0.515	0.623
	XGamma	0.212	0.215	0.212	0.254
	Xcat	0.303	0.308	0.303	0.366

For 30% MCAR, Ogundimu & Collins' MI had the most variables with the highest bias overall while Rubin's MI had the most variables with the lowest bias overall. Galimard et. al's MI had higher biases for Xunif and XGamma. For MSE, all 4 methods had very low values for all variables. For Coverage of True Regression Coefficients, for all 4 methods, all variable coefficients had more than 90% coverage with Ogundimu and Collins' MI having the most variables with the lowest coverage. For the average 95% CI length, Ogundimu and Collins' MI had the most variables with the longest CI lengths while CC and Galimard et. al's MI had the shortest CI for most variables.

5.6.1.5 30% MAR Data

Table 5.21 Simulation results of CC, Rubin's MI, Galimard et. al's MI, and Ogundimu and Collins' MI on 30% MAR Data

30% MAR, simulated data					
Assessment Statisticcs	**Variables**	**CC**	**Rubin's MI**	**Galimard**	**Ogundimu**
Bias of regression coefficients	**Intercept**	0.001	0.001	0.001	0.043
	Xnorm	-0.001	-0.001	-0.001	-0.007
	Xunif	0.002	0.001	0.002	0.001
	XGamma	0.001	0.001	0.001	-0.003
	Xcat	0.001	0.001	0.001	0.006

Table 5.21 (cont'd) Simulation results of CC, Rubin's MI, Galimard et. al's MI, and Ogundimu and Collins' MI on 30% MAR Data

30% MAR, simulated data					
Assessment Statisticcs	**Variables**	**CC**	**Rubin's MI**	**Galimard**	**Ogundimu**
Relative bias of regression coefficients	**Intercept**	0.106	0.103	0.106	4.251
	Xnorm	-0.070	-0.081	-0.070	-0.491
	Xunif	0.078	0.042	0.078	0.041
	XGamma	-0.127	-0.149	-0.127	0.299
	Xcat	-0.027	-0.063	-0.027	-0.277
Standardized bias of regression coefficients	**Intercept**	23.3	9.9	23.3	125.4
	Xnorm	-66.9	-34.5	-66.9	-172.6
	Xunif	34.3	7.3	34.3	6.6
	XGamma	30.3	24.3	30.3	-46.6
	Xcat	25.9	22.0	25.9	87.3
Mean Square Error (MSE) of regression coefficients	**Intercept**	0.000	0.000	0.000	0.003
	Xnorm	0.000	0.000	0.000	0.000
	Xunif	0.000	0.000	0.000	0.000
	XGamma	0.000	0.000	0.000	0.000
	Xcat	0.000	0.000	0.000	0.000
Coverage of true regression coefficients (%)	**Intercept**	0.939	0.944	0.939	0.923
	Xnorm	0.945	0.951	0.945	0.949
	Xunif	0.957	0.958	0.957	0.949
	XGamma	0.946	0.952	0.946	0.962
	Xcat	0.951	0.948	0.951	0.962
Average 95% confidence interval length among simulated datasets	**Intercept**	0.454	0.461	0.454	0.661
	Xnorm	0.150	0.152	0.150	0.161
	Xunif	0.519	0.526	0.519	0.555
	XGamma	0.213	0.216	0.213	0.231
	Xcat	0.306	0.309	0.306	0.325

For 30% MAR, Ogundimu & Collins' MI had the most variables with the highest bias overall while Rubin's MI had the most variables with the lowest bias overall. Galimard et. al's MI and CC had the exact same bias results as well as MSE, Coverage of True Regression Coefficients, and average 95% CI length. For MSE, all 4 methods had very low values for all variables. For Coverage of True Regression Coefficients, for all 4 methods, all variable coefficients had more than 90% coverage without significant difference between methods. For the average 95% CI length, Ogundimu and Collins' MI had the most variables with the longest CI lengths while CC and Galimard et. al's MI had the shortest CI for most variables.

5.6.1.6 30% MNAR Data

Table 5.22 Simulation results of CC, Rubin's MI, Galimard et. al's MI, and Ogundimu and Collins' MI on 30% MNAR Data

30% MNAR, simulated data					
Assessment Statisticcs	**Variables**	**CC**	**Rubin's MI**	**Galimard**	**Ogundimu**
Bias of regression coefficients	**Intercept**	0.585	0.614	0.585	-0.016
	Xnorm	-0.186	-0.195	-0.186	0.005
	Xunif	-0.245	-0.259	-0.245	0.008
	XGamma	0.146	0.154	0.146	-0.001
	Xcat	0.332	0.348	0.332	-0.007
Relative bias of regression coefficients	**Intercept**	58.5	61.4	58.5	-1.6
	Xnorm	-12.4	-13.0	-12.4	0.3
	Xunif	-12.3	-13.0	-12.3	0.4
	XGamma	-14.6	-15.4	-14.6	0.1
	Xcat	-14.5	-15.1	-14.5	0.3
Standardized bias of regression coefficients	**Intercept**	12473	4342	12473	-60
	Xnorm	-11601	-4175	-11601	51
	Xunif	-5984	-2586	-5984	61
	XGamma	3506	1488	3506	-11
	Xcat	12114	4060	12114	-46
Mean Square Error (MSE) of regression coefficients	**Intercept**	0.342	0.378	0.342	0.001
	Xnorm	0.035	0.038	0.035	0.000
	Xunif	0.060	0.067	0.060	0.000
	XGamma	0.021	0.024	0.021	0.000
	Xcat	0.111	0.121	0.111	0.000
Coverage of true regression coefficients (%)	**Intercept**	0.003	0.004	0.003	0.958
	Xnorm	0.005	0.005	0.005	0.954
	Xunif	0.493	0.457	0.493	0.965
	XGamma	0.374	0.359	0.374	0.968
	Xcat	0.014	0.006	0.014	0.968
Average 95% confidence interval length among simulated datasets	**Intercept**	0.469	0.478	0.469	0.553
	Xnorm	0.155	0.158	0.155	0.186
	Xunif	0.482	0.488	0.482	0.546
	XGamma	0.256	0.261	0.256	0.282
	Xcat	0.311	0.315	0.311	0.366

For 30% MNAR, the biases, MSE, average 95% CI length were higher for all 4 methods than the other two missing mechanisms for 30% missing data as well as 15% MNAR. The percent

coverage of the true regression coefficients were also significantly for lower CC, Rubin's MI, and Galimard et. al's MI. CC and Galimard et. al's MI had identical results for all 6 assessment statistics for all variables. Ogundimu and Collins' MI had the lowest biases and MSE for all variables while Rubin's MI had the highest. For coverage of true regression coefficient percentage, Ogundimu and Collins' MI had all variables with 95%+ coverage while the other 3 methods had significantly lower coverage for their variables with Rubin's MI having the most variables with the lowest coverage. For 95% CI length, Ogundimu and Collins' MI had the longest length while CC and Galimard et. al's MI had the shortest.

5.6.1.7 50% MCAR Data

Table 5.23 Simulation Results of CC, Rubin's MI, and Galimard et. al's MI on 50% MCAR Data

50% MCAR, simulated data				
Assessment Statisticcs	**Variables**	**CC**	**Rubin's MI**	**Galimard**
Bias of regression coefficients	**Intercept**	0.001	0.001	0.001
	Xnorm	-0.002	-0.001	-0.002
	Xunif	0.005	0.005	0.005
	XGamma	-0.001	-0.002	-0.001
	Xcat	-0.001	-0.001	-0.001
Relative bias of regression coefficients	**Intercept**	0.079	0.081	0.108
	Xnorm	-0.100	-0.083	-0.103
	Xunif	0.255	0.250	0.255
	XGamma	0.134	0.231	0.141
	Xcat	0.036	0.037	0.042
Standardized bias of regression coefficients	**Intercept**	12.0	4.3	16.4
	Xnorm	-67.9	-20.2	-69.5
	Xunif	77.7	23.8	77.6
	XGamma	-22.8	-23.7	-23.9
	Xcat	-23.5	-7.1	-27.5
Mean Square Error (MSE) of regression coefficients	**Intercept**	0.000	0.000	0.000
	Xnorm	0.000	0.000	0.000
	Xunif	0.000	0.000	0.000
	XGamma	0.000	0.000	0.000
	Xcat	0.000	0.000	0.000

Table 5.23 (cont'd) Simulation Results of CC, Rubin's MI, and Galimard et. al's MI on 50% MCAR Data

50% MCAR, simulated data				
Assessment Statisticcs	**Variables**	**CC**	**Rubin's MI**	**Galimard**
Coverage of true regression coefficients (%)	**Intercept**	0.955	0.944	0.955
	Xnorm	0.960	0.950	0.960
	Xunif	0.951	0.941	0.951
	XGamma	0.956	0.941	0.956
	Xcat	0.964	0.950	0.964
Average 95% confidence interval length among simulated datasets	**Intercept**	0.534	0.543	0.534
	Xnorm	0.177	0.180	0.177
	Xunif	0.611	0.620	0.611
	XGamma	0.251	0.256	0.251
	Xcat	0.360	0.367	0.360

For 50% MCAR, Galimard et. al's MI had the most variables with the highest bias while Rubin's MI had the lowest. For MSE, CC and Galimard et. al's MI had the lowest MSE while Rubin's MI had higher MSE. For the coverage of true regression coefficients, all 3 methods achieved 90%+ coverage for all variables with Galimard et. al's MI and CC having slightly higher coverages than Rubin's MI. For the average 95% CI, Rubin's MI had the longer CI lengths while CC and Galimard et. al's MI had the shorter CI lengths.

5.6.1.8 50% MAR Data

Table 5.24 Simulation Results of CC, Rubin's MI, and Galimard et. al's MI on 50% MAR Data

50% MAR, simulated data				
Assessment Statisticcs	**Variables**	**CC**	**Rubin's MI**	**Galimard**
Bias of regression coefficients	**Intercept**	0.004	0.004	0.004
	Xnorm	-0.001	-0.001	-0.001
	Xunif	-0.001	-0.002	-0.001
	XGamma	0.000	0.000	0.000
	Xcat	-0.001	-0.001	-0.001
Relative bias of regression coefficients	**Intercept**	0.422	0.360	0.422
	Xnorm	-0.091	-0.063	-0.091
	Xunif	-0.046	-0.094	-0.046
	XGamma	0.007	-0.019	0.007
	Xcat	0.050	0.042	0.050

Table 5.24 (cont'd) Simulation Results of CC, Rubin's MI, and Galimard et. al's MI on 50% MAR Data

50% MAR, simulated data				
Assessment Statisticcs	**Variables**	**CC**	**Rubin's MI**	**Galimard**
Standardized bias of regression coefficients	**Intercept**	60.9	19.0	60.9
	Xnorm	-58.8	-14.7	-58.8
	Xunif	-13.1	-8.4	-13.1
	XGamma	-1.2	1.8	-1.2
	Xcat	-31.4	-8.0	-31.4
Mean Square Error (MSE) of regression coefficients	**Intercept**	0.000	0.000	0.000
	Xnorm	0.000	0.000	0.000
	Xunif	0.000	0.000	0.000
	XGamma	0.000	0.000	0.000
	Xcat	0.000	0.000	0.000
Coverage of true regression coefficients (%)	**Intercept**	0.945	0.948	0.945
	Xnorm	0.951	0.944	0.951
	Xunif	0.946	0.937	0.946
	XGamma	0.958	0.944	0.958
	Xcat	0.960	0.954	0.960
Average 95% confidence interval length among simulated datasets	**Intercept**	0.546	0.557	0.546
	Xnorm	0.180	0.183	0.180
	Xunif	0.625	0.638	0.625
	XGamma	0.258	0.262	0.258
	Xcat	0.368	0.373	0.368

For 50% MAR, Galimard et. al's MI had the most variables with the highest bias while Rubin's MI had the lowest. For MSE, CC and Galimard et. al's MI had the lowest MSE while Rubin's MI had higher MSE. For the coverage of true regression coefficients, all 3 methods achieved 90%+ coverage for all variables with Galimard et. al's MI and CC having slightly higher coverages than Rubin's MI. For the average 95% CI, Rubin's MI had the longer CI lengths while CC and Galimard et. al's MI had the shorter CI lengths.

5.6.1.9 50% MNAR RANDHIE Data

Table 5.25 Simulation Results of CC, Rubin's MI, and Galimard et. al's MI on 50% MNAR Data

50% MNAR, simulated data				
Assessment Statisticcs	**Variables**	**CC**	**Rubin's MI**	**Galimard**
Bias of regression coefficients	**Intercept**	0.771	0.895	0.771
	Xnorm	-0.197	-0.229	-0.197
	Xunif	-0.265	-0.311	-0.265
	XGamma	0.146	0.175	0.146
	Xcat	0.336	0.391	0.336
Relative bias of regression coefficients	**Intercept**	77.1	89.5	77.1
	Xnorm	-13.1	-15.3	-13.1
	Xunif	-13.3	-15.6	-13.3
	XGamma	-14.6	-17.5	-14.6
	Xcat	-14.6	-17.0	-14.6
Standardized bias of regression coefficients	**Intercept**	10551	3588	10551
	Xnorm	-8293	-3144	-8293
	Xunif	-4385	-1616	-4385
	XGamma	2207	1051	2207
	Xcat	7506	2416	7506
Mean Square Error (MSE) of regression coefficients	**Intercept**	0.595	0.802	0.595
	Xnorm	0.039	0.053	0.039
	Xunif	0.070	0.097	0.070
	XGamma	0.021	0.031	0.021
	Xcat	0.113	0.153	0.113
Coverage of true regression coefficients (%)	**Intercept**	0.001	0.000	0.001
	Xnorm	0.011	0.001	0.011
	Xunif	0.559	0.455	0.559
	XGamma	0.576	0.434	0.576
	Xcat	0.084	0.036	0.084
Average 95% confidence interval length among simulated datasets	**Intercept**	0.589	0.611	0.589
	Xnorm	0.185	0.191	0.185
	Xunif	0.575	0.584	0.575
	XGamma	0.325	0.328	0.325
	Xcat	0.392	0.403	0.392

For 50% MNAR, the biases, MSE, average 95% CI length were higher for all 3 methods than the other two missing mechanisms for 50% missing data as well as 30% and 15% MNAR. The

percent coverage of the true regression coefficients was also significantly for lower for all 3 methods than 50% MAR and MCAR. CC and Galimard et. al's MI had identical results for all 6 assessment statistics for all variables. CC and Galimard et. al's MI had the lowest biases and MSE for all variables while Rubin's MI had the highest. For coverage of true regression coefficient percentage, Rubin's MI had the most variables with the lowest coverage. For 95% CI length, Rubin's MI had the longest length while CC and Galimard et. al's MI had shorter CI lengths.

CHAPTER 6

DISCUSSION AND FUTURE DIRECTIONS

6.1 Discussion & Conclusions

When the 4 missing data methods were applied to real-life datasets, the 95% confidence interval lengths of the estimated betas and the difference between the estimated betas with missing data and without missing data were used to assess how well the methods worked on two different datasets. Based on the beta differences in the RANDHIE dataset, the observed pattern was neither as clear-cut nor as absolute as expected. The CC method worked (as expected) best for MCAR data of all 3 data percentages. Rubin's MI worked best on 15% MCAR data; when the missing data increased to 30%, the method worked best on MAR data. However, for 50% missing data, the Rubin's MI worked best on MCAR data. Galimard et. al's MI, which was designed for handling MNAR data, performed best overall on MCAR data for all missing percentages, with its' performances on MNAR decreasing in quality as the missing data percentage increased. The results of Galimard et. al's MI and CC were identical. Ogundimu & Collins' MI performed better on MNAR data than the MCAR and MAR data mechanisms when the missing data percentages were low (15% and 30%) and performed better on the ignorable missing data mechanisms (MCAR and MAR) than MNAR data when the missing percentage was high (50%). This shows that the two newly designed MNAR methods' will decrease in performance when the missing data percentage is increased. Regarding the confidence interval lengths, all 4 methods have shown that the datasets with ignorable missing data had more variables with shorter confidence interval lengths. This shows that the 4 methods will deliver more precise results in ignorable missing data than in non-ignorable and the two newer methods are unlikely to produce more precise methods than CC and Rubin's MI for MNAR data in RANDHIE.

For application of the 4 methods on the SRHS datasets with missing data, like the RANDHIE datasets, the pattern of the beta differences and confidence interval lengths were also not as clear-cut as expected. For the beta differences of the CC analysis, the method performed better on 15%

MNAR data than MCAR and MAR datasets (the expected result), while for 30% and 50% missing, the method performed better on the ignorable missing data than the non-ignorable. For Rubin's MI, the method performed best on MCAR datasets for all missing percentages. Galimard et. al's MI for SRHS, like RANDHIE, delivered the same performance pattern as CC for different missing percentage and mechanism combinations. Ogundimu & Collins' MI also did not show the expected clear-cut superior performance on MNAR datasets. The method worked better on MCAR and MAR datasets when missing percentage was 15% and 30% while at 50% the method performed slightly better in the sense that none of the beta differences for the MNAR dataset were the highest. The RANDHIE dataset is significantly larger than the SRHS dataset; and its imputation and outcome models involved more independent variables than SRHS. The pattern of the confidence interval lengths for the SRHS datasets was similar to RANDHIE where CC, Rubin's MI, and Galimard et. al's MI delivered more precise results for MCAR and MAR data than MNAR, but for SRHS, Ogundimu & Collins' MI performed better for MNAR data when missing percentage was higher (50%) which is opposite to the pattern of Ogundimu & Collins' MI on RANDHIE. Because the RANDHIE and SRHS dataset has similar result patterns of CC, Rubin's MI, and Galimard et. al's MI, while Ogundimu & Collins' MI results were different, this suggests that CC, Rubin's MI and Galimard et. al's MI were more robust against differences in datasets' and models' sizes than Ogundimu & Collins' MI.

For the simulation study, 6 assessment statistics were used to assess how well the 4 missing data handling methods perform on simulated datasets. For each percentage of missing data, the biases of regression coefficients calculated from all 4 methods were consistently lowest for MCAR and highest for MNAR. For each missing mechanism, the biases of regression coefficients for the 4 methods were not consistently lower for lower missing percentage nor higher for higher missing percentage for MCAR and MAR, but for MNAR the lowest and highest missing percentages received lowest and highest biases respectively. This suggests that missing percentage may play a bigger role in the effectiveness of missing data handling methods than missing mechanism (which may play a secondary role than missing percentage). It was also shown in the simulation results that for all 4 methods, the biases were highest for MNAR data for all missing percentages. Consistent with the biases result, the MSE values for regression coefficients were highest, the percent coverage of true regression coefficients were the lowest, and average 95% CI lengths were longest for MNAR data for each missing percentage.

For each missing percentage, Galimard et. al's MI has consistently delivered results similar to CC for each missing mechanism. This suggests that Galimard et. al's MI (a method designed specifically for MNAR data) will probably not deliver results superior to methods that are deemed only appropriate for MCAR and MAR data. However, Galimard et. al's MI's results being similar to the CC results across all missing percentages and missing mechanisms shows consistency in the method's result delivery. The method, like CC in this study, delivered lower biase and MSE, and narrower average 95% CI length than Rubin's MI when the methods were applied on MNAR data, which suggests that its' usefulness with MNAR data will be superior than Rubin's MI. The method can also be applied to MCAR and MAR data without introducing more bias to the results compared to CC or Rubin's MI.

For Ogundimu and Collins' MI, the method delivered significant superior results for MNAR data for 15% and 30% missing data (the missing percentages where there was some convergence for the method). The biases and MSE were significantly lower than the other methods for MNAR data, the coverage of true regression coefficients was significantly higher. The average 95% CI length were slightly longer for Ogundimu and Collins' MI compared to the other methods, but the slight downpoint cannot overturn its superior performance for the other 5 assessment statistics. For MCAR and MAR data of all percentages, Ogundimu and Collins' MI performed slightly worse than the other 3 methods in the assessment statistics. This suggests that Ogundimu and Collins' MI performs specifically well in MNAR data when the missing percentages are 15 – 30%, in contrast to Galimard et. al's MI which delivered similar performance as CC and Rubin's MI in all missing percentages and mechanisms in the simulation. Nevertheless, although Ogundimu and Collins' MI worked well for MNAR data, unlike Galimard et. al's MI, it cannot be used on MCAR nor MAR data without introducing bias to the results. Also, results of Ogundimu and Collins' MI should be viewed with more skepticism than the results of Galimard et. al's MI because Galimard et. al's MI converged for all 1000 simulated datasets whereas Ogundimu and Collins' MI only had approximately 100 – 400 cases of convergences for 15% and 30% missing data. Also, Ogundimu and Collins' MI did not converge for any of the 1000 simulated datasets for 50% missing. This suggests that it is still early to use Ogundimu and Collins' MI to datasets with high percentage of missing data compared to Galimard et. al's MI.

The simulation part of this study contains a more varied group of covariates with different distributions (normal, uniform, gamma, and binary categorical) while the studies of Galimard et. al and Ogundimu and Collins used only covariates of normal distributions. The results of the simulation showed that the assessment statistics for the variables did not vary from eachother much for different missing percentage and mechanisms. Even though there were slight differences, these differences were consistent between different missing mechanisms for the missing percentages. Also, in Galimard et. al and Ogundimu and Collins' studies, there differences of biases and other assessment statistics were also found between their covariates of the same distributions with the magnitude of the differences being more noticeable than the differences found in this study. This suggests that Galimard et. al and Ogundimu and Collins' MI methods are robust against covariates of different distributions.

From the application and simulation studies' results, it can be concluded that:

1) Galimard et. al's MI is likely to deliver the same results as CC analysis can be used for 15 – 50% MCAR data when CC analysis is not a feasible option
2) Ogundimu and Collins' MI is likely to produce distinctly better results for 15 – 30% MNAR data than its MCAR and MAR counter parts. But the method should be used with skepticism and caution because of its tendency to not converge in the simulation study.

Overall, the results of this study did not confirm significantly stronger performance of Galimard et. al's MI and Ogundimu & Collins' MI compared to methods such as CC and Rubin's MI presented in their own studies, even though the methods were applied onto datasets with similar outcome variables and the performance of the application in the real-life datasets can be verified with the results of the simulations. The results of this study raised some skepticisms of the promises of Galimard et. al's MI and Ogundimu & Collins' MI for their abilities to deliver unbiased linear regression estimates for cross-sectional data when there are MNAR data in the dataset.

6.2 Future Directions

In this study, Galimard et. al and Ogundimu & Collins' MI methods were put to the test through application to two new datasets; Overall, the results of these newer MI methods were not found to be consistently superior for MNAR data in every aspect compared to older methods such

as CC and Rubin's MI. However, despite of the results of this study, it is still very early to make a claim that Galimard et. al's MI and Ogundimu & Collins' MI are not superior to the other more classical methods for datasets. The methods were applied to RANDHIE and SRHS datasets in a health and a monetary outcome study respectively. There is still a possibility that the methods can work well in datasets other than RANDHIE and SRHS. Methods like CC analysis and Rubin's MI did not reach a status where their performances are well known by being applied to only a few datasets (Sterne et al., 2009; White, Royston, & Wood, 2011). In order to see how well the newer MI methods can handle MNAR data, they must be widely applied to more datasets of different kinds and modified to suit the different datasets to improve their performances.

This study also used two relatively complicated imputation and outcome models where 6 and 5 variables were used for the outcome models. The reason for using these complicated imputation models was partly because it was difficult to find variables in RANDHIE and SRHS which strongly influenced their individual outcome variables. Therefore, more significant independents were chosen to ensure the correctness of the imputation. In future studies, it may be wise to try applying the two newer MI methods with simpler imputation and outcome models and choose more datasets which contains independent variables that strongly influence on the outcome to see how well the two newer methods can perform. Also, the missing data in the RANDHIE and SRHS in this study were introduced into the dataset using independent variables and outcome variables with the purpose of being able to directly compare the imputation results with true results. In real life data, missing data (even for MAR and MNAR) are usually caused by more than one variable. Therefore, in future studies, it may be interesting to see how well Galimard et. al and Ogundimu & Collins' MI can perform on missing data introduced using more than one variable or on datasets with missing data that were not introduced artificially.

The difficulty of handling missing data lies mainly in the values of the missing data being unknown. When obtaining the missing information is difficult, handling missing data statistically is important for the accuracy of the information obtained from the dataset. The lack of knowledge for the missing data is the main issue which makes handling missing data a difficult task. In real life, even though there can be strong speculations of why and how the data were missing, the only knowledge we have for sure is how much data were missing in the datasets. It is highly possible that the missing data in a dataset do not all follow just one missing mechanism (MCAR, MAR, or MNAR). This possibility reveals the greatest weakness of MI - it expects that by imputing data

multiple times the uncertainty of the imputed data can also handle multiple missing data mechanisms in a dataset, which may not be the case.

When early statisticians first chose to impute missing data, they opted to impute each individual datum using one value that can describe the value of the variable overall (eg. mean, median, mode) or using the value from the individuals similar to the individuals with the missing data as imputes (eg. hot and cold deck imputation). These methods focus on the idea that we need to gather as much information about the missing data as possible before filling the missing data with placeholders. Multiple imputation strayed from this idea inadvertently through its design where it focuses more on how the imputed values are influenced by the values of other variables in the dataset and how through generating multiple placeholders using values of other variables it can compensate for the inherent inaccuracies caused by ignoring the characteristics of the variable with the missing data. In the study presented in chapter 4, it was shown that the MI methods (older like Rubin's MI and newer ones like Galimard et. al and Ogundimu & Collins' MI) do not always outperform complete case analysis in generating the best coefficients. This difficulty to predict MI's performance is probably due to the missing data not all following just one missing mechanism. For Rubin's MI, normal distribution is assumed for the variable containing missing data to be imputed; Galimard et. al's MI also has a normal assumption, Ogundimu & Collins' MI is the only one which paid some attention to the characteristics of the variable with missing data by taking into consideration that the variable could be skewed. All 3 methods focused strongly on the uncertainty of the missing data by focusing on the imputation models and not much on the characteristics of the missing data and the variable. Hence, it is not surprising that using MI to handle the missing data can lead to inconsistent results. Because the missing data in the datasets considered by this study were introduced into the datasets, which makes the missing mechanisms more clear-cut than real-life missing data, the inconsistent results further validates the idea that MI's over-emphasis on the uncertainties of the missing data rather than attempting to precisely deduce the values of the missing data, which will not serve well in handling missing data.

Although this study's attempt to compare two newly developed MI methods for MNAR data with classic Rubin's MI and CC analysis exposed some weaknesses of MI, it does not undermine MI and its variants' creative design and its robustness which made it so widely use since its conception. This study also does not undermine the efforts made by Galimard et. al and Ogundimu & Collins to extend the MI technique to MNAR data. Just as George Box claimed "All models are

wrong but some are useful", all missing data methods are wrong but some are better at handling certain types of missing data than others (Box & Tiao, 1992). Galimard et. al and Ogundimu & Collins' MI may not work best for MNAR for continuous data for all datasets, but the designs of their methods were logical and innovative. No statistical methods are designed perfectly where it needs no improvement and the subtle imperfections found for these methods found in these methods for certain dataset types can serve as a motivation to design better methods for handling missing data in the future.

The best method for designing better methods for an uncertain area like handling missing data is to improve upon existing methods for handling MNAR data such as Galimard et. al and Ogundimu & Collins' methods. Ogundimu & Collins' MI did not run successfully with 50% missing percentage. The first step in designing a better method is to run another simulation where simulated data generated based this dataset can accommodate all 4 methods considered in this study (CC, Rubin's MI, Galimard et. al's MI, and Ogundimu & Collins' MI). Galimard et. al's MI and Ogundimu & Collins' MI should also be applied to more datasets (of different types) and have the results compared to results of CC and Rubin's MI to find their positive and negative sides for handling MNAR data that were not uncovered by this study. The methods can also be extended to other types of missing data (Galimard et. al and Ogundimu & Collins' MI considered only missing continuous variables) to their effectiveness. Also, in this study, the imputation models included more independent variables (6 for RANDHIE and 5 for SRHS) than in other studies (which only uses 1 or 2). It may be more revealing in future studies to use less variables which are more effective in influencing the values of the outcome variables than multiple with only average influencing power because it is likely that this was the reason why the MI method did not perform as well as expected in the study. (But in defense of this study, it is difficult to find real life datasets where there are variables that strongly influence the outcome variable chosen without the variables being variants of eachother.) In addition, Galimard et. al and Ogundimu & Collins' MI both used parametric selection models to help build their imputation models. This is unwise because the distribution of the missingness data are unknown and modeling their missingness in such a concrete manner could potentially lead to an unsuitable imputation model which lead to inaccurate imputed values. It may be interesting to see in the future that semi- and non-parametric selection models can be used to develop imputation models for MNAR data and if they can produce better results than parametric.

Other than making improvements upon the existing methods, another way of improving the methods of handling MNAR data is by going back to the methods which emphasize deducing the values of the individual missing values (like hot and cold deck imputations). Deducing the values of individual missing values rather than imputing each missing value multiple times and take the average can potentially bypass the issue that there could be groups of missing data in the dataset with different missing mechanisms, which cannot be solved by MI with one imputation model. Currently, disciplines such as machine learning and deep learning have developed strongly capable methods for learning from information in a dataset, and these methods have the potential to be extended to solve statistical problems such as handling missing data. The randomforest method from machine learning can be applied to a larger dataset where there are missing data with more than one missing mechanisms by dividing the dataset into multiple datasets in multiple ways and multiple times with each subdataset's missing data being imputed with an imputation model suitable for it(Ostmann & Martínez Arbizu, 2018). The imputed subdatasets can then be merged into multiple "complete" datasets which can then be analyzed and merged with Rubin's method (Tang & Ishwaran, 2017). The deep learning method generative adversarial networks (GANs) can calculate the relationship of the outcome variable and the independent variables and how they are related can be learned down to the individual level, which will make more accurate predictions for the missing values than using a statistical model (Yoon, Jordon, & van der Schaar, 2018). Through using a deep neural network built using the dataset with missing data, the missing pattern, mechanism, and most likely values can be learned at the individual level, thereby generating more likely values for the missing data.

Missing data is a hydra-like issue in statistics. Because of the values of the missing data are unknown, each method designed to handle missing values is like using one sword used to cut off one hydra-head; and just like a sword cutting off one hydra head will only cause two more to grow in its place, one single missing data method will introduce more inaccuracies to the study results than the ones caused by the missing data. Galimard et. al and Ogundimu & Collins' MI, regardless of how well they were designed, are merely singly forged swords to individual hydra heads which can introduce more inaccuracies into the study results. When Heracles slayed the hydra, he called upon his cousin Iolaus to cauterize each hydra head after it has been cut so it will not grow back, thereby successfully slaying the hydra. The "cauterization method" for handling missing data is unlikely to be a single method, but a method that involves using multiple well-suited imputation

methods at different sections of the dataset at the same time. For this to happen, not only do we need an Iolaus of statistics in the future to figure out a way to combination of missing data methods properly for different datasets, we need more Heracles to design better and better methods to handle single types of missing data. As of now, all statisticians and data scientists can be either Heraculeses or Iolauses of missing data, or even both. By developing newer single methods and newer ways to combine these better methods, the issue of missing data can (if not solved) be better controlled.

BIBLIOGRAPHY

Andridge, R. R. (2011). A Review of Hot Deck Imputation for Survey Non-Response. *International Statistical Review*, *78*(1), 40–64. https://doi.org/10.1111/j.1751-5823.2010.00103.x.A

Azur, M. J., Stuart, E. A., Frangakis, C., & Leaf, P. J. (2012). Multiple Imputation by Chained Equations: What is it and how does it work? *International Journal of Methods in Psychiatric Research*, *20*(1), 40–49. https://doi.org/10.1002/mpr.329.Multiple

Bartlett, J. W., Seaman, S. R., White, I. R., & Carpenter, J. R. (2015). Multiple imputation of covariates by fully conditional specification: Accommodating the substantive model. *Statistical Methods in Medical Research*, *24*(4), 462–487. https://doi.org/10.1177/0962280214521348

Box, G. E. P., & Tiao, G. G. (1992). *Bayesian inference in statistical analysis*. Reading: Wiley.

Carpenter, J. R., & Kenward, M. G. (2012). *Multiple Imputation and Its Applications*. Chichester, UK: John Wiley & Sons, Ltd.

Chen, Q., Williams, S. Z., Liu, Y., Chihuri, S. T., & Li, G. (2018). Multiple imputation of missing marijuana data in the Fatality Analysis Reporting System using a Bayesian multilevel model. *Accident Analysis and Prevention*, *120*(January), 262–269. https://doi.org/10.1016/j.aap.2018.08.021

Demissie, S., LaValley, M. P., Horton, N. J., Glynn, R. J., & Cupples, L. A. (2003). Bias due to missing exposure data using complete-case analysis in the proportional hazards regression model. *Statistics in Medicine*, *22*(4), 545–557. https://doi.org/10.1002/sim.1340

Duan, Manning, W., Morris, C., & Newhouse, J. (1985). Comments on Selectivity Bias. *Advances in Health Economics and Health Services Research*, *6*, 19–24.

Duan, N., Manning, W. G., Morris, C. N., Newhouse, J. P., Duan, N., Manning, W. G., … Monica, S. (2017). A Comparison of Alternative Models for the Demand for Medical Care A Comparison of Alternative Models for the Demand for Medical Care, *0015*(March), 115–

126. https://doi.org/10.1080/07350015.1983.10509330

Duan Naihua, et al. (1984). Choosing between the Sample-Selection Model and the Multi-part Model. *Journal of Business & Economic Statistics*, *2*(3), 283–289. Retrieved from http://ideas.repec.org/a/bes/jnlbes/v2y1984i3p283-89.html

Durrant, G. B., & Skinner, C. (2006). Using data augmentation to correct for non-ignorable non-response when surrogate data are available: An application to the distribution of hourly pay. *Journal of the Royal Statistical Society. Series A: Statistics in Society*, *169*(3), 605–623. https://doi.org/10.1111/j.1467-985X.2006.00398.x

Enders, C. K. (2017). Multiple imputation as a flexible tool for missing data handling in clinical research. *Behaviour Research and Therapy*, *98*, 4–18. https://doi.org/10.1016/j.brat.2016.11.008

Galimard, J. E., Chevret, S., Protopopescu, C., & Resche-Rigon, M. (2016). A multiple imputation approach for MNAR mechanisms compatible with Heckman's model. *Statistics in Medicine*, *35*(17), 2907–2920. https://doi.org/10.1002/sim.6902

Greenlees, W. S., Reece, J. S., & Ziexhang, K. D. (1982). Imputation of missing values when the probability of nonreponse depends on the variable being imputed. *Journal of the American Statistical Association*, *77*(378), 251–256.

Heckman, J. J. (1979). Sample Selection Bias as a Specification Error. *Econometrica*, *47*(1), 153–161.

Hossain, A., & Pahwa, P. (2010). A Comparison between Standard Regression and Multilevel Modeling Techniques based on Longitudinal Complex Survey. *Journal of Statistics and Applications*, *5*(3), 243–262.

Im, J., & Kim, S. (2017). Multiple imputation for nonignorable missing data. *Journal of the Korean Statistical Society*, 1–10. https://doi.org/10.1016/j.jkss.2017.05.001

Kenward, M. G., & Molenberghs, G. (2007). *Missing Data in Clinical Studies*. West Sussex, England: John Wiley & Sons, Ltd.

Kim, J. K. (2011). Parametric fractional imputation for missing data analysis. *Biometrika*, *98*(1), 119–132. https://doi.org/10.1093/biomet/asq073

Knol, M. J., Janssen, K. J. M., Donders, A. R. T., Egberts, A. C. G., Heerdink, E. R., Grobbee, D. E., … Geerlings, M. I. (2010). Unpredictable bias when using the missing indicator method or complete case analysis for missing confounder values: an empirical example. *Journal of Clinical Epidemiology*, *63*(7), 728–736. https://doi.org/10.1016/j.jclinepi.2009.08.028

Lee, K. J., & Carlin, J. B. (2010). Multiple imputation for missing data: Fully conditional specification versus multivariate normal imputation. *American Journal of Epidemiology*, *171*(5), 624–632. https://doi.org/10.1093/aje/kwp425

Lee, M. (1996). Nonparametric Two-Stage Estimation of Simultaneous Equations with Limited Endogenous Regressors Author (s): Myoung-jae Lee Stable URL : https://www.jstor.org/stable/3532833 NONPARAMETRIC TWO-STAGE ESTIMATION OF SIMULTANEOUS EQUATIONS WITH LIMITED ENDOGE, *12*(2), 305–330.

Little, R. J. A., & Rubin, D. B. (2002). *Statistical Analysis with Missing Data* (2nd ed.). Hoboken, NJ: Wiley-Interscience.

Marchenko, Y. V., & Genton, M. G. (2012). A heckman selection-t model. *Journal of the American Statistical Association*, *107*(497), 304–317. https://doi.org/10.1080/01621459.2012.656011

Masconi, K. L., Matsha, T. E., Erasmus, R. T., & Kengne, A. P. (2015). Effects of different missing data imputation techniques on the performance of undiagnosed diabetes risk prediction models in a mixed-ancestry population of South Africa. *PLoS ONE*, *10*(9), 1–13. https://doi.org/10.1371/journal.pone.0139210

Mühlenbruch, K., Kuxhaus, O., di Giuseppe, R., Boeing, H., Weikert, C., & Schulze, M. B. (2017). Multiple imputation was a valid approach to estimate absolute risk from a prediction model based on case–cohort data. *Journal of Clinical Epidemiology*, *84*, 130–141. https://doi.org/10.1016/j.jclinepi.2016.12.019

Newey, W. K., Powell, J. L., & Walker, J. R. (1990). American Economic Association Semiparametric Estimation of Selection Models : Some Empirical Results Author (s): Whitney K . Newey , James L . Powell and James R . Walker Source : The American Economic Review , Vol . 80 , No . 2 , Papers and Proceedings, *80*(2), 324–328.

Newhouse, J. P. (2005). RAND Health Insurance Experiment [in Metropolitan and Non-

Metropolitan Areas of the United States], 1974-1982. Retrieved November 10, 2018, from https://www.icpsr.umich.edu/icpsrweb/NACDA/studies/6439

Nocedal, J., & Wright, S. (2000). *Numerical Optimization*. New York: Springer-Verlag.

Ogundimu, E. O., & Collins, G. S. (2017). A robust imputation method for missing responses and covariates in sample selection models. https://doi.org/10.1177/0962280217715663

Ostmann, A., & Martínez Arbizu, P. (2018). Predictive models using randomForest regression for distribution patterns of meiofauna in Icelandic waters. *Marine Biodiversity*, *48*(2), 719–735. https://doi.org/10.1007/s12526-018-0882-9

Pahwa, P, Karunanayake, C. P., Hagel, L., Janzen, B., Pickett, W., Rennie, D., … Dosman, J. (2012). The Saskatchewan rural health study: an application of a population health framework to understand respiratory health outcomes. *BMC Research Notes*, *5*, 400. https://doi.org/10.1186/1756-0500-5-400

Pahwa, Punam, Rana, M., Pickett, W., Karunanayake, C. P., Amin, K., Rennie, D., … Dosman, J. (2017). Cohort profile: The Saskatchewan Rural Health Study - Adult component. *BMC Research Notes*, *10*(1), 1–7. https://doi.org/10.1186/s13104-017-3047-1

Puhani, P. (2000). The Heckman Correction for Sample Selection and its Critique. *Journal of Economic Surveys*, *14*(1), 53–68.

Qin, J., Leung, D., & Shao, J. (2002). Estimation With Survey Data Under Nonignorable Nonresponse or Informative Sampling. *Journal of the American Statistical Association*, *97*(February 2015), 193–200. https://doi.org/10.1198/016214502753479338

Riddles, M. K., Kim, J. K., & Im, J. (2016). A propensity-score-adjustment method for nonignorable nonresponse. *Journal of Survey Statistics and Methodology*, *4*(2), 215–245. https://doi.org/10.1093/jssam/smv047

Rubin, D. B. (1976). Inference and Missing Data. *Biometrika*, *63*(3), 581–592.

Rubin, D. B. (1978). Multiple imputation in sample surveys - a phenomenological Bayesian approach to nonresponse. *Proceedings of the Survey Resarch Methods Section of the American Statistical Association*, 20–34.

Rubin, D. B. (1987). *Multiple Imputations for Nonresponse in Surveys*. New York: John Wiley &

Sons, Inc.

Rubin, D. B. (1996). Multiple Imputation after 18+ Years. *Journal of the American Statistical Association*, *91*(434), 473–489. https://doi.org/10.1080/01621459.1996.10476908

Schafer, J. L., & Graham, J. W. (2002). Missing data: Our view of the state of the art. *Psychological Methods*, *7*(2), 147–177. https://doi.org/10.1037//1082-989X.7.2.147

Sterne, J. A. C., White, I. R., Carlin, J. B., Spratt, M., Royston, P., Kenward, M. G., … Carpenter, J. R. (2009). Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls. *Bmj*, *338*(jun29 1), b2393–b2393. https://doi.org/10.1136/bmj.b2393

Tang, F., & Ishwaran, H. (2017). Random forest missing data algorithms. *Statistical Analysis and Data Mining*, *10*(6), 363–377. https://doi.org/10.1002/sam.11348

Tseng, C., & Chen, Y.-H. (2017). Regularized approach for data missing not at random. *Statistical Methods in Medical Research*, *0*(0), 1–17. https://doi.org/10.1177/0962280217717760

Van Buuren, S., Boshuizen, H. C., & Knook, D. L. (1999). Multiple imputation of missing blood pressure covariates in survival analysis. *Statistics in Medicine*, *18*(6), 681–694. https://doi.org/10.1002/(SICI)1097-0258(19990330)18:6<681::AID-SIM71>3.0.CO;2-R

van Buuren, Stef. (2012). *Flexible Imputation of Missing Data*. Boca Raton, FL: Chapman & Hall/CRC.

White, I. R., Royston, P., & Wood, A. M. (2011). Multiple imputation using chained equations: Issues and guidance for practice. *Statistics in Medicine*, *30*(4), 377–399. https://doi.org/10.1002/sim.4067

Yoon, J., Jordon, J., & van der Schaar, M. (2018). GAIN: Missing Data Imputation using Generative Adversarial Nets. *ArXiv*.

Zhang, Z. (2016). Missing data imputation: focusing on single imputation. *Annals of Translational Medicine*, *4*(1), 9. https://doi.org/10.3978/j.issn.2305-5839.2015.12.38

APPENDIX A:

Results of Application

A.1 Results of the Outcome Models For Complete Datasets

Table A1. Results of the outcome model for the individuals in RANDHIE with complete data

	Est.	S.E.	t-value	p-value
Intercept	3.45	0.19	18.02	<0.0001
logc	-0.06	0.01	-5.72	<0.0001
linc	0.07	0.02	3.63	0.0003
lfam	-0.18	0.04	-4.16	<0.0001
xage	0.01	0.00	5.55	<0.0001
female	0.39	0.05	7.27	<0.0001
child	-0.20	0.09	-2.30	0.0218

Table A2. Results of the outcome model for the individuals in SRHS with complete data

	Est.	S.E.	t-value	p-value
Intercept	5.81	0.09	65.56	< 0.000 1
Age	-0.03	0	-31.73	< 0.0001
BMI	-0.01	0	-4.56	< 0.0001
Sex	-1.02	0.03	-38.3	< 0.0001
Packyears	-0.01	0	-7.35	< 0.0001
Livestock	0.05	0.03	2.01	0.0449

A.2 15% Missing, RANDHIE

Table A3. Results of applying CC, Rubin's MI, Galimard et. al's MI and Ogundimu & Collins' MI to RANDHIE data with 15% MCAR data

MCAR	Complete Case		Rubin's MI		Galimard's MI		Ogundimu's MI	
	Est.	95% CI	Est.	95% CI	Est.	95% CI	Est.	95% CI
Intercept	3.41	(3.00, 3.82)	3.41	(2.95, 3.88)	3.41	(2.93, 3.89)	3.15	(2.63, 3.66)
Logc	-0.05	(-0.08, -0.03)	-0.06	(-0.08, -0.03)	-0.05	(-0.08, -0.03)	-0.06	(-0.08, -0.03)
Linc	0.09	(0.05, 0.13)	0.09	(0.04, 0.14)	0.09	(0.04, 0.14)	0.09	(0.04, 0.15)
Lfam	-0.19	(-0.28, -0.10)	-0.2	(-0.31, -0.09)	-0.19	(-0.30, -0.08)	-0.21	(-0.33, -0.08)
Xage	0.01	(0.00, 0.01)	0.01	(0.00, 0.01)	0.01	(0.00, 0.01)	0.01	(0.00, 0.01)
Female	0.36	(0.25, 0.48)	0.37	(0.23, 0.51)	0.36	(0.23, 0.50)	0.38	(0.24, 0.53)
Child	-0.29	(-0.47, -0.11)	-0.29	(-0.50, -0.08)	-0.29	(-0.51, -0.08)	-0.29	(-0.51, -0.06)

Table A4. Results of applying CC, Rubin's MI, Galimard et. al's MI and Ogundimu & Collins' MI to RANDHIE data with 15% MAR data

MAR	Complete Case		Rubin's MI		Galimard's MI		Ogundimu's MI	
	Est.	95% CI	Est.	95% CI	Est.	95% CI	Est.	95% CI
Intercept	3.33	(2.89, 3.77)	3.32	(2.87, 3.77)	3.33	(2.81, 3.85)	3.19	(2.69, 3.70)
Logc	-0.06	(-0.08, -0.03)	-0.06	(-0.09, -0.03)	-0.06	(-0.09, -0.03)	-0.07	(-0.09, -0.04)
Linc	0.11	(0.06, 0.16)	0.11	(0.06, 0.16)	0.11	(0.06, 0.17)	0.08	(0.03, 0.13)
Lfam	-0.21	(-0.31, -0.11)	-0.22	(-0.34, -0.10)	-0.21	(-0.32, -0.09)	-0.16	(-0.26, -0.05)
Xage	0.01	(0.00, 0.01)	0.01	(0.00, 0.01)	0.01	(0.00, 0.01)	0.02	(0.01, 0.02)
Female	0.30	(0.18, 0.43)	0.32	(0.17, 0.46)	0.30	(0.16, 0.45)	0.38	(0.25, 0.50)
Child	-0.35	(-0.55, -0.15)	-0.31	(-0.55, -0.07)	-0.35	(-0.59, -0.11)	-0.37	(-0.61, -0.14)

Table A5. Results of applying CC, Rubin's MI, Galimard et. al's MI and Ogundimu & Collins' MI to RANDHIE data with 15% MNAR data

MNAR	Complete Case		Rubin's MI		Galimard's MI		Ogundimu's MI	
	Est.	95% CI	Est.	95% CI	Est.	95% CI	Est.	95% CI
Intercept	3.46	(2.93, 3.99)	3.43	(2.80, 4.06)	3.46	(2.84, 4.09)	3.36	(2.84, 3.88)
logc	-0.05	(-0.07, -0.02)	-0.05	(-0.08, -0.02)	-0.05	(-0.08, -0.01)	-0.07	(-0.10, -0.04)
linc	0.06	(0.00, 0.11)	0.06	(0.00, 0.12)	0.06	(-0.01, 0.12)	0.07	(0.01, 0.12)
lfam	-0.15	(-0.27, -0.02)	-0.15	(-0.30, 0.00)	-0.15	(-0.29, 0.00)	-0.17	(-0.29, -0.05)
xage	0.01	(0.01, 0.02)	0.01	(0.00, 0.02)	0.01	(0.01, 0.02)	0.01	(0.01, 0.02)
female	0.43	(0.28, 0.58)	0.43	(0.26, 0.60)	0.43	(0.26, 0.61)	0.40	(0.26, 0.54)
child	-0.11	(-0.34, 0.12)	-0.09	(-0.37, 0.20)	-0.11	(-0.38, 0.16)	-0.24	(-0.47, -0.01)

A.3 15% Missing, SRHS

Table A6. Results of applying CC, Rubin's MI, Galimard et. al's MI and Ogundimu & Collins' MI to SRHSdata with 15% MCAR data

MCAR	Complete Case		Rubin's MI		Galimard's MI		Ogundimu's MI	
	Est.	95% CI	Est.	95% CI	Est.	95% CI	Est.	95% CI
Intercept	5.78	(5.59, 5.97)	5.76	(5.54, 5.98)	5.78	(5.56, 6.00)	5.89	(5.68, 6.11)
Age	-0.03	(-0.04, -0.03)	-0.03	(-0.04, -0.03)	-0.03	(-0.04, -0.03)	-0.03	(-0.04, -0.03)
BMI	-0.01	(-0.01, 0.00)	-0.01	(-0.02, 0.00)	-0.01	(-0.02, 0.00)	-0.01	(-0.02, -0.01)
Sex	-1.02	(-1.08, -0.97)	-1.02	(-1.09, -0.96)	-1.02	(-1.09, -0.96)	-1.06	(-1.13, -1.00)
Packyears	-0.01	(-0.01, -0.01)	-0.01	(-0.01, 0.00)	-0.01	(-0.01, 0.00)	-0.01	(-0.01, -0.01)
Livestock	0.05	(0.00, 0.11)	0.05	(-0.02, 0.12)	0.05	(-0.01, 0.12)	0.03	(-0.03, 0.10)

Table A7. Results of applying CC, Rubin's MI, Galimard et. al's MI and Ogundimu & Collins' MI to SRHS data with 15% MAR data

MAR	Complete Case		Rubin's MI		Galimard's MI		Ogundimu's MI	
	Est.	95% CI	Est.	95% CI	Est.	95% CI	Est.	95% CI
Intercept	5.71	(5.51, 5.91)	5.73	(5.51, 5.95)	5.71	(5.47, 5.95)	6.00	(5.75, 6.24)
Age	-0.03	(-0.04, -0.03)	-0.03	(-0.04, -0.03)	-0.03	(-0.04, -0.03)	-0.04	(-0.04, -0.03)
BMI	-0.01	(-0.01, 0.00)	-0.01	(-0.01, 0.00)	-0.01	(-0.01, 0.00)	-0.01	(-0.02, -0.01)
Sex	-1.03	(-1.09, -0.97)	-1.05	(-1.11, -0.98)	-1.03	(-1.10, -0.96)	-1.05	(-1.11, -0.98)
Packyears	-0.01	(-0.01, -0.01)	-0.01	(-0.01, -0.01)	-0.01	(-0.01, 0.00)	-0.01	(-0.01, 0.00)
Livestock	0.06	(0.00, 0.12)	0.05	(-0.01, 0.12)	0.06	(-0.01, 0.13)	0.03	(-0.03, 0.09)

Table A8. Results of applying CC, Rubin's MI, Galimard et. al's MI and Ogundimu & Collins' MI to SRHS data with 15% MNAR data

MNAR	Complete Case		Rubin's MI		Galimard's MI		Ogundimu's MI	
	Est.	95% CI	Est.	95% CI	Est.	95% CI	Est.	95% CI
Intercept	5.83	(5.58, 6.08)	5.87	(5.52, 6.21)	5.83	(5.54, 6.12)	5.85	(5.62, 6.08)
Age	-0.03	(-0.04, -0.03)	-0.03	(-0.04, -0.03)	-0.03	(-0.04, -0.03)	-0.03	(-0.03, -0.03)
BMI	-0.01	(-0.02, 0.00)	-0.01	(-0.02, 0.00)	-0.01	(-0.02, 0.00)	-0.01	(-0.02, -0.01)
Sex	-1.03	(-1.10, -0.95)	-1.03	(-1.11, -0.95)	-1.03	(-1.11, -0.94)	-1.04	(-1.10, -0.97)
Packyears	-0.01	(-0.01, 0.00)	-0.01	(-0.01, 0.00)	-0.01	(-0.01, 0.00)	-0.01	(-0.01, 0.00)
Livestock	0.05	(-0.02, 0.13)	0.06	(-0.04, 0.16)	0.05	(-0.03, 0.14)	0.01	(-0.06, 0.07)

A.4 30% Missing, RANDHIE

Table A9. Results of applying CC, Rubin's MI, Galimard et. al's MI and Ogundimu & Collins' MI to RANDHIE data with 30% MCAR data

MCAR	Complete Case		Rubin's MI		Galimard's MI		Ogundimu's MI	
	Est.	95% CI	Est.	95% CI	Est.	95% CI	Est.	95% CI
Intercept	3.45	(3.03, 3.87)	3.48	(2.98, 3.98)	3.45	(2.96, 3.94)	2.88	(2.21, 3.55)
Logc	-0.06	(-0.08, -0.04)	-0.06	(-0.09, -0.04)	-0.06	(-0.09, -0.04)	-0.06	(-0.09, -0.03)
Linc	0.06	(0.02, 0.11)	0.06	(0.01, 0.12)	0.06	(0.01, 0.12)	0.11	(0.04, 0.18)
Lfam	-0.16	(-0.25, -0.07)	-0.16	(-0.27, -0.06)	-0.16	(-0.27, -0.06)	-0.24	(-0.38, -0.11)
Xage	0.01	(0.01, 0.02)	0.01	(0.01, 0.02)	0.01	(0.01, 0.02)	0.01	(0.00, 0.01)
Female	0.39	(0.28, 0.49)	0.39	(0.26, 0.51)	0.39	(0.26, 0.51)	0.32	(0.15, 0.48)
Child	-0.18	(-0.37, 0.00)	-0.20	(-0.40, 0.01)	-0.18	(-0.40, 0.03)	-0.39	(-0.67, -0.11)

Table A10. Results of applying CC, Rubin's MI, Galimard et. al's MI and Ogundimu & Collins' MI to RANDHIE data with 30% MAR data

MAR	Complete Case		Rubin's MI		Galimard's MI		Ogundimu's MI	
	Est.	95% CI	Est.	95% CI	Est.	95% CI	Est.	95% CI
Intercept	3.43	(2.97, 3.88)	3.41	(2.84, 3.97)	3.43	(2.89, 3.96)	2.70	(2.07, 3.33)
Logc	-0.07	(-0.09, -0.04)	-0.06	(-0.09, -0.04)	-0.07	(-0.09, -0.04)	-0.07	(-0.10, -0.04)
Linc	0.07	(0.02, 0.12)	0.08	(0.02, 0.14)	0.07	(0.02, 0.13)	0.09	(0.03, 0.15)
Lfam	-0.15	(-0.25, -0.06)	-0.17	(-0.28, -0.06)	-0.15	(-0.27, -0.04)	-0.16	(-0.28, -0.04)
Xage	0.01	(0.01, 0.02)	0.01	(0.01, 0.02)	0.01	(0.01, 0.02)	0.02	(0.02, 0.03)
Female	0.36	(0.25, 0.47)	0.36	(0.22, 0.49)	0.36	(0.23, 0.49)	0.33	(0.18, 0.47)
Child	-0.23	(-0.44, -0.02)	-0.23	(-0.47, 0.01)	-0.23	(-0.47, 0.01)	-0.49	(-0.75, -0.24)

Table A11. Results of applying CC, Rubin's MI, Galimard et. al's MI and Ogundimu & Collins' MI to RANDHIE data with 30% MNAR data

MNAR	Complete Case		Rubin's MI		Galimard's MI		Ogundimu's MI	
	Est.	95% CI	Est.	95% CI	Est.	95% CI	Est.	95% CI
Intercept	3.59	(3.00, 4.17)	3.60	(2.79, 4.41)	3.59	(2.89, 4.28)	3.22	(2.52, 3.92)
Logc	-0.06	(-0.08, -0.03)	-0.06	(-0.09, -0.03)	-0.06	(-0.09, -0.02)	-0.07	(-0.10, -0.04)
Linc	0.06	(0.00, 0.13)	0.06	(-0.02, 0.15)	0.06	(-0.01, 0.14)	0.06	(-0.01, 0.13)
Lfam	-0.16	(-0.27, -0.04)	-0.14	(-0.27, -0.01)	-0.16	(-0.30, -0.02)	-0.17	(-0.31, -0.04)
Xage	0.01	(0.01, 0.02)	0.01	(0.00, 0.02)	0.01	(0.00, 0.02)	0.01	(0.01, 0.02)
Female	0.26	(0.12, 0.40)	0.23	(0.09, 0.38)	0.26	(0.10, 0.42)	0.46	(0.28, 0.63)
Child	-0.30	(-0.54, -0.06)	-0.28	(-0.56, -0.01)	-0.30	(-0.59, -0.02)	-0.28	(-0.54, -0.02)

A.5 30% Missing, SRHS

Table A12. Results of applying CC, Rubin's MI, Galimard et. al's MI and Ogundimu & Collins' MI to SRHS data with 30% MCAR data

MCAR	Complete Case		Rubin's MI		Galimard's MI		Ogundimu's MI	
	Est.	95% CI	Est.	95% CI	Est.	95% CI	Est.	95% CI
Intercept	5.80	(5.60, 5.99)	5.81	(5.58, 6.04)	5.80	(5.57, 6.03)	5.82	(5.53, 6.12)
Age	-0.03	(-0.04, -0.03)	-0.03	(-0.04, -0.03)	-0.03	(-0.04, -0.03)	-0.03	(-0.04, -0.03)
BMI	-0.01	(-0.02, -0.01)	-0.01	(-0.02, 0.00)	-0.01	(-0.02, 0.00)	-0.01	(-0.02, 0.00)
Sex	-1.01	(-1.06, -0.95)	-1.01	(-1.08, -0.95)	-1.01	(-1.07, -0.94)	-1.07	(-1.14, -0.99)
Packyears	-0.01	(-0.01, -0.01)	-0.01	(-0.01, -0.01)	-0.01	(-0.01, -0.01)	-0.01	(-0.01, -0.01)
Livestock	0.05	(-0.01, 0.11)	0.05	(-0.01, 0.11)	0.05	(-0.02, 0.12)	0.03	(-0.03, 0.10)

Table A13. Results of applying CC, Rubin's MI, Galimard et. al's MI and Ogundimu & Collins' MI to SRHS data with 30% MAR data

MAR	Complete Case		Rubin's MI		Galimard's MI		Ogundimu's MI	
	Est.	95% CI	Est.	95% CI	Est.	95% CI	Est.	95% CI
Intercept	5.86	(5.64, 6.08)	5.87	(5.65, 6.10)	5.86	(5.60, 6.11)	5.82	(5.53, 6.12)
Age	-0.03	(-0.04, -0.03)	-0.03	(-0.04, -0.03)	-0.03	(-0.04, -0.03)	-0.03	(-0.04, -0.03)
BMI	-0.01	(-0.02, -0.01)	-0.01	(-0.02, -0.01)	-0.01	(-0.02, -0.01)	-0.01	(-0.02, 0.00)
Sex	-1.01	(-1.07, -0.94)	-1.02	(-1.10, -0.94)	-1.01	(-1.08, -0.93)	-1.07	(-1.14, -0.99)
Packyears	-0.01	(-0.01, -0.01)	-0.01	(-0.01, -0.01)	-0.01	(-0.01, 0.00)	-0.01	(-0.01, -0.01)
Livestock	0.04	(-0.03, 0.10)	0.03	(-0.05, 0.10)	0.04	(-0.04, 0.11)	0.03	(-0.03, 0.10)

Table A14. Results of applying CC, Rubin's MI, Galimard et. al's MI and Ogundimu & Collins' MI to SRHS data with 30% MNAR data

MNAR	Complete Case		Rubin's MI		Galimard's MI		Ogundimu's MI	
	Est.	95% CI	Est.	95% CI	Est.	95% CI	Est.	95% CI
Intercept	5.89	(5.64, 6.14)	5.89	(5.57, 6.21)	5.89	(5.60, 6.18)	5.86	(5.62, 6.10)
Age	-0.04	(-0.04, -0.03)	-0.04	(-0.04, -0.03)	-0.04	(-0.04, -0.03)	-0.03	(-0.04, -0.03)
BMI	-0.01	(-0.02, 0.00)	-0.01	(-0.02, 0.00)	-0.01	(-0.02, 0.00)	-0.01	(-0.02, 0.00)
Sex	-1.00	(-1.08, -0.93)	-1.01	(-1.09, -0.92)	-1.00	(-1.09, -0.92)	-1.06	(-1.15, -0.97)
Packyears	-0.01	(-0.01, -0.01)	-0.01	(-0.01, -0.01)	-0.01	(-0.01, -0.01)	-0.01	(-0.01, 0.00)
Livestock	0.05	(-0.02, 0.12)	0.03	(-0.06, 0.11)	0.05	(-0.04, 0.13)	0.01	(-0.08, 0.11)

A.6 50% Missing, RANDHIE

Table A15. Results of applying CC, Rubin's MI, Galimard et. al's MI and Ogundimu & Collins' MI to RANDHIE data with 50% MCAR data

MCAR	Complete Case		Rubin's MI		Galimard's MI		Ogundimu's MI	
	Est.	95% CI	Est.	95% CI	Est.	95% CI	Est.	95% CI
Intercept	3.41	(3.00, 3.82)	3.39	(2.94, 3.84)	3.41	(2.93, 3.89)	2.28	(1.58, 2.98)
Logc	-0.05	(-0.08, -0.03)	-0.05	(-0.08, -0.03)	-0.05	(-0.08, -0.03)	-0.06	(-0.10, -0.02)
Linc	0.09	(0.05, 0.13)	0.09	(0.04, 0.14)	0.09	(0.04, 0.14)	0.08	(0.01, 0.16)
Lfam	-0.19	(-0.28, -0.10)	-0.20	(-0.30, -0.09)	-0.19	(-0.30, -0.08)	-0.14	(-0.32, 0.04)
Xage	0.01	(0.00, 0.01)	0.01	(0.00, 0.01)	0.01	(0.00, 0.01)	0.01	(0.01, 0.02)
Female	0.36	(0.25, 0.48)	0.37	(0.25, 0.50)	0.36	(0.23, 0.50)	0.44	(0.23, 0.65)
Child	-0.29	(-0.47, -0.11)	-0.28	(-0.48, -0.07)	-0.29	(-0.51, -0.08)	0.03	(-0.31, 0.36)

Table A16. Results of applying CC, Rubin's MI, Galimard et. al's MI and Ogundimu & Collins' MI to RANDHIE data with 50% MAR data

MAR	Complete Case		Rubin's MI		Galimard's MI		Ogundimu's MI	
	Est.	95% CI	Est.	95% CI	Est.	95% CI	Est.	95% CI
Intercept	3.84	(3.38, 4.29)	3.78	(3.13, 4.42)	3.84	(3.30, 4.37)	2.16	(1.16, 3.16)
logc	-0.05	(-0.08, -0.03)	-0.05	(-0.08, -0.02)	-0.05	(-0.08, -0.03)	-0.06	(-0.10, -0.03)
linc	0.05	(0.01, 0.10)	0.05	(-0.01, 0.12)	0.05	(0.00, 0.11)	0.10	(0.02, 0.17)
lfam	-0.13	(-0.23, -0.03)	-0.13	(-0.24, -0.01)	-0.13	(-0.25, -0.02)	-0.13	(-0.29, 0.02)
xage	0.01	(0.00, 0.01)	0.01	(0.00, 0.02)	0.01	(0.00, 0.01)	0.02	(0.01, 0.03)
female	0.42	(0.30, 0.54)	0.44	(0.29, 0.58)	0.42	(0.28, 0.56)	0.22	(0.03, 0.41)
child	-0.24	(-0.44, -0.04)	-0.22	(-0.47, 0.02)	-0.24	(-0.47, -0.01)	-0.27	(-0.59, 0.05)

Table A17. Results of applying CC, Rubin's MI, Galimard et. al's MI and Ogundimu & Collins' MI to RANDHIE data with 50% MNAR data

MNAR	Complete Case		Rubin's MI		Galimard's MI		Ogundimu's MI	
	Est.	95% CI	Est.	95% CI	Est.	95% CI	Est.	95% CI
Intercept	3.82	(3.32, 4.33)	3.77	(3.15, 4.40)	3.82	(3.23, 4.42)	2.47	(1.50, 3.45)
Logc	-0.03	(-0.06, 0.00)	-0.03	(-0.06, 0.00)	-0.03	(-0.06, 0.00)	-0.07	(-0.10, -0.03)
Linc	0.06	(0.00, 0.11)	0.06	(-0.02, 0.13)	0.06	(-0.01, 0.12)	0.07	(-0.02, 0.16)
Lfam	-0.10	(-0.22, 0.01)	-0.09	(-0.23, 0.04)	-0.10	(-0.24, 0.04)	-0.13	(-0.32, 0.07)
Xage	0.01	(0.00, 0.02)	0.01	(0.00, 0.02)	0.01	(0.00, 0.02)	0.02	(0.01, 0.02)
Female	0.31	(0.17, 0.46)	0.35	(0.18, 0.52)	0.31	(0.14, 0.48)	0.45	(0.25, 0.66)
Child	-0.42	(-0.65, -0.18)	-0.40	(-0.73, -0.06)	-0.42	(-0.70, -0.14)	-0.29	(-0.60, 0.02)

A.7 50% Missing, SRHS

Table A18. Results of applying CC, Rubin's MI, Galimard et. al's MI and Ogundimu & Collins' MI to SRHS data with 50% MCAR data

MCAR	Complete Case		Rubin's MI		Galimard's MI		Ogundimu's MI	
	Est.	95% CI	Est.	95% CI	Est.	95% CI	Est.	95% CI
Intercept	5.81	(5.62, 6.00)	5.82	(5.58, 6.05)	5.81	(5.59, 6.04)	5.86	(5.54, 6.19)
Age	-0.03	(-0.04, -0.03)	-0.03	(-0.04, -0.03)	-0.03	(-0.04, -0.03)	-0.03	(-0.04, -0.03)
BMI	-0.01	(-0.02, -0.01)	-0.01	(-0.02, -0.01)	-0.01	(-0.02, -0.01)	-0.01	(-0.02, 0.00)
Sex	-1.04	(-1.10, -0.99)	-1.04	(-1.11, -0.98)	-1.04	(-1.11, -0.98)	-1.08	(-1.18, -0.99)
Packyears	-0.01	(-0.01, -0.01)	-0.01	(-0.01, 0.00)	-0.01	(-0.01, 0.00)	-0.01	(-0.01, 0.00)
Livestock	0.04	(-0.01, 0.10)	0.04	(-0.02, 0.11)	0.04	(-0.02, 0.11)	0.03	(-0.04, 0.10)

Table A19. Results of applying CC, Rubin's MI, Galimard et. al's MI and Ogundimu & Collins' MI to SRHS data with 50% MAR data

MAR	Complete Case		Rubin's MI		Galimard's MI		Ogundimu's MI	
	Est.	95% CI	Est.	95% CI	Est.	95% CI	Est.	95% CI
Intercept	5.88	(5.67, 6.10)	5.91	(5.68, 6.13)	5.88	(5.63, 6.13)	5.84	(5.32, 6.35)
Age	-0.03	(-0.04, -0.03)	-0.03	(-0.04, -0.03)	-0.03	(-0.04, -0.03)	-0.03	(-0.04, -0.03)
BMI	-0.01	(-0.02, -0.01)	-0.01	(-0.02, -0.01)	-0.01	(-0.02, 0.00)	-0.01	(-0.02, 0.00)
Sex	-1.06	(-1.12, -1.00)	-1.06	(-1.13, -0.99)	-1.06	(-1.13, -0.98)	-1.04	(-1.12, -0.96)
Packyears	-0.01	(-0.01, 0.00)	-0.01	(-0.01, 0.00)	-0.01	(-0.01, 0.00)	-0.01	(-0.01, -0.01)
Livestock	0.03	(-0.03, 0.09)	0.03	(-0.05, 0.10)	0.03	(-0.05, 0.10)	0.02	(-0.06, 0.10)

Table A20. Results of applying CC, Rubin's MI, Galimard et. al's MI and Ogundimu & Collins' MI to SRHS data with 50% MNAR data

MNAR	Complete Case		Rubin's MI		Galimard's MI		Ogundimu's MI	
	Est.	95% CI	Est.	95% CI	Est.	95% CI	Est.	95% CI
Intercept	5.82	(5.58, 6.06)	5.87	(5.60, 6.14)	5.82	(5.54 6.10)	5.80	(5.52, 6.08)
Age	-0.04	(-0.04, -0.03)	-0.04	(-0.04, -0.03)	-0.04	(-0.04 -0.03)	-0.04	(-0.04, -0.03)
BMI	-0.01	(-0.01, 0.00)	-0.01	(-0.02, 0.00)	-0.01	(-0.01 0.00)	-0.01	(-0.01, 0.00)
Sex	-1.00	(-1.07, -0.92)	-1.01	(-1.10, -0.91)	-1.00	(-1.08 -0.91)	-1.01	(-1.08, -0.93)
Packyears	-0.01	(-0.01, -0.01)	-0.01	(-0.01, 0.00)	-0.01	(-0.01 0.00)	-0.01	(-0.01, -0.01)
Livestock	0.09	(0.01, 0.16)	0.08	(0.00, 0.17)	0.09	(0.00 0.18)	0.08	(0.01, 0.16)

APPENDIX B:

R-Code for Study

B.1 Creating Datasets with Missing Data

```
#######################
#RANDHIE              #
#######################

#Data Cleaning and rearranging

install.packages("sampleSelection")
library(sampleSelection)
data(package = "sampleSelection")
data(RANDHIE)
RANDHIE_baseline <- RANDHIE[RANDHIE$year==1,] #5638 individuals, 4451 observed
individuals
RANDHIE_baseline_data <- data.frame(RANDHIE_baseline$lnmeddol,
RANDHIE_baseline$logc, RANDHIE_baseline$disea,RANDHIE_baseline$linc,
RANDHIE_baseline$lfam, RANDHIE_baseline$xage,
RANDHIE_baseline$female,RANDHIE_baseline$child, RANDHIE_baseline$educdec,
RANDHIE_baseline$idp)
names(RANDHIE_baseline_data) <- c("lnmeddol", "logc", "disea", "linc", "lfam", "xage",
"female", “child”, "educdec", "idp")
RANDHIE_baseline_data_complete <-
RANDHIE_baseline_data[!is.na(RANDHIE_baseline_data$lnmeddol), ]
attach(RANDHIE_baseline_data_complete)

write.csv(RANDHIE_baseline_data_complete, "C:/Users/April/Google
Drive/Thesis/FinalCodes/DatasetsWithMissing/RANDHIE_baseline_data_complete.csv")

#Outcome and Selection Model Regressions
selection <- ry1~ logc +  disea +
  linc + lfam  + xage + female +
  child  + educdec + idp

OutcomeModel <- lm(outcome, data = RANDHIE_baseline_data_complete)
summary(OutcomeModel)

outcome <- lnmeddol ~logc + disea +
```

```
  linc + lfam + xage + female +
  child

#Introducing MCAR, MAR, and MNAR data into
#Missing 15%
RANDHIE_baseline_data_complete$lnmeddol15.mcar <-
RANDHIE_baseline_data_complete$lnmeddol
RANDHIE_baseline_data_complete$lnmeddol15.mar <-
RANDHIE_baseline_data_complete$lnmeddol
RANDHIE_baseline_data_complete$lnmeddol15.mnar <-
RANDHIE_baseline_data_complete$lnmeddol
set.seed(1234)
mcar <- rbinom(n = 4451, size = 1, prob = 1/(1+exp(1-2.75)))
sum(mcar)/length(mcar)
mar <- rbinom(n = 4451, size = 1, prob = 1/(1+exp(1-0.19*xage)))
sum(mar)/length(mar)
mnar <- rbinom(n = 4451, size = 1, prob = 1/(1+exp(1-0.75*lnmeddol)))
sum(mnar)/length(mnar)


mcar_vec <- rep(NA, 1000)

for (i in 1:1000){
  mcar<- rbinom(n = 4451, size = 1, prob = 1/(1+exp(1-2.75)))
  p <- sum(mcar)/length(mcar)
  mcar_vec[i] <- p
}
hist(mcar_vec)

mar_vec <- rep(NA, 1000)

for (i in 1:1000){
  mar <- rbinom(n = 4451, size = 1, prob = 1/(1+exp(1-0.19*xage)))
  p <- sum(mar)/length(mar)
  mar_vec[i] <- p
}
hist(mar_vec)

mnar_vec <- rep(NA, 1000)

for (i in 1:1000){
  mnar <- rbinom(n = 4451, size = 1, prob = 1/(1+exp(1-0.75*lnmeddol)))
  p <- sum(mnar)/length(mnar)
  mnar_vec[i] <- p
}
```

```
hist(mnar_vec)


for (i in 1:4451){
  if (mcar[i]==0){
    RANDHIE_baseline_data_complete$lnmeddol15.mcar[i] <- NA
  }
  if (mar[i]==0){
    RANDHIE_baseline_data_complete$lnmeddol15.mar[i] <- NA
  }
  if (mnar[i]==0){
    RANDHIE_baseline_data_complete$lnmeddol15.mnar[i] <- NA
  }
}

RANDHIE_baseline_data_complete$lnmeddolBMCAR15 <- 0
RANDHIE_baseline_data_complete$lnmeddolBMCAR15[!is.na(RANDHIE_baseline_data_
complete$lnmeddol15.mcar)] <- 1

RANDHIE_baseline_data_complete$lnmeddolBMAR15 <- 0
RANDHIE_baseline_data_complete$lnmeddolBMAR15[!is.na(RANDHIE_baseline_data_c
omplete$lnmeddol15.mar)] <- 1

RANDHIE_baseline_data_complete$lnmeddolBMNAR15 <- 0
RANDHIE_baseline_data_complete$lnmeddolBMNAR15[!is.na(RANDHIE_baseline_data
_complete$lnmeddol15.mnar)] <- 1

attach(RANDHIE_baseline_data_complete)

#Creating new MCAR, MAR, and MNAR 15% alternatives of the RANDHIE dataset
#Write them into csv datasets

RANDHIE_baseline_data_complete_mcar15 <- data.frame(lnmeddol15.mcar, logc, disea,
linc, lfam, xage, female, child, educdec, idp, lnmeddolBMCAR15)

names(RANDHIE_baseline_data_complete_mcar15) <- c("lnmeddol", "logc", "disea",
"linc", "lfam", "xage", "female", “child”, "educdec", "idp", "lnmeddolMO")

write.csv(RANDHIE_baseline_data_complete_mcar15, "C:/Users/April/Google
Drive/Thesis/FinalCodes/DatasetsWithMissing/RANDHIE_baseline_data_complete_mcar15.csv
")

RANDHIE_baseline_data_complete_mar15 <- data.frame(lnmeddol15.mar, logc, disea, linc,
lfam, xage, female, child, educdec, idp, lnmeddolBMAR15)
```

```
    names(RANDHIE_baseline_data_complete_mar15) <- c("lnmeddol", "logc", "disea", "linc",
"lfam", "xage", "female", “child”, "educdec", "idp", "lnmeddolMO")

    write.csv(RANDHIE_baseline_data_complete_mar15, "C:/Users/April/Google
Drive/Thesis/FinalCodes/DatasetsWithMissing/RANDHIE_baseline_data_complete_mar15.csv"
)

    RANDHIE_baseline_data_complete_mnar15 <-data.frame(lnmeddol15.mnar, logc, disea,
linc, lfam, xage, female, child, educdec, idp, lnmeddolBMNAR15)

    names(RANDHIE_baseline_data_complete_mnar15) <- c("lnmeddol", "logc", "disea",
"linc", "lfam", "xage", "female", “child”, "educdec", "idp", "lnmeddolMO")

    write.csv(RANDHIE_baseline_data_complete_mnar15, "C:/Users/April/Google
Drive/Thesis/FinalCodes/DatasetsWithMissing/RANDHIE_baseline_data_complete_mnar15.csv
")




    ############################
    # 30% and 50% Missing      #
    ############################


    #30% Missing

    RANDHIE_baseline_data_complete$lnmeddol30.mcar <-
RANDHIE_baseline_data_complete$lnmeddol
    RANDHIE_baseline_data_complete$lnmeddol30.mar <-
RANDHIE_baseline_data_complete$lnmeddol
    RANDHIE_baseline_data_complete$lnmeddol30.mnar <-
RANDHIE_baseline_data_complete$lnmeddol

    set.seed(1234)
    mcar <- rbinom(n = 4451, size = 1, prob = 1/(1+exp(1-1.84)))
    sum(mcar)/length(mcar)
    mar <- rbinom(n = 4451, size = 1, prob = 1/(1+exp(1-0.088*xage)))
    sum(mar)/length(mar)
    mnar <- rbinom(n = 4451, size = 1, prob = 1/(1+exp(1-0.47*lnmeddol)))
    sum(mnar)/length(mnar)


    mcar_vec <- rep(NA, 1000)

    for (i in 1:1000){
```

```
  mcar<- rbinom(n = 4451, size = 1, prob = 1/(1+exp(1-1.84)))
  p <- sum(mcar)/length(mcar)
  mcar_vec[i] <- p
}
hist(mcar_vec)

mar_vec <- rep(NA, 1000)

for (i in 1:1000){
  mar <- rbinom(n = 4451, size = 1, prob = 1/(1+exp(1-0.088*xage)))
  p <- sum(mar)/length(mar)
  mar_vec[i] <- p
}
hist(mar_vec)

mnar_vec <- rep(NA, 1000)

for (i in 1:1000){
  mnar <- rbinom(n = 4451, size = 1, prob = 1/(1+exp(1-0.47*lnmeddol)))
  p <- sum(mnar)/length(mnar)
  mnar_vec[i] <- p
}

hist(mnar_vec)


for (i in 1:4451){
  if (mcar[i]==0){
    RANDHIE_baseline_data_complete$lnmeddol30.mcar[i] <- NA
  }
  if (mar[i]==0){
    RANDHIE_baseline_data_complete$lnmeddol30.mar[i] <- NA
  }
  if (mnar[i]==0){
    RANDHIE_baseline_data_complete$lnmeddol30.mnar[i] <- NA
  }
}

RANDHIE_baseline_data_complete$lnmeddolBMCAR30 <- 0
RANDHIE_baseline_data_complete$lnmeddolBMCAR30[!is.na(RANDHIE_baseline_data_
complete$lnmeddol30.mcar)] <- 1

RANDHIE_baseline_data_complete$lnmeddolBMAR30 <- 0
RANDHIE_baseline_data_complete$lnmeddolBMAR30[!is.na(RANDHIE_baseline_data_c
omplete$lnmeddol30.mar)] <- 1
```

```
RANDHIE_baseline_data_complete$lnmeddolBMNAR30 <- 0
RANDHIE_baseline_data_complete$lnmeddolBMNAR30[!is.na(RANDHIE_baseline_data
_complete$lnmeddol30.mnar)] <- 1

attach(RANDHIE_baseline_data_complete)

#Creating new MCAR, MAR, and MNAR 30% alternatives of the RANDHIE dataset
#Write them into csv datasets

RANDHIE_baseline_data_complete_mcar30 <- data.frame(lnmeddol30.mcar, logc, disea,
linc, lfam, xage, female, child, educdec, idp, lnmeddolBMCAR30)

names(RANDHIE_baseline_data_complete_mcar30) <- c("lnmeddol", "logc", "disea",
"linc", "lfam", "xage", "female", “child”, "educdec", "idp", "lnmeddolMO")

write.csv(RANDHIE_baseline_data_complete_mcar30, "C:/Users/April/Google
Drive/Thesis/FinalCodes/DatasetsWithMissing/RANDHIE_baseline_data_complete_mcar30.csv
")

RANDHIE_baseline_data_complete_mar30 <- data.frame(lnmeddol30.mar, logc, disea, linc,
lfam, xage, female, child, educdec, idp, lnmeddolBMAR30)

names(RANDHIE_baseline_data_complete_mar30) <- c("lnmeddol", "logc", "disea", "linc",
"lfam", "xage", "female", “child”, "educdec", "idp", "lnmeddolMO")


write.csv(RANDHIE_baseline_data_complete_mar30, "C:/Users/April/Google
Drive/Thesis/FinalCodes/DatasetsWithMissing/RANDHIE_baseline_data_complete_mar30.csv"
)

RANDHIE_baseline_data_complete_mnar30 <-data.frame(lnmeddol30.mnar, logc, disea,
linc, lfam, xage, female, child, educdec, idp, lnmeddolBMNAR30)

names(RANDHIE_baseline_data_complete_mnar30) <- c("lnmeddol", "logc", "disea",
"linc", "lfam", "xage", "female", “child”, "educdec", "idp", "lnmeddolMO")


write.csv(RANDHIE_baseline_data_complete_mnar30, "C:/Users/April/Google
Drive/Thesis/FinalCodes/DatasetsWithMissing/RANDHIE_baseline_data_complete_mnar30.csv
")



#50% Missing
```

```
    RANDHIE_baseline_data_complete$lnmeddol50.mcar <-
RANDHIE_baseline_data_complete$lnmeddol
    RANDHIE_baseline_data_complete$lnmeddol50.mar <-
RANDHIE_baseline_data_complete$lnmeddol
    RANDHIE_baseline_data_complete$lnmeddol50.mnar <-
RANDHIE_baseline_data_complete$lnmeddol

    set.seed(1234)
    mcar <- rbinom(n = 4451, size = 1, prob = 1/(1+exp(1-0.99)))
    sum(mcar)/length(mcar)
    mar <- rbinom(n = 4451, size = 1, prob = 1/(1+exp(1-0.038*xage)))
    sum(mar)/length(mar)
    mnar <- rbinom(n = 4451, size = 1, prob = 1/(1+exp(1-0.25*lnmeddol)))
    sum(mnar)/length(mnar)


    mcar_vec <- rep(NA, 1000)

    for (i in 1:1000){
      mcar<- rbinom(n = 4451, size = 1, prob = 1/(1+exp(1-0.99)))
      p <- sum(mcar)/length(mcar)
      mcar_vec[i] <- p
    }
    hist(mcar_vec)

    mar_vec <- rep(NA, 1000)

    for (i in 1:1000){
      mar <- rbinom(n = 4451, size = 1, prob = 1/(1+exp(1-0.038*xage)))
      p <- sum(mar)/length(mar)
      mar_vec[i] <- p
    }
    hist(mar_vec)

    mnar_vec <- rep(NA, 1000)

    for (i in 1:1000){
      mnar <- rbinom(n = 4451, size = 1, prob = 1/(1+exp(1-0.25*lnmeddol)))
      p <- sum(mnar)/length(mnar)
      mnar_vec[i] <- p
    }

    hist(mnar_vec)


    for (i in 1:4451){
```

```
  if (mcar[i]==0){
    RANDHIE_baseline_data_complete$lnmeddol50.mcar[i] <- NA
  }
  if (mar[i]==0){
    RANDHIE_baseline_data_complete$lnmeddol50.mar[i] <- NA
  }
  if (mnar[i]==0){
    RANDHIE_baseline_data_complete$lnmeddol50.mnar[i] <- NA
  }
}

RANDHIE_baseline_data_complete$lnmeddolBMCAR50 <- 0
RANDHIE_baseline_data_complete$lnmeddolBMCAR50[!is.na(RANDHIE_baseline_data_
complete$lnmeddol50.mcar)] <- 1

RANDHIE_baseline_data_complete$lnmeddolBMAR50 <- 0
RANDHIE_baseline_data_complete$lnmeddolBMAR50[!is.na(RANDHIE_baseline_data_c
omplete$lnmeddol50.mar)] <- 1

RANDHIE_baseline_data_complete$lnmeddolBMNAR50 <- 0
RANDHIE_baseline_data_complete$lnmeddolBMNAR50[!is.na(RANDHIE_baseline_data
_complete$lnmeddol50.mnar)] <- 1

attach(RANDHIE_baseline_data_complete)

#Creating new MCAR, MAR, and MNAR 50% alternatives of the RANDHIE dataset
#Write them into csv datasets

RANDHIE_baseline_data_complete_mcar50 <- data.frame(lnmeddol50.mcar, logc, disea,
linc, lfam, xage, female, child, educdec, idp, lnmeddolBMCAR50)

names(RANDHIE_baseline_data_complete_mcar50) <- c("lnmeddol", "logc", "disea",
"linc", "lfam", "xage", "female", “child”, "educdec", "idp","lnmeddolMO")


write.csv(RANDHIE_baseline_data_complete_mcar50, "C:/Users/April/Google
Drive/Thesis/FinalCodes/DatasetsWithMissing/RANDHIE_baseline_data_complete_mcar50.csv
")

RANDHIE_baseline_data_complete_mar50 <- data.frame(lnmeddol50.mar, logc, disea, linc,
lfam, xage, female, child, educdec, idp, lnmeddolBMAR50)

names(RANDHIE_baseline_data_complete_mar50) <- c("lnmeddol", "logc", "disea", "linc",
"lfam", "xage", "female", “child”, "educdec", "idp", "lnmeddolMO")
```

```
    write.csv(RANDHIE_baseline_data_complete_mar50, "C:/Users/April/Google
Drive/Thesis/FinalCodes/DatasetsWithMissing/RANDHIE_baseline_data_complete_mar50.csv"
)

    RANDHIE_baseline_data_complete_mnar50 <-data.frame(lnmeddol50.mnar, logc, disea,
linc, lfam, xage, female, child, educdec, idp, lnmeddolBMNAR50)

    names(RANDHIE_baseline_data_complete_mnar50) <- c("lnmeddol", "logc", "disea",
"linc", "lfam", "xage", "female", "child", "educdec", "idp", "lnmeddolMO")

    write.csv(RANDHIE_baseline_data_complete_mnar50, "C:/Users/April/Google
Drive/Thesis/FinalCodes/DatasetsWithMissing/RANDHIE_baseline_data_complete_mnar50.csv
")




    ##################
    #SRHS            #
    ##################

    SRHS <- SRHS_Variables
    SRHS <- read.csv("//cabinet/work$/axl807/Desktop/Thesis/SRHS.csv", head = TRUE)
    ModelFrame <- SRHS[!is.na(SRHS$c_FEV1OBSER),]

    #ModelFrame <- data.frame(SRHS_lung$c_FEV1OBSER,
as.numeric(SRHS_lung$i_AGE), as.numeric(SRHS_lung$i_BMI),
SRHS_lung$PACKYEARS,SRHS_lung$ri_GRAINDUST, SRHS_lung$h_HOMEPESTICIDE,
SRHS_lung$ri_LIVESTOCK, SRHS_lung$h_LOCATION)

    #names(ModelFrame) <- c("c_FEV1OBSER", "i_AGE", "i_BMI", "PACKYEARS",
"ri_GRAINDUST", "h_HOMEPESTICIDE", "ri_LIVESTOCK", "h_LOCATION")


    ModelFrame$ri_GRAINDUSTRecode <- rep(NA, 1608)
    ModelFrame$ri_GRAINDUSTRecode[ModelFrame$ri_GRAINDUST == 'No'] <- 0
    ModelFrame$ri_GRAINDUSTRecode[ModelFrame$ri_GRAINDUST == 'Yes'] <- 1

    ModelFrame$ri_LIVESTOCKRecode <- rep(NA, 1608)
    ModelFrame$ri_LIVESTOCKRecode[ModelFrame$ri_LIVESTOCK == "No"] <- 0
    ModelFrame$ri_LIVESTOCKRecode[ModelFrame$ri_LIVESTOCK == "Yes"] <- 1

    ModelFrame$h_HOMEPESTICIDERecode <- rep(NA, 1608)
    ModelFrame$h_HOMEPESTICIDERecode[ModelFrame$h_HOMEPESTICIDE == "No"]
<- 0
```

```
    ModelFrame$h_HOMEPESTICIDERecode[ModelFrame$h_HOMEPESTICIDE == "Yes"]
<- 1

    ModelFrame$h_LOCATIONRecode <- rep(NA, 1608)
    ModelFrame$h_LOCATIONRecode[ModelFrame$h_LOCATION == "Farm"] <- 0
    ModelFrame$h_LOCATIONRecode[ModelFrame$h_LOCATION == "In town" |
ModelFrame$h_LOCATION == "Acreage" ] <- 1

    ModelFrame$i_SEXRecode <- rep(NA, 1608)
    ModelFrame$i_SEXRecode[ModelFrame$i_SEX == "Male"] <- 0
    ModelFrame$i_SEXRecode[ModelFrame$i_SEX  == "Female"] <- 1



    NewModelFrame <- data.frame(ModelFrame$c_FEV1OBSER, ModelFrame$i_AGE,
ModelFrame$i_BMI, ModelFrame$i_SEXRecode, ModelFrame$PACKYEARS,
ModelFrame$ri_GRAINDUSTRecode, ModelFrame$h_HOMEPESTICIDERecode,
ModelFrame$ri_LIVESTOCKRecode, ModelFrame$h_LOCATIONRecode)

    names(NewModelFrame) <- c("c_FEV1OBSER", "i_AGE", "i_BMI", "i_SEXRecode",
"PACKYEARS", "ri_GRAINDUSTRecode", "h_HOMEPESTICIDERecode",
"ri_LIVESTOCKRecode", "h_LOCATIONRecode")

    #Get Rid of Missing Data in Independent Variables#

    NewModelFrame2 <- NewModelFrame[!is.na(NewModelFrame$PACKYEARS),]
    NewModelFrame3 <-
NewModelFrame2[!is.na(NewModelFrame2$ri_GRAINDUSTRecode),]
    NewModelFrame4 <-
NewModelFrame3[!is.na(NewModelFrame3$h_HOMEPESTICIDERecode),]
    NewModelFrame5 <-
NewModelFrame4[!is.na(NewModelFrame4$ri_LIVESTOCKRecode),]
    NewModelFrame6 <-
NewModelFrame5[!is.na(NewModelFrame5$h_LOCATIONRecode),]
    NewModelFrame7 <- NewModelFrame6[!is.na(NewModelFrame6$i_BMI),]
    NewModelFrame8 <- NewModelFrame7[!is.na(NewModelFrame7$i_SEXRecode),]
    NewModelFrameFINAL <-NewModelFrame8[!is.na(NewModelFrame8$i_AGE),]

    write.csv(NewModelFrameFINAL, "C:/Users/April/Google
Drive/Thesis/FinalCodes/DatasetsWithMissing/SRHS_complete.csv")

    #Introducing MCAR, MAR, and MNAR Data to ModelFrame_completeFINAL

    #Adding 15% Missing Data
    NewModelFrameFINAL$c_FEV1OBSER15.mcar <-
NewModelFrameFINAL$c_FEV1OBSER
```

```
    NewModelFrameFINAL$c_FEV1OBSER15.mar <-
NewModelFrameFINAL$c_FEV1OBSER
    NewModelFrameFINAL$c_FEV1OBSER15.mnar <-
NewModelFrameFINAL$c_FEV1OBSER


    set.seed(1234)
    mcar <- rbinom(n = 1495, size = 1, prob = 1/(1+exp(1-2.73)))
    sum(mcar)/length(mcar)
    mar <- rbinom(n = 1495, size = 1, prob = 1/(1+exp(1-
0.053*NewModelFrameFINAL$i_AGE)))
    sum(mar)/length(mar)
    mnar <- rbinom(n = 1495, size = 1, prob = 1/(1+exp(1-
0.93*NewModelFrameFINAL$c_FEV1OBSER)))
    sum(mnar)/length(mnar)


    mcar_vec <- rep(NA, 1000)

    for (i in 1:1000){
      mcar<- rbinom(n = 4451, size = 1, prob = 1/(1+exp(1-2.73)))
      p <- sum(mcar)/length(mcar)
      mcar_vec[i] <- p
    }
    hist(mcar_vec)

    mar_vec <- rep(NA, 1000)

    for (i in 1:1000){
      mar <- rbinom(n = 4451, size = 1, prob = 1/(1+exp(1-
0.053*NewModelFrameFINAL$i_AGE)))
      p <- sum(mar)/length(mar)
      mar_vec[i] <- p
    }
    hist(mar_vec)

    mnar_vec <- rep(NA, 1000)

    for (i in 1:1000){
      mnar <- rbinom(n = 4451, size = 1, prob = 1/(1+exp(1-
0.93*NewModelFrameFINAL$c_FEV1OBSER)))
      p <- sum(mnar)/length(mnar)
      mnar_vec[i] <- p
    }

    hist(mnar_vec)
```

```
for (i in 1:1495){
  if (mcar[i]==0){
    NewModelFrameFINAL$c_FEV1OBSER15.mcar[i] <- NA
  }
  if (mar[i]==0){
    NewModelFrameFINAL$c_FEV1OBSER15.mar[i] <- NA
  }
  if (mnar[i]==0){
    NewModelFrameFINAL$c_FEV1OBSER15.mnar[i] <- NA
  }
}

NewModelFrameFINAL$c_FEV1OBSER15.mcarMO <- rep(0, 1495)
NewModelFrameFINAL$c_FEV1OBSER15.mcarMO[!is.na(NewModelFrameFINAL$c_FEV1OBSER15.mcar)] <- 1

NewModelFrameFINAL$c_FEV1OBSER15.marMO <- rep(0, 1495)
NewModelFrameFINAL$c_FEV1OBSER15.marMO[!is.na(NewModelFrameFINAL$c_FEV1OBSER15.mar)] <- 1

NewModelFrameFINAL$c_FEV1OBSER15.mnarMO <- rep(0, 1495)
NewModelFrameFINAL$c_FEV1OBSER15.mnarMO[!is.na(NewModelFrameFINAL$c_FEV1OBSER15.mnar)] <- 1

#Get model frames ready for imputation#

SRHS15_MCAR <- data.frame(NewModelFrameFINAL$c_FEV1OBSER15.mcar,
NewModelFrameFINAL$i_AGE, NewModelFrameFINAL$i_BMI,
NewModelFrameFINAL$i_SEXRecode, NewModelFrameFINAL$PACKYEARS,
NewModelFrameFINAL$ri_GRAINDUSTRecode,
NewModelFrameFINAL$h_HOMEPESTICIDERecode,
NewModelFrameFINAL$ri_LIVESTOCKRecode,
NewModelFrameFINAL$h_LOCATIONRecode,
NewModelFrameFINAL$c_FEV1OBSER15.mcarMO)

names(SRHS15_MCAR) <- c("c_FEV1OBSER", "i_AGE", "i_BMI", "i_SEXRecode",
"PACKYEARS", "ri_GRAINDUSTRecode", "h_HOMEPESTICIDERecode",
"ri_LIVESTOCKRecode", "h_LOCATIONRecode", "c_FEV1OBSERMO")

write.csv(SRHS15_MCAR, "C:/Users/April/Google
Drive/Thesis/FinalCodes/DatasetsWithMissing/SRHS15_MCAR.csv")

SRHS15_MAR <- data.frame(NewModelFrameFINAL$c_FEV1OBSER15.mar,
NewModelFrameFINAL$i_AGE, NewModelFrameFINAL$i_BMI,
```

```
NewModelFrameFINAL$i_SEXRecode, NewModelFrameFINAL$PACKYEARS,
NewModelFrameFINAL$ri_GRAINDUSTRecode,
NewModelFrameFINAL$h_HOMEPESTICIDERecode,
NewModelFrameFINAL$ri_LIVESTOCKRecode,
NewModelFrameFINAL$h_LOCATIONRecode,
NewModelFrameFINAL$c_FEV1OBSER15.marMO)

    names(SRHS15_MAR) <- c("c_FEV1OBSER", "i_AGE", "i_BMI", "i_SEXRecode",
"PACKYEARS", "ri_GRAINDUSTRecode", "h_HOMEPESTICIDERecode",
"ri_LIVESTOCKRecode", "h_LOCATIONRecode", "c_FEV1OBSERMO")

    write.csv(SRHS15_MAR, "C:/Users/April/Google
Drive/Thesis/FinalCodes/DatasetsWithMissing/SRHS15_MAR.csv")


    SRHS15_MNAR <- data.frame(NewModelFrameFINAL$c_FEV1OBSER15.mnar,
NewModelFrameFINAL$i_AGE, NewModelFrameFINAL$i_BMI,
NewModelFrameFINAL$i_SEXRecode, NewModelFrameFINAL$PACKYEARS,
NewModelFrameFINAL$ri_GRAINDUSTRecode,
NewModelFrameFINAL$h_HOMEPESTICIDERecode,
NewModelFrameFINAL$ri_LIVESTOCKRecode,
NewModelFrameFINAL$h_LOCATIONRecode,
NewModelFrameFINAL$c_FEV1OBSER15.mnarMO)

    names(SRHS15_MNAR) <- c("c_FEV1OBSER", "i_AGE", "i_BMI", "i_SEXRecode",
"PACKYEARS", "ri_GRAINDUSTRecode", "h_HOMEPESTICIDERecode",
"ri_LIVESTOCKRecode", "h_LOCATIONRecode", "c_FEV1OBSERMO")

    write.csv(SRHS15_MNAR, "C:/Users/April/Google
Drive/Thesis/FinalCodes/DatasetsWithMissing/SRHS15_MNAR.csv")

    ##########################
    #Adding 30% Missing Data   #
    ##########################
    NewModelFrameFINAL$c_FEV1OBSER30.mcar <-
NewModelFrameFINAL$c_FEV1OBSER
    NewModelFrameFINAL$c_FEV1OBSER30.mar <-
NewModelFrameFINAL$c_FEV1OBSER
    NewModelFrameFINAL$c_FEV1OBSER30.mnar <-
NewModelFrameFINAL$c_FEV1OBSER

    set.seed(1234)
    mcar <- rbinom(n = 1495, size = 1, prob = 1/(1+exp(1-1.9)))
    sum(mcar)/length(mcar)
    mar <- rbinom(n = 1495, size = 1, prob = 1/(1+exp(1-
0.035*NewModelFrameFINAL$i_AGE)))
```

```
sum(mar)/length(mar)
mnar <- rbinom(n = 1495, size = 1, prob = 1/(1+exp(1-
0.6*NewModelFrameFINAL$c_FEV1OBSER)))
sum(mnar)/length(mnar)


mcar_vec <- rep(NA, 1000)

for (i in 1:1000){
  mcar<- rbinom(n = 4451, size = 1, prob = 1/(1+exp(1-1.9)))
  p <- sum(mcar)/length(mcar)
  mcar_vec[i] <- p
}
hist(mcar_vec)

mar_vec <- rep(NA, 1000)

for (i in 1:1000){
  mar <- rbinom(n = 4451, size = 1, prob = 1/(1+exp(1-
0.035*NewModelFrameFINAL$i_AGE)))
  p <- sum(mar)/length(mar)
  mar_vec[i] <- p
}
hist(mar_vec)

mnar_vec <- rep(NA, 1000)

for (i in 1:1000){
  mnar <- rbinom(n = 4451, size = 1, prob = 1/(1+exp(1-
0.6*NewModelFrameFINAL$c_FEV1OBSER)))
  p <- sum(mnar)/length(mnar)
  mnar_vec[i] <- p
}

hist(mnar_vec)


for (i in 1:1495){
  if (mcar[i]==0){
    NewModelFrameFINAL$c_FEV1OBSER30.mcar[i] <- NA
  }
  if (mar[i]==0){
    NewModelFrameFINAL$c_FEV1OBSER30.mar[i] <- NA
  }
  if (mnar[i]==0){
    NewModelFrameFINAL$c_FEV1OBSER30.mnar[i] <- NA
```

```
  }
}

NewModelFrameFINAL$c_FEV1OBSER30.mcarMO <- rep(0, 1495)
NewModelFrameFINAL$c_FEV1OBSER.mcarMO[!is.na(NewModelFrameFINAL$c_FEV
1OBSER30.mcar)] <- 1

NewModelFrameFINAL$c_FEV1OBSER30.marMO <- rep(0, 1495)
NewModelFrameFINAL$c_FEV1OBSER.marMO[!is.na(NewModelFrameFINAL$c_FEV1
OBSER30.mar)] <- 1

NewModelFrameFINAL$c_FEV1OBSER30.mnarMO <- rep(0, 1495)
NewModelFrameFINAL$c_FEV1OBSER30.mnarMO[!is.na(NewModelFrameFINAL$c_F
EV1OBSER30.mnar)] <- 1

#Get model frames ready for imputation#

SRHS30_MCAR <- data.frame(NewModelFrameFINAL$c_FEV1OBSER30.mcar,
NewModelFrameFINAL$i_AGE, NewModelFrameFINAL$i_BMI,
NewModelFrameFINAL$i_SEXRecode, NewModelFrameFINAL$PACKYEARS,
NewModelFrameFINAL$ri_GRAINDUSTRecode,
NewModelFrameFINAL$h_HOMEPESTICIDERecode,
NewModelFrameFINAL$ri_LIVESTOCKRecode,
NewModelFrameFINAL$h_LOCATIONRecode,
NewModelFrameFINAL$c_FEV1OBSER30.mcarMO)

names(SRHS30_MCAR) <- c("c_FEV1OBSER", "i_AGE", "i_BMI", "i_SEXRecode",
"PACKYEARS", "ri_GRAINDUSTRecode", "h_HOMEPESTICIDERecode",
"ri_LIVESTOCKRecode", "h_LOCATIONRecode", "c_FEV1OBSERMO")

write.csv(SRHS30_MCAR, "C:/Users/April/Google
Drive/Thesis/FinalCodes/DatasetsWithMissing/SRHS30_MCAR.csv")

SRHS30_MAR <- data.frame(NewModelFrameFINAL$c_FEV1OBSER30.mar,
NewModelFrameFINAL$i_AGE, NewModelFrameFINAL$i_BMI,
NewModelFrameFINAL$i_SEXRecode, NewModelFrameFINAL$PACKYEARS,
NewModelFrameFINAL$ri_GRAINDUSTRecode,
NewModelFrameFINAL$h_HOMEPESTICIDERecode,
NewModelFrameFINAL$ri_LIVESTOCKRecode,
NewModelFrameFINAL$h_LOCATIONRecode,
NewModelFrameFINAL$c_FEV1OBSER30.marMO)

names(SRHS30_MAR) <- c("c_FEV1OBSER", "i_AGE", "i_BMI", "i_SEXRecode",
"PACKYEARS", "ri_GRAINDUSTRecode", "h_HOMEPESTICIDERecode",
"ri_LIVESTOCKRecode", "h_LOCATIONRecode", "c_FEV1OBSERMO")
```

```
    write.csv(SRHS30_MAR, "C:/Users/April/Google
Drive/Thesis/FinalCodes/DatasetsWithMissing/SRHS30_MAR.csv")

    SRHS30_MNAR <- data.frame(NewModelFrameFINAL$c_FEV1OBSER30.mnar,
NewModelFrameFINAL$i_AGE, NewModelFrameFINAL$i_BMI,
NewModelFrameFINAL$i_SEXRecode, NewModelFrameFINAL$PACKYEARS,
NewModelFrameFINAL$ri_GRAINDUSTRecode,
NewModelFrameFINAL$h_HOMEPESTICIDERecode,
NewModelFrameFINAL$ri_LIVESTOCKRecode,
NewModelFrameFINAL$h_LOCATIONRecode,
NewModelFrameFINAL$c_FEV1OBSER30.mnarMO)

    names(SRHS30_MNAR) <- c("c_FEV1OBSER", "i_AGE", "i_BMI", "i_SEXRecode",
"PACKYEARS", "ri_GRAINDUSTRecode", "h_HOMEPESTICIDERecode",
"ri_LIVESTOCKRecode", "h_LOCATIONRecode", "c_FEV1OBSERMO")

    write.csv(SRHS30_MNAR, "C:/Users/April/Google
Drive/Thesis/FinalCodes/DatasetsWithMissing/SRHS30_MNAR.csv")



    ##########################
    #Adding 50% Missing Data#
    ##########################
    NewModelFrameFINAL$c_FEV1OBSER50.mcar <-
NewModelFrameFINAL$c_FEV1OBSER
    NewModelFrameFINAL$c_FEV1OBSER50.mar <-
NewModelFrameFINAL$c_FEV1OBSER
    NewModelFrameFINAL$c_FEV1OBSER50.mnar <-
NewModelFrameFINAL$c_FEV1OBSER

    set.seed(1234)
    mcar <- rbinom(n = 1495, size = 1, prob = 1/(1+exp(1-0.99)))
    sum(mcar)/length(mcar)
    mar <- rbinom(n = 1495, size = 1, prob = 1/(1+exp(1-
0.0185*NewModelFrameFINAL$i_AGE)))
    sum(mar)/length(mar)
    mnar <- rbinom(n = 1495, size = 1, prob = 1/(1+exp(1-
0.33*NewModelFrameFINAL$c_FEV1OBSER)))
    sum(mnar)/length(mnar)


    mcar_vec <- rep(NA, 1000)

    for (i in 1:1000){
      mcar<- rbinom(n = 4451, size = 1, prob = 1/(1+exp(1-0.99)))
```

```
  p <- sum(mcar)/length(mcar)
  mcar_vec[i] <- p
 }
 hist(mcar_vec)

 mar_vec <- rep(NA, 1000)

 for (i in 1:1000){
  mar <- rbinom(n = 4451, size = 1, prob = 1/(1+exp(1-
0.0185*NewModelFrameFINAL$i_AGE)))
  p <- sum(mar)/length(mar)
  mar_vec[i] <- p
 }
 hist(mar_vec)

 mnar_vec <- rep(NA, 1000)

 for (i in 1:1000){
  mnar <- rbinom(n = 4451, size = 1, prob = 1/(1+exp(1-
0.33*NewModelFrameFINAL$c_FEV1OBSER)))
  p <- sum(mnar)/length(mnar)
  mnar_vec[i] <- p
 }

 hist(mnar_vec)


 for (i in 1:1495){
  if (mcar[i]==0){
   NewModelFrameFINAL$c_FEV1OBSER50.mcar[i] <- NA
  }
  if (mar[i]==0){
   NewModelFrameFINAL$c_FEV1OBSER50.mar[i] <- NA
  }
  if (mnar[i]==0){
   NewModelFrameFINAL$c_FEV1OBSER50.mnar[i] <- NA
  }
 }

 NewModelFrameFINAL$c_FEV1OBSER50.mcarMO <- rep(0, 1495)
 NewModelFrameFINAL$c_FEV1OBSER50.mcarMO[!is.na(NewModelFrameFINAL$c_F
EV1OBSER50.mcar)] <- 1

 NewModelFrameFINAL$c_FEV1OBSER50.marMO <- rep(0, 1495)
 NewModelFrameFINAL$c_FEV1OBSER50.marMO[!is.na(NewModelFrameFINAL$c_FE
V1OBSER50.mar)] <- 1
```

```
    NewModelFrameFINAL$c_FEV1OBSER50.mnarMO <- rep(0, 1495)
    NewModelFrameFINAL$c_FEV1OBSER50.mnarMO[!is.na(NewModelFrameFINAL$c_F
EV1OBSER50.mnar)] <- 1

    #Get model frames ready for imputation#

    SRHS50_MCAR <- data.frame(NewModelFrameFINAL$c_FEV1OBSER50.mcar,
NewModelFrameFINAL$i_AGE, NewModelFrameFINAL$i_BMI,
NewModelFrameFINAL$i_SEXRecode, NewModelFrameFINAL$PACKYEARS,
NewModelFrameFINAL$ri_GRAINDUSTRecode,
NewModelFrameFINAL$h_HOMEPESTICIDERecode,
NewModelFrameFINAL$ri_LIVESTOCKRecode,
NewModelFrameFINAL$h_LOCATIONRecode,
NewModelFrameFINAL$c_FEV1OBSER50.mcarMO)

    names(SRHS50_MCAR) <- c("c_FEV1OBSER", "i_AGE", "i_BMI", "i_SEXRecode",
"PACKYEARS", "ri_GRAINDUSTRecode", "h_HOMEPESTICIDERecode",
"ri_LIVESTOCKRecode", "h_LOCATIONRecode", "c_FEV1OBSERMO")

    write.csv(SRHS50_MCAR, "C:/Users/April/Google
Drive/Thesis/FinalCodes/DatasetsWithMissing/SRHS50_MCAR.csv")

    SRHS50_MAR <- data.frame(NewModelFrameFINAL$c_FEV1OBSER50.mar,
NewModelFrameFINAL$i_AGE, NewModelFrameFINAL$i_BMI,
NewModelFrameFINAL$i_SEXRecode, NewModelFrameFINAL$PACKYEARS,
NewModelFrameFINAL$ri_GRAINDUSTRecode,
NewModelFrameFINAL$h_HOMEPESTICIDERecode,
NewModelFrameFINAL$ri_LIVESTOCKRecode,
NewModelFrameFINAL$h_LOCATIONRecode,
NewModelFrameFINAL$c_FEV1OBSER50.marMO)

    names(SRHS50_MAR) <- c("c_FEV1OBSER", "i_AGE", "i_BMI", "i_SEXRecode",
"PACKYEARS", "ri_GRAINDUSTRecode", "h_HOMEPESTICIDERecode",
"ri_LIVESTOCKRecode", "h_LOCATIONRecode", "c_FEV1OBSERMO")

    write.csv(SRHS50_MAR, "C:/Users/April/Google
Drive/Thesis/FinalCodes/DatasetsWithMissing/SRHS50_MAR.csv")

    SRHS50_MNAR <- data.frame(NewModelFrameFINAL$c_FEV1OBSER50.mnar,
NewModelFrameFINAL$i_AGE, NewModelFrameFINAL$i_BMI,
NewModelFrameFINAL$i_SEXRecode, NewModelFrameFINAL$PACKYEARS,
NewModelFrameFINAL$ri_GRAINDUSTRecode,
NewModelFrameFINAL$h_HOMEPESTICIDERecode,
NewModelFrameFINAL$ri_LIVESTOCKRecode,
```

```
NewModelFrameFINAL$h_LOCATIONRecode,
NewModelFrameFINAL$c_FEV1OBSER50.mnarMO)

    names(SRHS50_MNAR) <- c("c_FEV1OBSER", "i_AGE", "i_BMI", "i_SEXRecode",
"PACKYEARS", "ri_GRAINDUSTRecode", "h_HOMEPESTICIDERecode",
"ri_LIVESTOCKRecode", "h_LOCATIONRecode", "c_FEV1OBSERMO")

    write.csv(SRHS50_MNAR, "C:/Users/April/Google
Drive/Thesis/FinalCodes/DatasetsWithMissing/SRHS50_MNAR.csv")
```

B.2 Applying Missing Data Methods to Datasets

```
    folder <- "C:/Users/April/Desktop/PhD_Analysis/DatasetsWithMissing/"
    # path to folder that holds multiple .csv files

    file_list <- list.files(path=folder, pattern="*.csv")
    # create list of all .csv files in folder

    # read in each .csv file in file_list and create a data frame with the same name as the .csv file

    for (i in 1:length(file_list)){

     assign(file_list[i],

          read.csv(paste(folder, file_list[i], sep=''))

     )}

    RANDHIE_Datasets <- list(RANDHIE_baseline_data_complete_mcar15,
                RANDHIE_baseline_data_complete_mcar30,
                RANDHIE_baseline_data_complete_mcar50,
                RANDHIE_baseline_data_complete_mar15,
                RANDHIE_baseline_data_complete_mar30,
                RANDHIE_baseline_data_complete_mar50,
                RANDHIE_baseline_data_complete_mcar15,
                RANDHIE_baseline_data_complete_mnar30,
                RANDHIE_baseline_data_complete_mnar50)


    SRHS_Datasets <- list(SRHS15_MCAR,
                SRHS30_MCAR,
                SRHS50_MCAR,
```

```
                SRHS15_MAR,
                SRHS30_MAR,
                SRHS50_MAR,
                SRHS15_MNAR,
                SRHS30_MNAR,
                SRHS50_MNAR)

    ##################################
    #RANDHIE Complete Data Analysis#
    ##################################

    Complete_Analysis_RANDHIE <- lm(lnmeddol ~ logc +
                    linc + lfam + xage + female +
                    child , data = RANDHIE_baseline_data_complete)

    summary(Complete_Analysis_RANDHIE)


    Complete_Analysis_SRHS <- lm(c_FEV1OBSER ~ i_AGE + i_BMI + i_SEXRecode +
PACKYEARS + ri_LIVESTOCKRecode, data = SRHS_complete)

    summary(Complete_Analysis_SRHS)

    #Selection model contains h_LOCATIONRecode and ri_GRAINDUSTRecode

    ##########################################################################
    #Complete Case Analysis for SRHS and RANDHIE missing 15%, 30%, and 50%#
    ##########################################################################

    #Complete Case Analysis

    RANDHIE_models_CC <-   lapply(RANDHIE_Datasets, function(data){
      lm(reformulate(c("logc", "linc", "lfam", "xage", "female", "child"),
response=names(data)[2]), data)})

    RANDHIE_ModelsSummary_CC <- lapply(RANDHIE_models_CC,
function(data){summary(data)})
    RANDHIE_ModelsSummary_CC

    betas.Confint.RANDHIE_models_CC <- cbind(coef(RANDHIE_models_CC[[1]]),
confint(RANDHIE_models_CC[[1]], level = 0.95),
                        coef(RANDHIE_models_CC[[2]]),
confint(RANDHIE_models_CC[[2]], level = 0.95),
                        coef(RANDHIE_models_CC[[3]]),
confint(RANDHIE_models_CC[[3]], level = 0.95),
```

```
                              coef(RANDHIE_models_CC[[4]]),
confint(RANDHIE_models_CC[[4]], level = 0.95),
                              coef(RANDHIE_models_CC[[5]]),
confint(RANDHIE_models_CC[[5]], level = 0.95),
                              coef(RANDHIE_models_CC[[6]]),
confint(RANDHIE_models_CC[[6]], level = 0.95),
                              coef(RANDHIE_models_CC[[7]]),
confint(RANDHIE_models_CC[[7]], level = 0.95),
                              coef(RANDHIE_models_CC[[8]]),
confint(RANDHIE_models_CC[[8]], level = 0.95),
                              coef(RANDHIE_models_CC[[9]]),
confint(RANDHIE_models_CC[[9]], level = 0.95))

    write.csv(betas.Confint.RANDHIE_models_CC,
"C:/Users/axl807/Downloads/Tables/betas.Confint.RANDHIE_models_CC.csv")



    SRHS_models_CC <- lapply(SRHS_Datasets,
                  function(data){
                    lm(reformulate(c("i_AGE", "i_BMI", "i_SEXRecode", "PACKYEARS",
                          "ri_LIVESTOCKRecode"), response=names(data)[2]), data)
                  })
    SRHS_ModelsSummary_CC <- lapply(SRHS_models_CC,
function(model){summary(model)})
    SRHS_ModelsSummary_CC

    betas.Confint.SRHS_models_CC <- cbind(coef(SRHS_models_CC[[1]]),
confint(SRHS_models_CC[[1]], level = 0.95),
                              coef(SRHS_models_CC[[2]]), confint(SRHS_models_CC[[2]],
level = 0.95),
                              coef(SRHS_models_CC[[3]]), confint(SRHS_models_CC[[3]],
level = 0.95),
                              coef(SRHS_models_CC[[4]]), confint(SRHS_models_CC[[4]],
level = 0.95),
                              coef(SRHS_models_CC[[5]]), confint(SRHS_models_CC[[5]],
level = 0.95),
                              coef(SRHS_models_CC[[6]]), confint(SRHS_models_CC[[6]],
level = 0.95),
                              coef(SRHS_models_CC[[7]]), confint(SRHS_models_CC[[7]],
level = 0.95),
                              coef(SRHS_models_CC[[8]]), confint(SRHS_models_CC[[8]],
level = 0.95),
                              coef(SRHS_models_CC[[9]]), confint(SRHS_models_CC[[9]],
level = 0.95))
```

```
    write.csv(betas.Confint.SRHS_models_CC,
"C:/Users/axl807/Downloads/Tables/betas.Confint.SRHS_models_CC.csv")

    ############
    #Rubin's MI#
    ############
    install.packages("mi")
    library(mi)
    install.packages("betareg")
    library(betareg)

    #SRHS_Rubin <- mi(SRHS15_MCAR.csv[, 2:10], n.iter = 30, n.chains = 10, seed = 1234)

    #Result_RANDHIE_Rubin <- pool(c_FEV1OBSER~i_AGE + i_BMI + i_SEXRecode +
PACKYEARS + ri_LIVESTOCKRecode, data = SRHS_Rubin)
    #pool combines estimates by Rubin's rules
    #summary(Result_RANDHIE_Rubin)
    RANDHIE_Models_Rubin <- lapply(RANDHIE_Datasets, function(data){
      RANDHIE_Rubin <- mi(data[, 2:13], n.iter = 30, n.chains = 10, seed = 1234)
      Result_SRHS_Rubin <- pool(lnmeddol~logc +  disea + linc + lfam + xage + female + child
, data = RANDHIE_Rubin)
    })

    RANDHIE_ModelsSummary_Rubin  <- lapply(RANDHIE_Models_Rubin,
function(model){summary(model)})
    RANDHIE_ModelsSummary_Rubin



    SRHS_Models_Rubin <- lapply(SRHS_Datasets, function(data){
      SRHS_Rubin <- mi(data[, 2:10], n.iter = 30, n.chains = 10, seed = 1234)

      Result_SRHS_Rubin <- pool(c_FEV1OBSER~i_AGE + i_BMI + i_SEXRecode +
PACKYEARS + ri_LIVESTOCKRecode, data = SRHS_Rubin)
    })

    SRHS_ModelsSummary_Rubin <- lapply(SRHS_Models_Rubin,
function(model){summary(model)})
    SRHS_ModelsSummary_Rubin


    ###############
    #Galimard's MI#
    ###############
    install.packages("miceMNAR")
    library(miceMNAR)
```

```
install.packages("mice")
library(mice)

RANDHIE_Models_Galimard <- lapply(RANDHIE_Datasets, function(data){
  JointModelEq <- generate_JointModelEq(data=data,varMNAR = "lnmeddol")

  JointModelEq[,"lnmeddol_var_sel"] <- c(0,1,1,1,1,1,1,1,1,1,1)
  JointModelEq[,"lnmeddol_var_out"] <- c(0,1,0, 1,1,1,1,1,0,0,0)


 names(RANDHIE_baseline_data_complete_mar30) <- c("lnmeddol", "logc", "disea", "linc",
"lfam", "xage", "female", “child”, "educdec", "idp", "lnmeddolMO")

  arg <- MNARargument(data=data, varMNAR="lnmeddol", JointModelEq=JointModelEq)
  arg$method["lnmeddol"] <- "hecknorm2step"

  RANDHIE_Imputation_Galimard <- mice(data = arg$data_mod,
                                      method = arg$method,
                                      predictorMatrix = arg$predictorMatrix,
                                      JointModelEq=arg$JointModelEq,
                                      control=arg$control,
                                      maxit=30,m=10)

   RANDHIE_analysis_Galimard <-
with(RANDHIE_Imputation_Galimard,lm(lnmeddol~logc +  disea + linc + lfam + xage +
female + child , data = data))
   RANDHIE_analysis_Galimard_pool <- pool(RANDHIE_analysis_Galimard)
   summary(RANDHIE_analysis_Galimard_pool)

 })

 RANDHIE_Models_Galimard


 SRHS_Models_Galimard <- lapply(SRHS_Datasets, function(data){
  JointModelEq <- generate_JointModelEq(data=data,varMNAR = "c_FEV1OBSER")

  JointModelEq[,"c_FEV1OBSER_var_sel"] <- c(0,1,1,1,1,1,1,1,1,1,0)
  JointModelEq[,"c_FEV1OBSER_var_out"] <- c(0,1,1,1,1,0,0,1,0,0)
```

Age, **BMI**, **Sex**, **Packyears**, **LIVESTOCK**, **Graindust**, and **Pesticide** while the outcome model excluded **Graindust** and **Pesticide**.

```
 names(SRHS15_MCAR) <- c("c_FEV1OBSER", "i_AGE", "i_BMI", "i_SEXRecode",
"PACKYEARS", "ri_GRAINDUSTRecode", "h_HOMEPESTICIDERecode",
"ri_LIVESTOCKRecode", "h_LOCATIONRecode", "c_FEV1OBSERMO")
```

```
      arg <-
MNARargument(data=data,varMNAR="c_FEV1OBSER",JointModelEq=JointModelEq)
      arg$method["c_FEV1OBSER"] <- "hecknorm2step"

      SRHS_Imputation_Galimard <- mice(data = arg$data_mod,
                                  method = arg$method,
                                  predictorMatrix = arg$predictorMatrix,
                                  JointModelEq=arg$JointModelEq,
                                  control=arg$control,
                                  maxit=30,m=10)

      SRHS_analysis_Galimard <- with(SRHS_Imputation_Galimard,
lm(c_FEV1OBSER~i_AGE + i_BMI + i_SEXRecode + PACKYEARS +
ri_LIVESTOCKRecode, data = data))
      SRHS_analysis_Galimard_pool <- pool(SRHS_analysis_Galimard)
      summary(SRHS_analysis_Galimard_pool)

    })

    SRHS_Models_Galimard

    ################
    #Ogundimu's MI#
    ################
    install.packages("gamlss")
    library(gamlss)
    library(sampleSelection)


    tselectEst<-function (selection,outcome,data = sys.frame(sys.parent()),YS, XS, YO, XO,
start=NULL,print.level=0,
                      maxMethod = "BFGS",...)
    {
     if (match("sampleSelection",.packages(),0)==0) require(sampleSelection)
     if (match("mnormt",.packages(),0)==0) require(mnormt)
     if (match("mvtnorm",.packages(),0)==0) require(mvtnorm)
     if (!missing(data)) {
       if (!inherits(data, "environment") & !inherits(data,
                                          "data.frame") & !inherits(data, "list")) {
         stop("'data' must be either environment, data.frame, or list (currently a ",
             class(data), ")")
       }
     }
```

```
mf <- match.call(expand.dots = FALSE)
m <- match(c("selection", "data", "subset"), names(mf), 0)
mfS <- mf[c(1, m)]
mfS$drop.unused.levels <- TRUE
mfS$na.action <- na.pass
mfS[[1]] <- as.name("model.frame")
names(mfS)[2] <- "formula"
mfS <- eval(mfS, parent.frame())
mtS <- attr(mfS, "terms")
XS <- model.matrix(mtS, mfS)
YS <- model.response(mfS)
YSLevels <- levels(as.factor(YS))
YS <- as.integer(YS == tail(YSLevels, 1))
badRow <- is.na(YS)
badRow <- badRow | apply(XS, 1, function(v) any(is.na(v)))
oArg <- match("outcome", names(mf), 0)
m <- match(c("outcome", "data", "subset", "offset"),
        names(mf), 0)
mfO <- mf[c(1, m)]
mfO$drop.unused.levels <- TRUE
mfO$na.action <- na.pass
mfO[[1]] <- as.name("model.frame")
names(mfO)[2] <- "formula"
mfO <- eval(mfO, parent.frame())
mtO <- attr(mfO, "terms")
XO <- model.matrix(mtO, mfO)
YO <- model.response(mfO)
badRow <- badRow | (is.na(YO) & (!is.na(YS) & YS == 1))
badRow <- badRow | (apply(XO, 1, function(v) any(is.na(v))) &
                  (!is.na(YS) & YS == 1))

if (length(YSLevels) != 2) {
  stop("the left hand side of the 'selection' formula\n",
      "has to contain", " exactly two levels (e.g. FALSE and TRUE)")
}
XS <- XS[!badRow, , drop = FALSE]
YS <- YS[!badRow]
XO <- XO[!badRow, , drop = FALSE]
YO <- YO[!badRow]
YO[YS == 0] <- NA
XO[YS == 0, ] <- NA

loglik <- function(bstart) {

  p <- ncol(XS); k=ncol(XO)
```

```
  b1 =bstart[1:p];b2 =bstart[(p+1):(k+p)]
  sigma <- bstart[(k+p+1)]
  if (sigma < 0)
    return(NA)
  rho <- bstart[k+p+2]
  if ((rho < -1) || (rho > 1))
    return(NA)
  nu <- bstart[k+p+3]
  ll <- vector()
  if( nu >2 ){
    XS.g <- XS %*% b1
    XO.b <- XO %*% b2
    u2 <- YO - XO.b
    r <- sqrt(1 - rho^2)
    z <- u2/sigma
    B<- (XS.g + rho/sigma * u2)/r
    K <- ((nu+1)/(nu+(z^2)))^0.5
    K1 <- K*B
    l1 <- log(dt(z,nu))-log(sigma)+log(pt(K1,nu+1))
    ll<- ifelse(YS==0,log(pt(-XS.g,nu)),l1)
    return(-sum(ll))
  }
  else return(Inf)

}
if (is.null(start))
  tobit2 <- selection(selection,outcome, data=data)
coefs <- coef(tobit2, part = "full")
bstart1 <- coefs[tobit2$param$index$betaS]
bstart2 <- coefs[tobit2$param$index$betaO]
bstart3 <- coefs['sigma']
bstart4 <- coefs['rho']
start <- c(bstart1,bstart2,bstart3,bstart4)
startt <- c(start,nu=5)
fit <- optim(startt,loglik,control=list(maxit=1000),method="BFGS", hessian=TRUE)
loglike <- fit$value
nn <- length(YS)
nParam <- length(startt)

aic <- 2*fit$value + 2*nParam
bic <- 2*fit$value + nParam*log(nn)

df <- nn-nParam
N0 <-sum(YS == 0); N1 <- sum(YS == 1);nObs = length(YS)
coef <- fit$par
vcov <- solve(fit$hessian)
```

```
  hessian <- fit$hessian

return(list(coefficients=coef,hessian=hessian,vcov=vcov,level=YSLevels,aic=aic,bic=bic,df=df,
            loglike=-loglike,N0=N0,N1=N1,NObs=nObs,initial.values=startt))
}



tselect <- function(selection, outcome, data,...) UseMethod("tselect")


tselect.default <- function(selection, outcome,data, start = NULL, verbose = FALSE, ...)
{
 mfs <- model.frame(selection, data)
 mts <- attr(mfs, "terms")
 YS <- model.response(mfs, "numeric")
 XS <- model.matrix(selection, data = data)

 mfo <- model.frame(outcome, data)
 mto <- attr(mfo, "terms")
 YO <- model.response(mfo, "numeric")
 XO <- model.matrix(outcome, data = data)
 est <- tselectEst(selection, outcome, data,start = NULL, verbose = FALSE)
 co <- est$coefficients
 NXS <- ncol(XS)
 NXO <- ncol(XO)
 iGamma <- 1:NXS
 iBeta <- max(iGamma) + seq(length = NXO)
 iSigma <- max(iBeta) + 1
 iRho <- max(iSigma) + 1
 iNu <- max(iSigma) + 2

 betaS <- co[iGamma]
 betaO <- co[iBeta]
 sigma <- co[iSigma]
 rho <- co[iRho]
 nu <- co[iNu]

 aic <- est$aic
 bic <- est$bic
 initial.values <- est$initial.values
 loglik <- est$loglik

 est$call <- match.call()
 class(est) <- "tselect"
 est
```

```
}

print.tselect <- function(formula, ...)
{
 cat("Call:\n")
 print(formula$call)
 cat("\nCoefficients:\n")
 print(formula$coefficients)
}


summary.tselect <- function(object, ...)
{
 se <- sqrt(diag(object$vcov))
 tval <- coef(object) / se
 TAB <- cbind(Estimate = coef(object),
              StdErr = se,
              t.value = tval,
              p.value = 2*pt(-abs(tval), df=object$df))
 res <- list(call=object$call,
             coefficients=TAB)
 class(res) <- "summary.tselect"
 res
}


print.summary.tselect <- function(formula, ...)
{
 cat("Call:\n")
 print(formula$call)
 cat("\n")
 printCoefmat(formula$coefficients, P.value=TRUE, has.Pvalue=TRUE)
}


##############################
#Imputation code             #
##############################
install.packages("mice")
library(mice)
require(sampleSelection)

#For RANDHIE Dataset

mice.impute.heckST <- function(y, ry, x,...)
{
```

```
  ry1 <- ry
  data <- data.frame(ry1,x,y)

  selection <- ry1~ logc +
    linc + lfam  + xage + female +
    child  + educdec + idp

  outcome <- y~logc +linc +
    lfam + xage + female +
    child

  mle2 <- tselect(selection, outcome, data=data)
  meane <- coef(mle2)
  sig <- solve(mle2$hessian)
  rv <- t(chol(sig))
  b.star <- meane+rv%*%rnorm(ncol(rv))
  xo <- model.matrix(outcome, data = data)
  xs <- model.matrix(selection, data = data)
  ng <- ncol(xs)
  nb <- ncol(xo)
  igamma <- 1:ng
  ibeta <- max(igamma) + seq(length = nb)
  isigma <- max(ibeta) + 1
  irho <- max(isigma) + 1
  inu <- max(isigma) + 2
  ggamma <- b.star[igamma]
  beta <- b.star[ibeta]
  sigma <- b.star[isigma]
  rho <- b.star[irho]
  nu <- b.star[inu]
  nu <- ifelse(nu<=2,3,nu)
  xb <- xo%*%beta
  xg <- xs%*%ggamma
  ivmT <- ((nu+(-xg[!ry,])^2)/(nu-1))*dt(-xg[!ry,],nu)/pt(-xg[!ry,],nu)
  return(xb[!ry,]-sigma*rho*ivmT+ sigma*rt(sum(!ry),nu))
}

#for SRHS dataset

mice.impute.heckST <- function(y, ry, x,...)
{
  ry1 <- ry
  data <- data.frame(ry1,x,y)
```

```
      selection <- ry1~ i_AGE + i_BMI + i_SEXRecode + PACKYEARS +
ri_GRAINDUSTRecode + h_HOMEPESTICIDERecode + ri_LIVESTOCKRecode +
h_LOCATIONRecode
      outcome <- y~i_AGE + i_BMI + i_SEXRecode + PACKYEARS +
ri_LIVESTOCKRecode

      mle2 <- tselect(selection, outcome, data=data)
      meane <- coef(mle2)
      sig <- solve(mle2$hessian)
      rv <- t(chol(sig))
      b.star <- meane+rv%*%rnorm(ncol(rv))
      xo <- model.matrix(outcome, data = data)
      xs <- model.matrix(selection, data = data)
      ng <- ncol(xs)
      nb <- ncol(xo)
      igamma <- 1:ng
      ibeta <- max(igamma) + seq(length = nb)
      isigma <- max(ibeta) + 1
      irho <- max(isigma) + 1
      inu <- max(isigma) + 2
      ggamma <- b.star[igamma]
      beta <- b.star[ibeta]
      sigma <- b.star[isigma]
      rho <- b.star[irho]
      nu <- b.star[inu]
      nu <- ifelse(nu<=2,3,nu)
      xb <- xo%*%beta
      xg <- xs%*%ggamma
      ivmT <- ((nu+(-xg[!ry,])^2)/(nu-1))*dt(-xg[!ry,],nu)/pt(-xg[!ry,],nu)
      return(xb[!ry,]-sigma*rho*ivmT+ sigma*rt(sum(!ry),nu))
    }


    ######################################################################
    # R-codes for fitting the analysis model and combining results
    ######################################################################

    ttEst <- function(formula, data,start = NULL, verbose = FALSE){
      if (match("gamlss",.packages(),0)==0) require(gamlss)
      mf <- model.frame(formula, data)
      mt <- attr(mf, "terms")
      y <- model.response(mf, "numeric")
      X <- model.matrix(formula, data = data)
      n <- length(y)
      k<-ncol(X)
```

```
  tlog <- function(B){
    beta <- B[1:k]
    sigma <- B[k+1]
    if (sigma < 0)
      return(NA)
    nu <- B[k+2]
    mu <- X%*%beta
    tempval <- vector()
    if( nu >2 ){
      z <- (y-mu)/sigma
      tempval <- log(dt(z,nu))-log(sigma)
      return( -sum(tempval) )
    }
    else return(Inf)
  }
  if (is.null(start))
    ml1 <- gamlss(formula,  data = data,family=TF)
  ak1 <- coef(ml1)
  start<- c(ak1,exp(ml1$sigma.coefficients),exp(ml1$nu.coefficients))
  options(warn=2)
  fit <- try(optim(start,fn=tlog,control=list(maxit=1000),method="BFGS", hessian=T))


  loglike <- fit$value
  nn <- nrow(X); nParam <- length(start)
  aic <- 2*fit$value + 2*nParam
  bic <- 2*fit$value + nParam*log(nn)
  df <- nn-nParam
  coef <- fit$par
  vcov <- solve(fit$hessian)
  h <- colnames(X); hh <- c(h,"sigma","nu")
  colnames(vcov) <- rownames(vcov) <- hh
  names(coef) <- hh
  list(coefficients = coef,vcov = vcov,df=df,aic=aic,nu=tail(coef,1),
       bic=bic,initial.value=start,loglik = -fit$value)
}



tt <- function(formula, ...) UseMethod("tt")


tt.default <- function(formula,data,start = NULL, verbose = FALSE, ...)
{
  mf <- model.frame(formula, data)
  mt <- attr(mf, "terms")
```

```
  y <- model.response(mf, "numeric")
  X <- model.matrix(formula, data = data)
  est <- ttEst(formula, data,start = NULL, verbose = FALSE)
  co <- est$coefficients
  NB <- ncol(X)
  iBeta <- 1:NB
  coe <- co[iBeta]
  est$fitted.values <- as.vector(X %*%coe)
  est$residuals <- y - est$fitted.values
  est$linear.predictors <- est$fitted.values
  aic <- est$aic
  bic <- est$bic
  nu <- est$nu
  initial.values <- est$initial.values
  loglik <- est$loglik
  est$call <- match.call()
  class(est) <- "tt"
  est
}



print.tt <- function(formula, ...)
{
  cat("Call:\n")
  print(formula$call)
  cat("\nCoefficients:\n")
  print(formula$coefficients)
}


summary.tt <- function(object, ...)
{
  se <- sqrt(diag(object$vcov))
  tval <- coef(object) / se
  TAB <- cbind(Estimate = coef(object),
               StdErr = se,
               t.value = tval,
               p.value = 2*pt(-abs(tval), df=object$df))
  res <- list(call=object$call,
              coefficients=TAB)
  class(res) <- "summary.tt"
  res
}

vcov.tt <- function(object){
```

```
  return(object$vcov)
}

coef.tt <- function(object){
  return(object$coef)
}

tt.mids <- function (formula, data, ...) {

  call <- match.call()
  if (!is.mids(data)) stop("The data must have class mids")

  analyses <- as.list(1:data$m)

  for (i in 1:data$m) {
    data.i       <- complete(data, i)
    analyses[[i]] <- tt(formula, data = data.i, ...)
  }

  object <- list(call = call, call1 = data$call,
                 nmis = data$nmis, analyses = analyses)

  return(object)
}

pool.impute <- function (object) {

  if ((m <- length(object$analyses)) < 2)
    stop("At least two imputations are needed for pooling.\n")

  analyses <- object$analyses

  k     <- length(coef(analyses[[1]]))
  names <- names(coef(analyses[[1]]))
  qhat  <- matrix(NA, nrow = m, ncol = k, dimnames = list(1:m,names))
  u     <- array(NA, dim = c(m, k, k),
                 dimnames = list(1:m, names, names))

  for (i in 1:m) {
    fit       <- analyses[[i]]
    qhat[i, ] <- coef(fit)
    u[i, , ]  <- vcov(fit)
  }

  qbar <- apply(qhat, 2, mean)
  ubar <- apply(u, c(2, 3), mean)
```

```
  e <- qhat - matrix(qbar, nrow = m, ncol = k, byrow = TRUE)
  b <- (t(e) %*% e)/(m - 1)
  t <- ubar + (1 + 1/m) * b
  r <- (1 + 1/m) * diag(b/ubar)
  f <- (1 + 1/m) * diag(b/t)
  df <- (m - 1) * (1 + 1/r)^2


  names(r) <- names(df) <- names(f) <- names
  fit <- list(call = call, call1 = object$call, call2 = object$call1,
              nmis = object$nmis, m = m, qhat = qhat, u = u,
              qbar = qbar, ubar = ubar, b = b, t = t, r = r, df = df,
              f = f)
  return(fit)
}

summary.impute <- function(object){

  est  <- object$qbar
  se   <- sqrt(diag(object$t))
  tval <- est/se
  df   <- object$df
  pval <- 2 * pt(abs(tval), df, lower.tail = FALSE)

  coefmat <- cbind(est, se, tval, pval)
  colnames(coefmat) <- c("Estimate", "Std. Error",
                         "t value", "Pr(>|t|)")

  ans <- list( coefficients=coefmat, df=df,
               call=object$call1, fracinfo.miss=object$f )
  #invisible( ans )
  class(ans) <- "summary.impute"
  ans

}

print.summary.impute <- function(object)
{
  if (!is.null(object$call1)){
    cat("Call: ")
    dput(object$call1)
  }
  cat("\nCoefficients:\n")
  printCoefmat(object$coefficients, P.values=T, has.Pvalue=T, signif.legend=T )
  cat("\nFraction of information about the coefficients
      missing due to nonresponse:","\n")
```

```
  print(object$f)
}

########################
# 15% Missing RANDHIE #
########################

attach(RANDHIE_baseline_data_complete_mcar15.csv)
dataset <- data.frame(lnmeddol, lnmeddolMO, logc, disea,
                linc, lfam, xage, female,
                child, educdec, idp)
ab <- mice(dataset, maxit = 30, m = 10, seed = 1234,method=c("heckST", "", "",
"","","","","", "","","", ""))
outcomeEq <- lnmeddol~logc+linc+lfam+xage+female+child
fit <- tt.mids(outcomeEq, data=ab)
ak <- pool.impute(fit)
RANDHIE15_MCAR_Results <- summary.impute(ak)

RANDHIE15_MCAR_Results <- data.frame(RANDHIE15_MCAR_Results$coefficients)
RANDHIE15_MCAR_Results <- RANDHIE15_MCAR_Results[, 1:2]
RANDHIE15_MCAR_LowerCI <- RANDHIE15_MCAR_Results[, 1] -
2.306*RANDHIE15_MCAR_Results[, 2]
RANDHIE15_MCAR_UpperCI <- RANDHIE15_MCAR_Results[, 1] +
2.306*RANDHIE15_MCAR_Results[, 2]
RANDHIE15_MCAR_Results <- cbind(RANDHIE15_MCAR_Results,
data.frame(RANDHIE15_MCAR_LowerCI), data.frame(RANDHIE15_MCAR_UpperCI))
RANDHIE15_MCAR_Results

write.csv(RANDHIE15_MCAR_Results,
"C:/Users/axl807/Downloads/Ogundimu_Outputs/RANDHIE15_MCAR_Results.csv")

attach(RANDHIE_baseline_data_complete_mar15.csv)
dataset <- data.frame(lnmeddol, lnmeddolMO, logc, disea,
                linc, lfam, xage, female,
                child, educdec, idp)
ab <- mice(dataset, maxit = 30, m = 10, seed = 1234,method=c("heckST", "", "",
"","","","","", "","","", ""))
outcomeEq <- lnmeddol~logc+linc+lfam+xage+female+child
fit <- tt.mids(outcomeEq, data=ab)
ak <- pool.impute(fit)
RANDHIE15_MAR_Results <- summary.impute(ak)

RANDHIE15_MAR_Results <- data.frame(RANDHIE15_MAR_Results$coefficients)
RANDHIE15_MAR_Results <- RANDHIE15_MAR_Results[, 1:2]
RANDHIE15_MAR_LowerCI <- RANDHIE15_MAR_Results[, 1] -
2.306*RANDHIE15_MAR_Results[, 2]
```

```
    RANDHIE15_MAR_UpperCI <- RANDHIE15_MAR_Results[, 1] +
2.306*RANDHIE15_MAR_Results[, 2]
    RANDHIE15_MAR_Results <- cbind(RANDHIE15_MAR_Results,
data.frame(RANDHIE15_MAR_LowerCI), data.frame(RANDHIE15_MAR_UpperCI))
    RANDHIE15_MAR_Results

    write.csv(RANDHIE15_MAR_Results,
"C:/Users/axl807/Downloads/Ogundimu_Outputs/RANDHIE15_MAR_Results.csv")


    attach(RANDHIE_baseline_data_complete_mnar15.csv)
    dataset <- data.frame(lnmeddol, lnmeddolMO, logc, disea,
                    linc, lfam, xage, female,
                    child, educdec, idp)
    ab <- mice(dataset, maxit = 30, m = 10, seed = 1234,method=c("heckST", "", "",
"","","","","", "","","", ""))
    outcomeEq <- lnmeddol~logc+linc+lfam+xage+female+child
    fit <- tt.mids(outcomeEq, data=ab)
    ak <- pool.impute(fit)
    RANDHIE15_MNAR_Results <- summary.impute(ak)

    RANDHIE15_MNAR_Results <- data.frame(RANDHIE15_MNAR_Results$coefficients)
    RANDHIE15_MNAR_Results <- RANDHIE15_MNAR_Results[, 1:2]
    RANDHIE15_MNAR_LowerCI <- RANDHIE15_MNAR_Results[, 1] -
2.306*RANDHIE15_MNAR_Results[, 2]
    RANDHIE15_MNAR_UpperCI <- RANDHIE15_MNAR_Results[, 1] +
2.306*RANDHIE15_MNAR_Results[, 2]
    RANDHIE15_MNAR_Results <- cbind(RANDHIE15_MNAR_Results,
data.frame(RANDHIE15_MNAR_LowerCI), data.frame(RANDHIE15_MNAR_UpperCI))
    RANDHIE15_MNAR_Results

    write.csv(RANDHIE15_MNAR_Results,
"C:/Users/axl807/Downloads/Ogundimu_Outputs/RANDHIE15_MNAR_Results.csv")

    ######################
    # 30% Missing RANDHIE #
    ######################

    attach(RANDHIE_baseline_data_complete_mcar30.csv)
    dataset <- data.frame(lnmeddol, lnmeddolMO, logc, disea,
                    linc, lfam, xage, female,
                    child, educdec, idp)
    ab <- mice(dataset, maxit = 30, m = 10, seed = 1234,method=c("heckST", "", "",
"","","","","", "","","", ""))
    outcomeEq <- lnmeddol~logc+linc+lfam+xage+female+child
    fit <- tt.mids(outcomeEq, data=ab)
```

```
ak <- pool.impute(fit)
RANDHIE30_MCAR_Results <- summary.impute(ak)

RANDHIE30_MCAR_Results <- data.frame(RANDHIE30_MCAR_Results$coefficients)
RANDHIE30_MCAR_Results <- RANDHIE30_MCAR_Results[, 1:2]
RANDHIE30_MCAR_LowerCI <- RANDHIE30_MCAR_Results[, 1] -
2.306*RANDHIE30_MCAR_Results[, 2]
RANDHIE30_MCAR_UpperCI <- RANDHIE30_MCAR_Results[, 1] +
2.306*RANDHIE30_MCAR_Results[, 2]
RANDHIE30_MCAR_Results <- cbind(RANDHIE30_MCAR_Results,
data.frame(RANDHIE30_MCAR_LowerCI), data.frame(RANDHIE30_MCAR_UpperCI))
RANDHIE30_MCAR_Results

write.csv(RANDHIE30_MCAR_Results,
"C:/Users/axl807/Downloads/Ogundimu_Outputs/RANDHIE30_MCAR_Results.csv")

attach(RANDHIE_baseline_data_complete_mar30.csv)
dataset <- data.frame(lnmeddol, lnmeddolMO, logc, disea,
                      linc, lfam, xage, female,
                      child, educdec, idp)
ab <- mice(dataset, maxit = 30, m = 10, seed = 1234,method=c("heckST", "", "",
"","","","","","","","","","","",""))
outcomeEq <- lnmeddol~logc+linc+lfam+xage+female+child
fit <- tt.mids(outcomeEq, data=ab)
ak <- pool.impute(fit)
RANDHIE30_MAR_Results <- summary.impute(ak)

RANDHIE30_MAR_Results <- data.frame(RANDHIE30_MAR_Results$coefficients)
RANDHIE30_MAR_Results <- RANDHIE30_MAR_Results[, 1:2]
RANDHIE30_MAR_LowerCI <- RANDHIE30_MAR_Results[, 1] -
2.306*RANDHIE30_MAR_Results[, 2]
RANDHIE30_MAR_UpperCI <- RANDHIE30_MAR_Results[, 1] +
2.306*RANDHIE30_MAR_Results[, 2]
RANDHIE30_MAR_Results <- cbind(RANDHIE30_MAR_Results,
data.frame(RANDHIE30_MAR_LowerCI), data.frame(RANDHIE30_MAR_UpperCI))
RANDHIE30_MAR_Results

write.csv(RANDHIE30_MAR_Results,
"C:/Users/axl807/Downloads/Ogundimu_Outputs/RANDHIE30_MAR_Results.csv")


attach(RANDHIE_baseline_data_complete_mnar30.csv)
dataset <- data.frame(lnmeddol, lnmeddolMO, logc, disea,
                      linc, lfam, xage, female,
                      child, educdec, idp)
```

```
    ab <- mice(dataset, maxit = 30, m = 10, seed = 1234,method=c("heckST", "", "",
"","","","","","","","","", ""))
    outcomeEq <- lnmeddol~logc+linc+lfam+xage+female+child
    fit <- tt.mids(outcomeEq, data=ab)
    ak <- pool.impute(fit)
    RANDHIE30_MNAR_Results <- summary.impute(ak)

    RANDHIE30_MNAR_Results <- data.frame(RANDHIE30_MNAR_Results$coefficients)
    RANDHIE30_MNAR_Results <- RANDHIE30_MNAR_Results[, 1:2]
    RANDHIE30_MNAR_LowerCI <- RANDHIE30_MNAR_Results[, 1] -
2.306*RANDHIE30_MNAR_Results[, 2]
    RANDHIE30_MNAR_UpperCI <- RANDHIE30_MNAR_Results[, 1] +
2.306*RANDHIE30_MNAR_Results[, 2]
    RANDHIE30_MNAR_Results <- cbind(RANDHIE30_MNAR_Results,
data.frame(RANDHIE30_MNAR_LowerCI), data.frame(RANDHIE30_MNAR_UpperCI))
    RANDHIE30_MNAR_Results

    write.csv(RANDHIE30_MNAR_Results,
"C:/Users/axl807/Downloads/Ogundimu_Outputs/RANDHIE30_MNAR_Results.csv")

    ########################
    # 50% Missing RANDHIE #
    ########################

    attach(RANDHIE_baseline_data_complete_mcar50.csv)
    dataset <- data.frame(lnmeddol, lnmeddolMO, logc, disea,
                linc, lfam, xage, female,
                child, educdec, idp)
    ab <- mice(dataset, maxit = 30, m = 10, seed = 1234,method=c("heckST", "", "",
"","","","","","","","","", ""))
    outcomeEq <- lnmeddol~logc+linc+lfam+xage+female+child
    fit <- tt.mids(outcomeEq, data=ab)
    ak <- pool.impute(fit)
    RANDHIE50_MCAR_Results <- summary.impute(ak)

    RANDHIE50_MCAR_Results <- data.frame(RANDHIE50_MCAR_Results$coefficients)
    RANDHIE50_MCAR_Results <- RANDHIE50_MCAR_Results[, 1:2]
    RANDHIE50_MCAR_LowerCI <- RANDHIE50_MCAR_Results[, 1] -
2.306*RANDHIE50_MCAR_Results[, 2]
    RANDHIE50_MCAR_UpperCI <- RANDHIE50_MCAR_Results[, 1] +
2.306*RANDHIE50_MCAR_Results[, 2]
    RANDHIE50_MCAR_Results <- cbind(RANDHIE50_MCAR_Results,
data.frame(RANDHIE50_MCAR_LowerCI), data.frame(RANDHIE50_MCAR_UpperCI))
    RANDHIE50_MCAR_Results
```

```
    write.csv(RANDHIE50_MCAR_Results,
"C:/Users/axl807/Downloads/Ogundimu_Outputs/RANDHIE50_MCAR_Results.csv")

    attach(RANDHIE_baseline_data_complete_mar50.csv)
    dataset <- data.frame(lnmeddol, lnmeddolMO, logc, disea,
                    linc, lfam, xage, female,
                    child, educdec, idp)
    ab <- mice(dataset, maxit = 30, m = 10, seed = 1234,method=c("heckST", "", "",
"","","","","","","","","", ""))
    outcomeEq <- lnmeddol~logc+linc+lfam+xage+female+child
    fit <- tt.mids(outcomeEq, data=ab)
    ak <- pool.impute(fit)
    RANDHIE50_MAR_Results <- summary.impute(ak)

    RANDHIE50_MAR_Results <- data.frame(RANDHIE50_MAR_Results$coefficients)
    RANDHIE50_MAR_Results <- RANDHIE50_MAR_Results[, 1:2]
    RANDHIE50_MAR_LowerCI <- RANDHIE50_MAR_Results[, 1] -
2.306*RANDHIE50_MAR_Results[, 2]
    RANDHIE50_MAR_UpperCI <- RANDHIE50_MAR_Results[, 1] +
2.306*RANDHIE50_MAR_Results[, 2]
    RANDHIE50_MAR_Results <- cbind(RANDHIE50_MAR_Results,
data.frame(RANDHIE50_MAR_LowerCI), data.frame(RANDHIE50_MAR_UpperCI))
    RANDHIE50_MAR_Results

    write.csv(RANDHIE50_MAR_Results,
"C:/Users/axl807/Downloads/Ogundimu_Outputs/RANDHIE50_MAR_Results.csv")


    attach(RANDHIE_baseline_data_complete_mnar50.csv)
    dataset <- data.frame(lnmeddol, lnmeddolMO, logc, disea,
                    linc, lfam, xage, female,
                    child, educdec, idp)
    ab <- mice(dataset, maxit = 30, m = 10, seed = 1234,method=c("heckST", "", "",
"","","","","","","","","", ""))
    outcomeEq <- lnmeddol~logc+linc+lfam+xage+female+child
    fit <- tt.mids(outcomeEq, data=ab)
    ak <- pool.impute(fit)
    RANDHIE50_MNAR_Results <- summary.impute(ak)

    RANDHIE50_MNAR_Results <- data.frame(RANDHIE50_MNAR_Results$coefficients)
    RANDHIE50_MNAR_Results <- RANDHIE50_MNAR_Results[, 1:2]
    RANDHIE50_MNAR_LowerCI <- RANDHIE50_MNAR_Results[, 1] -
2.306*RANDHIE50_MNAR_Results[, 2]
    RANDHIE50_MNAR_UpperCI <- RANDHIE50_MNAR_Results[, 1] +
2.306*RANDHIE50_MNAR_Results[, 2]
```

```
    RANDHIE50_MNAR_Results <- cbind(RANDHIE50_MNAR_Results,
data.frame(RANDHIE50_MNAR_LowerCI), data.frame(RANDHIE50_MNAR_UpperCI))
    RANDHIE50_MNAR_Results

    write.csv(RANDHIE50_MNAR_Results,
"C:/Users/axl807/Downloads/Ogundimu_Outputs/RANDHIE50_MNAR_Results.csv")



    ####################
    # 15% Missing SRHS #
    ####################

    attach(SRHS15_MCAR)
    dataset <- data.frame(c_FEV1OBSER, c_FEV1OBSERMO, i_AGE, i_BMI, i_SEXRecode,
PACKYEARS, ri_GRAINDUSTRecode, h_HOMEPESTICIDERecode,
ri_LIVESTOCKRecode, h_LOCATIONRecode)
    ab <- mice(dataset, maxit = 30, m = 10, seed = 1234,method=c("heckST", "", "",
"","","","","","", "",""))
    outcomeEq <- c_FEV1OBSER~i_AGE + i_BMI + i_SEXRecode + PACKYEARS +
ri_LIVESTOCKRecode
    fit <- tt.mids(outcomeEq, data=ab)
    ak <- pool.impute(fit)
    SRHS15_MCAR_Results <- summary.impute(ak)

    SRHS15_MCAR_Results <- data.frame(SRHS15_MCAR_Results$coefficients[, 1:2])
    SRHS15_MCAR_LowerCI <- SRHS15_MCAR_Results[, 1] -
2.306*SRHS15_MCAR_Results[, 2]
    SRHS15_MCAR_UpperCI <- SRHS15_MCAR_Results[, 1] +
2.306*SRHS15_MCAR_Results[, 2]
    SRHS15_MCAR_Results <- cbind(SRHS15_MCAR_Results,
data.frame(SRHS15_MCAR_LowerCI), data.frame(SRHS15_MCAR_UpperCI))
    SRHS15_MCAR_Results

    write.csv(SRHS15_MCAR_Results,
"C:/Users/axl807/Downloads/Ogundimu_Outputs/SRHS15_MCAR_Results.csv")

    attach(SRHS15_MAR)
    dataset <- data.frame(c_FEV1OBSER, c_FEV1OBSERMO, i_AGE, i_BMI, i_SEXRecode,
PACKYEARS, ri_GRAINDUSTRecode, h_HOMEPESTICIDERecode,
ri_LIVESTOCKRecode, h_LOCATIONRecode)
    ab <- mice(dataset, maxit = 30, m = 10, seed = 1234,method=c("heckST", "", "",
"","","","","","", "",""))
    outcomeEq <- c_FEV1OBSER~i_AGE + i_BMI + i_SEXRecode + PACKYEARS +
ri_LIVESTOCKRecode
    fit <- tt.mids(outcomeEq, data=ab)
```

```
    ak <- pool.impute(fit)
    SRHS15_MAR_Results <- summary.impute(ak)

    SRHS15_MAR_Results <- data.frame(SRHS15_MAR_Results$coefficients[, 1:2])
    SRHS15_MAR_LowerCI <- SRHS15_MAR_Results[, 1] - 2.306*SRHS15_MAR_Results[,
2]
    SRHS15_MAR_UpperCI <- SRHS15_MAR_Results[, 1] + 2.306*SRHS15_MAR_Results[,
2]
    SRHS15_MAR_Results <- cbind(SRHS15_MAR_Results,
data.frame(SRHS15_MAR_LowerCI), data.frame(SRHS15_MAR_UpperCI))
    SRHS15_MAR_Results

    write.csv(SRHS15_MAR_Results,
"C:/Users/axl807/Downloads/Ogundimu_Outputs/SRHS15_MAR_Results.csv")

    attach(SRHS15_MNAR)
    dataset <- data.frame(c_FEV1OBSER, c_FEV1OBSERMO, i_AGE, i_BMI, i_SEXRecode,
PACKYEARS, ri_GRAINDUSTRecode, h_HOMEPESTICIDERecode,
ri_LIVESTOCKRecode, h_LOCATIONRecode)
    ab <- mice(dataset, maxit = 30, m = 10, seed = 1234,method=c("heckST", "", "",
"","","","","", "","",""))
    outcomeEq <- c_FEV1OBSER~i_AGE + i_BMI + i_SEXRecode + PACKYEARS +
ri_LIVESTOCKRecode
    fit <- tt.mids(outcomeEq, data=ab)
    ak <- pool.impute(fit)
    SRHS15_MNAR_Results <- summary.impute(ak)
    SRHS15_MNAR_Results <- data.frame(SRHS15_MNAR_Results$coefficients)

    SRHS15_MNAR_Results <- SRHS15_MNAR_Results[, 1:2]
    SRHS15_MNAR_LowerCI <- SRHS15_MNAR_Results[, 1] -
2.306*SRHS15_MNAR_Results[, 2]
    SRHS15_MNAR_UpperCI <- SRHS15_MNAR_Results[, 1] +
2.306*SRHS15_MNAR_Results[, 2]
    SRHS15_MNAR_Results <- cbind(SRHS15_MNAR_Results,
data.frame(SRHS15_MNAR_LowerCI), data.frame(SRHS15_MNAR_UpperCI))
    SRHS15_MNAR_Results

    write.csv(SRHS15_MNAR_Results,
"C:/Users/axl807/Downloads/Ogundimu_Outputs/SRHS15_MNAR_Results.csv")

    ####################
    # 30% Missing SRHS #
    ####################
    attach(SRHS30_MCAR)
```

```
    dataset <- data.frame(c_FEV1OBSER, c_FEV1OBSERMO, i_AGE, i_BMI, i_SEXRecode,
PACKYEARS, ri_GRAINDUSTRecode, h_HOMEPESTICIDERecode,
ri_LIVESTOCKRecode, h_LOCATIONRecode)
    ab <- mice(dataset, maxit = 30, m = 10, seed = 1234,method=c("heckST", "", "",
"","","","","","","",""))
    outcomeEq <- c_FEV1OBSER~i_AGE + i_BMI + i_SEXRecode + PACKYEARS +
ri_LIVESTOCKRecode
    fit <- tt.mids(outcomeEq, data=ab)
    ak <- pool.impute(fit)
    SRHS30_MCAR_Results <- summary.impute(ak)

    SRHS30_MCAR_Results <- data.frame(SRHS30_MCAR_Results$coefficients[, 1:2])
    SRHS30_MCAR_LowerCI <- SRHS30_MCAR_Results[, 1] -
2.306*SRHS30_MCAR_Results[, 2]
    SRHS30_MCAR_UpperCI <- SRHS30_MCAR_Results[, 1] +
2.306*SRHS30_MCAR_Results[, 2]
    SRHS30_MCAR_Results <- cbind(SRHS30_MCAR_Results,
data.frame(SRHS30_MCAR_LowerCI), data.frame(SRHS30_MCAR_UpperCI))
    SRHS30_MCAR_Results

    write.csv(SRHS30_MCAR_Results,
"C:/Users/axl807/Downloads/Ogundimu_Outputs/SRHS30_MCAR_Results.csv")

    attach(SRHS30_MAR)
    dataset <- data.frame(c_FEV1OBSER, c_FEV1OBSERMO, i_AGE, i_BMI, i_SEXRecode,
PACKYEARS, ri_GRAINDUSTRecode, h_HOMEPESTICIDERecode,
ri_LIVESTOCKRecode, h_LOCATIONRecode)
    ab <- mice(dataset, maxit = 30, m = 10, seed = 1234,method=c("heckST", "", "",
"","","","","","","",""))
    outcomeEq <- c_FEV1OBSER~i_AGE + i_BMI + i_SEXRecode + PACKYEARS +
ri_LIVESTOCKRecode
    fit <- tt.mids(outcomeEq, data=ab)
    ak <- pool.impute(fit)
    SRHS30_MAR_Results <- summary.impute(ak)

    SRHS30_MAR_Results <- data.frame(SRHS30_MAR_Results$coefficients[, 1:2])
    SRHS30_MAR_LowerCI <- SRHS30_MAR_Results[, 1] - 2.306*SRHS30_MAR_Results[,
2]
    SRHS30_MAR_UpperCI <- SRHS30_MAR_Results[, 1] + 2.306*SRHS30_MAR_Results[,
2]
    SRHS30_MAR_Results <- cbind(SRHS30_MAR_Results,
data.frame(SRHS30_MAR_LowerCI), data.frame(SRHS30_MAR_UpperCI))
    SRHS30_MAR_Results

    write.csv(SRHS30_MAR_Results,
"C:/Users/axl807/Downloads/Ogundimu_Outputs/SRHS30_MAR_Results.csv")
```

```
attach(SRHS30_MNAR)
dataset <- data.frame(c_FEV1OBSER, c_FEV1OBSERMO, i_AGE, i_BMI, i_SEXRecode,
PACKYEARS, ri_GRAINDUSTRecode, h_HOMEPESTICIDERecode,
ri_LIVESTOCKRecode, h_LOCATIONRecode)
ab <- mice(dataset, maxit = 30, m = 10, seed = 1234,method=c("heckST", "", "",
"","","","","","", "",""))
outcomeEq <- c_FEV1OBSER~i_AGE + i_BMI + i_SEXRecode + PACKYEARS +
ri_LIVESTOCKRecode
fit <- tt.mids(outcomeEq, data=ab)
ak <- pool.impute(fit)
SRHS30_MNAR_Results <- summary.impute(ak)

SRHS30_MNAR_Results <- data.frame(SRHS30_MNAR_Results$coefficients[, 1:2])
SRHS30_MNAR_LowerCI <- SRHS30_MNAR_Results[, 1] -
2.306*SRHS30_MNAR_Results[, 2]
SRHS30_MNAR_UpperCI <- SRHS30_MNAR_Results[, 1] +
2.306*SRHS30_MNAR_Results[, 2]
SRHS30_MNAR_Results <- cbind(SRHS30_MNAR_Results,
data.frame(SRHS30_MNAR_LowerCI), data.frame(SRHS30_MNAR_UpperCI))
SRHS30_MNAR_Results

write.csv(SRHS30_MNAR_Results,
"C:/Users/axl807/Downloads/Ogundimu_Outputs/SRHS30_MNAR_Results.csv")

####################
# 50% Missing SRHS #
####################

attach(SRHS50_MCAR)
dataset <- data.frame(c_FEV1OBSER, c_FEV1OBSERMO, i_AGE, i_BMI, i_SEXRecode,
PACKYEARS, ri_GRAINDUSTRecode, h_HOMEPESTICIDERecode,
ri_LIVESTOCKRecode, h_LOCATIONRecode)
ab <- mice(dataset, maxit = 30, m = 10, seed = 1234,method=c("heckST", "", "",
"","","","","","", "",""))
outcomeEq <- c_FEV1OBSER~i_AGE + i_BMI + i_SEXRecode + PACKYEARS +
ri_LIVESTOCKRecode
fit <- tt.mids(outcomeEq, data=ab)
ak <- pool.impute(fit)
SRHS50_MCAR_Results <- summary.impute(ak)
SRHS50_MCAR_Results <- data.frame(SRHS50_MCAR_Results$coefficients)

SRHS50_MCAR_Results <- SRHS50_MCAR_Results[, 1:2]
SRHS50_MCAR_LowerCI <- SRHS50_MCAR_Results[, 1] -
2.306*SRHS50_MCAR_Results[, 2]
```

```
SRHS50_MCAR_UpperCI <- SRHS50_MCAR_Results[, 1] +
2.306*SRHS50_MCAR_Results[, 2]
SRHS50_MCAR_Results <- cbind(SRHS50_MCAR_Results,
data.frame(SRHS50_MCAR_LowerCI), data.frame(SRHS50_MCAR_UpperCI))
SRHS50_MCAR_Results

write.csv(SRHS50_MCAR_Results,
"C:/Users/axl807/Downloads/Ogundimu_Outputs/SRHS50_MCAR_Results.csv")

attach(SRHS50_MAR)
dataset <- data.frame(c_FEV1OBSER, c_FEV1OBSERMO, i_AGE, i_BMI, i_SEXRecode,
PACKYEARS, ri_GRAINDUSTRecode, h_HOMEPESTICIDERecode,
ri_LIVESTOCKRecode, h_LOCATIONRecode)
ab <- mice(dataset, maxit = 30, m = 10, seed = 1234,method=c("heckST", "", "",
"","","","","", "",""))
outcomeEq <- c_FEV1OBSER~i_AGE + i_BMI + i_SEXRecode + PACKYEARS +
ri_LIVESTOCKRecode
fit <- tt.mids(outcomeEq, data=ab)
ak <- pool.impute(fit)
SRHS50_MAR_Results <- summary.impute(ak)
SRHS50_MAR_Results <- data.frame(SRHS50_MAR_Results$coefficients)

SRHS50_MAR_Results <- SRHS50_MAR_Results[, 1:2]
SRHS50_MAR_LowerCI <- SRHS50_MAR_Results[, 1] - 2.306*SRHS50_MAR_Results[,
2]
SRHS50_MAR_UpperCI <- SRHS50_MAR_Results[, 1] + 2.306*SRHS50_MAR_Results[,
2]
SRHS50_MAR_Results <- cbind(SRHS50_MAR_Results,
data.frame(SRHS50_MAR_LowerCI), data.frame(SRHS50_MAR_UpperCI))
SRHS50_MAR_Results

write.csv(SRHS50_MAR_Results,
"C:/Users/axl807/Downloads/Ogundimu_Outputs/SRHS50_MAR_Results.csv")

attach(SRHS50_MNAR)
dataset <- data.frame(c_FEV1OBSER, c_FEV1OBSERMO, i_AGE, i_BMI, i_SEXRecode,
PACKYEARS, ri_GRAINDUSTRecode, h_HOMEPESTICIDERecode,
ri_LIVESTOCKRecode, h_LOCATIONRecode)
ab <- mice(dataset, maxit = 30, m = 10, seed = 1234,method=c("heckST", "", "",
"","","","","", "",""))
outcomeEq <- c_FEV1OBSER~i_AGE + i_BMI + i_SEXRecode + PACKYEARS +
ri_LIVESTOCKRecode
fit <- tt.mids(outcomeEq, data=ab)
ak <- pool.impute(fit)
SRHS50_MNAR_Results <- summary.impute(ak)
SRHS50_MNAR_Results <- data.frame(SRHS50_MNAR_Results$coefficients)
```

```
    SRHS50_MNAR_Results <- SRHS50_MNAR_Results[, 1:2]
    SRHS50_MNAR_LowerCI <- SRHS50_MNAR_Results[, 1] -
2.306*SRHS50_MNAR_Results[, 2]
    SRHS50_MNAR_UpperCI <- SRHS50_MNAR_Results[, 1] +
2.306*SRHS50_MNAR_Results[, 2]
    SRHS50_MNAR_Results <- cbind(SRHS50_MNAR_Results,
data.frame(SRHS50_MNAR_LowerCI), data.frame(SRHS50_MNAR_UpperCI))
    SRHS50_MNAR_Results

    write.csv(SRHS50_MNAR_Results,
"C:/Users/axl807/Downloads/Ogundimu_Outputs/SRHS50_MNAR_Results.csv")
```

B.3 Simulation Codes

```
    #Generating 15% Missing, MCAR, MAR, MNAR, Complete Case Analysis
    f <- function(seed)
    {
     #Generate the independent variables separately using rnorm() or rbinom()
     i_AGE <- runif(1495, min = 18, max = 83)
     i_BMI <- rgamma(1495,  shape = 28.2618/1.12 , rate = 1/1.12)
     i_SEXRecode <- sample(c(0,1),1495, replace = TRUE, prob = c(711/1495, 784/1495))
     PACKYEARS <-rgamma(1495, shape = 7.041/23.01 , rate = 1/23.01)
     ri_GRAINDUSTRecode <-sample(c(0,1), 1495, replace = TRUE, prob = c(428/1495,
1067/1495))
     h_HOMEPESTICIDERecode <-sample(c(0,1), 1495, replace = TRUE, prob =
c(1124/1495, 371/1495))
     ri_LIVESTOCKRecode <- sample(c(0,1),1495, replace = TRUE, prob = c(670/1495,
825/1495))
     h_LOCATIONRecode <- sample(c(0,1),1495, replace = TRUE, prob = c(732/1495,
763/1495))
     c_FEV1OBSER <- rnorm(1495, mean = 3.100334)
     SRHS_Simulation <- data.frame(c_FEV1OBSER, i_AGE, i_BMI, i_SEXRecode,
PACKYEARS, ri_GRAINDUSTRecode, h_HOMEPESTICIDERecode,
ri_LIVESTOCKRecode, h_LOCATIONRecode)
     SRHS_Simulation$c_FEV1OBSER15.mcar <- SRHS_Simulation$c_FEV1OBSER
     SRHS_Simulation$c_FEV1OBSER15.mar <- SRHS_Simulation$c_FEV1OBSER
     SRHS_Simulation$c_FEV1OBSER15.mnar <-SRHS_Simulation$c_FEV1OBSER
     mcar <- rbinom(n = 1495, size = 1, prob = 1/(1+exp(1-2.73)))
     mar <- rbinom(n = 1495, size = 1, prob = 1/(1+exp(1-0.053*SRHS_Simulation$i_AGE)))
     mnar <- rbinom(n = 1495, size = 1, prob = 1/(1+exp(1-
0.93*SRHS_Simulation$c_FEV1OBSER)))
     for (i in 1:1495){
```

```
        if (mcar[i]==0){
          SRHS_Simulation$c_FEV1OBSER15.mcar[i] <- NA
        }
        #if (mar[i]==0){
          #SRHS_Simulation$c_FEV1OBSER15.mar[i] <- NA
        #}
        #if (mnar[i]==0){
          #SRHS_Simulation$c_FEV1OBSER15.mnar[i] <- NA
        #}
      }
      #fit15.mcar <-
lm(c_FEV1OBSER15.mcar~i_AGE+i_BMI+i_SEXRecode+PACKYEARS+ri_LIVESTOCKRe
code, data=SRHS_Simulation)
      fit15.mar <-
lm(c_FEV1OBSER15.mar~i_AGE+i_BMI+i_SEXRecode+PACKYEARS+ri_LIVESTOCKRec
ode, data=SRHS_Simulation)
      #fit15.mnar <-
lm(c_FEV1OBSER15.mnar~i_AGE+i_BMI+i_SEXRecode+PACKYEARS+ri_LIVESTOCKRe
code, data=SRHS_Simulation)

      return(summary(fit15.mcar))
      #return(summary(fit15.mar))
      #return(summary(fit15.mnar))
    }

    summaries_CompleteCase15 <- lapply(1:1000,f)
    coef_CompleteCase15 <- lapply(summaries_CompleteCase15, coef)
    sd_CompleteCase15 <- lapply(summaries_CompleteCase15, function(data){
      data$coefficient[,2]
    })
    t_coef <- unlist(coef_CompleteCase15 )
    t_sd <- unlist(sd_CompleteCase15 )
    M_coef15mcar <- matrix(data = t_coef,nrow = 1000, ncol = 6, byrow = TRUE)
    M_sd15mcar <- matrix(data = t_sd,nrow = 1000, ncol = 6, byrow = TRUE)
    #M_coef15mar <- matrix(data = t_coef,nrow = 1000, ncol = 6, byrow = TRUE)
    #M_sd15mar <- matrix(data = t_sd,nrow = 1000, ncol = 6, byrow = TRUE)
    #M_coef15mnar <- matrix(data = t_coef,nrow = 1000, ncol = 6, byrow = TRUE)
    #M_sd15mcar <- matrix(data = t_sd,nrow = 1000, ncol = 6, byrow = TRUE)
    colnames(M_coef15mcar) <- c("beta0", "beta_iAGE", "beta_iBMI", "beta_iSEX",
"beta_PACKYEARS", "beta_LIVESTOCK")
    colnames(M_sd15mcar) <- c("beta0", "beta_iAGE", "beta_iBMI", "beta_iSEX",
"beta_PACKYEARS", "beta_LIVESTOCK")
    #colnames(M_coef15mar) <- c("beta0", "beta_iAGE", "beta_iBMI", "beta_iSEX",
"beta_PACKYEARS", "beta_LIVESTOCK")
    #colnames(M_sd15mar) <- c("beta0", "beta_iAGE", "beta_iBMI", "beta_iSEX",
"beta_PACKYEARS", "beta_LIVESTOCK")
```

```
    #colnames(M_coef15mnar) <- c("beta0", "beta_iAGE", "beta_iBMI", "beta_iSEX",
"beta_PACKYEARS", "beta_LIVESTOCK")
    #colnames(M_sd15mnar) <- c("beta0", "beta_iAGE", "beta_iBMI", "beta_iSEX",
"beta_PACKYEARS", "beta_LIVESTOCK")
    MeanBeta.15mcar <- colMeans(data.frame(M_coef15mcar))
    MeanSD.15mcar <- colMeans(data.frame(M_sd15mcar))
    #MeanBeta.15mar <- colMeans(data.frame(M_coef15mar))
    #MeanSD.15mar <- colMeans(data.frame(M_sd15mar))
    #MeanBeta.15mnar <- colMeans(data.frame(M_coef15mnar))
    #MeanSD.15mnar <- colMeans(data.frame(M_sd15mnar))

    write.csv(M_coef15mcar, "E:/MatricesOf1000Coef&SD/M_coef15mcar.csv")
    write.csv(M_sd15mcar, "E:/MatricesOf1000Coef&SD/M_sd15mcar.csv")
    write.csv(MeanBeta.15mcar, "E:/MatricesOf1000Coef&SD/MeanBeta.15mcar.csv")
    write.csv(MeanSD.15mcar, "E:/MatricesOf1000Coef&SD/MeanSD.15mcar.csv")

    #write.csv(M_coef15mar, "E:/MatricesOf1000Coef&SD/M_coef15mar.csv")
    #write.csv(M_sd15mar, "E:/MatricesOf1000Coef&SD/M_sd15mar.csv")
    #write.csv(MeanBeta.15mar, "E:/MatricesOf1000Coef&SD/MeanBeta.15mar.csv")
    #write.csv(MeanSD.15mar, "E:/MatricesOf1000Coef&SD/MeanSD.15mar.csv")

    #write.csv(M_coef15mnar, "E:/MatricesOf1000Coef&SD/M_coef15mnar.csv")
    #write.csv(M_sd15mnar, "E:/MatricesOf1000Coef&SD/M_sd15mnar.csv")
    #write.csv(MeanBeta.15mnar, "E:/MatricesOf1000Coef&SD/MeanBeta.15mar.csv")
    #write.csv(MeanSD.15mnar, "E:/MatricesOf1000Coef&SD/MeanSD.15mar.csv")

    f <- function(seed)
    {
     #Generate the independent variables separately using rnorm() or rbinom()
     i_AGE <- runif(1495, min = 18, max = 83)
     i_BMI <- rgamma(1495,  shape = 28.2618/1.12 , rate = 1/1.12)
     i_SEXRecode <- sample(c(0,1),1495, replace = TRUE, prob = c(711/1495, 784/1495))
     PACKYEARS <-rgamma(1495, shape = 7.041/23.01 , rate = 1/23.01)
     ri_GRAINDUSTRecode <-sample(c(0,1), 1495, replace = TRUE, prob = c(428/1495,
1067/1495))
     h_HOMEPESTICIDERecode <-sample(c(0,1), 1495, replace = TRUE, prob =
c(1124/1495, 371/1495))
     ri_LIVESTOCKRecode <- sample(c(0,1),1495, replace = TRUE, prob = c(670/1495,
825/1495))
     h_LOCATIONRecode <- sample(c(0,1),1495, replace = TRUE, prob = c(732/1495,
763/1495))
     c_FEV1OBSER <- rnorm(1495, mean = 3.100334)
     SRHS_Simulation <- data.frame(c_FEV1OBSER, i_AGE, i_BMI, i_SEXRecode,
PACKYEARS, ri_GRAINDUSTRecode, h_HOMEPESTICIDERecode,
ri_LIVESTOCKRecode, h_LOCATIONRecode)
     SRHS_Simulation$c_FEV1OBSER30.mcar <- SRHS_Simulation$c_FEV1OBSER
```

```
      SRHS_Simulation$c_FEV1OBSER30.mar <- SRHS_Simulation$c_FEV1OBSER
      SRHS_Simulation$c_FEV1OBSER30.mnar <-SRHS_Simulation$c_FEV1OBSER
      mcar <- rbinom(n = 1495, size = 1, prob = 1/(1+exp(1-1.9)))
      mar <- rbinom(n = 1495, size = 1, prob = 1/(1+exp(1-0.035*SRHS_Simulation$i_AGE)))
      mnar <- rbinom(n = 1495, size = 1, prob = 1/(1+exp(1-
0.6*SRHS_Simulation$c_FEV1OBSER)))
      for (i in 1:1495){
       #if (mcar[i]==0){
        #SRHS_Simulation$c_FEV1OBSER30.mcar[i] <- NA
       #}
       #if (mar[i]==0){
        #SRHS_Simulation$c_FEV1OBSER30.mar[i] <- NA
       #}
       if (mnar[i]==0){
        SRHS_Simulation$c_FEV1OBSER30.mnar[i] <- NA
       }
      }
      #fit30.mcar <-
lm(c_FEV1OBSER30.mcar~i_AGE+i_BMI+i_SEXRecode+PACKYEARS+ri_LIVESTOCKRe
code, data=SRHS_Simulation)
      #fit30.mar <-
lm(c_FEV1OBSER30.mar~i_AGE+i_BMI+i_SEXRecode+PACKYEARS+ri_LIVESTOCKRec
ode, data=SRHS_Simulation)
      fit30.mnar <-
lm(c_FEV1OBSER30.mnar~i_AGE+i_BMI+i_SEXRecode+PACKYEARS+ri_LIVESTOCKRe
code, data=SRHS_Simulation)

      #return(summary(fit30.mcar))
      #return(summary(fit30.mar))
      return(summary(fit30.mnar))
     }

     summaries_CompleteCase30 <- lapply(1:1000,f)
     coef_CompleteCase30 <- lapply(summaries_CompleteCase30, coef)
     sd_CompleteCase30 <- lapply(summaries_CompleteCase30, function(data){
      data$coefficient[,2]
     })
     t_coef <- unlist(coef_CompleteCase30)
     t_sd <- unlist(sd_CompleteCase30)

     #M_coef30mcar <- matrix(data = t_coef,nrow = 1000, ncol = 6, byrow = TRUE)
     #M_sd30mcar <- matrix(data = t_sd,nrow = 1000, ncol = 6, byrow = TRUE)
     #M_coef30mar <- matrix(data = t_coef,nrow = 1000, ncol = 6, byrow = TRUE)
     #M_sd30mar <- matrix(data = t_sd,nrow = 1000, ncol = 6, byrow = TRUE)
     M_coef30mnar <- matrix(data = t_coef,nrow = 1000, ncol = 6, byrow = TRUE)
     M_sd30mnar <- matrix(data = t_sd,nrow = 1000, ncol = 6, byrow = TRUE)
```

```
    #colnames(M_coef30mcar) <- c("beta0", "beta_iAGE", "beta_iBMI", "beta_iSEX",
"beta_PACKYEARS", "beta_LIVESTOCK")
    #colnames(M_sd30mcar) <- c("beta0", "beta_iAGE", "beta_iBMI", "beta_iSEX",
"beta_PACKYEARS", "beta_LIVESTOCK")
    #colnames(M_coef30mar) <- c("beta0", "beta_iAGE", "beta_iBMI", "beta_iSEX",
"beta_PACKYEARS", "beta_LIVESTOCK")
    #colnames(M_sd30mar) <- c("beta0", "beta_iAGE", "beta_iBMI", "beta_iSEX",
"beta_PACKYEARS", "beta_LIVESTOCK")
    colnames(M_coef30mnar) <- c("beta0", "beta_iAGE", "beta_iBMI", "beta_iSEX",
"beta_PACKYEARS", "beta_LIVESTOCK")
    colnames(M_sd30mnar) <- c("beta0", "beta_iAGE", "beta_iBMI", "beta_iSEX",
"beta_PACKYEARS", "beta_LIVESTOCK")

    #MeanBeta.30mcar <- colMeans(data.frame(M_coef30mcar))
    #MeanSD.30mcar <- colMeans(data.frame(M_sd30mcar))
    #MeanBeta.30mar <- colMeans(data.frame(M_coef30mar))
    #MeanSD.30mar <- colMeans(data.frame(M_sd30mar))
    MeanBeta.30mnar <- colMeans(data.frame(M_coef30mnar))
    MeanSD.30mnar <- colMeans(data.frame(M_sd30mnar))

    #write.csv(M_coef30mcar, "E:/MatricesOf1000Coef&SD/M_coef30mcar.csv")
    #write.csv(M_sd30mcar, "E:/MatricesOf1000Coef&SD/M_sd30mcar.csv")
    #write.csv(MeanBeta.30mcar, "E:/MatricesOf1000Coef&SD/MeanBeta.30mcar.csv")
    #write.csv(MeanSD.30mcar, "E:/MatricesOf1000Coef&SD/MeanSD.30mcar.csv")

    #write.csv(M_coef30mar, "E:/MatricesOf1000Coef&SD/M_coef30mar.csv")
    #write.csv(M_sd30mar, "E:/MatricesOf1000Coef&SD/M_sd30mar.csv")
    #write.csv(MeanBeta.30mar, "E:/MatricesOf1000Coef&SD/MeanBeta.30mar.csv")
    #write.csv(MeanSD.30mar, "E:/MatricesOf1000Coef&SD/MeanSD.30mar.csv")

    write.csv(M_coef30mnar, "E:/MatricesOf1000Coef&SD/M_coef30mnar.csv")
    write.csv(M_sd30mnar, "E:/MatricesOf1000Coef&SD/M_sd30mnar.csv")
    write.csv(MeanBeta.30mnar, "E:/MatricesOf1000Coef&SD/MeanBeta.30mnar.csv")
    write.csv(MeanSD.30mnar, "E:/MatricesOf1000Coef&SD/MeanSD.30mnar.csv")

    f <- function(seed)
    {
     #Generate the independent variables separately using rnorm() or rbinom()
     i_AGE <- runif(1495, min = 18, max = 83)
     i_BMI <- rgamma(1495,  shape = 28.2618/1.12 , rate = 1/1.12)
     i_SEXRecode <- sample(c(0,1),1495, replace = TRUE, prob = c(711/1495, 784/1495))
     PACKYEARS <-rgamma(1495, shape = 7.041/23.01 , rate = 1/23.01)
     ri_GRAINDUSTRecode <-sample(c(0,1), 1495, replace = TRUE, prob = c(428/1495,
1067/1495))
```

```
      h_HOMEPESTICIDERecode <-sample(c(0,1), 1495, replace = TRUE, prob =
c(1124/1495, 371/1495))
      ri_LIVESTOCKRecode <- sample(c(0,1),1495, replace = TRUE, prob = c(670/1495,
825/1495))
      h_LOCATIONRecode <- sample(c(0,1),1495, replace = TRUE, prob = c(732/1495,
763/1495))
      c_FEV1OBSER <- rnorm(1495, mean = 3.100334)
      SRHS_Simulation <- data.frame(c_FEV1OBSER, i_AGE, i_BMI, i_SEXRecode,
PACKYEARS, ri_GRAINDUSTRecode, h_HOMEPESTICIDERecode,
ri_LIVESTOCKRecode, h_LOCATIONRecode)
      SRHS_Simulation$c_FEV1OBSER50.mcar <- SRHS_Simulation$c_FEV1OBSER
      SRHS_Simulation$c_FEV1OBSER50.mar <- SRHS_Simulation$c_FEV1OBSER
      SRHS_Simulation$c_FEV1OBSER50.mnar <-SRHS_Simulation$c_FEV1OBSER
      mcar <- rbinom(n = 1495, size = 1, prob = 1/(1+exp(1-0.99)))
      mar <- rbinom(n = 1495, size = 1, prob = 1/(1+exp(1-0.0185*SRHS_Simulation$i_AGE)))
      mnar <- rbinom(n = 1495, size = 1, prob = 1/(1+exp(1-
0.33*SRHS_Simulation$c_FEV1OBSER)))
      for (i in 1:1495){
        if (mcar[i]==0){
          SRHS_Simulation$c_FEV1OBSER50.mcar[i] <- NA
        }
        #if (mar[i]==0){
          #SRHS_Simulation$c_FEV1OBSER50.mar[i] <- NA
        #}
        #if (mnar[i]==0){
          #SRHS_Simulation$c_FEV1OBSER50.mnar[i] <- NA
        #}
      }
      fit50.mcar <-
lm(c_FEV1OBSER50.mcar~i_AGE+i_BMI+i_SEXRecode+PACKYEARS+ri_LIVESTOCKRe
code, data=SRHS_Simulation)
      #fit50.mar <-
lm(c_FEV1OBSER50.mar~i_AGE+i_BMI+i_SEXRecode+PACKYEARS+ri_LIVESTOCKRec
ode, data=SRHS_Simulation)
      #fit50.mnar <-
lm(c_FEV1OBSER50.mnar~i_AGE+i_BMI+i_SEXRecode+PACKYEARS+ri_LIVESTOCKRe
code, data=SRHS_Simulation)

      return(summary(fit50.mcar))
      #return(summary(fit50.mar))
      #return(summary(fit50.mnar))
    }

    summaries_CompleteCase50 <- lapply(1:1000,f)
    coef_CompleteCase50 <- lapply(summaries_CompleteCase50, coef)
    sd_CompleteCase50 <- lapply(summaries_CompleteCase50, function(data){
```

```
  data$coefficient[,2]
})
t_coef <- unlist(coef_CompleteCase50)
t_sd <- unlist(sd_CompleteCase50)
M_coef50mcar <- matrix(data = t_coef,nrow = 1000, ncol = 6, byrow = TRUE)
M_sd50mcar <- matrix(data = t_sd,nrow = 1000, ncol = 6, byrow = TRUE)
#M_coef50mar <- matrix(data = t_coef,nrow = 1000, ncol = 6, byrow = TRUE)
#M_sd50mar <- matrix(data = t_sd,nrow = 1000, ncol = 6, byrow = TRUE)
#M_coef50mnar <- matrix(data = t_coef,nrow = 1000, ncol = 6, byrow = TRUE)
#M_sd50mnar <- matrix(data = t_sd,nrow = 1000, ncol = 6, byrow = TRUE)

colnames(M_coef50mcar) <- c("beta0", "beta_iAGE", "beta_iBMI", "beta_iSEX",
"beta_PACKYEARS", "beta_LIVESTOCK")
colnames(M_sd50mcar) <- c("beta0", "beta_iAGE", "beta_iBMI", "beta_iSEX",
"beta_PACKYEARS", "beta_LIVESTOCK")
#colnames(M_coef50mar) <- c("beta0", "beta_iAGE", "beta_iBMI", "beta_iSEX",
"beta_PACKYEARS", "beta_LIVESTOCK")
#colnames(M_sd50mar) <- c("beta0", "beta_iAGE", "beta_iBMI", "beta_iSEX",
"beta_PACKYEARS", "beta_LIVESTOCK")
#colnames(M_coef50mnar) <- c("beta0", "beta_iAGE", "beta_iBMI", "beta_iSEX",
"beta_PACKYEARS", "beta_LIVESTOCK")
#colnames(M_sd50mnar) <- c("beta0", "beta_iAGE", "beta_iBMI", "beta_iSEX",
"beta_PACKYEARS", "beta_LIVESTOCK")

MeanBeta.50mcar <- colMeans(data.frame(M_coef50mcar))
MeanSD.50mcar <- colMeans(data.frame(M_sd50mcar))
#MeanBeta.50mar <- colMeans(data.frame(M_coef50mar))
#MeanSD.50mar <- colMeans(data.frame(M_sd50mar))
#MeanBeta.50mnar <- colMeans(data.frame(M_coef50mnar))
#MeanSD.50mnar <- colMeans(data.frame(M_sd50mnar))

write.csv(M_coef50mcar, "E:/MatricesOf1000Coef&SD/M_coef50mcar.csv")
write.csv(M_sd50mcar, "E:/MatricesOf1000Coef&SD/M_sd50mcar.csv")
write.csv(MeanBeta.50mcar, "E:/MatricesOf1000Coef&SD/MeanBeta.50mcar.csv")
write.csv(MeanSD.50mcar, "E:/MatricesOf1000Coef&SD/MeanSD.50mcar.csv")

#write.csv(M_coef50mar, "E:/MatricesOf1000Coef&SD/M_coef50mar.csv")
#write.csv(M_sd50mar, "E:/MatricesOf1000Coef&SD/M_sd50mar.csv")
#write.csv(MeanBeta.50mar, "E:/MatricesOf1000Coef&SD/MeanBeta.50mar.csv")
#write.csv(MeanSD.50mar, "E:/MatricesOf1000Coef&SD/MeanSD.50mar.csv")

#write.csv(M_coef50mnar, "E:/MatricesOf1000Coef&SD/M_coef50mnar.csv")
#write.csv(M_sd50mnar, "E:/MatricesOf1000Coef&SD/M_sd50mnar.csv")
#write.csv(MeanBeta.50mnar, "E:/MatricesOf1000Coef&SD/MeanBeta.50mnar.csv")
#write.csv(MeanSD.50mnar, "E:/MatricesOf1000Coef&SD/MeanSD.50mnar.csv")
```

```
#Confidence Interval Length and Beta Differences
#Results For Article

# CI length, RANDHIE

Galimard_CILength_RANDHIE <- RANDHIE_Galimard_CI_upper_csv[,2:10] -
RANDHIE_Galimard_CI_lower_csv[,2:10]

Rubin_CILength_RANDHIE <- RANDHIE_Rubin_CIupper_csv[, 2:10] -
RANDHIE_Rubin_CIlower_csv[, 2:10]

Ogundimu_CILength_RANDHIE_15MCAR <-
RANDHIE15_MCAR_Results$RANDHIE15_MCAR_UpperCI -
RANDHIE15_MCAR_Results$RANDHIE15_MCAR_LowerCI
Ogundimu_CILength_RANDHIE_15MAR <-
RANDHIE15_MAR_Results$RANDHIE15_MAR_UpperCI -
RANDHIE15_MAR_Results$RANDHIE15_MAR_LowerCI
Ogundimu_CILength_RANDHIE_15MNAR <-
RANDHIE15_MNAR_Results$RANDHIE15_MNAR_UpperCI -
RANDHIE15_MNAR_Results$RANDHIE15_MNAR_LowerCI
Ogundimu_CILength_RANDHIE_30MCAR <-
RANDHIE30_MCAR_Results$RANDHIE30_MCAR_UpperCI -
RANDHIE30_MCAR_Results$RANDHIE30_MCAR_LowerCI
Ogundimu_CILength_RANDHIE_30MAR <-
RANDHIE30_MAR_Results$RANDHIE30_MAR_UpperCI -
RANDHIE30_MAR_Results$RANDHIE30_MAR_LowerCI
Ogundimu_CILength_RANDHIE_30MNAR <-
RANDHIE30_MNAR_Results$RANDHIE30_MNAR_UpperCI -
RANDHIE30_MNAR_Results$RANDHIE30_MNAR_LowerCI
Ogundimu_CILength_RANDHIE_50MCAR <-
RANDHIE50_MCAR_Results$RANDHIE50_MCAR_UpperCI -
RANDHIE50_MCAR_Results$RANDHIE50_MCAR_LowerCI
Ogundimu_CILength_RANDHIE_50MAR <-
RANDHIE50_MAR_Results$RANDHIE50_MAR_UpperCI -
RANDHIE50_MAR_Results$RANDHIE50_MAR_LowerCI
Ogundimu_CILength_RANDHIE_50MNAR <-
RANDHIE50_MNAR_Results$RANDHIE50_MNAR_UpperCI -
RANDHIE50_MNAR_Results$RANDHIE50_MNAR_LowerCI

Ogundimu_CILength_RANDHIE <-
data.frame(Ogundimu_CILength_RANDHIE_15MCAR,
Ogundimu_CILength_RANDHIE_15MAR, Ogundimu_CILength_RANDHIE_15MNAR,
Ogundimu_CILength_RANDHIE_30MCAR, Ogundimu_CILength_RANDHIE_30MAR,
Ogundimu_CILength_RANDHIE_30MNAR, Ogundimu_CILength_RANDHIE_50MCAR,
Ogundimu_CILength_RANDHIE_50MAR, Ogundimu_CILength_RANDHIE_50MNAR)
```

```
    write.csv(Galimard_CILength_RANDHIE,
"E:/MatricesOf1000Coef&SD/ResultsForArticle2/Galimard_CILength_RANDHIE.csv")
    write.csv(Rubin_CILength_RANDHIE,
"E:/MatricesOf1000Coef&SD/ResultsForArticle2/Rubin_CILength_RANDHIE.csv")
    write.csv(Ogundimu_CILength_RANDHIE,
"E:/MatricesOf1000Coef&SD/ResultsForArticle2/Ogundimu_CILength_RANDHIE.csv")

    # CI length, SRHS
    empty <- rep(NA, 6)

    for (i in seq(4, 28, 3)){
      diff <- betas_Confint_SRHS_models_CC[,i] - betas_Confint_SRHS_models_CC[, i-1]
      empty <- cbind(empty, diff)
    }

    CC_CILength_SRHS <- empty[, 2:10]

    Rubin_CILength_SRHS <- SRHS_Rubin_CIupper[, 2:10] - SRHS_Rubin_CIlower[, 2:10]
    Galimard_CILength_SRHS <- SRHS_Galimard_CI_upper[, 2:10] -
SRHS_Galimard_CI_lower[, 2:10]


    Ogundimu_CILength_SRHS_15MCAR <-
SRHS15_MCAR_Results$SRHS15_MCAR_UpperCI -
SRHS15_MCAR_Results$SRHS15_MCAR_LowerCI
    Ogundimu_CILength_SRHS_15MAR <- SRHS15_MAR_Results$SRHS15_MAR_UpperCI
- SRHS15_MAR_Results$SRHS15_MAR_LowerCI
    Ogundimu_CILength_SRHS_15MNAR <-
SRHS15_MNAR_Results$SRHS15_MNAR_UpperCI -
SRHS15_MNAR_Results$SRHS15_MNAR_LowerCI
    Ogundimu_CILength_SRHS_30MCAR <-
SRHS30_MCAR_Results$SRHS30_MCAR_UpperCI -
SRHS30_MCAR_Results$SRHS30_MCAR_LowerCI
    Ogundimu_CILength_SRHS_30MAR <- SRHS30_MAR_Results$SRHS30_MAR_UpperCI
- SRHS30_MAR_Results$SRHS30_MAR_LowerCI
    Ogundimu_CILength_SRHS_30MNAR <-
SRHS30_MNAR_Results$SRHS30_MNAR_UpperCI -
SRHS30_MNAR_Results$SRHS30_MNAR_LowerCI
    Ogundimu_CILength_SRHS_50MCAR <-
SRHS50_MCAR_Results$SRHS50_MCAR_UpperCI -
SRHS50_MCAR_Results$SRHS50_MCAR_LowerCI
    Ogundimu_CILength_SRHS_50MAR <- SRHS50_MAR_Results$SRHS50_MAR_UpperCI
- SRHS50_MAR_Results$SRHS50_MAR_LowerCI
```

```
    Ogundimu_CILength_SRHS_50MNAR <-
SRHS50_MNAR_Results$SRHS50_MNAR_UpperCI -
SRHS50_MNAR_Results$SRHS50_MNAR_LowerCI
    Ogundimu_CILength_SRHS <- data.frame(Ogundimu_CILength_SRHS_15MCAR,
Ogundimu_CILength_SRHS_15MAR, Ogundimu_CILength_SRHS_15MNAR,
Ogundimu_CILength_SRHS_30MCAR, Ogundimu_CILength_SRHS_30MAR,
Ogundimu_CILength_SRHS_30MNAR, Ogundimu_CILength_SRHS_50MCAR,
Ogundimu_CILength_SRHS_50MAR, Ogundimu_CILength_SRHS_50MNAR)

    write.csv(CC_CILength_SRHS,
"E:/MatricesOf1000Coef&SD/ResultsForArticle2/CC_CILength_SRHS.csv")
    write.csv(Rubin_CILength_SRHS,
"E:/MatricesOf1000Coef&SD/ResultsForArticle2/Rubin_CILength_SRHS.csv")
    write.csv(Galimard_CILength_SRHS,
"E:/MatricesOf1000Coef&SD/ResultsForArticle2/Galimard_CILength_SRHS.csv")
    write.csv(Ogundimu_CILength_SRHS,
"E:/MatricesOf1000Coef&SD/ResultsForArticle2/Ogundimu_CILength_SRHS.csv")

    #Beta Difference (between impute and complete), RANDHIE

    empty_beta <- rep(NA, 8)#repeat for every dataset

    for (i in seq(2, 28, 3)){
      diff <- betas_Confint_RANDHIE_models_CC_csv[,i] - CompleteDataBetasRANDHIE
      empty_beta <- cbind(empty_beta, diff)
    }

    CC_BetaDiff_RANDHIE <- empty_beta[, 2:10]

    for (i in seq(2, 10, 1)){
      diff <- RANDHIE_Rubin_betas_csv[,i] - CompleteDataBetasRANDHIE
      empty_beta <- cbind(empty_beta, diff)
    }

    Rubin_BetaDiff_RANDHIE <- empty_beta[, 2:10]


    for (i in seq(2, 10, 1)){
      diff <- RANDHIE_Galimard_beta_ONLY_csv[,i] - CompleteDataBetasRANDHIE
      empty_beta <- cbind(empty_beta, diff)
    }

    Galimard_BetaDiff_RANDHIE <- empty_beta[, 2:10]

    Ogundimu_Betas_RANDHIE <- data.frame(RANDHIE15_MCAR_Results$Estimate,
RANDHIE15_MAR_Results$Estimate, RANDHIE15_MNAR_Results$Estimate,
```

```
RANDHIE30_MCAR_Results$Estimate, RANDHIE30_MAR_Results$Estimate,
RANDHIE30_MNAR_Results$Estimate, RANDHIE50_MCAR_Results$Estimate,
RANDHIE50_MAR_Results$Estimate, RANDHIE50_MNAR_Results$Estimate)

    for (i in seq(1, 9, 1)){
      diff <- Ogundimu_Betas_RANDHIE[,i] - CompleteDataBetasRANDHIE
      empty_beta <- cbind(empty_beta, diff)
    }

    Ogundimu_BetaDiff_RANDHIE <- empty_beta[, 2:10]

    write.csv(CC_BetaDiff_RANDHIE,
"E:/MatricesOf1000Coef&SD/ResultsForArticle2/CC_BetaDiff_RANDHIE.csv")
    write.csv(Rubin_BetaDiff_RANDHIE,
"E:/MatricesOf1000Coef&SD/ResultsForArticle2/Rubin_BetaDiff_RANDHIE.csv")
    write.csv(Galimard_BetaDiff_RANDHIE,
"E:/MatricesOf1000Coef&SD/ResultsForArticle2/Galimard_BetaDiff_RANDHIE.csv")
    write.csv(Ogundimu_BetaDiff_RANDHIE,
"E:/MatricesOf1000Coef&SD/ResultsForArticle2/Ogundimu_BetaDiff_RANDHIE.csv")

    #Beta Difference (between impute and complete), SRHS
    empty_beta <- rep(NA, 6)
    for (i in seq(2, 28, 3)){
      diff <- betas_Confint_SRHS_models_CC[,i] - CompleteDataBetasSRHS
      empty_beta <- cbind(empty_beta, diff)
    }

    CC_BetaDiff_SRHS <- empty_beta[, 2:10]

    for (i in seq(2, 10, 1)){
      diff <- SRHS_Rubin_betas[,i] - CompleteDataBetasSRHS
      empty_beta <- cbind(empty_beta, diff)
    }

    Rubin_BetaDiff_SRHS <- empty_beta[, 2:10]

    for (i in seq(2, 10, 1)){
      diff <- SRHS_Galimard_beta_ONLY[,i] - CompleteDataBetasSRHS
      empty_beta <- cbind(empty_beta, diff)
    }

    Galimard_BetaDiff_SRHS <- empty_beta[, 2:10]

    Ogundimu_Betas_SRHS <- data.frame(SRHS15_MCAR_Results$Estimate,
SRHS15_MAR_Results$Estimate, SRHS15_MNAR_Results$Estimate,
SRHS30_MCAR_Results$Estimate, SRHS30_MAR_Results$Estimate,
```

```
SRHS30_MNAR_Results$Estimate, SRHS50_MCAR_Results$Estimate,
SRHS50_MAR_Results$Estimate, SRHS50_MNAR_Results$Estimate)

    for (i in seq(1, 9, 1)){
      diff <- Ogundimu_Betas_SRHS[,i] - CompleteDataBetasSRHS
      empty_beta <- cbind(empty_beta, diff)
    }

    Ogundimu_BetaDiff_SRHS <- empty_beta[, 2:10]

    write.csv(CC_BetaDiff_SRHS,
"E:/MatricesOf1000Coef&SD/ResultsForArticle2/CC_BetaDiff_SRHS.csv")
    write.csv(Rubin_BetaDiff_SRHS,
"E:/MatricesOf1000Coef&SD/ResultsForArticle2/Rubin_BetaDiff_SRHS.csv")
    write.csv(Galimard_BetaDiff_SRHS,
"E:/MatricesOf1000Coef&SD/ResultsForArticle2/Galimard_BetaDiff_SRHS.csv")
    write.csv(Ogundimu_BetaDiff_SRHS,
"E:/MatricesOf1000Coef&SD/ResultsForArticle2/Ogundimu_BetaDiff_SRHS.csv")
```

B.4 Calculating the Simulation Results

```
    #folder <- "C:/Users/April/Google Drive/Thesis/Outputs"
    folder <- "E:/MatricesOf1000Coef&SD/SRHS/CC/"
    folder <- "E:/MatricesOf1000Coef&SD/SRHS/Rubin/"
    folder <- "E:/MatricesOf1000Coef&SD/SRHS/Galimard/"

    # path to folder that holds multiple .csv files

    file_list <- list.files(path=folder, pattern="MeanBeta*")
    file_list <- list.files(path=folder, pattern="MeanSD*")
    file_list <- list.files(path=folder, pattern="M_*")
    file_list <- list.files(path=folder, pattern="M_Rubin*")
    file_list <- list.files(path=folder, pattern="M_Galimard*")


    # create list of all .csv files in folder

    # read in each .csv file in file_list and create a data frame with the same name as the .csv file

    for (i in 1:length(file_list)){

      assign(file_list[i],
```

```
        read.csv(paste(folder, file_list[i], sep=''))

    )}

RANDHIE_Datasets <- list(RANDHIE_baseline_data_complete_mcar15,
                         RANDHIE_baseline_data_complete_mcar30,
                         RANDHIE_baseline_data_complete_mcar50,
                         RANDHIE_baseline_data_complete_mar15,
                         RANDHIE_baseline_data_complete_mar30,
                         RANDHIE_baseline_data_complete_mar50,
                         RANDHIE_baseline_data_complete_mcar15,
                         RANDHIE_baseline_data_complete_mnar30,
                         RANDHIE_baseline_data_complete_mnar50)


SRHS_Datasets <- list(SRHS15_MCAR,
                      SRHS30_MCAR,
                      SRHS50_MCAR,
                      SRHS15_MAR,
                      SRHS30_MAR,
                      SRHS50_MAR,
                      SRHS15_MNAR,
                      SRHS30_MNAR,
                      SRHS50_MNAR)

#1. Bias of Regression coefficient
CC15MCARBeta <- MeanBeta.15mcar.csv[, 2] - betas_Confint_SRHS_models_CC[, 2]
CC30MCARBeta <- MeanBeta.30mcar.csv[, 2] - betas_Confint_SRHS_models_CC[, 5]
CC50MCARBeta <- MeanBeta.50mcar.csv[, 2] - betas_Confint_SRHS_models_CC[, 8]

CC15MARBeta <- MeanBeta.15mar.csv[, 2] - betas_Confint_SRHS_models_CC[, 11]
CC30MARBeta <- MeanBeta.30mar.csv[, 2] - betas_Confint_SRHS_models_CC[, 14]
CC50MARBeta <- MeanBeta.50mar.csv[, 2] - betas_Confint_SRHS_models_CC[, 17]

CC15MNARBeta <- MeanBeta.15mnar.csv[, 2] - betas_Confint_SRHS_models_CC[, 20]
CC30MNARBeta <- MeanBeta.30mnar.csv[, 2] - betas_Confint_SRHS_models_CC[, 23]
CC50MNARBeta <- MeanBeta.50mnar.csv[, 2] - betas_Confint_SRHS_models_CC[, 26]

CCBiasRegCoef <- cbind(CC15MCARBeta, CC30MCARBeta, CC50MCARBeta,
                       CC15MARBeta, CC30MARBeta, CC50MARBeta,
                       CC15MNARBeta, CC30MNARBeta, CC30MNARBeta)

write.csv(CCBiasRegCoef,
"D:/MatricesOf1000Coef&SD/ResultsForTable/CCBiasRegCoef.csv")
```

```
    Rubin15MCARBeta <- MeanBeta.Rubin15mcar.csv[,2] - SRHS_Rubin_betas[,2]
    Rubin30MCARBeta <- MeanBeta.Rubin30mcar.csv[,2] - SRHS_Rubin_betas[,3]
    Rubin50MCARBeta <- MeanBeta.Rubin50mcar.csv[,2] - SRHS_Rubin_betas[,4]

    Rubin15MARBeta <- MeanBeta.Rubin15mar.csv[,2] - SRHS_Rubin_betas[,5]
    Rubin30MARBeta <- MeanBeta.Rubin30mar.csv[,2] - SRHS_Rubin_betas[,6]
    Rubin50MARBeta <- MeanBeta.Rubin50mar.csv[,2] - SRHS_Rubin_betas[,7]

    Rubin15MNARBeta <- MeanBeta.Rubin15mnar.csv[,2] - SRHS_Rubin_betas[,8]
    Rubin30MNARBeta <- MeanBeta.Rubin30mnar.csv[,2] - SRHS_Rubin_betas[,9]
    Rubin50MNARBeta <- MeanBeta.Rubin50mnar.csv[,2] - SRHS_Rubin_betas[,10]

    RubinBiasRegCoef <- cbind(Rubin15MCARBeta,
Rubin30MCARBeta,Rubin50MCARBeta,
                    Rubin15MARBeta, Rubin30MARBeta, Rubin50MARBeta,
                    Rubin15MNARBeta, Rubin30MNARBeta, Rubin50MNARBeta)

    write.csv(RubinBiasRegCoef,
"D:/MatricesOf1000Coef&SD/ResultsForTable/RubinBiasRegCoef.csv")

    Galimard15MCARBeta <- MeanBeta.Galimard15mcar.csv[,2] -
SRHS_Galimard_beta_ONLY[, 2]
    Galimard30MCARBeta <- MeanBeta.Galimard30mcar.csv[,2] -
SRHS_Galimard_beta_ONLY[, 3]
    Galimard50MCARBeta <- MeanBeta.Galimard50mcar.csv[,2] -
SRHS_Galimard_beta_ONLY[, 4]

    Galimard15MARBeta <- MeanBeta.Galimard15mar.csv[,2] -
SRHS_Galimard_beta_ONLY[, 5]
    Galimard30MARBeta <- MeanBeta.Galimard30mar.csv[,2] -
SRHS_Galimard_beta_ONLY[, 6]
    Galimard50MARBeta <- MeanBeta.Galimard50mar.csv[,2] -
SRHS_Galimard_beta_ONLY[, 7]

    Galimard15MNARBeta <- MeanBeta.Galimard15mnar.csv[,2] -
SRHS_Galimard_beta_ONLY[, 8]
    Galimard30MNARBeta <- MeanBeta.Galimard30mnar.csv[,2] -
SRHS_Galimard_beta_ONLY[, 9]
    Galimard50MNARBeta <- MeanBeta.Galimard50mnar.csv[,2] -
SRHS_Galimard_beta_ONLY[, 10]

    GalimardBiasRegCoef <- cbind(Galimard15MCARBeta, Galimard30MCARBeta,
Galimard50MCARBeta,
                    Galimard15MARBeta, Galimard30MARBeta, Galimard50MARBeta,
                    Galimard15MNARBeta, Galimard30MNARBeta,
Galimard50MNARBeta)
```

```
    write.csv(GalimardBiasRegCoef,
"D:/MatricesOf1000Coef&SD/ResultsForTable/GalimardBiasRegCoef.csv")

    #2. Relative Bias of Regression Coefficient

    RB_BetaCC15MCAR <- (CCBiasRegCoef[,1]/betas_Confint_SRHS_models_CC[, 2])*100
    RB_BetaCC30MCAR <- (CCBiasRegCoef[,2]/betas_Confint_SRHS_models_CC[, 5])*100
    RB_BetaCC50MCAR <- (CCBiasRegCoef[,3]/betas_Confint_SRHS_models_CC[, 8])*100

    RB_BetaCC15MAR <- (CCBiasRegCoef[,4]/betas_Confint_SRHS_models_CC[, 11])*100
    RB_BetaCC30MAR <- (CCBiasRegCoef[,5]/betas_Confint_SRHS_models_CC[, 14])*100
    RB_BetaCC50MAR <- (CCBiasRegCoef[,6]/betas_Confint_SRHS_models_CC[, 17])*100

    RB_BetaCC15MNAR <- (CCBiasRegCoef[,7]/betas_Confint_SRHS_models_CC[,
20])*100
    RB_BetaCC30MNAR <- (CCBiasRegCoef[,8]/betas_Confint_SRHS_models_CC[,
23])*100
    RB_BetaCC50MNAR <- (CCBiasRegCoef[,9]/betas_Confint_SRHS_models_CC[,
26])*100

    CCRelatBiasRegCoef <- cbind(RB_BetaCC15MCAR, RB_BetaCC30MCAR,
RB_BetaCC50MCAR,
                        RB_BetaCC15MAR, RB_BetaCC30MAR, RB_BetaCC50MAR,
                        RB_BetaCC15MNAR, RB_BetaCC30MNAR, RB_BetaCC50MNAR)

    write.csv(CCRelatBiasRegCoef,
"D:/MatricesOf1000Coef&SD/ResultsForTable/CCRelatBiasRegCoef.csv")

    RB_BetaRubin15MCAR <- (RubinBiasRegCoef[,1]/SRHS_Rubin_betas[,2])*100
    RB_BetaRubin30MCAR <- (RubinBiasRegCoef[,2]/SRHS_Rubin_betas[,3])*100
    RB_BetaRubin50MCAR <- (RubinBiasRegCoef[,3]/SRHS_Rubin_betas[,4])*100

    RB_BetaRubin15MAR <- (RubinBiasRegCoef[,4]/SRHS_Rubin_betas[,5])*100
    RB_BetaRubin30MAR <- (RubinBiasRegCoef[,5]/SRHS_Rubin_betas[,6])*100
    RB_BetaRubin50MAR <- (RubinBiasRegCoef[,6]/SRHS_Rubin_betas[,7])*100

    RB_BetaRubin15MNAR <- (RubinBiasRegCoef[,7]/SRHS_Rubin_betas[,8])*100
    RB_BetaRubin30MNAR <- (RubinBiasRegCoef[,8]/SRHS_Rubin_betas[,9])*100
    RB_BetaRubin50MNAR <- (RubinBiasRegCoef[,9]/SRHS_Rubin_betas[,10])*100

    RubinRelatBiasRegCoef <- cbind(RB_BetaRubin15MCAR, RB_BetaRubin30MCAR,
RB_BetaRubin50MCAR,
                        RB_BetaRubin15MAR, RB_BetaRubin30MAR, RB_BetaRubin50MAR,
                        RB_BetaRubin15MNAR, RB_BetaRubin30MNAR,
RB_BetaRubin50MNAR)
```

```
    write.csv(RubinRelatBiasRegCoef,
"D:/MatricesOf1000Coef&SD/ResultsForTable/RubinRelatBiasRegCoef.csv")

    RB_BetaGalimard15MCAR <- (GalimardBiasRegCoef[,1]/SRHS_Galimard_beta_ONLY[,
2])*100
    RB_BetaGalimard30MCAR <- (GalimardBiasRegCoef[,2]/SRHS_Galimard_beta_ONLY[,
3])*100
    RB_BetaGalimard50MCAR <- (GalimardBiasRegCoef[,3]/SRHS_Galimard_beta_ONLY[,
4])*100

    RB_BetaGalimard15MAR <- (GalimardBiasRegCoef[,4]/SRHS_Galimard_beta_ONLY[,
5])*100
    RB_BetaGalimard30MAR <- (GalimardBiasRegCoef[,5]/SRHS_Galimard_beta_ONLY[,
6])*100
    RB_BetaGalimard50MAR <- (GalimardBiasRegCoef[,6]/SRHS_Galimard_beta_ONLY[,
7])*100

    RB_BetaGalimard15MNAR <- (GalimardBiasRegCoef[,7]/SRHS_Galimard_beta_ONLY[,
8])*100
    RB_BetaGalimard30MNAR <- (GalimardBiasRegCoef[,8]/SRHS_Galimard_beta_ONLY[,
9])*100
    RB_BetaGalimard50MNAR <-
(GalimardBiasRegCoef[,9]/SRHS_Galimard_beta_ONLY[,10])*100

    GalimardRelatBiasRegCoef <- cbind(RB_BetaGalimard15MCAR,
RB_BetaGalimard30MCAR, RB_BetaGalimard50MCAR,
                          RB_BetaGalimard15MAR, RB_BetaGalimard30MAR,
RB_BetaGalimard50MAR,
                          RB_BetaGalimard15MNAR, RB_BetaGalimard30MNAR,
RB_BetaGalimard50MNAR)

    write.csv(GalimardRelatBiasRegCoef,
"D:/MatricesOf1000Coef&SD/ResultsForTable/GalimardRelatBiasRegCoef.csv")


    #3. Standardized bias of regression coefficient

    SB_BetaCC15MCAR <- (CCBiasRegCoef[,1]/SE_SRHS_models_CC[,2])*100
    SB_BetaCC30MCAR <- (CCBiasRegCoef[,2]/SE_SRHS_models_CC[,3])*100
    SB_BetaCC50MCAR <- (CCBiasRegCoef[,3]/SE_SRHS_models_CC[,4])*100

    SB_BetaCC15MAR <- (CCBiasRegCoef[,4]/SE_SRHS_models_CC[,5])*100
    SB_BetaCC30MAR <- (CCBiasRegCoef[,5]/SE_SRHS_models_CC[,6])*100
    SB_BetaCC50MAR <- (CCBiasRegCoef[,6]/SE_SRHS_models_CC[,7])*100
```

```
    SB_BetaCC15MNAR <- (CCBiasRegCoef[,7]/SE_SRHS_models_CC[,8])*100
    SB_BetaCC30MNAR <- (CCBiasRegCoef[,8]/SE_SRHS_models_CC[,9])*100
    SB_BetaCC50MNAR <- (CCBiasRegCoef[,9]/SE_SRHS_models_CC[,10])*100

    CCStandBiasRegCoef <- cbind(SB_BetaCC15MCAR, SB_BetaCC30MCAR,
SB_BetaCC50MCAR,
                        SB_BetaCC15MAR, SB_BetaCC30MAR, SB_BetaCC50MAR,
                        SB_BetaCC15MNAR, SB_BetaCC30MNAR, SB_BetaCC50MNAR)
    write.csv(CCStandBiasRegCoef,
"D:/MatricesOf1000Coef&SD/ResultsForTable/CCStandBiasRegCoef.csv")

    SB_BetaRubin15MCAR <- (RubinBiasRegCoef[,1]/SRHS_Rubin_SE[,2])*100
    SB_BetaRubin30MCAR <- (RubinBiasRegCoef[,2]/SRHS_Rubin_SE[,3])*100
    SB_BetaRubin50MCAR <- (RubinBiasRegCoef[,3]/SRHS_Rubin_SE[,4])*100

    SB_BetaRubin15MAR <- (RubinBiasRegCoef[,4]/SRHS_Rubin_SE[,5])*100
    SB_BetaRubin30MAR <- (RubinBiasRegCoef[,5]/SRHS_Rubin_SE[,6])*100
    SB_BetaRubin50MAR <- (RubinBiasRegCoef[,6]/SRHS_Rubin_SE[,7])*100

    SB_BetaRubin15MNAR <- (RubinBiasRegCoef[,7]/SRHS_Rubin_SE[,8])*100
    SB_BetaRubin30MNAR <- (RubinBiasRegCoef[,8]/SRHS_Rubin_SE[,9])*100
    SB_BetaRubin50MNAR <- (RubinBiasRegCoef[,9]/SRHS_Rubin_SE[,10])*100

    RubinStandBiasRegCoef <- cbind(SB_BetaRubin15MCAR, SB_BetaRubin30MCAR,
SB_BetaRubin50MCAR,
                        SB_BetaRubin15MAR, SB_BetaRubin30MAR,SB_BetaRubin50MAR,
                        SB_BetaRubin15MNAR, SB_BetaRubin30MNAR,
SB_BetaRubin50MNAR)

    write.csv(RubinStandBiasRegCoef,
"D:/MatricesOf1000Coef&SD/ResultsForTable/RubinStandBiasRegCoef.csv")


    SB_BetaGalimard15MCAR <-
(GalimardBiasRegCoef[,1]/SRHS_Galimard_SE_ONLY[,2])*100
    SB_BetaGalimard30MCAR <-
(GalimardBiasRegCoef[,2]/SRHS_Galimard_SE_ONLY[,3])*100
    SB_BetaGalimard50MCAR <-
(GalimardBiasRegCoef[,3]/SRHS_Galimard_SE_ONLY[,4])*100

    SB_BetaGalimard15MAR <-
(GalimardBiasRegCoef[,4]/SRHS_Galimard_SE_ONLY[,5])*100
    SB_BetaGalimard30MAR <-
(GalimardBiasRegCoef[,5]/SRHS_Galimard_SE_ONLY[,6])*100
    SB_BetaGalimard50MAR <-
(GalimardBiasRegCoef[,6]/SRHS_Galimard_SE_ONLY[,7])*100
```

```
    SB_BetaGalimard15MNAR <-
(GalimardBiasRegCoef[,7]/SRHS_Galimard_SE_ONLY[,8])*100
    SB_BetaGalimard30MNAR <-
(GalimardBiasRegCoef[,8]/SRHS_Galimard_SE_ONLY[,9])*100
    SB_BetaGalimard50MNAR <-
(GalimardBiasRegCoef[,9]/SRHS_Galimard_SE_ONLY[,10])*100

    GalimardStandBiasRegCoef <- cbind(SB_BetaGalimard15MCAR,
SB_BetaGalimard30MCAR, SB_BetaGalimard50MCAR,
                          SB_BetaGalimard15MAR, SB_BetaGalimard30MAR,
SB_BetaGalimard50MAR,
                          SB_BetaGalimard15MNAR, SB_BetaGalimard30MNAR,
SB_BetaGalimard50MNAR)

    write.csv(GalimardStandBiasRegCoef,
"D:/MatricesOf1000Coef&SD/ResultsForTable/GalimardStandBiasRegCoef.csv")

    #4. Mean Square Error (MSE) of regression coefficient

    MSE_BetaCC15MCAR <- (CCBiasRegCoef[,1])**2/(MeanSD.15mcar.csv[,2])**2
    MSE_BetaCC30MCAR <- (CCBiasRegCoef[,2])**2/(MeanSD.30mcar.csv[,2])**2
    MSE_BetaCC50MCAR <- (CCBiasRegCoef[,3])**2/(MeanSD.50mcar.csv[,2])**2

    MSE_BetaCC15MAR <- (CCBiasRegCoef[,4])**2/(MeanSD.15mar.csv[,2])**2
    MSE_BetaCC30MAR <- (CCBiasRegCoef[,5])**2/(MeanSD.30mar.csv[,2])**2
    MSE_BetaCC50MAR <- (CCBiasRegCoef[,6])**2/(MeanSD.50mar.csv[,2])**2

    MSE_BetaCC15MNAR <- (CCBiasRegCoef[,7])**2/(MeanSD.15mnar.csv[,2])**2
    MSE_BetaCC30MNAR <- (CCBiasRegCoef[,8])**2/(MeanSD.30mnar.csv[,2])**2
    MSE_BetaCC50MNAR <- (CCBiasRegCoef[,9])**2/(MeanSD.50mnar.csv[,2])**2

    CCMSE_Beta <- cbind(MSE_BetaCC15MCAR, MSE_BetaCC30MCAR,
MSE_BetaCC50MCAR,
                MSE_BetaCC15MAR, MSE_BetaCC30MAR, MSE_BetaCC50MAR,
                MSE_BetaCC15MNAR, MSE_BetaCC30MNAR, MSE_BetaCC50MNAR)

    write.csv(CCMSE_Beta,
"D:/MatricesOf1000Coef&SD/ResultsForTable/CCMSE_Beta.csv")

    MSE_BetaRubin15MCAR <-
(RubinBiasRegCoef[,1])**2/(MeanSD.Rubin15mcar.csv[,2])**2
    MSE_BetaRubin30MCAR <-
(RubinBiasRegCoef[,2])**2/(MeanSD.Rubin30mcar.csv[,2])**2
    MSE_BetaRubin50MCAR <-
(RubinBiasRegCoef[,3])**2/(MeanSD.Rubin50mcar.csv[,2])**2
```

```
    MSE_BetaRubin15MAR <- (RubinBiasRegCoef[,4])**2/(MeanSD.Rubin15mar.csv[,2])**2
    MSE_BetaRubin30MAR <- (RubinBiasRegCoef[,5])**2/(MeanSD.Rubin30mar.csv[,2])**2
    MSE_BetaRubin50MAR <- (RubinBiasRegCoef[,6])**2/(MeanSD.Rubin50mar.csv[,2])**2

    MSE_BetaRubin15MNAR <-
(RubinBiasRegCoef[,7])**2/(MeanSD.Rubin15mnar.csv[,2])**2
    MSE_BetaRubin30MNAR <-
(RubinBiasRegCoef[,8])**2/(MeanSD.Rubin30mnar.csv[,2])**2
    MSE_BetaRubin50MNAR <-
(RubinBiasRegCoef[,9])**2/(MeanSD.Rubin50mnar.csv[,2])**2

    RubinMSE_Beta <- cbind(MSE_BetaRubin15MCAR, MSE_BetaRubin30MCAR,
MSE_BetaRubin50MCAR,
                           MSE_BetaRubin15MAR, MSE_BetaRubin30MAR, MSE_BetaRubin50MAR,
                           MSE_BetaRubin15MNAR, MSE_BetaRubin30MNAR,
MSE_BetaRubin50MNAR)
    write.csv(RubinMSE_Beta,
"D:/MatricesOf1000Coef&SD/ResultsForTable/RubinMSE_Beta.csv")

    MSE_BetaGalimard15MCAR <-
(GalimardBiasRegCoef[,1])**2/(MeanSD.Galimard15mcar.csv[,2])**2
    MSE_BetaGalimard30MCAR <-
(GalimardBiasRegCoef[,2])**2/(MeanSD.Galimard30mcar.csv[,2])**2
    MSE_BetaGalimard50MCAR <-
(GalimardBiasRegCoef[,3])**2/(MeanSD.Galimard50mcar.csv[,2])**2

    MSE_BetaGalimard15MAR <-
(GalimardBiasRegCoef[,4])**2/(MeanSD.Galimard15mar.csv[,2])**2
    MSE_BetaGalimard30MAR <-
(GalimardBiasRegCoef[,5])**2/(MeanSD.Galimard30mar.csv[,2])**2
    MSE_BetaGalimard50MAR <-
(GalimardBiasRegCoef[,6])**2/(MeanSD.Galimard50mar.csv[,2])**2

    MSE_BetaGalimard15MNAR <-
(GalimardBiasRegCoef[,7])**2/(MeanSD.Galimard15mnar.csv[,2])**2
    MSE_BetaGalimard30MNAR <-
(GalimardBiasRegCoef[,8])**2/(MeanSD.Galimard30mnar.csv[,2])**2
    MSE_BetaGalimard50MNAR <-
(GalimardBiasRegCoef[,9])**2/(MeanSD.Galimard50mnar.csv[,2])**2

    GalimardMSE_Beta <- cbind(MSE_BetaGalimard15MCAR, MSE_BetaGalimard30MCAR,
MSE_BetaGalimard50MCAR,
                          MSE_BetaGalimard15MAR, MSE_BetaGalimard30MAR,
MSE_BetaGalimard50MAR,
```

```
                    MSE_BetaGalimard15MNAR, MSE_BetaGalimard30MNAR,
MSE_BetaGalimard50MNAR)
    write.csv(GalimardMSE_Beta,
"D:/MatricesOf1000Coef&SD/ResultsForTable/GalimardMSE_Beta.csv")

    #5. Average 95% Confidence Interval

    Avg95CI_CC15MCAR <- colSums(2*1.96*M_sd15mcar.csv[,2:7])/1000
    Avg95CI_CC30MCAR <- colSums(2*1.96*M_sd30mcar.csv[,2:7])/1000
    Avg95CI_CC50MCAR <- colSums(2*1.96*M_sd50mcar.csv[,2:7])/1000

    Avg95CI_CC15MAR <- colSums(2*1.96*M_sd15mar.csv[,2:7])/1000
    Avg95CI_CC30MAR <- colSums(2*1.96*M_sd30mar.csv[,2:7])/1000
    Avg95CI_CC50MAR <-colSums(2*1.96*M_sd50mar.csv[,2:7])/1000

    Avg95CI_CC15MNAR <- colSums(2*1.96*M_sd15mnar.csv[,2:7])/1000
    Avg95CI_CC30MNAR <- colSums(2*1.96*M_sd30mnar.csv[,2:7])/1000
    Avg95CI_CC50MNAR <-colSums(2*1.96*M_sd50mnar.csv[,2:7])/1000

    CCAvg95CI <- cbind(Avg95CI_CC15MCAR, Avg95CI_CC30MCAR,
Avg95CI_CC50MCAR,
                Avg95CI_CC15MAR, Avg95CI_CC30MAR, Avg95CI_CC50MAR,
                Avg95CI_CC15MNAR, Avg95CI_CC30MNAR, Avg95CI_CC50MNAR)
    write.csv(CCAvg95CI, "D:/MatricesOf1000Coef&SD/ResultsForTable/CCAvg95CI.csv")

    Avg95CI_Rubin15MCAR <- colSums(2*1.96*M_Rubinsd15mcar.csv[,2:7])/1000
    Avg95CI_Rubin30MCAR <- colSums(2*1.96*M_Rubinsd30mcar.csv[,2:7])/1000
    Avg95CI_Rubin50MCAR <- colSums(2*1.96*M_Rubinsd50mcar.csv[,2:7])/1000

    Avg95CI_Rubin15MAR <- colSums(2*1.96*M_Rubinsd15mar.csv[,2:7])/1000
    Avg95CI_Rubin30MAR <- colSums(2*1.96*M_Rubinsd30mar.csv[,2:7])/1000
    Avg95CI_Rubin50MAR <-colSums(2*1.96*M_Rubinsd50mar.csv[,2:7])/1000

    Avg95CI_Rubin15MNAR <- colSums(2*1.96*M_Rubinsd15mnar.csv[,2:7])/1000
    Avg95CI_Rubin30MNAR <- colSums(2*1.96*M_Rubinsd30mnar.csv[,2:7])/1000
    Avg95CI_Rubin50MNAR <-colSums(2*1.96*M_Rubinsd50mnar.csv[,2:7])/1000

    RubinAvg95CI <- cbind(Avg95CI_Rubin15MCAR, Avg95CI_Rubin30MCAR,
Avg95CI_Rubin50MCAR,
                  Avg95CI_Rubin15MAR, Avg95CI_Rubin30MAR, Avg95CI_Rubin50MAR,
                  Avg95CI_Rubin15MNAR, Avg95CI_Rubin30MNAR,
Avg95CI_Rubin50MNAR)
    write.csv(RubinAvg95CI,
"D:/MatricesOf1000Coef&SD/ResultsForTable/RubinAvg95CI.csv")

    Avg95CI_Galimard15MCAR <- colSums(2*1.96*M_Galimardsd15mcar.csv[,2:7])/1000
```

```
    Avg95CI_Galimard30MCAR <- colSums(2*1.96*M_Galimardsd30mcar.csv[,2:7])/1000
    Avg95CI_Galimard50MCAR <- colSums(2*1.96*M_Galimardsd50mcar.csv[,2:7])/1000

    Avg95CI_Galimard15MAR <- colSums(2*1.96*M_Galimardsd15mar.csv[,2:7])/1000
    Avg95CI_Galimard30MAR <- colSums(2*1.96*M_Galimardsd30mar.csv[,2:7])/1000
    Avg95CI_Galimard50MAR <-colSums(2*1.96*M_Galimardsd50mar.csv[,2:7])/1000

    Avg95CI_Galimard15MNAR <- colSums(2*1.96*M_Galimardsd15mnar.csv[,2:7])/1000
    Avg95CI_Galimard30MNAR <- colSums(2*1.96*M_Galimardsd30mnar.csv[,2:7])/1000
    Avg95CI_Galimard50MNAR <-colSums(2*1.96*M_Galimardsd50mnar.csv[,2:7])/1000

    GalimardAvg95CI <- cbind(Avg95CI_Galimard15MCAR, Avg95CI_Galimard30MCAR,
Avg95CI_Galimard50MCAR,
                    Avg95CI_Galimard15MAR, Avg95CI_Galimard30MAR,
Avg95CI_Galimard50MAR,
                    Avg95CI_Galimard15MNAR, Avg95CI_Galimard30MNAR,
Avg95CI_Galimard50MNAR)
    write.csv(GalimardAvg95CI,
"D:/MatricesOf1000Coef&SD/ResultsForTable/GalimardAvg95CI.csv")

    #6. Coverage of True Regression Coefficients

    #UpperCI

    SimUpperCI_CC15MCAR_SRHS <- M_coef15mcar.csv[, 2:7] + 1.96*M_sd15mcar.csv[,
2:7]
    SimUpperCI_CC30MCAR_SRHS <- M_coef30mcar.csv[, 2:7] + 1.96*M_sd30mcar.csv[,
2:7]
    SimUpperCI_CC50MCAR_SRHS <- M_coef50mcar.csv[, 2:7] + 1.96*M_sd50mcar.csv[,
2:7]

    SimUpperCI_CC15MAR_SRHS <- M_coef15mar.csv[, 2:7] + 1.96*M_sd15mar.csv[, 2:7]
    SimUpperCI_CC30MAR_SRHS <- M_coef30mar.csv[, 2:7] + 1.96*M_sd30mar.csv[, 2:7]
    SimUpperCI_CC50MAR_SRHS <- M_coef50mar.csv[, 2:7] + 1.96*M_sd50mar.csv[, 2:7]

    SimUpperCI_CC15MNAR_SRHS <- M_coef15mnar.csv[, 2:7] + 1.96*M_sd15mnar.csv[,
2:7]
    SimUpperCI_CC30MNAR_SRHS <- M_coef30mnar.csv[, 2:7] + 1.96*M_sd30mnar.csv[,
2:7]
    SimUpperCI_CC50MNAR_SRHS <- M_coef50mnar.csv[, 2:7] + 1.96*M_sd50mnar.csv[,
2:7]

    SimUpperCI_Rubin15MCAR_SRHS <- M_Rubincoef15mcar.csv[, 2:7] +
1.96*M_Rubinsd15mcar.csv[, 2:7]
    SimUpperCI_Rubin30MCAR_SRHS <- M_Rubincoef30mcar.csv[, 2:7] +
1.96*M_Rubinsd30mcar.csv[, 2:7]
```

```
    SimUpperCI_Rubin50MCAR_SRHS <- M_Rubincoef50mcar.csv[, 2:7] +
1.96*M_Rubinsd50mcar.csv[, 2:7]

    SimUpperCI_Rubin15MAR_SRHS <- M_Rubincoef15mar.csv[, 2:7] +
1.96*M_Rubinsd15mar.csv[, 2:7]
    SimUpperCI_Rubin30MAR_SRHS <- M_Rubincoef30mar.csv[, 2:7] +
1.96*M_Rubinsd30mar.csv[, 2:7]
    SimUpperCI_Rubin50MAR_SRHS <- M_Rubincoef50mar.csv[, 2:7] +
1.96*M_Rubinsd50mar.csv[, 2:7]

    SimUpperCI_Rubin15MNAR_SRHS <- M_Rubincoef15mnar.csv[, 2:7] +
1.96*M_Rubinsd15mnar.csv[, 2:7]
    SimUpperCI_Rubin30MNAR_SRHS <- M_Rubincoef30mnar.csv[, 2:7] +
1.96*M_Rubinsd30mnar.csv[, 2:7]
    SimUpperCI_Rubin50MNAR_SRHS <- M_Rubincoef50mnar.csv[, 2:7] +
1.96*M_Rubinsd50mnar.csv[, 2:7]

    SimUpperCI_Galimard15MCAR_SRHS <- M_Galimardcoef15mcar.csv[, 2:7] +
1.96*M_Galimardsd15mcar.csv[, 2:7]
    SimUpperCI_Galimard30MCAR_SRHS <- M_Galimardcoef30mcar.csv[, 2:7] +
1.96*M_Galimardsd30mcar.csv[, 2:7]
    SimUpperCI_Galimard50MCAR_SRHS <- M_Galimardcoef50mcar.csv[, 2:7] +
1.96*M_Galimardsd50mcar.csv[, 2:7]

    SimUpperCI_Galimard15MAR_SRHS <- M_Galimardcoef15mar.csv[, 2:7] +
1.96*M_Galimardsd15mar.csv[, 2:7]
    SimUpperCI_Galimard30MAR_SRHS <- M_Galimardcoef30mar.csv[, 2:7] +
1.96*M_Galimardsd30mar.csv[, 2:7]
    SimUpperCI_Galimard50MAR_SRHS <- M_Galimardcoef50mar.csv[, 2:7] +
1.96*M_Galimardsd50mar.csv[, 2:7]

    SimUpperCI_Galimard15MNAR_SRHS <- M_Galimardcoef15mnar.csv[, 2:7] +
1.96*M_Galimardsd15mnar.csv[, 2:7]
    SimUpperCI_Galimard30MNAR_SRHS <- M_Galimardcoef30mnar.csv[, 2:7] +
1.96*M_Galimardsd30mnar.csv[, 2:7]
    SimUpperCI_Galimard50MNAR_SRHS <- M_Galimardcoef50mnar.csv[, 2:7] +
1.96*M_Galimardsd50mnar.csv[, 2:7]

    #LowerCI

    SimLowerCI_CC15MCAR_SRHS <- M_coef15mcar.csv[, 2:7] - 1.96*M_sd15mcar.csv[,
2:7]
    SimLowerCI_CC30MCAR_SRHS <- M_coef30mcar.csv[, 2:7] - 1.96*M_sd30mcar.csv[,
2:7]
    SimLowerCI_CC50MCAR_SRHS <- M_coef50mcar.csv[, 2:7] - 1.96*M_sd50mcar.csv[,
2:7]
```

```
    SimLowerCI_CC15MAR_SRHS <- M_coef15mar.csv[, 2:7] - 1.96*M_sd15mar.csv[, 2:7]
    SimLowerCI_CC30MAR_SRHS <- M_coef30mar.csv[, 2:7] - 1.96*M_sd30mar.csv[, 2:7]
    SimLowerCI_CC50MAR_SRHS <- M_coef50mar.csv[, 2:7] - 1.96*M_sd50mar.csv[, 2:7]

    SimLowerCI_CC15MNAR_SRHS <- M_coef15mnar.csv[, 2:7] - 1.96*M_sd15mnar.csv[,
2:7]
    SimLowerCI_CC30MNAR_SRHS <- M_coef30mnar.csv[, 2:7] - 1.96*M_sd30mnar.csv[,
2:7]
    SimLowerCI_CC50MNAR_SRHS <- M_coef50mnar.csv[, 2:7] - 1.96*M_sd50mnar.csv[,
2:7]

    SimLowerCI_Rubin15MCAR_SRHS <- M_Rubincoef15mcar.csv[, 2:7] -
1.96*M_Rubinsd15mcar.csv[, 2:7]
    SimLowerCI_Rubin30MCAR_SRHS <- M_Rubincoef30mcar.csv[, 2:7] -
1.96*M_Rubinsd30mcar.csv[, 2:7]
    SimLowerCI_Rubin50MCAR_SRHS <- M_Rubincoef50mcar.csv[, 2:7] -
1.96*M_Rubinsd50mcar.csv[, 2:7]

    SimLowerCI_Rubin15MAR_SRHS <- M_Rubincoef15mar.csv[, 2:7] -
1.96*M_Rubinsd15mar.csv[, 2:7]
    SimLowerCI_Rubin30MAR_SRHS <- M_Rubincoef30mar.csv[, 2:7] -
1.96*M_Rubinsd30mar.csv[, 2:7]
    SimLowerCI_Rubin50MAR_SRHS <- M_Rubincoef50mar.csv[, 2:7] -
1.96*M_Rubinsd50mar.csv[, 2:7]

    SimLowerCI_Rubin15MNAR_SRHS <- M_Rubincoef15mnar.csv[, 2:7] -
1.96*M_Rubinsd15mnar.csv[, 2:7]
    SimLowerCI_Rubin30MNAR_SRHS <- M_Rubincoef30mnar.csv[, 2:7] -
1.96*M_Rubinsd30mnar.csv[, 2:7]
    SimLowerCI_Rubin50MNAR_SRHS <- M_Rubincoef50mnar.csv[, 2:7] -
1.96*M_Rubinsd50mnar.csv[, 2:7]

    SimLowerCI_Galimard15MCAR_SRHS <- M_Galimardcoef15mcar.csv[, 2:7] -
1.96*M_Galimardsd15mcar.csv[, 2:7]
    SimLowerCI_Galimard30MCAR_SRHS <- M_Galimardcoef30mcar.csv[, 2:7] -
1.96*M_Galimardsd30mcar.csv[, 2:7]
    SimLowerCI_Galimard50MCAR_SRHS <- M_Galimardcoef50mcar.csv[, 2:7] -
1.96*M_Galimardsd50mcar.csv[, 2:7]

    SimLowerCI_Galimard15MAR_SRHS <- M_Galimardcoef15mar.csv[, 2:7] -
1.96*M_Galimardsd15mar.csv[, 2:7]
    SimLowerCI_Galimard30MAR_SRHS <- M_Galimardcoef30mar.csv[, 2:7] -
1.96*M_Galimardsd30mar.csv[, 2:7]
    SimLowerCI_Galimard50MAR_SRHS <- M_Galimardcoef50mar.csv[, 2:7] -
1.96*M_Galimardsd50mar.csv[, 2:7]
```

```
    SimLowerCI_Galimard15MNAR_SRHS <- M_Galimardcoef15mnar.csv[, 2:7] -
1.96*M_Galimardsd15mnar.csv[, 2:7]
    SimLowerCI_Galimard30MNAR_SRHS <- M_Galimardcoef30mnar.csv[, 2:7] -
1.96*M_Galimardsd30mnar.csv[, 2:7]
    SimLowerCI_Galimard50MNAR_SRHS <- M_Galimardcoef50mnar.csv[, 2:7] -
1.96*M_Galimardsd50mnar.csv[, 2:7]

    CoveragePercentageOfSimCI <- function(UpperCI1000, LowerCI1000, Beta){
      In_OR_Out <- rep(NA, 1000)
      for (i in 1:1000){
        if ((Beta <UpperCI1000[i])&(Beta> LowerCI1000[i])){
          In_OR_Out[i] <-1}
        else In_OR_Out[i] <- 0
      }
      return(In_OR_Out)
    }



    CICoveragePercentageSingleCombination <- function(UpperCI1000Matrix,
LowerCI1000Matrix, BetaColumn){
      CICoveragePercentagesList <- list()
      for (i in 1:6){
        Percentage <- sum(CoveragePercentageOfSimCI(UpperCI1000Matrix[,i],
LowerCI1000Matrix[,i], BetaColumn[i, ]))/1000
        CICoveragePercentagesList[i] <- Percentage
      }
      return(CICoveragePercentagesList)
    }

    CC_Upper_SRHS <- list(SimUpperCI_CC15MCAR_SRHS,
SimUpperCI_CC15MAR_SRHS, SimUpperCI_CC15MNAR_SRHS,
                          SimUpperCI_CC30MCAR_SRHS, SimUpperCI_CC30MAR_SRHS,
SimUpperCI_CC30MNAR_SRHS,
                          SimUpperCI_CC50MCAR_SRHS, SimUpperCI_CC50MAR_SRHS,
SimUpperCI_CC50MNAR_SRHS)

    CC_Lower_SRHS <- list(SimLowerCI_CC15MCAR_SRHS,
SimUpperCI_CC15MAR_SRHS, SimUpperCI_CC15MNAR_SRHS,
                          SimLowerCI_CC30MCAR_SRHS, SimUpperCI_CC30MAR_SRHS,
SimUpperCI_CC30MNAR_SRHS,
                          SimLowerCI_CC50MCAR_SRHS, SimUpperCI_CC50MAR_SRHS,
SimUpperCI_CC50MNAR_SRHS)
```

```
    SRHS_CICoverPercentageFinal <- function(SimUpperVector, SimLowerVector,
BetaMatrix){
      SRHS_CICoveragePercentages <- c()
      for (i in 1:9){
        SRHS_CICoveragePercentagesList <-
CICoveragePercentageSingleCombination(SimUpperVector[[i]], SimLowerVector[[i]],
BetaMatrix[, i])
        SRHS_CICoveragePercentages <- cbind(SRHS_CICoveragePercentages,
as.vector(unlist(SRHS_CICoveragePercentagesList)))
      }
      return(SRHS_CICoveragePercentages)
    }

    SRHS_CC_betas <- betas_Confint_SRHS_models_CC[, seq(2, 28, 3)]
    SRHS_Rubin_betas <- SRHS_Rubin_betas[, 2:10]
    SRHS_Galimard_betas <- SRHS_Galimard_beta_ONLY[, 2:10]
    SRHS_CC_PercentageCoverages <- SRHS_CICoverPercentageFinal(CC_Upper_SRHS,
CC_Lower_SRHS, SRHS_CC_betas)
    SRHS_Rubin_PercentageCoverages <- SRHS_CICoverPercentageFinal(CC_Upper_SRHS,
CC_Lower_SRHS, SRHS_Rubin_betas)
    SRHS_galimard_PercentageCoverages <-
SRHS_CICoverPercentageFinal(CC_Upper_SRHS, CC_Lower_SRHS, SRHS_Galimard_betas)
    write.csv(SRHS_CC_PercentageCoverages,
"E:/MatricesOf1000Coef&SD/SRHS_CC_PercentageCoverages.csv")
    write.csv(SRHS_Rubin_PercentageCoverages,
"E:/MatricesOf1000Coef&SD/SRHS_Rubin_PercentageCoverages.csv")
    write.csv(SRHS_galimard_PercentageCoverages,
"E:/MatricesOf1000Coef&SD/SRHS_galimard_PercentageCoverages.csv")



    ##############################################################################
#########################################################

    CCSRHS1 <- CICoveragePercentageSingleCombination(SimUpperCI_CC15MCAR_SRHS,
SimLowerCI_CC15MCAR_SRHS, SRHS_CC_betas[, 1])
    CCSRHS2 <- CICoveragePercentageSingleCombination(SimUpperCI_CC15MAR_SRHS,
SimLowerCI_CC15MAR_SRHS, SRHS_CC_betas[, 2])
    CCSRHS3 <- CICoveragePercentageSingleCombination(SimUpperCI_CC15MNAR_SRHS,
SimLowerCI_CC15MNAR_SRHS, SRHS_CC_betas[, 3])
    CCSRHS4 <- CICoveragePercentageSingleCombination(SimUpperCI_CC30MCAR_SRHS,
SimLowerCI_CC15MCAR_SRHS, SRHS_CC_betas[, 4])
    CCSRHS5 <- CICoveragePercentageSingleCombination(SimUpperCI_CC30MAR_SRHS,
SimLowerCI_CC15MAR_SRHS, SRHS_CC_betas[, 5])
```

```
    CCSRHS6 <- CICoveragePercentageSingleCombination(SimUpperCI_CC30MNAR_SRHS,
SimLowerCI_CC15MNAR_SRHS, SRHS_CC_betas[, 6])
    CCSRHS7 <- CICoveragePercentageSingleCombination(SimUpperCI_CC50MCAR_SRHS,
SimLowerCI_CC50MCAR_SRHS, SRHS_CC_betas[, 7])
    CCSRHS8 <- CICoveragePercentageSingleCombination(SimUpperCI_CC50MAR_SRHS,
SimLowerCI_CC50MAR_SRHS, SRHS_CC_betas[, 8])
    CCSRHS9 <- CICoveragePercentageSingleCombination(SimUpperCI_CC50MNAR_SRHS,
SimLowerCI_CC50MNAR_SRHS, SRHS_CC_betas[, 9])

    RubinSRHS1 <-
CICoveragePercentageSingleCombination(SimUpperCI_Rubin15MCAR_SRHS,
SimLowerCI_Rubin15MCAR_SRHS, SRHS_Rubin_betas[, 1])
    RubinSRHS2 <-
CICoveragePercentageSingleCombination(SimUpperCI_Rubin15MAR_SRHS,
SimLowerCI_Rubin15MAR_SRHS, SRHS_Rubin_betas[, 2])
    RubinSRHS3 <-
CICoveragePercentageSingleCombination(SimUpperCI_Rubin15MNAR_SRHS,
SimLowerCI_Rubin15MNAR_SRHS, SRHS_Rubin_betas[, 3])
    RubinSRHS4 <-
CICoveragePercentageSingleCombination(SimUpperCI_Rubin30MCAR_SRHS,
SimLowerCI_Rubin15MCAR_SRHS, SRHS_Rubin_betas[, 4])
    RubinSRHS5 <-
CICoveragePercentageSingleCombination(SimUpperCI_Rubin30MAR_SRHS,
SimLowerCI_Rubin15MAR_SRHS, SRHS_Rubin_betas[, 5])
    RubinSRHS6 <-
CICoveragePercentageSingleCombination(SimUpperCI_Rubin30MNAR_SRHS,
SimLowerCI_Rubin15MNAR_SRHS, SRHS_Rubin_betas[, 6])
    RubinSRHS7 <-
CICoveragePercentageSingleCombination(SimUpperCI_Rubin50MCAR_SRHS,
SimLowerCI_Rubin50MCAR_SRHS, SRHS_Rubin_betas[, 7])
    RubinSRHS8 <-
CICoveragePercentageSingleCombination(SimUpperCI_Rubin50MAR_SRHS,
SimLowerCI_Rubin50MAR_SRHS, SRHS_Rubin_betas[, 8])
    RubinSRHS9 <-
CICoveragePercentageSingleCombination(SimUpperCI_Rubin50MNAR_SRHS,
SimLowerCI_Rubin50MNAR_SRHS, SRHS_Rubin_betas[, 9])

    GalimardSRHS1 <-
CICoveragePercentageSingleCombination(SimUpperCI_Galimard15MCAR_SRHS,
SimLowerCI_Galimard15MCAR_SRHS, SRHS_Galimard_betas[, 1])
    GalimardSRHS2 <-
CICoveragePercentageSingleCombination(SimUpperCI_Galimard15MAR_SRHS,
SimLowerCI_Galimard15MAR_SRHS, SRHS_Galimard_betas[, 2])
    GalimardSRHS3 <-
CICoveragePercentageSingleCombination(SimUpperCI_Galimard15MNAR_SRHS,
SimLowerCI_Galimard15MNAR_SRHS, SRHS_Galimard_betas[, 3])
```

```
    GalimardSRHS4 <-
CICoveragePercentageSingleCombination(SimUpperCI_Galimard30MCAR_SRHS,
SimLowerCI_Galimard15MCAR_SRHS, SRHS_Galimard_betas[, 4])
    GalimardSRHS5 <-
CICoveragePercentageSingleCombination(SimUpperCI_Galimard30MAR_SRHS,
SimLowerCI_Galimard15MAR_SRHS, SRHS_Galimard_betas[, 5])
    GalimardSRHS6 <-
CICoveragePercentageSingleCombination(SimUpperCI_Galimard30MNAR_SRHS,
SimLowerCI_Galimard15MNAR_SRHS, SRHS_Galimard_betas[, 6])
    GalimardSRHS7 <-
CICoveragePercentageSingleCombination(SimUpperCI_Galimard50MCAR_SRHS,
SimLowerCI_Galimard50MCAR_SRHS, SRHS_Galimard_betas[, 7])
    GalimardSRHS8 <-
CICoveragePercentageSingleCombination(SimUpperCI_Galimard50MAR_SRHS,
SimLowerCI_Galimard50MAR_SRHS, SRHS_Galimard_betas[, 8])
    GalimardSRHS9 <-
CICoveragePercentageSingleCombination(SimUpperCI_Galimard50MNAR_SRHS,
SimLowerCI_Galimard50MNAR_SRHS, SRHS_Galimard_betas[, 9])

    CCSRHSCoveragePercentage <- cbind(unlist(CCSRHS1), unlist(CCSRHS2),
unlist(CCSRHS3),
                                      unlist(CCSRHS4), unlist(CCSRHS5), unlist(CCSRHS6),
                                      unlist(CCSRHS1), unlist(CCSRHS1), unlist(CCSRHS9))
    CCSRHSCoveragePercentage <- CCSRHSCoveragePercentage*100

    RubinSRHSCoveragePercentage <- cbind(unlist(RubinSRHS1), unlist(RubinSRHS2),
unlist(RubinSRHS3),
                                         unlist(RubinSRHS4), unlist(RubinSRHS5), unlist(RubinSRHS6),
                                         unlist(RubinSRHS7), unlist(RubinSRHS8), unlist(RubinSRHS9))
    RubinSRHSCoveragePercentage <- RubinSRHSCoveragePercentage*100

    GalimardSRHSCoveragePercentage <- cbind(unlist(GalimardSRHS1),
unlist(GalimardSRHS2), unlist(GalimardSRHS3),
                                            unlist(GalimardSRHS4), unlist(GalimardSRHS5),
unlist(GalimardSRHS6),
                                            unlist(GalimardSRHS7), unlist(GalimardSRHS8),
unlist(GalimardSRHS9))

    GalimardSRHSCoveragePercentage <- GalimardSRHSCoveragePercentage*100
    write.csv(CCSRHSCoveragePercentage,
"E:/MatricesOf1000Coef&SD/CCSRHSCoveragePercentage.csv")
    write.csv(RubinSRHSCoveragePercentage,
"E:/MatricesOf1000Coef&SD/RubinSRHSCoveragePercentage.csv")
    write.csv(GalimardSRHSCoveragePercentage,
"E:/MatricesOf1000Coef&SD/GalimardSRHSCoveragePercentage.csv")
```

```
    ##############################################################################
#############
    folder <- "E:/MatricesOf1000Coef&SD/RANDHIE/CC/"
    folder <- "E:/MatricesOf1000Coef&SD/RANDHIE/Rubin/"
    folder <- "E:/MatricesOf1000Coef&SD/RANDHIE/Galimard/"

    # path to folder that holds multiple .csv files

    file_list <- list.files(path=folder, pattern="RandCC*")
    file_list <- list.files(path=folder, pattern="RandCCMeanSD*")
    file_list <- list.files(path=folder, pattern="RandMeanBeta.Rubin*")
    file_list <- list.files(path=folder, pattern="MeanBeta.Galimard*")
    file_list <- list.files(path=folder, pattern="RandCCMeanSD*")
    file_list <- list.files(path=folder, pattern="RandMeanSD.Rubin*")
    file_list <- list.files(path=folder, pattern="MeanSD.Galimard*")
    file_list <- list.files(path=folder, pattern="RandCC_sd*")
    file_list <- list.files(path=folder, pattern="Rand_Rubin*")
    file_list <- list.files(path=folder, pattern="Rand_Galimard*")

    # create list of all .csv files in folder

    # read in each .csv file in file_list and create a data frame with the same name as the .csv file

    for (i in 1:length(file_list)){

     assign(file_list[i],

          read.csv(paste(folder, file_list[i], sep=''))

     )}


    #1. Bias of Regression coefficient
    RandCC15MCARBeta <- RandCCMeanBeta.15mcar.csv[,2] -
betas_Confint_RANDHIE_models_CC_csv[,2]
    RandCC30MCARBeta <- RandCCMeanBeta.30mcar.csv[,2] -
betas_Confint_RANDHIE_models_CC_csv[,5]
    RandCC50MCARBeta <- RandCCMeanBeta.50mcar.csv[,2] -
betas_Confint_RANDHIE_models_CC_csv[,8]

    RandCC15MARBeta <- RandCCMeanBeta.15mar.csv[,2] -
betas_Confint_RANDHIE_models_CC_csv[,11]
    RandCC30MARBeta <- RandCCMeanBeta.30mar.csv[,2] -
betas_Confint_RANDHIE_models_CC_csv[,14]
```

```
    RandCC50MARBeta <- RandCCMeanBeta.50mar.csv[,2] -
betas_Confint_RANDHIE_models_CC_csv[,17]

    RandCC15MNARBeta <- RandCCMeanBeta.15mnar.csv[,2] -
betas_Confint_RANDHIE_models_CC_csv[,20]
    RandCC30MNARBeta <- RandCCMeanBeta.30mnar.csv[,2] -
betas_Confint_RANDHIE_models_CC_csv[,23]
    RandCC50MNARBeta <- RandCCMeanBeta.50mnar.csv[,2] -
betas_Confint_RANDHIE_models_CC_csv[,26]

    RandCCBiasRegCoef <- cbind(RandCC15MCARBeta, RandCC30MCARBeta,
RandCC50MCARBeta,
                        RandCC15MARBeta, RandCC30MARBeta, RandCC50MARBeta,
                        RandCC15MNARBeta, RandCC30MNARBeta, RandCC30MNARBeta)

    write.csv(RandCCBiasRegCoef,
"D:/MatricesOf1000Coef&SD/ResultsForTable/RandCCBiasRegCoef.csv")

    RandRubin15MCARBeta <- RandMeanBeta.Rubin15mcar.csv[,2] -
RANDHIE_Rubin_betas_csv[,2]
    RandRubin30MCARBeta <- RandMeanBeta.Rubin30mcar.csv[,2] -
RANDHIE_Rubin_betas_csv[,3]
    RandRubin50MCARBeta <- RandMeanBeta.Rubin50mcar.csv[,2]-
RANDHIE_Rubin_betas_csv[,4]

    RandRubin15MARBeta <- RandMeanBeta.Rubin15mar.csv[,2] -
RANDHIE_Rubin_betas_csv[,5]
    RandRubin30MARBeta <- RandMeanBeta.Rubin30mar.csv[,2] -
RANDHIE_Rubin_betas_csv[,6]
    RandRubin50MARBeta <- RandMeanBeta.Rubin50mar.csv[,2] -
RANDHIE_Rubin_betas_csv[,7]

    RandRubin15MNARBeta <- RandMeanBeta.Rubin15mnar.csv[,2] -
RANDHIE_Rubin_betas_csv[,8]
    RandRubin30MNARBeta <- RandMeanBeta.Rubin30mnar.csv[,2] -
RANDHIE_Rubin_betas_csv[,9]
    RandRubin50MNARBeta <- RandMeanBeta.Rubin50mnar.csv[,2] -
RANDHIE_Rubin_betas_csv[,10]

    RandRubinBiasRegCoef <- cbind(RandRubin15MCARBeta, RandRubin30MCARBeta,
RandRubin50MCARBeta,
                        RandRubin15MARBeta, RandRubin30MARBeta, RandRubin50MARBeta,
                        RandRubin15MNARBeta, RandRubin30MNARBeta,
RandRubin30MNARBeta)
```

```
    write.csv(RandRubinBiasRegCoef,
"D:/MatricesOf1000Coef&SD/ResultsForTable/RandRubinBiasRegCoef.csv")

    RandGalimard15MCARBeta <- MeanBeta.Galimard15mcar.csv[,2] -
RANDHIE_Galimard_beta_ONLY_csv[,2]
    RandGalimard30MCARBeta <- MeanBeta.Galimard30mcar.csv[,2] -
RANDHIE_Galimard_beta_ONLY_csv[,3]
    RandGalimard50MCARBeta <- MeanBeta.Galimard50mcar.csv[,2] -
RANDHIE_Galimard_beta_ONLY_csv[,4]

    RandGalimard15MARBeta <- MeanBeta.Galimard15mar.csv[,2] -
RANDHIE_Galimard_beta_ONLY_csv[,5]
    RandGalimard30MARBeta <- MeanBeta.Galimard30mar.csv[,2] -
RANDHIE_Galimard_beta_ONLY_csv[,6]
    RandGalimard50MARBeta <- MeanBeta.Galimard50mar.csv[,2] -
RANDHIE_Galimard_beta_ONLY_csv[,7]

    RandGalimard15MNARBeta <- MeanBeta.Galimard15mnar.csv[,2] -
RANDHIE_Galimard_beta_ONLY_csv[,8]
    RandGalimard30MNARBeta <- MeanBeta.Galimard30mnar.csv[,2] -
RANDHIE_Galimard_beta_ONLY_csv[,9]
    RandGalimard50MNARBeta <- MeanBeta.Galimard50mnar.csv[,2] -
RANDHIE_Galimard_beta_ONLY_csv[,10]

    RandGalimardBiasRegCoef <- cbind(RandGalimard15MCARBeta,
RandGalimard30MCARBeta, RandGalimard50MCARBeta,
                        RandGalimard15MARBeta, RandGalimard30MARBeta,
RandGalimard50MARBeta,
                        RandGalimard15MNARBeta, RandGalimard30MNARBeta,
RandGalimard30MNARBeta)

    write.csv(RandGalimardBiasRegCoef,
"D:/MatricesOf1000Coef&SD/ResultsForTable/RandGalimardBiasRegCoef.csv")


    #2. Relative Bias of Regression Coefficient
    RandRB_BetaCC15MCAR <-
(RandCCBiasRegCoef[,1]/betas_Confint_RANDHIE_models_CC_csv[,2])*100
    RandRB_BetaCC30MCAR <-
(RandCCBiasRegCoef[,2]/betas_Confint_RANDHIE_models_CC_csv[,5])*100
    RandRB_BetaCC50MCAR <-
(RandCCBiasRegCoef[,3]/betas_Confint_RANDHIE_models_CC_csv[,8])*100

    RandRB_BetaCC15MAR <-
(RandCCBiasRegCoef[,4]/betas_Confint_RANDHIE_models_CC_csv[,11])*100
```

```
    RandRB_BetaCC30MAR <-
(RandCCBiasRegCoef[,5]/betas_Confint_RANDHIE_models_CC_csv[,14])*100
    RandRB_BetaCC50MAR <-
(RandCCBiasRegCoef[,6]/betas_Confint_RANDHIE_models_CC_csv[,17])*100

    RandRB_BetaCC15MNAR <-
(RandCCBiasRegCoef[,7]/betas_Confint_RANDHIE_models_CC_csv[,20])*100
    RandRB_BetaCC30MNAR <-
(RandCCBiasRegCoef[,8]/betas_Confint_RANDHIE_models_CC_csv[,23])*100
    RandRB_BetaCC50MNAR <-
(RandCCBiasRegCoef[,9]/betas_Confint_RANDHIE_models_CC_csv[,26])*100

    RandCCRelatBiasRegCoef <- cbind(RandRB_BetaCC15MCAR,
RandRB_BetaCC30MCAR, RandRB_BetaCC50MCAR,
                        RandRB_BetaCC15MAR, RandRB_BetaCC30MAR,
RandRB_BetaCC50MAR,
                        RandRB_BetaCC15MNAR, RandRB_BetaCC30MNAR,
RandRB_BetaCC50MNAR)

    write.csv(RandCCRelatBiasRegCoef,
"D:/MatricesOf1000Coef&SD/ResultsForTable/RandCCRelatBiasRegCoef.csv")

    RandRB_BetaRubin15MCAR <-
(RandRubinBiasRegCoef[,1]/RANDHIE_Rubin_betas_csv[,2])*100
    RandRB_BetaRubin30MCAR <-
(RandRubinBiasRegCoef[,2]/RANDHIE_Rubin_betas_csv[,3])*100
    RandRB_BetaRubin50MCAR <-
(RandRubinBiasRegCoef[,3]/RANDHIE_Rubin_betas_csv[,4])*100

    RandRB_BetaRubin15MAR <-
(RandRubinBiasRegCoef[,4]/RANDHIE_Rubin_betas_csv[,5])*100
    RandRB_BetaRubin30MAR <-
(RandRubinBiasRegCoef[,5]/RANDHIE_Rubin_betas_csv[,6])*100
    RandRB_BetaRubin50MAR <-
(RandRubinBiasRegCoef[,6]/RANDHIE_Rubin_betas_csv[,7])*100

    RandRB_BetaRubin15MNAR <-
(RandRubinBiasRegCoef[,7]/RANDHIE_Rubin_betas_csv[,8])*100
    RandRB_BetaRubin30MNAR <-
(RandRubinBiasRegCoef[,8]/RANDHIE_Rubin_betas_csv[,9])*100
    RandRB_BetaRubin50MNAR <-
(RandRubinBiasRegCoef[,9]/RANDHIE_Rubin_betas_csv[,10])*100

    RandRubinRelatBiasRegCoef <- cbind(RandRB_BetaRubin15MCAR,
RandRB_BetaRubin30MCAR, RandRB_BetaRubin50MCAR,
```

```
                        RandRB_BetaRubin15MAR, RandRB_BetaRubin30MAR,
RandRB_BetaRubin50MAR,
                        RandRB_BetaRubin15MNAR, RandRB_BetaRubin30MNAR,
RandRB_BetaRubin50MNAR)

    write.csv(RandRubinRelatBiasRegCoef,
"D:/MatricesOf1000Coef&SD/ResultsForTable/RandRubinRelatBiasRegCoef.csv")

    RandRB_BetaRubin15MCAR <-
(RandGalimardBiasRegCoef[,1]/RANDHIE_Rubin_betas_csv[,2])*100
    RandRB_BetaRubin30MCAR <-
(RandGalimardBiasRegCoef[,2]/RANDHIE_Rubin_betas_csv[,3])*100
    RandRB_BetaRubin50MCAR <-
(RandGalimardBiasRegCoef[,3]/RANDHIE_Rubin_betas_csv[,4])*100

    RandRB_BetaRubin15MAR <-
(RandGalimardBiasRegCoef[,4]/RANDHIE_Rubin_betas_csv[,5])*100
    RandRB_BetaRubin30MAR <-
(RandGalimardBiasRegCoef[,5]/RANDHIE_Rubin_betas_csv[,6])*100
    RandRB_BetaRubin50MAR <-
(RandGalimardBiasRegCoef[,6]/RANDHIE_Rubin_betas_csv[,7])*100

    RandRB_BetaRubin15MNAR <-
(RandGalimardBiasRegCoef[,7]/RANDHIE_Rubin_betas_csv[,8])*100
    RandRB_BetaRubin30MNAR <-
(RandGalimardBiasRegCoef[,8]/RANDHIE_Rubin_betas_csv[,9])*100
    RandRB_BetaRubin50MNAR <-
(RandGalimardBiasRegCoef[,9]/RANDHIE_Rubin_betas_csv[,10])*100

    RandGalimardRelatBiasRegCoef <- cbind(RandRB_BetaRubin15MCAR,
RandRB_BetaRubin30MCAR, RandRB_BetaRubin50MCAR,
                        RandRB_BetaRubin15MAR, RandRB_BetaRubin30MAR,
RandRB_BetaRubin50MAR,
                        RandRB_BetaRubin15MNAR, RandRB_BetaRubin30MNAR,
RandRB_BetaRubin50MNAR)
    write.csv(RandGalimardRelatBiasRegCoef,
"D:/MatricesOf1000Coef&SD/ResultsForTable/RandGalimardRelatBiasRegCoef.csv")

    #3. Standardized bias of regression coefficient

    RandRB_BetaCC15MCAR <-
(RandCCBiasRegCoef[,1]/SE_RANDHIE_models_CC[,2])*100
    RandRB_BetaCC30MCAR <-
(RandCCBiasRegCoef[,2]/SE_RANDHIE_models_CC[,3])*100
    RandRB_BetaCC50MCAR <-
(RandCCBiasRegCoef[,3]/SE_RANDHIE_models_CC[,4])*100
```

```
    RandRB_BetaCC15MAR <-
(RandCCBiasRegCoef[,4]/SE_RANDHIE_models_CC[,5])*100
    RandRB_BetaCC30MAR <-
(RandCCBiasRegCoef[,5]/SE_RANDHIE_models_CC[,6])*100
    RandRB_BetaCC50MAR <-
(RandCCBiasRegCoef[,6]/SE_RANDHIE_models_CC[,7])*100

    RandRB_BetaCC15MNAR <-
(RandCCBiasRegCoef[,7]/SE_RANDHIE_models_CC[,8])*100
    RandRB_BetaCC30MNAR <-
(RandCCBiasRegCoef[,8]/SE_RANDHIE_models_CC[,9])*100
    RandRB_BetaCC50MNAR <-
(RandCCBiasRegCoef[,9]/SE_RANDHIE_models_CC[,10])*100

    RandCCStandBiasRegCoef <- cbind(RandRB_BetaCC15MCAR,
RandRB_BetaCC30MCAR, RandRB_BetaCC50MCAR,
                          RandRB_BetaCC15MAR, RandRB_BetaCC30MAR,
RandRB_BetaCC50MAR,
                          RandRB_BetaCC15MNAR, RandRB_BetaCC30MNAR,
RandRB_BetaCC50MNAR)
    write.csv(RandCCStandBiasRegCoef,
"D:/MatricesOf1000Coef&SD/ResultsForTable/RandCCStandBiasRegCoef.csv")

    RandRB_BetaRubin15MCAR <-
(RandRubinBiasRegCoef[,1]/RANDHIE_Rubin_SE_csv[,2])*100
    RandRB_BetaRubin30MCAR <-
(RandRubinBiasRegCoef[,2]/RANDHIE_Rubin_SE_csv[,3])*100
    RandRB_BetaRubin50MCAR <-
(RandRubinBiasRegCoef[,3]/RANDHIE_Rubin_SE_csv[,4])*100

    RandRB_BetaRubin15MAR <-
(RandRubinBiasRegCoef[,4]/RANDHIE_Rubin_SE_csv[,5])*100
    RandRB_BetaRubin30MAR <-
(RandRubinBiasRegCoef[,5]/RANDHIE_Rubin_SE_csv[,6])*100
    RandRB_BetaRubin50MAR <-
(RandRubinBiasRegCoef[,6]/RANDHIE_Rubin_SE_csv[,7])*100

    RandRB_BetaRubin15MNAR <-
(RandRubinBiasRegCoef[,7]/RANDHIE_Rubin_SE_csv[,8])*100
    RandRB_BetaRubin30MNAR <-
(RandRubinBiasRegCoef[,8]/RANDHIE_Rubin_SE_csv[,9])*100
    RandRB_BetaRubin50MNAR <-
(RandRubinBiasRegCoef[,9]/RANDHIE_Rubin_SE_csv[,10])*100
```

```
    RandRubinStandBiasRegCoef <- cbind(RandRB_BetaRubin15MCAR,
RandRB_BetaRubin30MCAR, RandRB_BetaRubin50MCAR,
                               RandRB_BetaRubin15MAR, RandRB_BetaRubin30MAR,
RandRB_BetaRubin50MAR,
                               RandRB_BetaRubin15MNAR, RandRB_BetaRubin30MNAR,
RandRB_BetaRubin50MNAR)
    write.csv(RandRubinStandBiasRegCoef,
"D:/MatricesOf1000Coef&SD/ResultsForTable/RandRubinStandBiasRegCoef.csv")

    RandRB_BetaRubin15MCAR <-
(RandGalimardBiasRegCoef[,1]/RANDHIE_Galimard_SE_ONLY_csv[,2])*100
    RandRB_BetaRubin30MCAR <-
(RandGalimardBiasRegCoef[,2]/RANDHIE_Galimard_SE_ONLY_csv[,3])*100
    RandRB_BetaRubin50MCAR <-
(RandGalimardBiasRegCoef[,3]/RANDHIE_Galimard_SE_ONLY_csv[,4])*100

    RandRB_BetaRubin15MAR <-
(RandGalimardBiasRegCoef[,4]/RANDHIE_Galimard_SE_ONLY_csv[,5])*100
    RandRB_BetaRubin30MAR <-
(RandGalimardBiasRegCoef[,5]/RANDHIE_Galimard_SE_ONLY_csv[,6])*100
    RandRB_BetaRubin50MAR <-
(RandGalimardBiasRegCoef[,6]/RANDHIE_Galimard_SE_ONLY_csv[,7])*100

    RandRB_BetaRubin15MNAR <-
(RandGalimardBiasRegCoef[,7]/RANDHIE_Galimard_SE_ONLY_csv[,8])*100
    RandRB_BetaRubin30MNAR <-
(RandGalimardBiasRegCoef[,8]/RANDHIE_Galimard_SE_ONLY_csv[,9])*100
    RandRB_BetaRubin50MNAR <-
(RandGalimardBiasRegCoef[,9]/RANDHIE_Galimard_SE_ONLY_csv[,10])*100

    RandGalimardStandBiasRegCoef <- cbind(RandRB_BetaRubin15MCAR,
RandRB_BetaRubin30MCAR, RandRB_BetaRubin50MCAR,
                               RandRB_BetaRubin15MAR, RandRB_BetaRubin30MAR,
RandRB_BetaRubin50MAR,
                               RandRB_BetaRubin15MNAR, RandRB_BetaRubin30MNAR,
RandRB_BetaRubin50MNAR)
    write.csv(RandGalimardStandBiasRegCoef,
"D:/MatricesOf1000Coef&SD/ResultsForTable/RandGalimardStandBiasRegCoef.csv")

    #4. Mean Square Error (MSE) of regression coefficient

    RandMSE_BetaCC15MCAR <-
(RandCCBiasRegCoef[,1])**2/(RandCCMeanSD.15mcar.csv[,2])**2
    RandMSE_BetaCC30MCAR <-
(RandCCBiasRegCoef[,2])**2/(RandCCMeanSD.30mcar.csv[,2])**2
```

```
    RandMSE_BetaCC50MCAR <-
(RandCCBiasRegCoef[,3])**2/(RandCCMeanSD.50mcar.csv[,2])**2

    RandMSE_BetaCC15MAR <-
(RandCCBiasRegCoef[,4])**2/(RandCCMeanSD.15mar.csv[,2])**2
    RandMSE_BetaCC30MAR <-
(RandCCBiasRegCoef[,5])**2/(RandCCMeanSD.30mar.csv[,2])**2
    RandMSE_BetaCC50MAR <-
(RandCCBiasRegCoef[,6])**2/(RandCCMeanSD.50mar.csv[,2])**2

    RandMSE_BetaCC15MNAR <-
(RandCCBiasRegCoef[,7])**2/(RandCCMeanSD.15mnar.csv[,2])**2
    RandMSE_BetaCC30MNAR <-
(RandCCBiasRegCoef[,8])**2/(RandCCMeanSD.30mnar.csv[,2])**2
    RandMSE_BetaCC50MNAR <-
(RandCCBiasRegCoef[,9])**2/(RandCCMeanSD.50mnar.csv[,2])**2

    RandCCMSE_Beta <- cbind(RandMSE_BetaCC15MCAR, RandMSE_BetaCC30MCAR,
RandMSE_BetaCC50MCAR,
                    RandMSE_BetaCC15MAR, RandMSE_BetaCC30MAR,
RandMSE_BetaCC50MAR,
                    RandMSE_BetaCC15MNAR, RandMSE_BetaCC30MNAR,
RandMSE_BetaCC50MNAR)
    write.csv(RandCCMSE_Beta,
"D:/MatricesOf1000Coef&SD/ResultsForTable/RandCCMSE_Beta.csv")

    RandMSE_BetaRubin15MCAR <-
(RandRubinBiasRegCoef[,1])**2/(RandMeanSD.Rubin15mcar.csv[,2])**2
    RandMSE_BetaRubin30MCAR <-
(RandRubinBiasRegCoef[,2])**2/(RandMeanSD.Rubin30mcar.csv[,2])**2
    RandMSE_BetaRubin50MCAR <-
(RandRubinBiasRegCoef[,3])**2/(RandMeanSD.Rubin30mcar.csv[,2])**2

    RandMSE_BetaRubin15MAR <-
(RandRubinBiasRegCoef[,4])**2/(RandMeanSD.Rubin15mar.csv[,2])**2
    RandMSE_BetaRubin30MAR <-
(RandRubinBiasRegCoef[,5])**2/(RandMeanSD.Rubin30mar.csv[,2])**2
    RandMSE_BetaRubin50MAR <-
(RandRubinBiasRegCoef[,6])**2/(RandMeanSD.Rubin50mar.csv[,2])**2

    RandMSE_BetaRubin15MNAR <-
(RandRubinBiasRegCoef[,7])**2/(RandMeanSD.Rubin15mnar.csv[,2])**2
    RandMSE_BetaRubin30MNAR <-
(RandRubinBiasRegCoef[,8])**2/(RandMeanSD.Rubin30mnar.csv[,2])**2
    RandMSE_BetaRubin50MNAR <-
(RandRubinBiasRegCoef[,9])**2/(RandMeanSD.Rubin50mnar.csv[,2])**2
```

```
    RandRubinMSE_Beta <- cbind(RandMSE_BetaRubin15MCAR,
RandMSE_BetaRubin30MCAR, RandMSE_BetaRubin50MCAR,
                        RandMSE_BetaRubin15MAR, RandMSE_BetaRubin30MAR,
RandMSE_BetaRubin50MAR,
                        RandMSE_BetaRubin15MNAR, RandMSE_BetaRubin30MNAR,
RandMSE_BetaRubin50MNAR)
    write.csv(RandRubinMSE_Beta,
"D:/MatricesOf1000Coef&SD/ResultsForTable/RandRubinMSE_Beta.csv")

    RandMSE_BetaGalimard15MCAR <-
(RandGalimardBiasRegCoef[,1])**2/(MeanSD.Galimard15mcar.csv[,2])**2
    RandMSE_BetaGalimard30MCAR <-
(RandGalimardBiasRegCoef[,2])**2/(MeanSD.Galimard30mcar.csv[,2])**2
    RandMSE_BetaGalimard50MCAR <-
(RandGalimardBiasRegCoef[,3])**2/(MeanSD.Galimard50mcar.csv[,2])**2

    RandMSE_BetaGalimard15MAR <-
(RandGalimardBiasRegCoef[,4])**2/(MeanSD.Galimard15mar.csv[,2])**2
    RandMSE_BetaGalimard30MAR <-
(RandGalimardBiasRegCoef[,5])**2/(MeanSD.Galimard30mar.csv[,2])**2
    RandMSE_BetaGalimard50MAR <-
(RandGalimardBiasRegCoef[,6])**2/(MeanSD.Galimard50mar.csv[,2])**2

    RandMSE_BetaGalimard15MNAR <-
(RandGalimardBiasRegCoef[,7])**2/(MeanSD.Galimard15mnar.csv[,2])**2
    RandMSE_BetaGalimard30MNAR <-
(RandGalimardBiasRegCoef[,8])**2/(MeanSD.Galimard30mnar.csv[,2])**2
    RandMSE_BetaGalimard50MNAR <-
(RandGalimardBiasRegCoef[,9])**2/(MeanSD.Galimard50mnar.csv[,2])**2

    RandGalimardMSE_Beta <- cbind(RandMSE_BetaGalimard15MCAR,
RandMSE_BetaGalimard30MCAR, RandMSE_BetaGalimard50MCAR,
                        RandMSE_BetaGalimard15MAR, RandMSE_BetaGalimard30MAR,
RandMSE_BetaGalimard50MAR,
                        RandMSE_BetaGalimard15MNAR, RandMSE_BetaGalimard30MNAR,
RandMSE_BetaGalimard50MNAR)
    write.csv(RandGalimardMSE_Beta,
"D:/MatricesOf1000Coef&SD/ResultsForTable/RandGalimardMSE_Beta.csv")


    #5. Average 95% Confidence Interval

    RandAvg95CI_CC15MCAR <- colSums(2*1.96*RandCC_sd15mcar.csv[,2:9])/1000
    RandAvg95CI_CC30MCAR <- colSums(2*1.96*RandCC_sd30mcar.csv[,2:9])/1000
    RandAvg95CI_CC50MCAR <- colSums(2*1.96*RandCC_sd50mcar.csv[,2:9])/1000
```

```
    RandAvg95CI_CC15MAR <- colSums(2*1.96*RandCC_sd15mar.csv[,2:9])/1000
    RandAvg95CI_CC30MAR <- colSums(2*1.96*RandCC_sd30mar.csv[,2:9])/1000
    RandAvg95CI_CC50MAR <-colSums(2*1.96*RandCC_sd50mar.csv[,2:9])/1000

    RandAvg95CI_CC15MNAR <- colSums(2*1.96*RandCC_sd15mnar.csv[,2:9])/1000
    RandAvg95CI_CC30MNAR <- colSums(2*1.96*RandCC_sd30mnar.csv[,2:9])/1000
    RandAvg95CI_CC50MNAR <-colSums(2*1.96*RandCC_sd50mnar.csv[,2:9])/1000

    RandCCAvg95CI <- cbind(RandAvg95CI_CC15MCAR, RandAvg95CI_CC30MCAR,
RandAvg95CI_CC50MCAR,
                  RandAvg95CI_CC15MAR, RandAvg95CI_CC30MAR,
RandAvg95CI_CC50MAR,
                  RandAvg95CI_CC15MNAR, RandAvg95CI_CC30MNAR,
RandAvg95CI_CC50MNAR)
    write.csv(RandCCAvg95CI,"D:/MatricesOf1000Coef&SD/ResultsForTable/RandCCAvg95
CI.csv")

    RandAvg95CI_Rubin15MCAR <- colSums(2*1.96*Rand_Rubinsd15mcar.csv[,2:9])/1000
    RandAvg95CI_Rubin30MCAR <- colSums(2*1.96*Rand_Rubinsd30mcar.csv[,2:9])/1000
    RandAvg95CI_Rubin50MCAR <- colSums(2*1.96*Rand_Rubinsd50mcar.csv[,2:9])/1000

    RandAvg95CI_Rubin15MAR <- colSums(2*1.96*Rand_Rubinsd15mar.csv[,2:9])/1000
    RandAvg95CI_Rubin30MAR <- colSums(2*1.96*Rand_Rubinsd30mar.csv[,2:9])/1000
    RandAvg95CI_Rubin50MAR <-colSums(2*1.96*Rand_Rubinsd50mar.csv[,2:9])/1000

    RandAvg95CI_Rubin15MNAR <- colSums(2*1.96*Rand_Rubinsd15mnar.csv[,2:9])/1000
    RandAvg95CI_Rubin30MNAR <- colSums(2*1.96*Rand_Rubinsd30mnar.csv[,2:9])/1000
    RandAvg95CI_Rubin50MNAR <-colSums(2*1.96*Rand_Rubinsd50mnar.csv[,2:9])/1000

    RandRubinAvg95CI <- cbind(RandAvg95CI_Rubin15MCAR,
RandAvg95CI_Rubin30MCAR, RandAvg95CI_Rubin50MCAR,
                    RandAvg95CI_Rubin15MAR, RandAvg95CI_Rubin30MAR,
RandAvg95CI_Rubin50MAR,
                    RandAvg95CI_Rubin15MNAR, RandAvg95CI_Rubin30MNAR,
RandAvg95CI_Rubin50MNAR)

    write.csv(RandRubinAvg95CI,"D:/MatricesOf1000Coef&SD/ResultsForTable/RandRubinA
vg95CI.csv" )

    RandAvg95CI_Galimard15MCAR <-
colSums(2*1.96*M_Galimardsd15mcar.csv[,2:9])/1000
    RandAvg95CI_Galimard30MCAR <-
colSums(2*1.96*M_Galimardsd30mcar.csv[,2:9])/1000
    RandAvg95CI_Galimard50MCAR <-
colSums(2*1.96*M_Galimardsd50mcar.csv[,2:9])/1000
```

```
    RandAvg95CI_Galimard15MAR <- colSums(2*1.96*M_Galimardsd15mar.csv[,2:9])/1000
    RandAvg95CI_Galimard30MAR <- colSums(2*1.96*M_Galimardsd30mar.csv[,2:9])/1000
    RandAvg95CI_Galimard50MAR <-colSums(2*1.96*M_Galimardsd50mar.csv[,2:9])/1000

    RandAvg95CI_Galimard15MNAR <-
colSums(2*1.96*M_Galimardsd15mnar.csv[,2:9])/1000
    RandAvg95CI_Galimard30MNAR <-
colSums(2*1.96*M_Galimardsd30mnar.csv[,2:9])/1000
    RandAvg95CI_Galimard50MNAR <-
colSums(2*1.96*M_Galimardsd50mnar.csv[,2:9])/1000

    RandGalimardAvg95CI <- cbind(RandAvg95CI_Galimard15MCAR,
RandAvg95CI_Galimard30MCAR, RandAvg95CI_Galimard50MCAR,
                        RandAvg95CI_Galimard15MAR, RandAvg95CI_Galimard30MAR,
RandAvg95CI_Galimard50MAR,
                        RandAvg95CI_Galimard15MNAR, RandAvg95CI_Galimard30MNAR,
RandAvg95CI_Galimard50MNAR)
    write.csv(RandGalimardAvg95CI,
"D:/MatricesOf1000Coef&SD/ResultsForTable/RandGalimardAvg95CI.csv")


    #6. Coverage of True Coefficients by 95% CI
    #UpperCI

    SimUpperCI_CC15MCAR_Rand <- RandCC_coef15mcar.csv[, 2:9] +
1.96*RandCC_sd15mcar.csv[, 2:9]
    SimUpperCI_CC30MCAR_Rand <- RandCC_coef30mcar.csv[, 2:9] +
1.96*RandCC_sd30mcar.csv[, 2:9]
    SimUpperCI_CC50MCAR_Rand <- RandCC_coef50mcar.csv[, 2:9] +
1.96*RandCC_sd50mcar.csv[, 2:9]

    SimUpperCI_CC15MAR_Rand <- RandCC_coef15mar.csv[, 2:9] +
1.96*RandCC_sd15mar.csv[, 2:9]
    SimUpperCI_CC30MAR_Rand <- RandCC_coef30mar.csv[, 2:9] +
1.96*RandCC_sd30mar.csv[, 2:9]
    SimUpperCI_CC50MAR_Rand <- RandCC_coef50mar.csv[, 2:9] +
1.96*RandCC_sd50mar.csv[, 2:9]

    SimUpperCI_CC15MNAR_Rand <- RandCC_coef15mnar.csv[, 2:9] +
1.96*RandCC_coef15mnar.csv[, 2:9]
    SimUpperCI_CC30MNAR_Rand <- RandCC_coef30mnar.csv[, 2:9] +
1.96*RandCC_coef30mnar.csv[, 2:9]
    SimUpperCI_CC50MNAR_Rand<- RandCC_coef50mnar.csv[, 2:9] +
1.96*RandCC_coef50mnar.csv[, 2:9]
```

```
    SimUpperCI_Rubin15MCAR_Rand <- Rand_Rubincoef15mcar.csv[, 2:9] +
1.96*Rand_Rubinsd15mcar.csv[, 2:9]
    SimUpperCI_Rubin30MCAR_Rand <- Rand_Rubincoef30mcar.csv[, 2:9] +
1.96*Rand_Rubinsd30mcar.csv[, 2:9]
    SimUpperCI_Rubin50MCAR_Rand <- Rand_Rubincoef50mcar.csv[, 2:9] +
1.96*Rand_Rubinsd50mcar.csv[, 2:9]

    SimUpperCI_Rubin15MAR_Rand <- Rand_Rubincoef15mar.csv[, 2:9] +
1.96*Rand_Rubinsd15mar.csv[, 2:9]
    SimUpperCI_Rubin30MAR_Rand <- Rand_Rubincoef30mar.csv[, 2:9] +
1.96*Rand_Rubinsd30mar.csv[, 2:9]
    SimUpperCI_Rubin50MAR_Rand <- Rand_Rubincoef50mar.csv[, 2:9] +
1.96*Rand_Rubinsd50mar.csv[, 2:9]

    SimUpperCI_Rubin15MNAR_Rand <- Rand_Rubincoef15mnar.csv[, 2:9] +
1.96*Rand_Rubinsd15mnar.csv[, 2:9]
    SimUpperCI_Rubin30MNAR_Rand <- Rand_Rubincoef30mnar.csv[, 2:9] +
1.96*Rand_Rubinsd30mnar.csv[, 2:9]
    SimUpperCI_Rubin50MNAR_Rand <- Rand_Rubincoef50mnar.csv[, 2:9] +
1.96*Rand_Rubinsd50mnar.csv[, 2:9]

    SimUpperCI_Galimard15MCAR_Rand <- Rand_Galimardcoef15mcar.csv[, 2:9] +
1.96*Rand_Galimardsd15mcar.csv[, 2:9]
    SimUpperCI_Galimard30MCAR_Rand <- Rand_Galimardcoef30mcar.csv[, 2:9] +
1.96*Rand_Galimardsd30mcar.csv[, 2:9]
    SimUpperCI_Galimard50MCAR_Rand <- Rand_Galimardcoef50mcar.csv[, 2:9] +
1.96*Rand_Galimardsd50mcar.csv[, 2:9]

    SimUpperCI_Galimard15MAR_Rand <- Rand_Galimardcoef15mar.csv[, 2:9] +
1.96*Rand_Galimardsd15mar.csv[, 2:9]
    SimUpperCI_Galimard30MAR_Rand <- Rand_Galimardcoef30mar.csv[, 2:9] +
1.96*Rand_Galimardsd30mar.csv[, 2:9]
    SimUpperCI_Galimard50MAR_Rand <- Rand_Galimardcoef50mar.csv[, 2:9] +
1.96*Rand_Galimardsd50mar.csv[, 2:9]

    SimUpperCI_Galimard15MNAR_Rand <- Rand_Galimardcoef15mnar.csv[, 2:9] +
1.96*Rand_Galimardsd15mnar.csv[, 2:9]
    SimUpperCI_Galimard30MNAR_Rand <- Rand_Galimardcoef30mnar.csv[, 2:9] +
1.96*Rand_Galimardsd30mnar.csv[, 2:9]
    SimUpperCI_Galimard50MNAR_Rand <- Rand_Galimardcoef50mnar.csv[, 2:9] +
1.96*Rand_Galimardsd50mnar.csv[, 2:9]

    #LowerCI

    SimLowerCI_CC15MCAR_Rand <- RandCC_coef15mcar.csv[, 2:9] -
1.96*RandCC_sd15mcar.csv[, 2:9]
```

```
    SimLowerCI_CC30MCAR_Rand <- RandCC_coef30mcar.csv[, 2:9] -
1.96*RandCC_sd30mcar.csv[, 2:9]
    SimLowerCI_CC50MCAR_Rand <- RandCC_coef50mcar.csv[, 2:9] -
1.96*RandCC_sd50mcar.csv[, 2:9]

    SimLowerCI_CC15MAR_Rand <- RandCC_coef15mar.csv[, 2:9] -
1.96*RandCC_sd15mar.csv[, 2:9]
    SimLowerCI_CC30MAR_Rand <- RandCC_coef30mar.csv[, 2:9] -
1.96*RandCC_sd30mar.csv[, 2:9]
    SimLowerCI_CC50MAR_Rand <- RandCC_coef50mar.csv[, 2:9] -
1.96*RandCC_sd50mar.csv[, 2:9]

    SimLowerCI_CC15MNAR_Rand <- RandCC_coef15mnar.csv[, 2:9] -
1.96*RandCC_coef15mnar.csv[, 2:9]
    SimLowerCI_CC30MNAR_Rand <- RandCC_coef30mnar.csv[, 2:9] -
1.96*RandCC_coef30mnar.csv[, 2:9]
    SimLowerCI_CC50MNAR_Rand<- RandCC_coef50mnar.csv[, 2:9] -
1.96*RandCC_coef50mnar.csv[, 2:9]

    SimLowerCI_Rubin15MCAR_Rand <- Rand_Rubincoef15mcar.csv[, 2:9] -
1.96*Rand_Rubinsd15mcar.csv[, 2:9]
    SimLowerCI_Rubin30MCAR_Rand <- Rand_Rubincoef30mcar.csv[, 2:9] -
1.96*Rand_Rubinsd30mcar.csv[, 2:9]
    SimLowerCI_Rubin50MCAR_Rand <- Rand_Rubincoef50mcar.csv[, 2:9] -
1.96*Rand_Rubinsd50mcar.csv[, 2:9]

    SimLowerCI_Rubin15MAR_Rand <- Rand_Rubincoef15mar.csv[, 2:9] -
1.96*Rand_Rubinsd15mar.csv[, 2:9]
    SimLowerCI_Rubin30MAR_Rand <- Rand_Rubincoef30mar.csv[, 2:9] -
1.96*Rand_Rubinsd30mar.csv[, 2:9]
    SimLowerCI_Rubin50MAR_Rand <- Rand_Rubincoef50mar.csv[, 2:9] -
1.96*Rand_Rubinsd50mar.csv[, 2:9]

    SimLowerCI_Rubin15MNAR_Rand <- Rand_Rubincoef15mnar.csv[, 2:9] -
1.96*Rand_Rubinsd15mnar.csv[, 2:9]
    SimLowerCI_Rubin30MNAR_Rand <- Rand_Rubincoef30mnar.csv[, 2:9] -
1.96*Rand_Rubinsd30mnar.csv[, 2:9]
    SimLowerCI_Rubin50MNAR_Rand <- Rand_Rubincoef50mnar.csv[, 2:9] -
1.96*Rand_Rubinsd50mnar.csv[, 2:9]

    SimLowerCI_Galimard15MCAR_Rand <- Rand_Galimardcoef15mcar.csv[, 2:9] -
1.96*Rand_Galimardsd15mcar.csv[, 2:9]
    SimLowerCI_Galimard30MCAR_Rand <- Rand_Galimardcoef30mcar.csv[, 2:9] -
1.96*Rand_Galimardsd30mcar.csv[, 2:9]
    SimLowerCI_Galimard50MCAR_Rand <- Rand_Galimardcoef50mcar.csv[, 2:9] -
1.96*Rand_Galimardsd50mcar.csv[, 2:9]
```

```
    SimLowerCI_Galimard15MAR_Rand <- Rand_Galimardcoef15mar.csv[, 2:9] -
1.96*Rand_Galimardsd15mar.csv[, 2:9]
    SimLowerCI_Galimard30MAR_Rand <- Rand_Galimardcoef30mar.csv[, 2:9] -
1.96*Rand_Galimardsd30mar.csv[, 2:9]
    SimLowerCI_Galimard50MAR_Rand <- Rand_Galimardcoef50mar.csv[, 2:9] -
1.96*Rand_Galimardsd50mar.csv[, 2:9]

    SimLowerCI_Galimard15MNAR_Rand <- Rand_Galimardcoef15mnar.csv[, 2:9] -
1.96*Rand_Galimardsd15mnar.csv[, 2:9]
    SimLowerCI_Galimard30MNAR_Rand <- Rand_Galimardcoef30mnar.csv[, 2:9] -
1.96*Rand_Galimardsd30mnar.csv[, 2:9]
    SimLowerCI_Galimard50MNAR_Rand <- Rand_Galimardcoef50mnar.csv[, 2:9] -
1.96*Rand_Galimardsd50mnar.csv[, 2:9]

    CoveragePercentageOfSimCI <- function(UpperCI1000, LowerCI1000, Beta){
      In_OR_Out <- rep(NA, 1000)
      for (i in 1:1000){
        if ((Beta <UpperCI1000[i])&(Beta> LowerCI1000[i])){
          In_OR_Out[i] <-1}
        else In_OR_Out[i] <- 0
      }
      return(In_OR_Out)
    }



    CICoveragePercentageSingleCombination <- function(UpperCI1000Matrix,
LowerCI1000Matrix, BetaColumn){
      CICoveragePercentagesList <- list()
      for (i in 1:8){
        Percentage <- sum(CoveragePercentageOfSimCI(UpperCI1000Matrix[,i],
LowerCI1000Matrix[,i], BetaColumn[i, ]))/1000
        CICoveragePercentagesList[[i]] <- Percentage
      }
      return(CICoveragePercentagesList)
    }

    CC_CI_Upper_RANDHIE <- list(SimUpperCI_CC15MCAR_Rand,
SimUpperCI_CC15MAR_Rand, SimUpperCI_CC15MNAR_Rand,
                        SimUpperCI_CC30MCAR_Rand, SimUpperCI_CC30MAR_Rand,
SimUpperCI_CC30MNAR_Rand,
                        SimUpperCI_CC50MCAR_Rand, SimUpperCI_CC50MAR_Rand,
SimUpperCI_CC50MNAR_Rand)
```

```
    CC_CI_Lower_RANDHIE <- list(SimLowerCI_CC15MCAR_Rand,
SimUpperCI_CC15MAR_Rand, SimUpperCI_CC15MNAR_Rand,
                    SimLowerCI_CC30MCAR_Rand, SimUpperCI_CC30MAR_Rand,
SimUpperCI_CC30MNAR_Rand,
                    SimLowerCI_CC50MCAR_Rand, SimUpperCI_CC50MAR_Rand,
SimUpperCI_CC50MNAR_Rand)

    Rubin_CI_Upper_RANDHIE <- list(SimUpperCI_Rubin15MCAR_Rand,
SimUpperCI_Rubin15MAR_Rand, SimUpperCI_Rubin15MNAR_Rand,
                        SimUpperCI_Rubin30MCAR_Rand, SimUpperCI_Rubin30MAR_Rand,
SimUpperCI_Rubin30MNAR_Rand,
                        SimUpperCI_Rubin50MCAR_Rand, SimUpperCI_Rubin50MAR_Rand,
SimUpperCI_Rubin50MNAR_Rand)

    Rubin_CI_Lower_RANDHIE <- list(SimLowerCI_Rubin15MCAR_Rand,
SimUpperCI_Rubin15MAR_Rand, SimUpperCI_Rubin15MNAR_Rand,
                        SimLowerCI_Rubin30MCAR_Rand, SimUpperCI_Rubin30MAR_Rand,
SimUpperCI_Rubin30MNAR_Rand,
                        SimLowerCI_Rubin50MCAR_Rand, SimUpperCI_Rubin50MAR_Rand,
SimUpperCI_Rubin50MNAR_Rand)

    Galimard_CI_Upper_RANDHIE <- list(SimUpperCI_Galimard15MCAR_Rand,
SimUpperCI_Galimard15MAR_Rand, SimUpperCI_Galimard15MNAR_Rand,
                        SimUpperCI_Galimard30MCAR_Rand,
SimUpperCI_Galimard30MAR_Rand, SimUpperCI_Galimard30MNAR_Rand,
                        SimUpperCI_Galimard50MCAR_Rand,
SimUpperCI_Galimard50MAR_Rand, SimUpperCI_Galimard50MNAR_Rand)

    Galimard_CI_Lower_RANDHIE <- list(SimLowerCI_Galimard15MCAR_Rand,
SimUpperCI_Galimard15MAR_Rand, SimUpperCI_Galimard15MNAR_Rand,
                        SimLowerCI_Galimard30MCAR_Rand,
SimUpperCI_Galimard30MAR_Rand, SimUpperCI_Galimard30MNAR_Rand,
                        SimLowerCI_Galimard50MCAR_Rand,
SimUpperCI_Galimard50MAR_Rand, SimUpperCI_Galimard50MNAR_Rand)

    RANDHIE_CICoverPercentageFinal <- function(SimUpperVector, SimLowerVector,
BetaMatrix){
      RANDHIE_CICoveragePercentages <- c()
      for (i in 1:9){
        RANDHIE_CICoveragePercentagesList <-
CICoveragePercentageSingleCombination(SimUpperVector[[i]], SimLowerVector[[i]],
BetaMatrix[, i])
        RANDHIE_CICoveragePercentages <- cbind(RANDHIE_CICoveragePercentages,
as.vector(unlist(RANDHIE_CICoveragePercentagesList)))
      }
      return(RANDHIE_CICoveragePercentages)
```

```
    }

    RANDHIE_CC_betas <- betas_Confint_RANDHIE_models_CC_csv[, seq(2, 28, 3)]
    RANDHIE_Rubin_betas <- RANDHIE_Rubin_betas_csv[, 2:10]
    RANDHIE_Galimard_betas <- RANDHIE_Galimard_beta_ONLY_csv[, 2:10]

    RANDHIE_CC_PercentageCoverages <-
RANDHIE_CICoverPercentageFinal(CC_CI_Upper_RANDHIE, CC_CI_Lower_RANDHIE,
RANDHIE_CC_betas)
    RANDHIE_Rubin_PercentageCoverages <-
RANDHIE_CICoverPercentageFinal(Rubin_CI_Upper_RANDHIE,
Rubin_CI_Upper_RANDHIE, RANDHIE_Rubin_betas)
    RANDHIE_galimard_PercentageCoverages <-
RANDHIE_CICoverPercentageFinal(Galimard_CI_Upper_RANDHIE,
Galimard_CI_Upper_RANDHIE, RANDHIE_Galimard_betas)
    write.csv(RANDHIE_CC_PercentageCoverages,
"E:/MatricesOf1000Coef&SD/RANDHIE_CC_PercentageCoverages.csv")
    write.csv(RANDHIE_Rubin_PercentageCoverages,
"E:/MatricesOf1000Coef&SD/RANDHIE_Rubin_PercentageCoverages.csv")
    write.csv(RANDHIE_galimard_PercentageCoverages,
"E:/MatricesOf1000Coef&SD/RANDHIE_galimard_PercentageCoverages.csv")

    ##########################################################################
##########################################################
    CCRand1 <- CICoveragePercentageSingleCombination(SimUpperCI_CC15MCAR_Rand,
SimLowerCI_CC15MCAR_Rand, RANDHIE_CC_betas[, 1])
    CCRand2 <- CICoveragePercentageSingleCombination(SimUpperCI_CC15MAR_Rand,
SimLowerCI_CC15MAR_Rand, RANDHIE_CC_betas[, 2])
    CCRand3 <- CICoveragePercentageSingleCombination(SimUpperCI_CC15MNAR_Rand,
SimLowerCI_CC15MNAR_Rand, RANDHIE_CC_betas[, 3])
    CCRand4 <- CICoveragePercentageSingleCombination(SimUpperCI_CC30MCAR_Rand,
SimLowerCI_CC15MCAR_Rand, RANDHIE_CC_betas[, 4])
    CCRand5 <- CICoveragePercentageSingleCombination(SimUpperCI_CC30MAR_Rand,
SimLowerCI_CC15MAR_Rand, RANDHIE_CC_betas[, 5])
    CCRand6 <- CICoveragePercentageSingleCombination(SimUpperCI_CC30MNAR_Rand,
SimLowerCI_CC15MNAR_Rand, RANDHIE_CC_betas[, 6])
    CCRand7 <- CICoveragePercentageSingleCombination(SimUpperCI_CC50MCAR_Rand,
SimLowerCI_CC50MCAR_Rand, RANDHIE_CC_betas[, 7])
    CCRand8 <- CICoveragePercentageSingleCombination(SimUpperCI_CC50MAR_Rand,
SimLowerCI_CC50MAR_Rand, RANDHIE_CC_betas[, 8])
    CCRand9 <- CICoveragePercentageSingleCombination(SimUpperCI_CC50MNAR_Rand,
SimLowerCI_CC50MNAR_Rand, RANDHIE_CC_betas[, 9])

    RubinRand1 <-
CICoveragePercentageSingleCombination(SimUpperCI_Rubin15MCAR_Rand,
SimLowerCI_Rubin15MCAR_Rand, RANDHIE_Rubin_betas[, 1])
```

```
    RubinRand2 <-
CICoveragePercentageSingleCombination(SimUpperCI_Rubin15MAR_Rand,
SimLowerCI_Rubin15MAR_Rand, RANDHIE_Rubin_betas[, 2])
    RubinRand3 <-
CICoveragePercentageSingleCombination(SimUpperCI_Rubin15MNAR_Rand,
SimLowerCI_Rubin15MNAR_Rand, RANDHIE_Rubin_betas[, 3])
    RubinRand4 <-
CICoveragePercentageSingleCombination(SimUpperCI_Rubin30MCAR_Rand,
SimLowerCI_Rubin15MCAR_Rand, RANDHIE_Rubin_betas[, 4])
    RubinRand5 <-
CICoveragePercentageSingleCombination(SimUpperCI_Rubin30MAR_Rand,
SimLowerCI_Rubin15MAR_Rand, RANDHIE_Rubin_betas[, 5])
    RubinRand6 <-
CICoveragePercentageSingleCombination(SimUpperCI_Rubin30MNAR_Rand,
SimLowerCI_Rubin15MNAR_Rand, RANDHIE_Rubin_betas[, 6])
    RubinRand7 <-
CICoveragePercentageSingleCombination(SimUpperCI_Rubin50MCAR_Rand,
SimLowerCI_Rubin50MCAR_Rand, RANDHIE_Rubin_betas[, 7])
    RubinRand8 <-
CICoveragePercentageSingleCombination(SimUpperCI_Rubin50MAR_Rand,
SimLowerCI_Rubin50MAR_Rand, RANDHIE_Rubin_betas[, 8])
    RubinRand9 <-
CICoveragePercentageSingleCombination(SimUpperCI_Rubin50MNAR_Rand,
SimLowerCI_Rubin50MNAR_Rand, RANDHIE_Rubin_betas[, 9])

    GalimardRand1 <-
CICoveragePercentageSingleCombination(SimUpperCI_Galimard15MCAR_Rand,
SimLowerCI_Galimard15MCAR_Rand, RANDHIE_Galimard_betas[, 1])
    GalimardRand2 <-
CICoveragePercentageSingleCombination(SimUpperCI_Galimard15MAR_Rand,
SimLowerCI_Galimard15MAR_Rand, RANDHIE_Galimard_betas[, 2])
    GalimardRand3 <-
CICoveragePercentageSingleCombination(SimUpperCI_Galimard15MNAR_Rand,
SimLowerCI_Galimard15MNAR_Rand, RANDHIE_Galimard_betas[, 3])
    GalimardRand4 <-
CICoveragePercentageSingleCombination(SimUpperCI_Galimard30MCAR_Rand,
SimLowerCI_Galimard15MCAR_Rand, RANDHIE_Galimard_betas[, 4])
    GalimardRand5 <-
CICoveragePercentageSingleCombination(SimUpperCI_Galimard30MAR_Rand,
SimLowerCI_Galimard15MAR_Rand, RANDHIE_Galimard_betas[, 5])
    GalimardRand6 <-
CICoveragePercentageSingleCombination(SimUpperCI_Galimard30MNAR_Rand,
SimLowerCI_Galimard15MNAR_Rand, RANDHIE_Galimard_betas[, 6])
    GalimardRand7 <-
CICoveragePercentageSingleCombination(SimUpperCI_Galimard50MCAR_Rand,
SimLowerCI_Galimard50MCAR_Rand, RANDHIE_Galimard_betas[, 7])
```

```
    GalimardRand8 <-
CICoveragePercentageSingleCombination(SimUpperCI_Galimard50MAR_Rand,
SimLowerCI_Galimard50MAR_Rand, RANDHIE_Galimard_betas[, 8])
    GalimardRand9 <-
CICoveragePercentageSingleCombination(SimUpperCI_Galimard50MNAR_Rand,
SimLowerCI_Galimard50MNAR_Rand, RANDHIE_Galimard_betas[, 9])

RandCCRandCoveragePercentage <- cbind(unlist(CCRand1), unlist(CCRand2),
unlist(CCRand3), unlist(CCRand4), unlist(CCRand5), unlist(CCRand6),
unlist(CCRand1), unlist(CCRand1), unlist(CCRand9))

RandCCRandCoveragePercentage <- RandCCRandCoveragePercentage*100

RandRubinRandCoveragePercentage <- cbind(unlist(RubinRand1), unlist(RubinRand2),
unlist(RubinRand3), unlist(RubinRand4), unlist(RubinRand5), unlist(RubinRand6),
unlist(RubinRand7), unlist(RubinRand8), unlist(RubinRand9))

RandRubinRandCoveragePercentage <- RandRubinRandCoveragePercentage*100

    RandGalimardRandCoveragePercentage <- cbind(unlist(GalimardRand1),
unlist(GalimardRand2), unlist(GalimardRand3),
                            unlist(GalimardRand4), unlist(GalimardRand5),
unlist(GalimardRand6),
                            unlist(GalimardRand7), unlist(GalimardRand8),
unlist(GalimardRand9))

    RandGalimardRandCoveragePercentage <- RandGalimardRandCoveragePercentage*100

    write.csv(RandCCRandCoveragePercentage,
"E:/MatricesOf1000Coef&SD/RandCCRandCoveragePercentage.csv")
    write.csv(RandRubinRandCoveragePercentage,
"E:/MatricesOf1000Coef&SD/RandRubinRandCoveragePercentage.csv")
    write.csv(RandGalimardRandCoveragePercentage,
"E:/MatricesOf1000Coef&SD/RandGalimardRandCoveragePercentage.csv
```