APPENDIX A

COVER SHEET TO BE APPENDED TO ALL THESES

By law, copyright in your thesis belongs to you, the author, including all rights of publication and reproduction. Only the copyright holder may determine what others may do with the thesis.

STEP 1. What should the library do with your thesis? (Choose one)

☒ Share my thesis
You authorize the library to share your thesis with others according to the Creative Commons license selected in step 2. (See Appendix B for an explanation of the three options.)

□ Place an optional embargo on my thesis
The library should not allow anybody except college officials to see its copy of my thesis until January 1st of the year ___________. (e.g., 1 year=2018, 5 years=2022, 100 years=2117)

Under this option, you ask the library to prevent anyone else from accessing its copy of your thesis. At the end of the lockdown period, the library will distribute its copy of your thesis according to the license you choose below. (See Appendix B for an explanation of the three options.)

STEP 2. Select a license for your thesis. (Choose one)

□ CC BY-NC-ND

□ CC BY-NC-SA

☒ CC BY

The type of Creative Commons license you choose determines what others may and may not do with your thesis. (See Appendix B for an explanation of the three options.)

STEP 3. Sign.

Please note that, by choosing to make your thesis available (whether now or at a future date), you are waiving any protections of the Family Educational Rights and Privacy Act (FERPA) that may apply with respect to your thesis as of the date specified. FERPA generally restricts disclosure by the college of records related to your education.

	Caleb Ki	4/27/17
Student Signature	Student Name	Date
	Nicholas Horton	4/27/17
Thesis Advisor Signature	Thesis Advisor Name	Date

Missing Data in Randomized Clinical Trials

Caleb Ki

April 26th, 2017

Submitted to the
Department of Mathematics and Statistics
of Amherst College
in Partial Fulfillment of the Requirements
for the Degree of
Bachelor of Arts with Honors

Faculty Advisor: Professor Nicholas J. Horton

Acknowledgements

First, I would like to express my gratitude to my thesis advisor, Nick Horton. I could not have written this thesis without his knowledge and passion for the study of statistics in medicine, his thoughtfulness, and his guidance. Nick has also helped me immensely with my graduate school applications, and for that I am very thankful. I would like to thank Professor Amanda Folsom who I did research under the summer after my Sophomore year. She has invested a significant amount of time and resources into my growth as a student and a researcher. Much of my success is owed to her. I would also like to show my appreciation for my former professor, Professor Susan Wang, who has always believed in my abilities even when I have doubted myself. It is hers and Nick's mentorship that led me to write a thesis and pursue graduate studies in statistics. I would also like to extend my thanks to all the professors in the math and stats department. Your skills in and zeal for teaching have not gone unnoticed. You all have contributed so much to my development here at Amherst. I would like to thank all of my friends. Thank you for asking me how I am doing, making me laugh, and picking me up. In particular, I would like to thank David Zhang for our shared experience in writing a thesis and Elaine Jeon for her patience and companionship. Last but certainly not least, I would like to thank my family: my sister, Michelle, for daring me to dream and my parents, Junghee and Ryun, for their relentless encouragement. Thank you all for your support and your love.

Table of Contents

List of Tables

List of Figures

Abstract

Today's health professionals are more evidence-based medicine oriented than ever. The medical world relies on clinical trials to provide knowledge in patient care. However, missing data has and continues to compromise inferences from clinical trials. In this thesis, I provide a background on clinical trials and the missing data problem. Then I review a few methods of handling missing data with an emphasis on multiple imputation. Throughout this thesis, I use an alcohol intervention clinical study as a motivating example; I illustrate the faults of complete case analysis and single imputation methods using data from the study. Following, I proceed to discuss sensitivity analysis in the context of missing data in clinical trials. I examine the delta adjustment procedure and the mean score method, two strategies for sensitivity analysis. Then I provide a motivating example using the same alcohol intervention study data as before and another using an artificial dataset. Finally, I present a simulation study uncovering several properties of multiple imputation under four different scenarios. The distinction in the scenarios are based on the type of outcome variable as well as whether the auxiliary variable has a relationship with the outcome variable. The exposition and simulation results show that the missing data problem in clinical trials cannot be completely solved. However, it is still important to direct further attention to multiple imputation and sensitivity analysis as they can improve inferences.

Chapter 1

Introduction

In statistics, missing data refers to when an observation within a dataset is missing a value for one or more of the variables in the dataset. In the context of clinical trials, missing data may occur for several reasons including patients withdrawing from the study, missing appointments, or failing to fill out the questionnaire properly. Simply put, missing data is unavoidable in real-world settings. For example, LaVange & Permutt (2016) reported that attrition in clinical trials for different drugs and diseases ranges from under 10% for cystic fibrosis to up to over 30% for schizophrenia. In their review of missing data in clinical trials in four top-tier medical journals, Bell, Fiero, Horton, & Hsu (2014) found that 73 (95%) of 77 trials reported missing outcome data.

Even though missing data is common, it is certainly not benign. In their paper, Barnard & Meng (1999) laid out the three main problems caused by missing data. First, missing data causes loss of information, efficiency, and power. Second, it increases the difficulty in handling, computing, and analyzing the data due to the irregularities found in the data and the inappropriateness of standard methods in that situation. Last, missing data will introduce bias in the analysis. Both in and out of the setting of clinical trials, large bias can and will lead to incorrect inferences being made. In turn, this could lead to detrimental consequences. As an example, imagine a drug with a harmful side effect is put on the market because during testing, all patients who felt the side effect dropped out, so the researchers were unable to collect the relevant data. When the drug reaches the market, it will inevitably have to be recalled because of its unintended side effect. The bias due to the missing data not only cost the drug manufacturer money, but more importantly, put people in danger. While the other two consequences of missing data complicate the process of data analysis, they are generally not as impactful as biased analyses. Thus, bias introduced by missing data is the most important of the three problems to address.

In randomized clinical trials (RCTs), there are two main types of analysis that are generally performed. When only the patients that successfully completed the trial are analyzed, the analysis is known as a per-protocol analysis. When per-protocol analysis is used in isolation, bias will be introduced in the analysis unless the data are missing independent of both observed and unobserved variables. It implies that there is no systematic difference between those who stay in the study and those who leave. It is a strong and often unrealistic assumption to make (White, Carpenter, Evans, & Schroter, 2007).

On the other hand, intention-to-treat (ITT) analysis is generally regarded as the most appropriate way to estimate efficacy of drugs and treatments (White, Carpenter, & Horton, 2012). In ITT analysis, patients are kept within the treatment group based on the randomization scheme regardless of what treatment the patient eventually received. Data are measured for all patients, and all patients are included in the analysis. Considering what was posited earlier about the pervasiveness of missing data, it is clear that ITT analysis cannot always be performed under normal circumstances.

To rectify the problem, assumptions on the missing data are made in order to perform ITT analysis. Since generally it is inherently impossible to verify the underlying relationship between the data and its missingness, it is important to understand what happens to the analysis if we deviate from the assumptions. Sensitivity analysis is a type of study which attempts to evaluate what occurs when assumptions are not necessarily satisfied.

The goal of this thesis is to provide an exposition of a variety of topics concerning missing data in clinical trials including multiple imputation and different methods of sensitivity analysis. In particular, multiple imputation via Bayesian inference, bootstrap, and predictive mean matching; the delta adjustment procedure; and the mean score method are discussed. Following the exposition, a simulation study, which uncovers a range of properties of multiple imputation under a medley of settings, is presented. Missingness was created in a way such that the true values were known, allowing any model fitted using a missing data method to be compared to models fit using the true data. To complement this simulation study, I have created a package in `R` called `rctmiss` providing a simulation environment that allows anyone to replicate my results and explore properties of multiple imputation. In addition, motivation for the mean score method and the delta adjustment procedure is given via artificial data and data from a clinical trial, the Kypri alcohol electronic brief intervention trial.

Chapter 2

Background

2.1 Randomized clinical trials

In the context of the medical world, interventions simply refer to procedures, medicines, or applications intended to ameliorate injuries, conditions, and diseases. An intervention could be a drug, a device, a type of surgery, or a biologic. While there is no universally agreed upon definition for biologics, the Food and Drug Administration (2016) defines them as substances derived from a natural source, made up of sugars, proteins, nucleic acids, cells, or tissues (or a combination of those components).

Generally, clinical researchers determine and vet the safety and effect of interventions through clinical trials. The National Institute of Health (2017) defines clinical trials as research experiments designed to examine how well interventions perform on humans. Each clinical trial has a corresponding document called a protocol which outlines the objective, design, methodology, statistical considerations, and organization of the clinical trial. This protocol not only provides clear direction on how the trial will be conducted, but it also keeps patients safer and ensures that the data collected are valid. An important function of the protocol is determining how the subjects within the trial are divided into different subgroups called treatment arms which receive different interventions (or a control).

In order to accurately evaluate an intervention, it is imperative that any observed differences in the outcomes between the treatment and control groups are due to the difference in the interventions received rather than any other difference between the two groups, observed or unobserved. Otherwise, incorrect conclusions concerning interventions can and will be drawn. Randomization is the most reliable way to ensure that the difference in the outcomes between the treatment arms is due to the differences in the interventions received. On average, randomization will balance out any known or unknown characteristics of trial patients between the treatment and control groups (CNStat, 2010). Without randomization, there is a large risk of selection or confounding bias. Randomization is the mechanism that allows clinical researchers to be able to draw inferences about medical interventions. Clinical trials that employ randomization to allocate their patients into different treatment groups are simply referred to as randomized clinical trials (or RCTs for short).

The three main types of clinical trials are superiority, equivalence, and non-inferiority trials. In a superiority clinical trial, the objective is to demonstrate that the intervention of interest is superior to a control treatment. There are no statistical tools that could prove that two treatments are equal in their capabilities (Lesaffre, 2008). Instead, treatments can only be shown to be equivalent. An equivalence trial's objective is to show that two treatments are not too different. Lastly, a non-inferiority trial attempts to establish that a new treatment is not inferior to the control treatment, in which the control is a proven standard treatment (otherwise known as an active control as opposed to a placebo control). For the rest of this thesis, the focus will be placed on superiority trials.

2.1.1 Estimands

Every clinical trial has an objective which is explicitly stated in its protocol. The objective could be to test whether a new procedure is effective at combating cancer or whether a new drug reduces acne as successfully as a drug already on the market, but with fewer side effects. In order to accomplish the objective of the trial, researchers must collect data on the trial outcomes to support their conclusions. Outcomes refer to quantitative measures of how a patient feels, functions, or survives while estimands are parameters that summarize the outcomes for the target population of the clinical trial. The NRC report describes estimands as a typical measure by which one can assess the effect of an intervention. As an example, consider a trial where a drug is being tested for its ability to reduce low-density lipoprotein (LDL) cholesterol over a period of three months. Only considering the case where there is just a control and a treatment group, the estimand of interest is the difference in the mean change of LDL cholesterol level from the start of the trial to three months later for the two populations (those who take the intervention and those who do not). The corresponding outcome measured for each patient in the trial would be the change in LDL cholesterol level from the start of the trial to the end of the intervention period three months later, and an estimate of the estimand would be the difference in the means of the outcomes for the two different samples in the study over the course of the trial.

When a patient deviates from their assigned treatment by stopping, switching, or adding another treatment, the patient is said to be nonadherent. The inevitability of nonadherence in trials necessitates that trial designers make an important protocol decision: Should the treatment effect be measured for only those who fully adhered to the protocol or should the treatment effect be measured for everyone regardless of adherence to the protocol? The first approach is referred to as efficacy. In a clinical trial where the objective is to determine the efficacy of the drug, the trial estimates the treatment effect of the drug in ideal circumstances. The second approach is referred to as effectiveness, and in a clinical trial where the objective is to determine the effectiveness of the drug, the trial estimates the treatment effect of the drug in practice. Predictably, the estimands in the diverging perspectives are known as efficacy and effectiveness estimands. The efficacy estimand does not account for tolerability or ease of use, whereas the effectiveness estimand encompasses those issues. These estimands are also known as *de jure* and *de facto* estimands, and based on the definitions given by M. Kenward (2013), the *de jure* estimand and *de facto* estimand are synonymous with the efficacy estimand and the effectiveness estimand, respectively.

2.1.2 Analysis in clinical trials

In clinical trials, the results of a trial are usually analyzed in one of two ways: intention-to-treat (ITT) or per-protocol. An intention-to-treat analysis sorts all trial participants into the treatment group into which they were randomized regardless of whether or not they received the designated treatment, whereas a per-protocol analysis only includes patients who have adhered to protocol. As stated in the introduction, of the two, ITT is the most appropriate method of analysis. On the other hand, in per-protocol analysis, post-randomization exclusions have to be made which often create bias and delegitimize the analysis. Thus, according the NRC report, in order to determine the success of an intervention in practice, RCTs should use ITT analysis. However, this is not to say that a per-protocol analysis is always inappropriate. Usually, an intervention will have to go through a preliminary clinical trial with a per-protocol analysis in order to establish evidence of a treatment effect before the intervention is taken to a clinical trial using ITT analysis (Food and Drug Administration, 2015).

Clearly, the types of analysis and estimands in clinical trials are related. When the method of analysis is ITT, the estimand of concern is the effectiveness (*de facto*) estimand, and when the method of analysis is per-protocol, the estimand of concern is the efficacy (*de jure*) estimand.

2.1.3 Alcohol intervention study

Throughout this chapter, data from a two-arm randomized clinical trial will be used repeatedly for different examples. The trial was designed to determine the effectiveness of a web-based alcohol screening and brief intervention program (Kypri et al., 2014).

Description

The trial participants came from seven different New Zealand universities. The trial design was double-blind, parallel-group, and individually randomized. To determine eligibility for the trial, the Alcohol Use Disorders Identification Test (AUDIT) was used. AUDIT was a product of a World Health Organization (collaborative) project between six countries (Saunders, Aasland, Babor, Fuente, & Grant, 1993). A 10-question questionnaire, AUDIT is designed to identify individuals who may have alcohol problems. AUDIT-C is a three-question test adapted from AUDIT. The questions are listed below:

1. How often do you have a drink containing alcohol?
   a. Never
   b. Monthly or less
   c. Two to four times a month
   d. Two to three times a week
   e. Four or more times a week
2. How many standard drinks containing alcohol do you have on a typical day?
   a. 0 to 2
   b. 3 or 4
   c. 5 or 6
   d. 7 to 9
   e. 10 or more
3. How often do you have six or more drinks on one occasion?
   a. Never
   b. Less than monthly
   c. Monthly

d. Weekly
e. Daily or almost daily

The scores of the AUDIT-C range from zero to 12 (individual questions can have scores between zero and four). For each question, answer choice *a* corresponds to a score of zero, answer choice *b* corresponds to a score of one, answer choice *c* corresponds to a score of two, answer choice *d* corresponds to a score of three, and answer choice *e* corresponds to a score of four.

In total, 14,991 students were asked to participate in the trial. Only 5,135 were screened for eligibility. To be eligible, the student had to have an AUDIT-C score ≥ 4. There were 3,422 students who screened positively and were randomized into the treatment arms. In addition to being asked the three questions from the AUDIT-C test, those in the intervention group were also asked to answer AUDIT questions 4 to 10. After completing the entire AUDIT questionnaire, participants in the intervention group answered the Leeds Dependence Questionnaire (also comprised of 10 questions). Based on their answers to the two tests, the participants were given personalized feedback concerning their drinking habits. This included explanations of the associated health risk of their drinking, how they might go about reducing that risk, information concerning behavior at different blood alcohol levels, estimates of their monthly expenditure on alcohol, comparisons of their drinking behavior to their peers, and links directing them to help with drinking.

All participants were asked to fill out a follow-up questionnaire five months after the initial screening. While there were six outcome measures, here we only consider the frequency of drinking (0-28 days) and the number of standard drinks (10g ethanol) per typical occasion. Other data collected during the trial include the age and the weight of the participant. A total of 269 students out of 1,706 in the treatment group did not complete the follow-up, while 303 students out of 1,716 in the control did not complete the follow-up (Kypri et al., 2014). Since the reasons for why they did not complete their follow-up are unknown, the missing data mechanism is also unknown.

2.2 Missing data mechanisms

In this section, I provide an overview of the nomenclature for missing data. Rubin (1976) introduced the now commonly used terminology concerning missing data mechanisms which are the relationships between the missingness and the values of the data. He created three different categories under which all missing data mechanisms fall. These are known as missing completely at random (MCAR), missing at random

(MAR), and missing not at random (MNAR or NMAR). When data are MCAR, the actual value of the data has no bearing on the probability that an observation is missing. This is to say that each observation is equally likely to be missing. When data are MAR, the probability of being missing is dependent only on the observed values. Lastly, MNAR means that the missing data mechanism is neither MCAR or MAR. This means that the probability of being missing is dependent on the missing values.

Consider a trial where there are p variables of interest and n observational units. Let Y indicate the matrix of values for the p different variables for the n different units. Then y_{ij} would indicate the value of the jth variable of the ith unit where $1 \leq j \leq p$ and $1 \leq i \leq n$. As explained in the introduction, missing data are inescapable in almost any clinical trial, so the matrix Y will almost inevitably contain missing entries. Let R be another matrix of the same size as Y where $r_{ij} = 0$ if y_{ij} is missing, and $r_{ij} = 1$ if y_{ij} is present. Let Y_{obs} indicate all the observed values of Y (i.e., the values y_{ij} of Y where the corresponding values of R, r_{ij}, equal 1) and Y_{mis} indicate all the missing values of Y (i.e., the values y_{ij} of Y where the corresponding values of R, r_{ij}, equal 0). Then Y can be split into its missing and observed parts so that $Y = (Y_{obs}, Y_{mis})$.

Returning to the definitions for the different missing data mechanisms, letting ψ be the parameters of the missing data model of R, then the missing data mechanism is MCAR if

$$P(R|Y_{obs}, Y_{mis}, \psi) = P(R|\psi).$$

MCAR is a very strong assumption on the data. Consider if every patient in an RCT rolled a die on the day of the followup, and if they rolled a six they decided to not show up to the follow-up. Although this scenario is very contrived, the missing data mechanism is MCAR because the roll of the die is independent of Y_{obs} and Y_{mis}.

The missing data mechanism is MAR if

$$P(R|Y_{obs}, Y_{mis}, \psi) = P(R|Y_{obs}, \psi).$$

Consider the example given in the estimands sections about the drug being tested for its capacity to reduce LDL cholesterol. Imagine if as part of the procedure, the height of every trial participant was taken, and that later, when it was time for the follow-up the taller people refused to go. Since their refusal was unrelated to their levels of LDL cholesterol, this is an example of the MAR missing data mechanism. Most methods of handling missing data (besides the very simple methods) start with the assumption that the missing data mechanism is MAR.

Lastly, the missing data mechanism is MNAR if $P(R|Y_{obs}, Y_{mis}, \psi)$ cannot be reduced. Returning to the LDL cholesterol example, the missing data mechanism would be MNAR if several trial participants decided to not go to the follow-up because the drug was not reducing their LDL cholesterol levels.

One important thing to note is that it is impossible to discern whether a missing data mechanism is MAR or MNAR without auxiliary data. This is because the values necessary to determine whether a missing data mechanism is MAR or MNAR are missing. Furthermore, it is much more difficult to model data when the missing data mechanism is MNAR which is why most methods of handling missing data start with the MAR assumption.

2.2.1 Ignorability

In this section, I introduce the notion of ignorability and nonignorability in missing data. Ignorability is a slightly stronger assumption than MAR. Under the ignorability condition/assumption, Rubin (1976) showed that any analysis performed on a set of missing data can ignore the underlying procedure by which the missing data is caused. This means that a legitimate analysis of the data does not require an explicit formulation of the missing data model.

Let θ be the set of parameters to the full data model Y. Generally, researchers are interested in drawing inference about θ, whereas ψ has no intrinsic scientific value. For a missing data mechanism to be ignorable, the data must be MAR and the parameters θ and ψ must be distinct. This means that the joint parameter space of ψ and θ is the Cartesian product of the parameter space of θ and the parameter space of ψ. Under Bayesian inference, there is an added condition: the prior distributions of θ and ψ must be independent (i.e., $p(\theta, \psi) = p(\theta)p(\psi)$).

When the missing data mechanism is ignorable, it is assumed that we can sufficiently impute the missing values from the nonmissing data. Furthermore, it means we can draw inferences on the parameter of the full data model θ without knowing ψ. When missing data mechanism is nonignorable, this is no longer the case.

2.3 Simple solutions

2.3.1 Complete case analysis

Complete case analysis, also known as listwise deletion, is a method of handling missing data when values missing for one or more variables are removed from the analysis. Complete case analysis is a quick and easy solution. In fact, in many statistical packages such as `R`, `SAS`, `SPSS`, and `Stata`, complete case analysis is the default method of handling missing data (van Buuren, 2012).

There is extensive literature denouncing the use of complete case analysis, but under certain circumstances, it may be a viable option. According to Schafer & Graham (2002), if only a small proportion of the sample is missing, complete case analysis can be effective. When the missing data mechanism is MCAR, complete case analysis will provide unbiased estimates. However, unless the amount of missing data is very low, it is best to avoid complete case analysis if possible.

There are two major disadvantages when using complete case analysis. The first is analyzing data under complete case could lead to losing a large portion of the observations. It is not uncommon to have a substantial proportion of observations with a missing value. This is especially true when there is a large number of variables in the dataset. In this respect, complete case analysis is both wasteful and inefficient. The second problem with using complete case analysis is that if the missing data mechanism is not MCAR, complete case analysis can significantly bias estimates of any estimand.

Another potential problem is that complete case analysis may cause erratic reporting of the data. If multiple complete case analyses are performed on the same set of data, it is likely that each subsample created through listwise deletion will be of different size. While this problem is minor relative to the other two, it still makes comparison of the analyses of the sample to the whole population more complicated.

Tables 2.1 and 2.2 exemplify how complete case could lead to biased results. Even though the alcohol intervention data are not complete, for the purpose of this example, the available data will be treated as the complete data. Moreover, while a negative binomial model was used in the actual analysis of the alcohol intervention trial, for the sake of simplicity, linear models were fit for this example. One model was fit with all available data and another was fit with data where it was more likely that values were missing for students aged 20 and over. Both models explore how the age and intervention affected the drinking frequency at the time of the follow-up.

	Estimate	Std. Error
(Intercept)	0.14	1.08
intervention_group1	-0.16	0.18
age	0.30	0.05

Table 2.1: Alcohol Intervention Trial: Model fit with all data

When the model is fit using all available data, the estimate for the coefficient of `age` is 0.30, so for a unit increase in `age` we would expect the frequency of drinking to increase by 0.30 days. In addition, the coefficient for `intervention_group1` is −0.16. This indicates that on average, if the person received the intervention we would expect their frequency of drinking to be 0.16 days lower than if they received in the control.

	Estimate	Std. Error
(Intercept)	-0.80	1.23
intervention_group1	0.03	0.21
age	0.34	0.06

Table 2.2: Alcohol Intervention Trial: Model fit using complete case analysis

For the model summarized in Table 2.2, missingness was created by simply setting around 40% of the values of the drinking frequency variable for those aged 20 and over to `NA`. In this case, the missing data mechanism is MAR as the missingness of drinking frequency is related to an observed variable, age. Using only the complete cases to create the model, the coefficients for the intervention and age are 0.03 and 0.34, respectively. Comparing with the results fit using all the data, complete case analysis overestimates the effect of age on drinking frequency at the time of follow-up and severely overestimates the effect of the intervention on the drinking frequency at the time of follow-up (so much so that the coefficient switched signs). Further, the standard errors of the coefficients of the intercept, intervention, and age are 14.24%, 14.37%, and 16.32% larger for the model using complete case than the model using all available data. The larger standard errors are indicative of the loss of efficiency due to the smaller sample size when using complete case analysis.

2.3.2 Mean imputation

Under mean imputation, missing values are replaced with the mean of the observed values. Like complete case, the advantage of mean imputation is that it is easily and quickly implemented. An added advantage it has over complete case is that all observations are retained. However, mean imputation has several flaws. First, it will always underestimate the variance of the imputed variable, and the more missing values there are, the more the variance will be underestimated. Second, it alters the relationship between the variables. Last, mean imputation leads to biased estimates of any estimate except the mean itself when the missing data mechanism is MCAR (Little, 1992).

In the previous example, when the missing data mechanism was MAR, complete case analysis produced biased estimates of the coefficients. In the following example, the same data are used to demonstrate the inaptitude of mean imputation and to show that complete case analysis can be effective when the data are MCAR. A model exploring the relationship of AUDIT-C score and the intervention with the quantity of drinks during a typical drinking occasion was fit with all available data. After introducing missingness, models were fit with complete case analysis and mean imputation. Around 40% of all the values of typical drinking occasion were randomly set to `NA`.

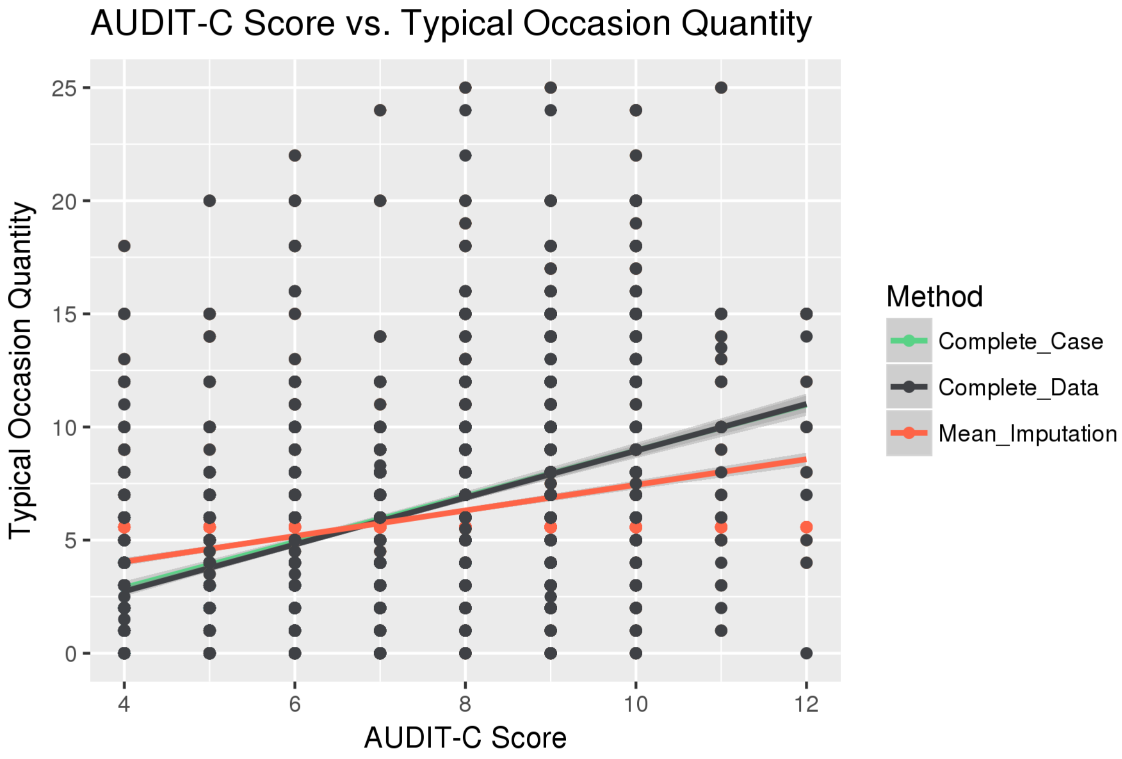

Figure 2.1: Mean Imputation with data from alcohol intervention trial

The scatterplot in Figure 2.1 displays the relationship between the typical occasion quantity and baseline AUDIT-C score. The black line, which is a linear model of the two variables using the complete data, almost entirely overlaps the green line, which represents the linear model fit using complete case analysis, indicating that complete case analysis provides an unbiased estimate of the relationship between the two variables. On the other hand, the orange line, which is the linear model fit using mean imputation, has a larger intercept and smaller slope. It is clear that the orange dots, which represent the mean imputed values, do not follow the overall pattern of the data. Using mean imputation clearly distorts the relationship between baseline AUDIT-C score and the typical occasion quantity. To quantify how biased the coefficient estimates are when using mean imputation, three models were fit with AUDIT-C score and intervention as predictors and typical occasion quantity as outcome. The coefficients of these models are summarized in tables 2.3, 2.4, and 2.5.

	Estimate	Std. Error
(Intercept)	-1.14	0.26
intervention_group1	-0.48	0.15
auditCscore	1.04	0.04

Table 2.3: Alcohol Intervention Trial: Model fit with complete data

	Estimate	Std. Error
(Intercept)	-0.88	0.35
intervention_group1	-0.42	0.20
auditCscore	1.02	0.05

Table 2.4: Alcohol Intervention Trial: Model fit with complete case

	Estimate	Std. Error
(Intercept)	1.93	0.21
intervention_group1	-0.26	0.12
auditCscore	0.57	0.03

Table 2.5: Alcohol Intervention Trial: Model fit with mean imputation

While the estimate of the coefficients of intervention and AUDIT-C score differ only by 0.06 between the complete data and complete case models, the estimates differ by 0.22 between the complete data and mean imputation methods. This example illustrates that when missing data are present, complete case could be used when the missing data mechanism is MCAR with minimal bias. However, the same can not be said of mean imputation. Even when the missing data mechanism is MCAR, mean imputation drastically biases the results. Mean imputation should generally be avoided. All that being said, there is one case where using mean imputation may be appropriate. If baseline predictors are missing values, then mean imputation can be a good method for handling the missingness (White & Thompson, 2004).

2.3.3 Missing indicator method

Another popular method of handling missing data is the missing indicator method. In this method, if one would like to run a regression, for an explanatory variable with missing values, all missing values are set to zero and the model is fit with an extra variable indicating whether there was a response or not. So if only one explanatory variable is missing, only one extra indicator variable is added, but if five explanatory variables are missing, then five extra indicator variables must be added. In the case of RCTs, the missing indicator method follows the ITT principle and can have unbiased estimates regardless of the underlying missing data mechanism (Groenwold et al., 2012). However, the conditions under which the missing indicator method is unbiased are difficult to find in practice (van Buuren, 2012). Specifically, no two predictors can be related (i.e., the covariance between the predictors is 0). Further, if the outcome variable is missing any values, then the missing indicator method should not be used. Because of its inflexibility, the missing indicator method cannot be used as an all-purpose method to handle missing data.

2.4 Multiple imputation

One flexible approach to handle missing data is multiple imputation (MI). This method was first introduced by Rubin in 1977 to address an issue noted by statisticians at the Social Security Administration and the U.S. Census Bureau. Single imputation was the primary means with which the U.S. Census Bureau handled nonresponse in the Current Population Survey.

Using single imputation, variance cannot be calculated correctly, so Scheuren, a statististician working with the Census Bureau, consulted Rubin, who in turn wrote a paper about multiple imputation to provide a solution (Scheuren, 2004). With single imputation, the imputed value was taken to be the truth. The uncertainty in the imputed values was not accounted for in any analysis with single imputation and the variability was always understated. Rubin realized that that no matter how a value was imputed, no single imputation procedure could always correctly pick the true value, and developed multiple imputation to account for this uncertainty (van Buuren, 2012). Rubin's paper originally written in 1977, in which he describes the general approach of multiple imputation, was republished in a 2004 article (Rubin, 2004).

In his book about multiple imputation, van Buuren (2012) outlines the history of multiple imputation. In the 1980s, near the inception of multiple imputation, there were concerns about the extra work needed to create the models and analysis especially for larger datasets as well as implementation in software. These issues have been substantially addressed, and today regardless of the size of the dataset, there are several software packages in which implementation of multiple imputation is possible (including but not limited to `R`, `SAS`, and `SPSS`) (Horton & Kleinman, 2007). In the 1990s and the early 2000s, there were several critics of multiple imputation (discussed in detail later). However, the perception of multiple imputation took a turn in the mid to late 2000s. Now multiple imputation is regarded as a powerful method to handle missing data and is widely used in practice. In fact, research regarding multiple imputation has been exponentially increasing over time (van Buuren, 2012).

One important thing to note about multiple imputation is that it generally starts with the assumption that data are MAR. In particular, it starts with the stronger assumption that the missing data mechanism is ignorable. However, there have been developments that allow multiple imputation even when the missing data mechanism is MNAR.

Multiple imputation consists of three steps:

1. Fill in missing values N times by taking N random draws from a conditional distribution of the missing data given the observed data.

2. Analyze the N completed datasets using standard methods.

3. Combine the N analyses using Rubin's rules for a single inference.

In the past, the recommended number of imputed datasets N was between 2 and 10 (Rubin, 1987). Recently, the number of imputations recommended has been updated to be between 20 and 100 (Graham, Olchowski, & Gilreath, 2007). The model used in step one to impute the values is referred to as the imputation model. The model fitted to each completed dataset is referred to as the substantive model. One critic of multiple imputation, Fay (1992), claimed to have produced examples that showed that multiple imputation overestimated variance and underestimated covariance. Fay went even as far as to say that multiple imputation was not suited for analysis of complex problems or large datasets. Meng (1994) addressed this issue, stating that the results found by Fay were due to uncongeniality between the imputation and substantive models. When an imputation and an analysis model are congenial, it means that covariates used in the analysis model should be found in the imputation model. This implies that imputation models should be very general even to the point of saturation. A reasonable concern with this strategy is that it may lead to several predictors with little to no association with the missingness of Y being included in the imputation model. In the simulation study run by Collins, Schafer, & Kam (2001), they found that including these unimportant variables had at worst a neutral effect and sometimes had an advantageous effect. Moreover, Meng stresses that imputation models should include covariates that are likely to be used in the analysis stage even if the covariates may not be good predictors for the missing observations.

The set of Rubin's rules referenced in step 3 as outlined by Carpenter & Kenward (2014) is laid out below.

Let β denote the scalar parameter of interest, and let σ^2 be its associated variance. Suppose we are imputing N times. We reference each of the N imputed datasets with a corresponding number $n \in 1, 2, ..., N$. Let $\hat{\beta_n}$ and $\hat{\sigma_n^2}$ denote the estimates of β and σ^2 by fitting the substantive model on the imputed dataset n. The multiple imputation estimator of β is simply the arithmetic mean of the $\hat{\beta}$'s,

$$\widehat{\beta_{MI}} = \frac{1}{N} \sum_{n=1}^{N} \hat{\beta}_n.$$

In order to calculate the variance of the estimates of β, it is important to account for both the variance *within* the imputations and *between* the imputations. Let the *within* imputation covariance matrix be denoted as

$$\widehat{W} = \frac{1}{N} \sum_{n=1}^{N} \hat{\sigma_n^2},$$

and the *between* imputation covariance matrix be denoted as

$$\hat{\beta} = \frac{1}{N-1} \sum_{n=1}^{N} (\hat{\beta_n} - \widehat{\beta_{MI}})^2.$$

Then the variance estimator is given by

$$\widehat{V_{MI}} = \widehat{W} + \left(1 + \frac{1}{K}\right)\hat{\beta}.$$

A t-test is used to test the hypothesis $\beta = 0$. The T statistic can be calculated with the following equation

$$T = \frac{\widehat{\beta_{MI}} - 0}{\sqrt{\widehat{V_{MI}}}}$$

where the degrees of freedom v is given by $v = (K-1)\left[1 + \frac{\widehat{W}}{(1+1/K)\hat{\beta}}\right]^2$.

2.4.1 Drawing imputations

Multiple imputation refers to a procedure or an approach to analyze the data. Within this framework, there are actually several different ways to impute the data. When only one variable has missing values then the missing data pattern is called univariate. In the following three sections, I will discuss three different methods to impute univariate missing data. I refer to the variable with missing values as Y.

Normal linear model

A naive approach for creating multiple imputations when there is only one variable with missing values is to simply use the other n completed variables to create a regression model, $Y = \beta_0 + \beta_1 X_1 + \ ... \ + \beta_n X_n$, and utilize the predicted values from the model as the imputations. While this seems like a reasonable method, this imputation strategy suffers from the same faults of single imputation: it treats the predicted value as the truth. The advantage of using a multiple imputation approach to create analysis is that it accounts for the uncertainty of the true value. Even if the predicted value is an extremely good estimate of the missing value, it does not mean that it is equal to the true value. Furthermore, using this strategy would create identical imputed datasets, defeating the purpose of multiple imputation.

Since the coefficients of the model are estimated from the data, not only are there questions concerning the validity of the predicted value, but in the parameter estimates as well. The solution to this problem would be to use the regression approach and implement a measure to account for the uncertainty in the predicted value. Two ways to do this would be to use Bayesian or bootstrap regression models to impute the values. Both methods impute missing values of Y using the model $Y = \beta_0 + \beta_1 X_1 + \beta_1 X_1 + \ ... \ + \beta_n X_n + \epsilon$ where $\epsilon \sim N(0, \sigma^2)$. In the Bayesian case, $\beta_0, \beta_1, \ ... \ , \beta_n$, and σ are drawn from their respective posterior distribution using non-informative priors and the observed data. The bootstrap method calculates $\beta_0, \beta_1, \ ... \ , \beta_n$, and σ using the standard least squares method using a bootstrap sample of the completed data.

Predictive mean matching

Originally proposed by Rubin (1986) but developed and coined by Little (1988), predictive mean matching (PMM) is a different and flexible method of imputing values. Under predictive mean matching, using an imputation model, the predicted value of Y, $\hat{Y}$, is calculated for every subject or observation (both complete or incomplete cases). For each incomplete observation, a set of complete observations with similar predicted values is used as candidate donors. From the candidate donors, an observed value of Y is drawn and used to replace the missing value (van Buuren, 2012). The metric being used to determine similarity is called the predictive mean metric. Since the values being imputed are actual observed values, predictive mean matching ensures that realistic values are being used in the imputation process. Compared to imputations based on the normal linear model, predictive mean matching is often more robust to model misspecification.

Formally, given an observation i and j where X_i and X_j are vectors of the completed covariates of Y for each observation respectively, the predictive mean metric is defined as: $d(i,j) = (\hat{Y}_i - \hat{Y}_j)^2$ where $\hat{Y}_i = X_i\hat{\beta}$ where $\hat{\beta}$ is the vector of the coefficients from the imputation model of Y using only the data from the completed cases.

An obvious concern surrounding predictive mean matching is determining the metric used and how many candidate donors are selected for the donor pool. Andridge & Little (2010) outline three different ways to pick donors using the predictive mean metric in their review of hot deck imputation methods. One way to pick donors is select a bound γ. Then for an incomplete observation i, all completed observations j that satisfy $d(i,j) < \gamma$ are selected as candidate donors. A simpler method is to just replace the missing value with the observed value of observation j that minimizes $d(i,j)$ (known as the "nearest neighbor hot deck imputation"). The generalized version of this method would be to specify a number of donors, d, and to find the d observations with the predictive mean metrics closest to zero. For the γ method, missing observations may have different numbers of donors, whereas for the latter method, all missing observations would have the same number of donors. Siddique & Belin (2008) introduce another method where all completed observations are considered candidate donors, but the probability of selecting a donor is inversely related to the predictive mean metric.

According to van Buuren (2012), donor pools are usually of size 1, 3, or 10. A problem with using only one donor is that this will often lead to one donor being used several times (this problem also occurs with small sample sizes and when the proportion of missing data is very high). On the other hand, using a large value of d

increases the probability of bad matches, which would lead to bias. Thus, picking a value that is neither too low or high like 3 or 10 alleviates both problems.

Overall, despite the fact that it may not impute values well when the sample size is small, predictive mean matching is a great method to impute values because of its robustness and reliability. Heitjan & Little (1991) found that multiple imputation via predictive mean matching is an improvement over complete case analysis and any single imputation method in their study using data from the Fatal Accident Reporting Service (FARS). Results from a simulation study by Marshall, Altman, Royston, & Holder (2010b) and a resampling study by Marshall, Altman, & Holder (2010a) suggest that multiple imputation using predictive mean matching should be the preferred approach under MAR missingness.

Categorical data

Thus far, all methods of imputation discussed have revolved around a continuous missing variable Y. However, imputation of missing categorical variables is possible using a generalized linear model. When the missing variable of interest is binary, values can be imputed under a logistic regression imputation model where

$$P(Y = 1) = \frac{\exp\left(\beta_0 + \beta_1 X_1 + ... + \beta_n X_n\right)}{1 + \exp\left(\beta_0 + \beta_1 X_1 + ... + \beta_n X_n\right)}.$$

When the number of categories $M > 2$, missing values can be imputed using the natural extension of the logistic regression model, the multinomial logit model where,

$$P(Y = m) = \frac{\exp\left(\beta_{m0} + \beta_{m1} X_1 + ... + \beta_{mn} X_n\right)}{\sum_{i=1}^{M} \exp\left(\beta_{i0} + \beta_{i1} X_1 + ... + \beta_{in} X_n\right)}.$$

Similar to the normal linear model, Bayesian and bootstrap methods should be used to fit the logistic and multinomial models to account for the uncertainty in the predictions.

In predictive mean matching, the idea is to only use the imputation model to map a missing value to an already observed one, and thus it follows that predictive mean matching works just as well when the missing variable is continuous or categorical.

2.4.2 Multivariate missing data

In the case when the missing data pattern is multivariate, the two main multiple imputation procedures are joint modeling, introduced by Schafer (1997), and fully conditional specification (FCS), developed independently by van Buuren, Boshuiezen,

& Knook (1999) and Raghunathan, Lepkowski, Hoewyk, & Solenberger (2001).

Joint modeling

Under joint modeling, the data are assumed to follow a multivariate distribution. While any multivariate distribution suffices, it is the multivariate normal distribution that is most often used (van Buuren, 2012). Imputations can then be drawn from this fitted multivariate distribution assuming the missing data mechanism is ignorable. Joint modeling is generally a two part procedure. We want to be able to draw imputations from $f(Y_{mis}|Y_{obs})$. Assuming ignorability,

$$f(Y_{mis}|Y_{obs}) = \int f(Y_{mis}|Y_{obs}, \psi) \cdot f(\psi|Y_{obs}) d\psi,$$

where ψ is the set of parameters from the missing data model. Since ψ is challenging to estimate from Y_{obs}, the procedure alternates from drawing $\psi*$ from $f(\psi|Y_{obs})$ and $Y_{mis}*$ from $f(Y_{mis}|Y_{obs}, \psi*)$.

Fully conditional specification

In contrast, fully conditional specification is an iterative method of drawing imputations; imputations are made one variable at a time. Rather than specifying a multivariate distribution for all the variables at once, a set of conditional densities is specified. The number of iterations T is usually set beforehand. The `mice` package in R multiply imputes missing data with a FCS approach. The process goes as follows:

Given a set of p partially observed variables $Y_1, Y_2, ..., Y_p$, an imputation model $P(Y_j^{mis}|Y_j^{obs}, Y_{-j})$ is fit where $j \in 1, 2, ..., p$ and $Y_{-j} = (Y_1, ..., Y_{j-1}, Y_{j+1}, ..., Y_p)$. After filling in missing values for each j with draws from the observed values, let Y_{-j}^t* be the completed dataset without Y_j at iteration t (similar to Y_{-j} from before). For each j, a parameter value ψ_j^t is drawn from $f(\psi_j^t|Y_j^{obs}, Y_{-j}^t)$ and then Y_{mis}^t is drawn from $f(Y_{mis}^t|Y_j^{obs}, Y_{-j}^t*, R, \psi_j^t)$. This process is repeated for $t = 1, ..., T$.

Joint modeling is a good approach if the distribution of the data is thought to follow the multivariate distribution because of how easily it can be implemented. On the other hand, FCS is the more flexible method of multiple imputation and can work even if no appropriate multivariate distribution exists. In spite of this, Lee & Carlin (2010) found that both joint modeling and FCS performed similarly even when the data did not follow a multivariate normal distribution.

2.5 Inference when missing data mechanism is MNAR

When the missing data mechanism is MNAR, the ignorability assumption no longer holds. In this case, the relationship between missingness and the data must be explicitly identified as a model. Selection models and pattern-mixture models are the two most frequently used ways to model non-ignorability.

A new, but similar set of notation will be used to introduce pattern mixture and selection models. Suppose there are n subjects. Let Y be a vector of outcomes where Y_i represents the outcome of the ith subject. Again, R is indicator for whether Y is observed. If Y_i is observed, $R_i = 1$. Otherwise, $R_i = 0$. Let X be a matrix of covariates where X_{ij} is the jth covariate of the ith subject. The joint distribution of Y and R is denoted by $f(Y_i, R_i|X_i, \psi)$ where ψ again represents the parameters of the missing data model. The difference between selection and pattern-mixture models is how the joint distribution is factored.

2.5.1 Pattern mixture models

For pattern mixture models, the factorization is:

$$f(Y_i, R_i|X_i, \psi) = f(Y_i|R_i, Xi, \psi) \cdot f(R_i|X_i, \psi).$$

Generally for any given dataset, the focal point of the inference lies in the relationship between the outcomes, Y, and the covariates X. An alternative explanation is that the full data marginal distribution of Y given X, $f(Y_i|X_i)$, is of interest. Under mixture models, $f(Y_i|X_i)$ is not modeled explicitly. Instead by averaging $f(Y_i|R_i, X_i, \psi)$ over the values that R_i can take, $f(Y_i|X_i, \psi)$ can be obtained. That is,

$$\sum_R f(Y_i|R_i, X_i) = f(Y_i|X_i, \psi).$$

The advantage that pattern mixture models have is that by establishing a distribution for each value of R_i, it is straightforward to see which parameters can and cannot be specified by the observed data at hand (Daniels & Hogan, 2008). In addition, Horton & Fitzmaurice (2002) show that it is easy to fit mixture models in the case that the data are actually ignorable. A key disadvantage of pattern mixture models is that the distribution and the parameters of interest require more work after the factorization.

2.5.2 Selection models

For selection models, the joint distribution is factored as

$$f(Y_i, R_i|X_i, \psi) = f(Y_i|X_i, \psi) \cdot f(R_i|Y_i, X_i, \psi).$$

Unlike mixture models, the marginal distribution of Y_i given X_i is immediately available after factorization. This is much more convenient than the mixture model approach where marginalization over another distribution must occur to get the distribution of interest. However, selection models do not come without their downsides. Daniels & Hogan (2008) lay out the two disadvantages to using selection models. First, figuring out the identifiability conditions for selection models is often troublesome. Second, the selection model approach is sensitive to the model specification. In addition, Horton & Fitzmaurice (2002) point out that with the available data, the assumptions necessary for selection models cannot be vetted.

2.6 Sensitivity analysis

As stated earlier, given a dataset with missing data, it is impossible to verify whether the missing data mechanism is MAR or MNAR. Since most missing data methods assume that data are MAR, it is important to check what happens to the model when the MAR assumption is removed. This process is known as sensitivity analysis. In particular, sensitivity analysis looks at how robust a model is and how much the model changes when the assumptions underpinning the model are no longer assumed to be true.

However, there is no established procedure for sensitivity analysis. According to Permutt (2016) sensitivity analysis consists of an inventory of other possible analyses that could be carried out in place of the primary analysis in most research protocols. For example, if the primary analysis was performed by fitting a model with multiple imputation under MAR, a typical sensitivity analysis would simply be fitting a model using complete case analysis. The role of assumptions, in particular, what happens when the assumptions do not hold, is sparsely discussed. On the other hand, the NRC report recommends that any analysis assuming that the data are MAR discuss what would occur if the data were MNAR (CNStat, 2010).

Two methods of sensitivity analysis that follow the NRC report's recommendation are the delta adjustment procedure and the mean score method.

2.6.1 Delta adjustment procedure

The delta adjustment procedure to multiple imputation is an easy method using a pattern mixture model to impute values under the MNAR assumption. The three steps to the delta adjustment procedure are:

1. Use multiple imputation under the MAR assumption to fit a model for outcome variable Y that has missing values.
2. Add a fixed quantity δ to the linear predictor.
3. Impute the missing values using the model from step 2.

As an example, consider the linear model $Y = \beta_0 + \beta_1 T + \beta_2 X$ fitted with multiple imputation where T is a treatment variable and X is an auxiliary variable. A model using the delta adjustment procedure could look like $Y = \beta_0 + \beta_1 T + \beta_2 X + \delta(1-R)$. If there is reason to believe that there is a difference in differences between the observed patients and those that dropped out within the different treatment arms in a clinical trial, then another model using the delta adjustment procedure could simply specify a unique δ_i for each treatment arm. In the case where there are only two treatment arms the model would look like $Y = \beta_0 + \beta_1 T + \beta_2 X + \delta_0 I_{\{T=-1\}}(1-R) + \delta_1 I_{\{T=1\}}(1-R)$.

Instead of offsetting the imputed values by a δ, there exists a similar procedure that scales the imputed values instead. The procedure is exactly the same as the one above except that instead of adding δ to a linear predictor, a coefficient is multiplied by δ. Using the same linear model as above, a model using this alternative delta adjustment procedure could look like $Y = \beta_0 + \delta(1-R)\beta_1 T + \beta_2 X$. Again, if we believe that there is a difference in differences between observed and unobserved within the different treatment arms, we can specify a δ_i for each treatment arm. In the two-arm case, the model would be $Y = \beta_0 + \delta_0 I_{\{T=-1\}}\beta_1(1-R)T + \delta_1 I_{\{T=1\}}\beta_1(1-R)T + \beta_2 X$.

The delta adjustment procedure can easily be modified to work for non-linear predictors. As an example, Leacy, Floyd, Yates, & White (2017) used the delta adjustment procedure to assess sensitivity to departures from the MAR assumption using a logistic regression imputation model.

2.6.2 Mean score method

An alternative to the delta adjustment is the mean score method. Pepe, Reilly, & Fleming (1994) introduced the mean score method to retrieve missing outcome data, and Pepe & Reilly (1995) adapted the method for missing covariate data. The basic idea is to use an auxiliary or surrogate variable (outcome or covariate) in order to recover information about the missing variable of interest.

White, Carpenter, & Horton (in revision) proposed using the mean score method for sensitivity analysis rather than its intended use for incomplete outcome and covariate data. Their proposed method uses a series of estimating equations implementing a pattern mixture model to model the MNAR missing data mechanism. Using notation from before, let the substantive model be a generalized linear model with canonical link,

$$E[Y_i|X_{Si};\beta_S] = h(\beta_S^T X_{Si})$$

where $h(.)$ is the inverse link function, X_{Si} is the p_S-dimensional fully-observed covariates in the substantive model, and β_S is the p_S-dimensional vector of coefficients corresponding to X_{Si}.

In most analyses, the vector of coefficients β_S is of the most interest. If the data were complete, β_S could be estimated by solving $U_S^*(\beta_S) = 0$ where $U_S^*(\beta_S) = \sum_i U_{Si}^*(\beta_S)$ and $U_{Si}^*(\beta_S) = \{Y_i - h(\beta_S^T X_{Si})\}X_{Si}$. The missing data are handled through substituting $U_{Si}^*(\beta_S)$ with its expectation over the distribution of missing data given the observed data X_i, $U_{Si}(\beta_S) = E\Big(U_{Si}^*(\beta_S)|X_i, R_i, R_iX_i\Big)$. Let $U_S(\beta_S) = \sum_i U_{Si}(\beta_S)$. By the law of total expectation, $E\Big(U_{Si}(\beta_S)|X_i\Big) = E\Big(U_{Si}^*(\beta_S)|X_i\Big)$.

$U_{Si}(\beta_S)$ is estimated using a pattern-mixture model:

$$E\Big(Y_i|X_i, R_i;\beta_P\Big) = h\Big(\beta_p^T X_{P_i} + \Delta(X_i)(1 - R_i)\Big)$$

where X_{P_i} includes all fully observed covariates (those in the substantive model and also auxiliary variables). $\Delta(X_i)$ is a value specified by the user which represents the difference between the observed and the missing data. Just like the delta adjustment procedure, $\Delta(X_i)$ can be specified to be a different value for each treatment arm.

For both methods, how do we decide which values of delta to select and whether there is a difference between nonresponders in the various treatment arms? Obviously, trying out all possible deltas for each treatment arm is impossible given time and resource constraints nor is it desirable. One approach suggested by White et al. (2007) is to consult experts about the value of δ. Eliciting expert opinion about δ should give the analyzer an idea of the plausible values that δ could take.

In their study of post-traumatic stress disorder, Kessler, Sonnega, Bromet, Hughes, & Nelson (1995) made use of another strategy where they sent a survey to nonresponders to try to recover a small percentage of missing values and then used those recovered values to inform a choice of δ. For this method, the recovered values from the nonresponders should be compared to the responders to see if there is an overall difference between responders and nonresponders and then within the sample of nonresponders, the recovered values should be compared across the treatment arms. These two comparisons would inform the analyzer whether to use the model with one δ or the model with multiple δ_is corresponding to each treatment arm. They would also provide an estimate of δ or the δ_is.

2.6.3 Return to alcohol intervention trial

One of the goals of the trial was to see whether those who received the intervention consumed less alcohol per drinking occasion. From the observed values, the median amount of drinks consumed per drinking occasion for those in the intervention group was four, whereas that value was five for those who did not receive the intervention. The α value for the hypothesis test was .05, but since there were six outcome variables measured, using a Bonferroni correction, the adjusted $alpha$ was $0.008\overline{3}$. Using a negative binomial regression, the primary analysis showed that there was a significant difference in this outcome between those in the intervention and the control group (p-value of .005) (Kypri et al., 2014).

A sensitivity analysis was performed in order to test whether this difference held under different MNAR assumptions. The mean score method was used via the `rctmiss` package in `Stata`. The values of δ used were 0.05, 0.10, and 0.30. The difference in the outcome between responders and nonresponders in the control group was assumed to be 0. The analysis proved to be robust when $\delta = .05$, but when the unobserved patients in the treatment group were assumed to drink 11% ($\delta = log(1.11) = 0.10$) more than their counterparts, the p-value was no longer significant using the specified level of α. The sensitivity analysis shows that the model was not robust to misspecification. The initial findings that there was a treatment effect were overturned by the results of this sensitivity analysis.

2.6.4 Example using artificial data

Imagine a scenario in a randomized clinical trial where on average there is no difference in the outcomes between the treatment and control groups. However, a sizable proportion of patients in both groups drop out. While there is no difference between responders and nonresponders in the control group, there is a difference between responders and nonresponders in the treatment group. Clearly, the missing data mechanism in this case is MNAR.

With this scenario in mind, an artificial dataset with sample size 5000 was created with 20% missingness in both the control and treatment groups (500 observations missing in each group). The true mean of the outcome variable Y is 10 in both the treatment and control groups, but when the groups are split into responders and nonresponders, the mean of Y is 9 for responders in the treatment group and 14 for nonresponders in the treatment group. There is no difference between the treatment and control groups, but the difference between the responders and nonresponders makes it appears as if there is a difference between the treatment arms. This information is neatly summarized in Table 2.6 and Table 2.7.

	Missing (.2)	Observed (.8)	True
Control	10.08	9.98	10.00
Treatment	13.98	9.02	10.01

Table 2.6: Summary of MNAR dataset

The goal of a RCT is generally to determine whether there is a treatment effect or not, so β_1 is more of concern than β_0. For this reason, we pay little attention to β_0. If the model $Y = \beta_0 + \beta_1 T$ were fit with all values of Y, the coefficient of T, β_1, would be 0 (or at least very close to 0). On the other hand, when a model is fit with only the observed values of y using complete case analysis or with multiple imputation assuming MAR, the estimated value for β_1 is $-.95$.

	Estimate	Std. Error
True Model	0.02	0.07
Complete Case	-0.95	0.06
Multiple Imputation MAR	-0.95	0.06

Table 2.7: Coefficient of t of models fit with MNAR data

As stated earlier, it is impossible to differentiate whether the missing data mechanism is MAR or MNAR from the data alone. In this scenario, without a sensitivity analysis to explore deviations from the MAR assumption, there is a real danger that the incorrect conclusion will be drawn. Even though there is no difference between the treatment and control group, there appears to be a treatment effect because of the nonignorable nonresponse.

Using the delta adjustment procedure, models were fit to the artificial data under various MNAR assumptions in order to analyze the sensitivity of the results. The model used was the additive delta adjustment model where we assume that there is

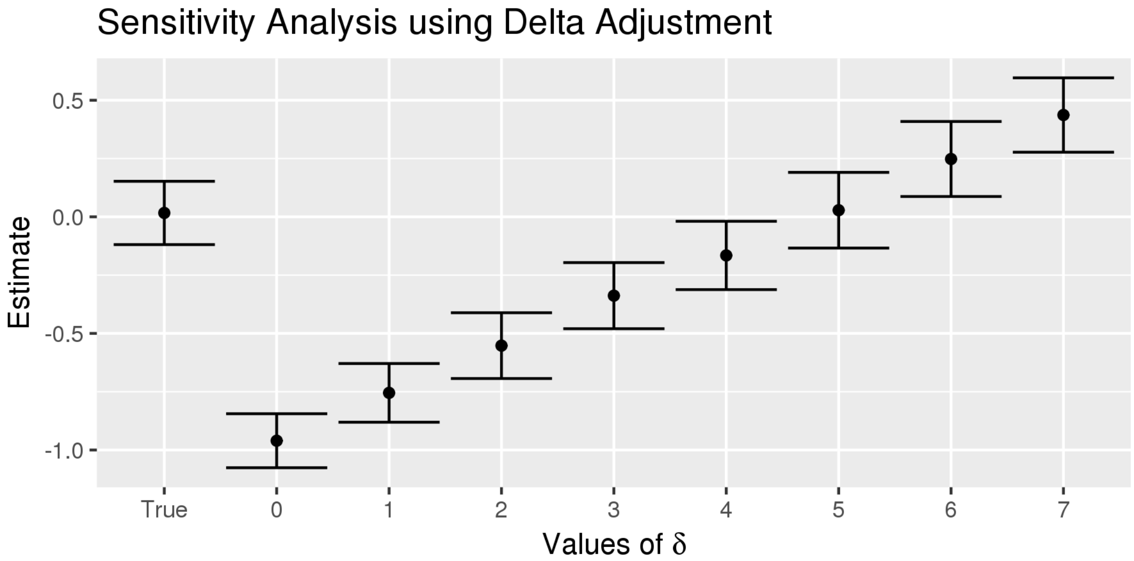

Figure 2.2: Confidence intervals of the coefficient of t using delta adjustment

no difference between responders and nonresponders in the control group. That is,

$$Y = \beta_0 + \beta_1 T + \delta I_{\{T=treatment\}}(1 - R).$$

The values of δ used were integers between zero and seven, where $\delta = 0$ corresponds to multiple imputation under the MAR assumption. Ordinarily, we would need to elicit expert opinion or try to recover information from nonresponders, but since this is a simulated scenario, we will assume that this portion was already completed, and the delta adjustment models used were the best option given the information we know.

Looking at Figure 2.2, the true treatment effect, $\beta_1 = 1$, can only be found in the confidence interval when $\delta = 5$. This shows that the model is not very robust. Changing the values of δ causes the values of β_1 to change by a relatively large margin. If the main analysis was performed using complete case analysis or multiple imputation assuming MAR, the results of this sensitivity analysis would suggest that perhaps further investigation is necessary. The sensitivity analysis should make whoever is analyzing the results of the clinical trial wary of concluding that there is a treatment effect (or not) if there is substantial missing data.

Exporting the data to `Stata` and using the mean score method via the `rctmiss` package to perform a sensitivity analysis gives a similar result. Looking at Figure 2.3 (paying attention only to the branch corresponding to intervention only), again, the true value of β_1 is contained in the confidence interval of the coefficient only when the

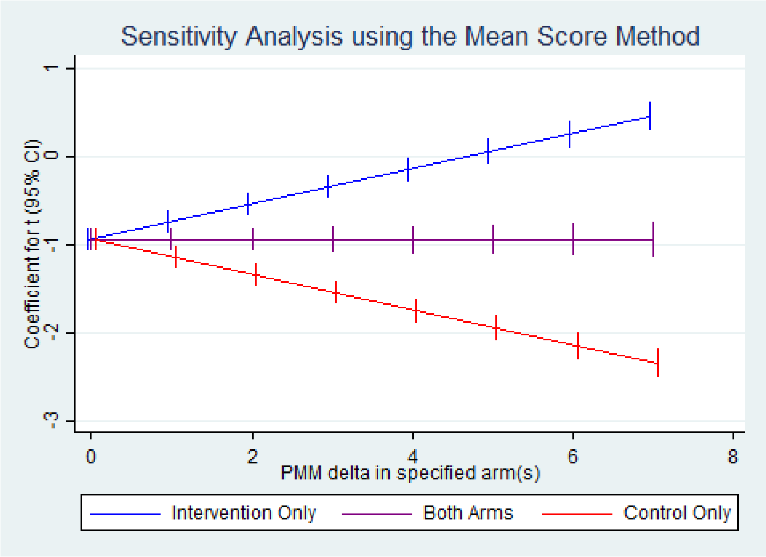

Figure 2.3: Confidence intervals of the coefficient of t using the mean score method

$\delta = 5$. Sensitivity analysis via the mean score method also shows that the model is not robust, and that the analyzer should be careful when deciding whether there is a treatment effect.

The previous two examples highlight why the NRC report recommends strategies to minimize missing data and to routinely performing sensitivity analyses. Yet, Bell et al. (2014) report that in recent analyses of RCTs in top-tier medical journals, only 35% of trials performed some sort of sensitivity analysis. Of those that did, 63% did not weaken assumptions concerning the missing data, and of the papers that conducted a sensitivity analysis, none used the MNAR assumption. While missing data research supports and encourages the use of sensitivity analysis, as of now, the implementation of sensitivity analysis has yet to catch up.

Chapter 3

Simulation study

To determine how well different missing data methods perform under various scenarios, a simulation environment was created to explore hypotheses regarding missing data when the true underlying setting is known. The simulation study, performed using `R`, put the methods detailed in the exposition into practice.

3.1 Simulation environment

In this section, I give an overview of the simulation environment. This simulation environment is intended to replicate a simple, large, two-arm randomized superiority clinical trial with one treatment, one placebo, and one follow-up at the end of an intervention period. All the estimands in the simulation environment are effectiveness (*de facto*) estimands and the method of analysis is intention-to-treat. There are four variables of concern: a variable for the outcome, a variable indicating the treatment group, an auxiliary variable, and an observed indicator for the outcome variable. The dataframes generated in the environment have a column for each of the four variables specified and a fifth column for observed values of the outcome variable.

The treatment variable is denoted by T, the auxiliary variable is denoted by X, the observed indicator is denoted by R, the outcome variable is denoted by Y, and the observed outcome variable is denoted by $ObsY$. For $i \in \{1, 2, ..., n\}$, $t_i, x_i, r_i, y_i, obsy_i$ represents the values of $T, X, R, Y, ObsY$ for the ith patient (ith row in the dataset).

T can only take values in the set $\{-1, 1\}$. If $t_i = 1$, then ith patient in the trial is placed in the treatment group. If $t_i = -1$, the ith patient is placed in the control group. Half the patients in each simulated dataset are placed in the treatment group and the other half in the control group. Unlike T, X is a continuous variable generated using a normal distribution with the mean and the variance of the distribution specified by the author. Simulations were performed for both when Y was a continuous outcome and when it was a dichotomous outcome.

Similar to T, R can only take two values. However, R takes values in the set $\{0, 1\}$ instead of $\{-1, 1\}$. Taken from normal convention, when $r_i = 1$, the outcome variable is observed for the ith subject, meaning $obsy_i = y_i$. Whereas if $r_i = 0$, the outcome variable is missing, meaning $obsy_i$ will be set to `NA`. R will be generated in a variety of ways depending on the missing data mechanism and the outcome variable which will be outlined later.

The values of Y are generated using the values of X and T. In the case when Y is a continuous variable, values of Y are generated using this simple equation:

$$Y = \beta_0 + \beta_1 T + \beta_2 X + \epsilon$$

where $\epsilon \sim N(0, \sigma_\epsilon^2)$ and σ_ϵ is a user-specified parameter. For the dichotomous case, let Y' be the probability that $Y = 1$. Values of Y' are produced using the equation

$$Y' = \frac{1}{1 + \exp\left(-(\beta_0 + \beta_1 T + \beta_2 X + \epsilon)\right)}$$

where $\epsilon \sim N(0, \sigma_\epsilon^2)$ and σ_ϵ is a user-specified parameter. Then for each observation, the value of Y' is compared to a realization of a random uniform variable in the range $[0, 1]$. If Y' is less than that value, Y is set to 1, and if Y' is greater than that value, Y is set to 0.

3.2 Missingness level specification

I created a function called `generateComplete`() was created to generate complete datasets. Missingness is controlled by specifying the expected value of the proportion of missingness (which will be referred to $propMiss$ hereafter). This is easy to implement in the MCAR scenario as in that case, for any i, y_i has $propMiss$ chance of being missing. For the other settings, the functions `generateMAR`() and `generateMNAR`() were designed to generate missingness for the MAR and MNAR settings, respectively. These functions are designed to retain the given expected proportion of missingness in the MAR and MNAR settings using logistic regression.

Given the ith subject, denote the probability that the outcome variable is missing as m_i. The probability that the outcome variable is missing is modeled using the following:

$$logit(m_i) = \log\left(\frac{m_i}{1-m_i}\right) = g_0 + g_1 \cdot t_i + g_2 \cdot y_i. \tag{3.1}$$

A function called `chanceMissing()` takes in the missing data mechanism, $propMiss$, and a risk ratio (or relative risk) RR, as well as the complete dataset generated using `generateComplete()` as parameters and calculates m_i for all observations i by specifying the values of g_0, g_1, and g_2 based on the parameters. As will be explained further along, the risk ratio parameter takes different meanings depending on whether Y is continuous or dichotomous and whether we are in the MAR or MNAR setting. The functions `generateMAR()` and `generateMNAR()` both use `chanceMissing()` in order to create missing values. After the value of m_i is determined using `chanceMissing()`, a random number in the set $[0, 1]$ is drawn using the `runif()` function. If the random number is less than m_i, y_i is set to be missing. Otherwise, y_i is set as observed. Explained below are how the values of g_0, g_1, and g_2 are determined in terms of RR and $propMissing$ in the `chanceMissing()` function under the MAR and MNAR settings.

3.2.1 MAR setting

In the MAR case, since missingness is independent of the value of the (possibly missing) Y, so $g_2 = 0$. Let $p_1 = m_i$ when $t_i = 1$ and $p_{-1} = m_i$ when $t_i = -1$. When generating missingness under MAR, the RR parameter refers to the probability of being missing given $t_i = 1$ divided by the probability of being missing given $t_i = -1$. Rearranging Equation (3.1) using the inverse-logit function and the fact that $g_2 = 0$ we have that,

$$m_i = \frac{\exp(g_0 + g_1 t_i)}{1 + \exp(g_0 + g_1 t_i)} = \begin{cases} p_1 = \dfrac{\exp(g_0 + g_1)}{1 + \exp(g_0 + g_1)} & \text{if } t_i = 1, \\ p_{-1} = \dfrac{\exp(g_0 - g_1)}{1 + \exp(g_0 - g_1)} & \text{if } t_i = -1. \end{cases}$$

Since the treatment and control groups are equal in proportion we have that,

$$\frac{1}{2}p_1 + \frac{1}{2}p_{-1} = propMiss.$$

From the definition of risk ratio, we have that $RR = \frac{p_1}{p_{-1}}$ which implies $p_{-1} + p_{-1} = 2propMiss$. Factoring p_{-1}, the relationship can be written as $p_{-1}(1 + RR) = 2propMiss$. Solving for p_{-1} and p_1 in terms of $propMiss$ and RR gives

$$\frac{\exp(g_0 - g_1)}{1+\exp(g_0 - g_1)} = p_1 = \frac{2propMiss}{1+RR} \cdot RR$$
$$\frac{\exp(g_0 + g_1)}{1+\exp(g_0 + g_1)} = p_{-1} = \frac{2propMiss}{1+RR}.$$

Thus, with some simple rearrangement it follows that

$$\exp(g_0 - g_1) = \frac{z}{1-z}$$
$$\exp(g_0 + g_1) = \frac{RRz}{1-RRz},$$

where $z = \frac{2propMiss}{1+RR}$.
What is left is a simple system of equations,

$$g_0 + g_1 = log\left(\frac{RRz}{1-RRz}\right)$$
$$g_0 - g_1 = log\left(\frac{z}{1-z}\right).$$

Solving for g_0 and g_1 gives

$$g_0 = \frac{1}{2}log\left(\frac{RRz^2}{(1-z)(1-RRz)}\right)$$
$$g_1 = \frac{1}{2}log\left(\frac{RR(1-z)}{1-RRz}\right).$$

3.2.2 MNAR setting

Derivation with dichotomous outcome variable

We start with the case where the outcome variable is dichotomous. So the following derivation of the formulas for g_0, g_1, and g_2 takes for granted that $y_i = 1$ or $y_i = -1$. In this case, the RR parameter refers to the probability of being missing given $y_i = 1$ divided by the probability of being missing given $y_i = 0$.

In the MNAR case, for this simulation, missingness is independent of T and X which means that $g_1 = 0$. Let $p_1 = m_i$ when $y_i = 1$ and $p_0 = m_i$ when $y_i = 0$.

Rearranging Equation (3.1) using the inverse-logit function and the fact that $g_1 = 0$ we have that,

$$m_i = \frac{\exp\left(g_0 + g_2 y_i\right)}{1 + \exp\left(g_0 + g_2 y_i\right)} = \begin{cases} p_1 = \dfrac{\exp(g_0 + g_2)}{1 + \exp(g_0 + g_2)} & \text{if } y_i = 1, \\ p_0 = \dfrac{\exp(g_0)}{1 + \exp(g_0)} & \text{if } y_i = 0. \end{cases}$$

Using the `prop()` function, the proportion of observations in the dataset where $y_i = 1$ and $y_i = 0$ can easily be obtained. The proportion of observations in the dataset such that $y_i = 1$ will be denoted as w. We can write $propMiss$ in terms of w, p_1, and p_0:

$$propMiss = wp_1 + (1 - w)p_0.$$

From the definition of risk ratio, we have that $RR = \frac{p_1}{p_0}$, which implies

$$wRRp_0 + (1 - w)p_0 = propMiss.$$

Factoring p_0, the relationship can be written as $p_0(wRR + (1 - w)) = propMiss$. Solving for p_0 in terms of $propMiss$ and RR gives

$$\frac{\exp(g_0)}{1 + \exp(g_0)} = p_0 = \frac{propMiss}{(1 - w) + wRR}.$$

With some rearrangement it follows that

$$exp(g_0) = \frac{z}{1 - z},$$

where $z = \frac{p}{(1-w)+wRR}$. Taking the log of both sides gives,

$$g_0 = log\left(\frac{z}{1 - z}\right)$$

Since $p_1 = p_0 RR$,

$$\frac{\exp(g_0 + g_2)}{1 + \exp(g_0 + g_2)} = p_1 = \frac{propMiss}{(1 - w) + wRR} \cdot RR.$$

With some manipulation and rearrangement, we have that

$$exp(g_0 + g_2) = \frac{RRz}{1 - RRz}.$$

Taking the log of both sides gives

$$g_0 + g_2 = log\left(\frac{RRz}{1 - RRz}\right).$$

Putting g_0 in terms of RR and $propMiss$ and solving for g_2 yields

$$g_2 = log\left(\frac{RR(1-z)}{z(1 - RRz)}\right).$$

Derivation with continuous outcome variable

In the continuous case, while it is much more difficult to get the exact proportion of missingness desired, the logistic regression and risk ratio framework still create missingness at a proportion extremely close to the one specified. If we condition Y on T, Y becomes a linear combination of normal variables, so Y conditioned on T follows a normal distribution as well. Specifically $Y \mid T = t \sim N(\beta_0 + \beta_1 t + \beta_2 \mu_x + \epsilon, \beta_2 \sigma_X^2 + \sigma_\epsilon^2)$. Because half the observations in the dataset have $t_i = -1$ and the other half $t_i = 1$, Y can be described as a mixture of the following distributions:

$$\begin{aligned} Y_1 &\sim N(\beta_0 + \beta_1 + \beta_2\mu_X, \beta_2^2\sigma_X^2 + \sigma_\epsilon^2) \\ Y_{-1} &\sim N(\beta_0 - \beta_1 + \beta_2\mu_X, \beta_2^2\sigma_X^2 + \sigma_\epsilon^2), \end{aligned}$$

where the weights of the distributions are both equal to 0.5.

The risk ratio parameter, RR, passed to the `makeMissing`() function, has a different meaning in the context of a continuous outcome variable. RR no longer indicates the ratio of the probability of being missing when $y_i = 1$ and $y_i = 0$. Instead, it refers to the ratio of the probability of being missing when $y_i = E(Y_1)$ and when $y_i = E(Y_{-1})$. Using notation,

$$RR = \frac{P(r_i = 0 | y_i = E(Y_1))}{P(r_i = 0 | y_i = E(Y_2))}.$$

In addition, the relationship to derive the probability of being missing for observation i, m_i, is modified from Equation 3.1 to

$$logit(m_i) = \frac{m_i}{1 - m_i} = g_0 + g_1 \cdot t_i + g_2 \cdot \frac{y_i - \beta_0 + \beta_1 - \beta_2\mu_X}{2\beta_1}. \tag{3.2}$$

This modification effectively scales the values of Y so that when $y_i = E(Y_1)$, m_i is the same as when $y_1 = 1$ in the dichotomous case and when $y_i = E(Y_{-1})$, m_i is the same as when $y_1 = 0$ in the dichotomous case. Since the values of Y are normally distributed around $E(Y_1)$ and $E(Y_{-1})$ (not simply equal to those values) and the inverse-logit function is not linear, the expected proportion of missingness will not be exactly equal to $propMiss$. However, as Figure 3.1 shows, the inverse-logit function is basically linear towards the center (the red line is the line $y = x$). This implies that the logistic regression risk ratio framework we have described will generate missingness at a closer rate to $propMiss$ when the risk ratio is closer to 1. Additionally, when the variance of Y_1 and Y_{-1} is smaller, the actual missing rate will be closer to $propMiss$.

As long as the risk ratios used in the study are not too extreme, the actual proportion of missing values in the continuous case will almost always be nearly equal to $propMiss$. Furthermore, the concern is not about how missing data methods perform at an exact proportion of missingness, but how they perform near a specified $propMiss$. Thus, the logistic regression risk ratio framework generates missingness with a satisfactory level of precision.

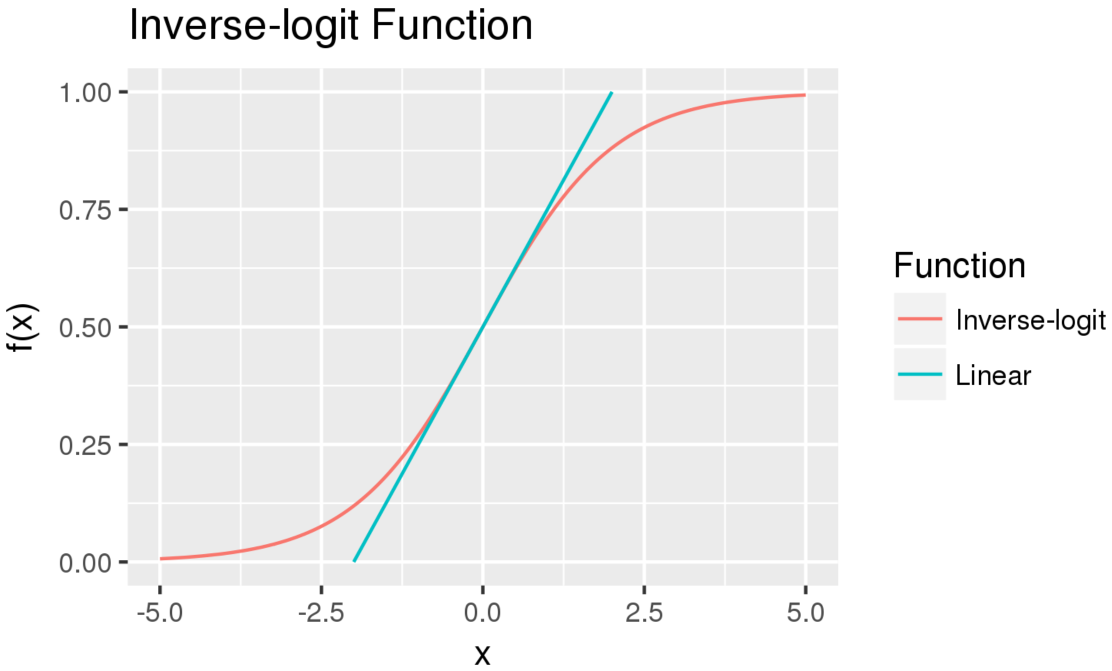

Figure 3.1: Comparison of the inverse-logit and identity functions

In order to validate the theory and the design behind creating missingness through a logistic regression risk ratio framework, a small empirical simulation was run. For risk ratios of 1, 3, 5, and 10, the value of $propMiss$ passed to the function was plotted against the actual proportion of values missing after the function was run.

Half the values of Y were generated under a normal distribution with mean

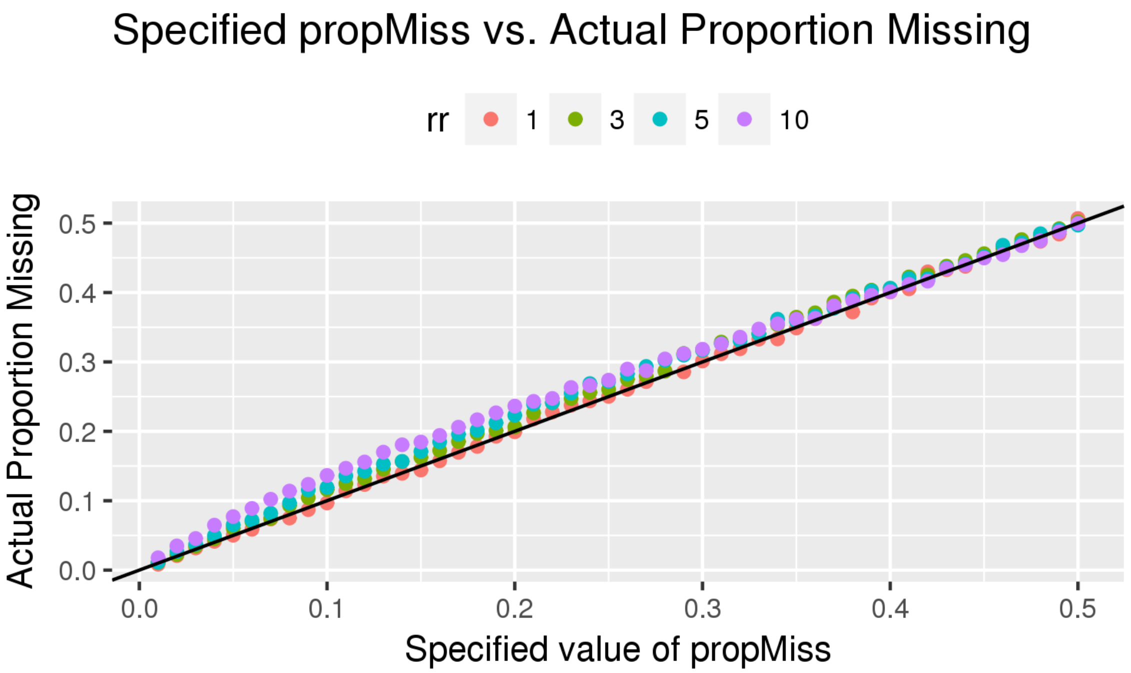

Figure 3.2: Performance of propMiss function

0 and standard deviation 0.5, and the other half were generated under a normal distribution with mean 1 and standard deviation 0.5. Equation 3.2 was used to generate the probability that Y was missing for an observation. Then this probability was compared to a realization of a random uniform from $[0, 1]$ as before to generate MNAR missingness with a continuous outcome variable. If missingness is generated perfectly by the function, then the scatterplot of $propMiss$ (user-specified proportion of missingness) against the actual missingness should be on the line $y = x$. Clearly this is not the case here as most of the points seem to lie above the line. However, even for larger risk ratio values, the actual observed proportion of missingness is almost always within 0.05 of the specified $propMiss$. Since the standard deviation of Y used in this simulation was 0.5, and values from the continuous Y values are scaled by $2\beta_1$, as long as the standard deviation of the two normal distributions is around or less than β_1, the actual proportion missing will be nearly equal to the $propMiss$ specified.

3.3 Method

In order to get a comprehensive examination of various methods of handling missing data, four different methods were assessed in four different scenarios. The methods were tested for both when Y was dichotomous and continuous and for when X provided information about Y and when it did not (i.e., when $\beta_2 = 0$ and when $\beta_2 \neq 0$). For each scenario, models were fit using complete case analysis, Bayesian multiple imputation, bootstrap multiple imputation, and predictive mean matching all of which can be implemented using the `mice` package in `R` (van Buuren & Groothuis-Oudshoorn, 2017). In each simulation, predictive mean matching was used with four different numbers of donors: 1, 3, 10, and 20.

Each method of handling missing data was tested under the three different missing data mechanisms and when 10%, 30%, and 50% of the observations were missing a value for Y. For the MAR case, the data was missing at random with respect to T. In the case of a dichotomous Y, logistic regression models were fit using the various missing data methods, and the values of the parameters were set as $\beta_0 = 1$, $\beta_1 = 2$, $\mu_X = 0$, $\sigma_x = 1$, $\sigma_\epsilon = .1$, and $RR = 5$. For the set of simulations where there existed a relationship between X and Y, $\beta_2 = 1$, and when there was no relationship, β_2 was set to 0. In the case of continuous Y, linear regression models were fit, and the values of the parameters were $\beta_0 = 5.7$, $\beta_1 = -1$, $\mu_X = 10$, $\sigma_x = 1.44$, $\sigma_\epsilon = 1$, and $RR = 5$. For the set of simulations where there existed a relationship between X and Y, $\beta_2 = 0.1$, and again, when there was no relationship, $\beta_2 = 0$. The parameter values chosen for the continuous case were loosely based on data from the alcohol intervention trial.

For each of the four methods, both X and T were used as predictors and $ObsY$ was used as the response variable. As a baseline of comparison for the models, since the values of Y are available regardless of the value of R, a model using the true values Y as the response variable were fit as well.

The coefficient estimates and standard errors were collected for each simulation and the average was taken across all simulation in each scenario. Confidence intervals were also generated using the standard symmetric method which are how confidence intervals are created using `mice` and Rubin's rules (Rubin, 1987).

3.4 Results

Results of the simulation study are presented in tables and graphs below. For the rest of this chapter, I will often refer to multiple imputation as MI. Since β_1 is the coefficient for the treatment variables, it is the coefficient of most concern, and thus more emphasis is placed on it.

3.4.1 Dichotomous outcome variable

Relationship between auxiliary and outcome variables exists ($\beta_2 \neq 0$)

	beta0	se0	beta1	se1	beta2	se2
Truth	1.0007	0.1090	2.0038	0.1201	1.0047	0.1067
Complete Case	1.0058	0.1311	2.0122	0.1446	1.0103	0.1282
Bayesian MI	1.0014	0.1331	2.0084	0.1472	1.0094	0.1303
Bootstrap MI	1.0090	0.1353	2.0191	0.1491	1.0135	0.1311
PMM (d = 1)	1.0078	0.1328	2.0156	0.1452	0.9979	0.1270
PMM (d = 3)	1.0064	0.1323	2.0124	0.1444	0.9968	0.1279
PMM (d = 10)	1.0077	0.1316	2.0110	0.1447	0.9924	0.1273
PMM (d = 20)	1.0083	0.1316	2.0123	0.1436	0.9892	0.1271

Table 3.1: Data are MCAR and the proportion missing is 0.3

	beta0	se0	beta1	se1	beta2	se2
Truth	1.0072	0.1095	2.0120	0.1206	1.0070	0.1070
Complete Case	1.0158	0.1467	2.0230	0.1576	1.0095	0.1209
Bayesian MI	1.0089	0.1498	2.0170	0.1612	1.0102	0.1228
Bootstrap MI	1.0241	0.1517	2.0324	0.1626	1.0106	0.1228
PMM (d = 1)	1.0294	0.1439	2.0348	0.1548	0.9998	0.1210
PMM (d = 3)	1.0285	0.1428	2.0325	0.1536	0.9972	0.1209
PMM (d = 10)	1.0272	0.1418	2.0321	0.1530	0.9973	0.1210
PMM (d = 20)	1.0256	0.1421	2.0292	0.1533	0.9934	0.1209

Table 3.2: Data are MAR and the proportion missing is 0.3

Tables 3.1, 3.2, and 3.3 display the mean of the coefficients and standard errors of the models fit for the 1,000 datasets with a dichotomous outcome variable and $\beta_2 = 1$ under MCAR, MAR, and MNAR, respectively. As expected, the bias of the coefficient estimates becomes larger as the missing data mechanism shifts from MCAR to MAR to MNAR. However, all methods of multiple imputation still have relatively little bias except when the missing data mechanism is MNAR and the coefficient in question is β_0. In general, as the number of donors increases for PMM, the coefficient estimates become closer to the true values, indicating that the performance of predictive mean matching increases with the number of donors. Another interesting (but perhaps unimportant) pattern is that while the estimate for β_2 under complete case analysis, Bayesian MI, and bootstrap MI is positively biased, the estimate for β_2 using predictive mean matching is negatively biased for all cases.

Looking at the third column in each table, with the exception of the true model, estimates of β_1 produced under Bayesian MI are the closest to the true value of $\beta_1 = 2$

	beta0	se0	beta1	se1	beta2	se2
Truth	1.0118	0.1097	2.0167	0.1210	1.0100	0.1072
Complete Case	0.5358	0.1222	2.0188	0.1382	1.0093	0.1270
Bayesian MI	0.5322	0.1243	2.0147	0.1412	1.0066	0.1302
Bootstrap MI	0.5391	0.1254	2.0256	0.1417	1.0116	0.1305
PMM (d = 1)	0.5359	0.1229	2.0203	0.1391	0.9954	0.1274
PMM (d = 3)	0.5366	0.1236	2.0200	0.1391	0.9940	0.1265
PMM (d = 10)	0.5351	0.1221	2.0173	0.1379	0.9889	0.1261
PMM (d = 20)	0.5343	0.1219	2.0167	0.1371	0.9828	0.1254

Table 3.3: Data are MNAR and the proportion missing is 0.3

(this is also true for β_0 and β_2 as well). When the missing data mechanism is MNAR, predictive mean matching with 10 or 20 donors seems to perform just as well as the Bayesian MI (biases of 0.0173, 0.0167, and 0.0147 respectively). Under this scenario, Bayesian MI performs the best, followed by complete case, predictive mean matching, and bootstrap MI in that order.

A few obvious patterns can be found in Figure 3.3 which displays the coverage of the true value of β_1 using the confidence intervals calculated using the various missing data methods. As the proportion of observations with missing values increases, the coverage of β_1 becomes more erratic. An interesting thing to note is while bootstrap MI seemed to produce the most biased coefficient estimates, the confidence intervals generated from bootstrap multiple imputation always have nearly the correct level of coverage (doing as well as complete case and Bayesian MI in this respect). Predictive mean matching always undercovers $\beta_1 = 2$ as the proportion missing increases (even when the missing data mechanism is MCAR) indicating that the confidence intervals of β_1 are much too narrow using this method (especially considering that bootstrap MI is just as biased as PMM but manages to get the correct coverage level). The coverage is lowest under all missing data mechanisms when the number of donors is just one, reinforcing the observation from Tables 3.1, 3.2, and 3.3 that information seems to be gained by increasing the number of donors. All other methods either have the correct coverage level or are very close.

No relationship between auxiliary and outcome variables ($\beta_2 \neq 0$)

Tables 3.4, 3.5, and 3.6 summarize the results of the simulations run with a dichotomous Y and no relationship between the auxiliary and outcome variables. The main story is the same: Bayesian MI provides the least biased estimates for virtually all coefficients under all missing data mechanisms (with the exception of β_0 when the data are

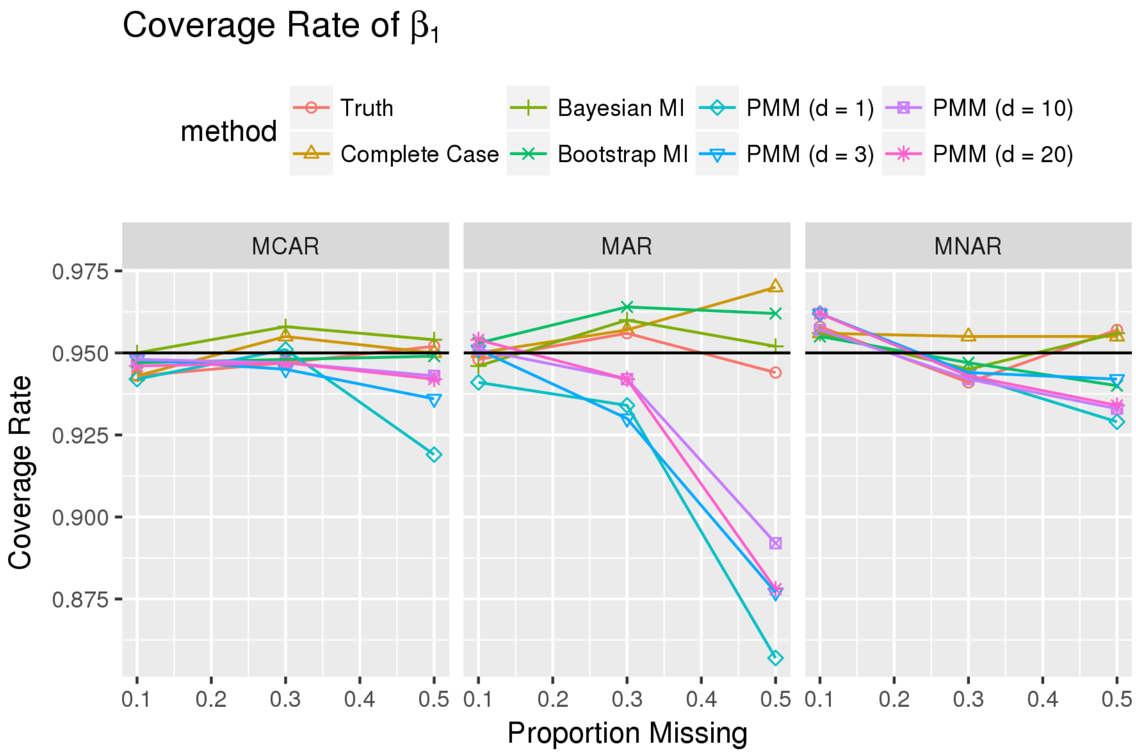

Figure 3.3: Dichotomous outcome with relationship with auxiliary variable

MNAR). A large difference between the two simulations is that while complete case analysis, Bayesian MI, and Bootstrap MI all perform similarly between the two different scenarios, predictive mean matching is much less effective when $\beta_0 = 0$. The bias in coefficient estimates increased and the variance of the estimates did as well. The drop in performance is steeper when the number of donors is lower, and there is a much more noticeable increase in the effectiveness of predictive mean matching as the number of donors increases. In this scenario, Bayesian MI once again provides the best results followed by complete case analysis, bootstrap MI, and predictive mean matching.

Figure 3.4 augments the observations made from gleaning at the tables. While complete case analysis, Bayesian MI, and bootstrap MI always have nearly the correct coverage levels under all levels of missingness and mechanisms, the same cannot be said of predictive mean matching. If the proportion of missingness is greater than 0.1, predictive mean matching undercovers the true values of β_1. The undercoverage worsens as the proportion of data observations with missing values increases and as the number of donors decreases. A parallel between this case and the previous one is that the confidence intervals created with predictive mean matching when the data

	beta0	se0	beta1	se1	beta2	se2
Truth	1.0116	0.1186	2.0166	0.1187	-0.0019	0.0916
Complete Case	1.0145	0.1427	2.0255	0.1430	-0.0014	0.1099
Bayesian MI	1.0091	0.1455	2.0216	0.1455	-0.0021	0.1125
Bootstrap MI	1.0185	0.1461	2.0315	0.1475	-0.0016	0.1125
PMM (d = 1)	1.0441	0.2118	2.0948	0.2115	0.0002	0.1702
PMM (d = 3)	1.0382	0.1904	2.0690	0.1904	0.0006	0.1403
PMM (d = 10)	1.0284	0.1716	2.0489	0.1704	0.0004	0.1202
PMM (d = 20)	1.0208	0.1609	2.0373	0.1613	0.0005	0.1141

Table 3.4: Data are MCAR and the proportion missing is 0.3

	beta0	se0	beta1	se1	beta2	se2
Truth	1.0086	0.1180	2.0093	0.1181	0.0006	0.0914
Complete Case	1.0163	0.1619	2.0170	0.1620	0.0021	0.1008
Bayesian MI	1.0052	0.1678	2.0065	0.1675	0.0018	0.1022
Bootstrap MI	1.0263	0.1703	2.0279	0.1706	0.0021	0.1021
PMM (d = 1)	1.1225	0.2406	2.1268	0.2388	0.0016	0.1317
PMM (d = 3)	1.1034	0.2386	2.1060	0.2391	0.0028	0.1186
PMM (d = 10)	1.0612	0.2220	2.0635	0.2216	0.0016	0.1082
PMM (d = 20)	1.0459	0.1996	2.0484	0.1996	0.0023	0.1055

Table 3.5: Data are MAR and the proportion missing is 0.3

are MAR are more prone to undercovering the true value of β_1 than when the data are MNAR.

3.4.2 Continuous outcome variable

Relationship between auxiliary and outcome variables exists ($\beta_2 \neq 0$)

Tables 3.7, 3.8, and 3.9 correspond to the simulations where there was a continuous outcome and $\beta_2 \neq 0$. After running a simulation with a continuous outcome, many of the observations made when there was a dichotomous outcome no longer apply. All the methods seem to give nearly the same results. Bayesian MI seems to have a lost any advantage it had over the other methods. Adding more donors does not improve the performance of predictive mean matching, and in fact, adding more donors seems to marginally increase the bias of the method. There does not seem to be any drop in performance from when the missing data mechanism is MCAR to when it is MAR. However, unlike when the outcome variable is dichotomous, when the missing data mechanism is MNAR and the the outcome variable is continuous, all the missing data methods have biased estimates for all coefficients (whereas when Y was dichotomous,

	beta0	se0	beta1	se1	beta2	se2
Truth	1.0056	0.1177	2.0053	0.1178	-0.0036	0.0912
Complete Case	0.5234	0.1301	2.0112	0.1304	-0.0041	0.1112
Bayesian MI	0.5192	0.1340	2.0075	0.1349	-0.0035	0.1135
Bootstrap MI	0.5259	0.1331	2.0173	0.1346	-0.0025	0.1130
PMM (d = 1)	0.5749	0.2136	2.0854	0.2137	-0.0053	0.1587
PMM (d = 3)	0.5554	0.2039	2.0534	0.2023	-0.0051	0.1377
PMM (d = 10)	0.5466	0.1713	2.0383	0.1721	-0.0034	0.1205
PMM (d = 20)	0.5343	0.1531	2.0265	0.1535	-0.0048	0.1161

Table 3.6: Data are MNAR and the proportion missing is 0.3

	beta0	se0	beta1	se1	beta2	se2
Truth	5.7010	0.2218	-0.9987	0.0316	0.0998	0.0220
Complete Case	5.6884	0.2654	-0.9975	0.0377	0.1010	0.0263
Bayesian MI	5.6910	0.2717	-0.9974	0.0387	0.1008	0.0269
Bootstrap MI	5.6893	0.2724	-0.9972	0.0388	0.1009	0.0270
PMM (d = 1)	5.6956	0.2821	-0.9980	0.0398	0.1003	0.0280
PMM (d = 3)	5.6951	0.2765	-0.9977	0.0392	0.1004	0.0274
PMM (d = 10)	5.7065	0.2704	-0.9970	0.0389	0.0992	0.0268
PMM (d = 20)	5.7212	0.2701	-0.9971	0.0389	0.0977	0.0267

Table 3.7: Data are MCAR and the proportion missing is 0.3

estimates of β_1 and β_2 were unbiased).

The coverage of the true value of β_1 is much less erratic with a continuous variable than when the outcome variable is dichotomous. When the missing data method is MCAR, all the methods have the correct coverage rate at all proportions of missingness. Coverage under MAR only becomes problematic when the proportion of observations with missing values is 0.5. All missing data methods undercover the true value of β_1 indicating that the variance of the coefficient estimates is less than what it should be. Since the coefficient estimates are biased when the missing data mechanism is MNAR, the coverage rate is much lower than when the mechanism is not MNAR. As the proportion of observations with missing values increases, the coverage rate actually begins to approach zero for nearly all methods reflecting how the results become more biased when there is more missing data and the missing data mechanism is MNAR.

No relationship between auxiliary and outcome variables ($\beta_2 = 0$)

Tables 3.10, 3.11, and 3.12 recap the simulations where there was a continuous outcome and $\beta_2 = 0$. There is little difference in the results between the simulation when $\beta_2 = 0.1$ and when $\beta_2 = 0$. Again, there is almost no difference in the coefficient

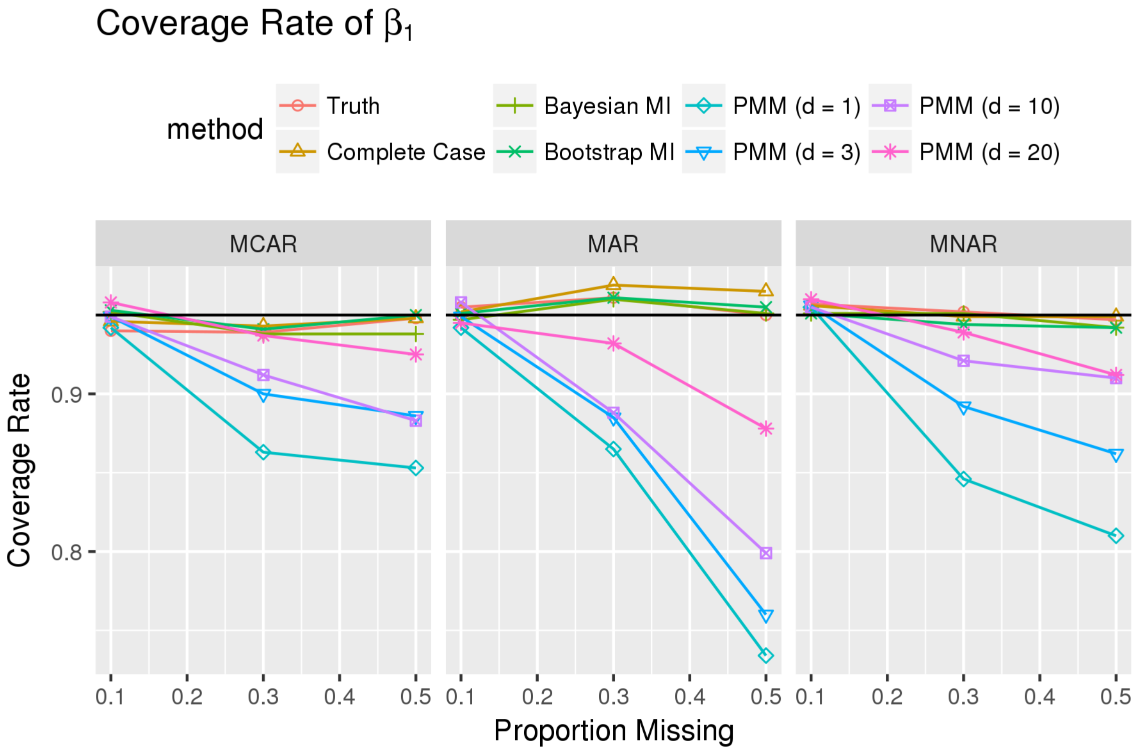

Figure 3.4: Dichotomous outcome without relationship with auxiliary variable

estimates of the various methods across all the missing data mechanisms. All methods provide biased results when the missing data are MNAR and there is no change in the performance of the missing data methods from when the data are MCAR to MAR. The estimates of β_0 are less biased in this case than in the previous one.

As with the case of the tables, the coverage plot (Figure 3.6) for this set of simulations share a similar story with the plot of the previous set of simulations. All methods seem to have the correct coverage rate at all proportions of missingness when the missing data mechanism is MNAR. The performance in coverage only drops under MAR when missingness proportion increases to 0.5. Again, because the estimates of β_1 are biased under MNAR, the coverage probability is very low. While the coverage rate approaches zero as the proportion of observations with missing values increases, the coverage rate under predictive mean matching converges much more slowly. Increasing the number of the donors actually causes the coverage rate of predictive mean matching to drop faster when missingness increases.

	beta0	se0	beta1	se1	beta2	se2
Truth	5.6952	0.2222	-1.0000	0.0316	0.1004	0.0220
Complete Case	5.6978	0.2659	-0.9980	0.0395	0.1003	0.0263
Bayesian MI	5.6979	0.2713	-0.9983	0.0403	0.1003	0.0268
Bootstrap MI	5.6959	0.2722	-0.9977	0.0404	0.1005	0.0269
PMM (d = 1)	5.6965	0.3025	-0.9988	0.0430	0.1004	0.0300
PMM (d = 3)	5.7085	0.2859	-0.9984	0.0417	0.0992	0.0283
PMM (d = 10)	5.7168	0.2793	-0.9977	0.0412	0.0984	0.0276
PMM (d = 20)	5.7344	0.2746	-0.9979	0.0407	0.0967	0.0272

Table 3.8: Data are MAR and the proportion missing is 0.3

	beta0	se0	beta1	se1	beta2	se2
Truth	5.6953	0.2222	-1.0004	0.0316	0.1005	0.0220
Complete Case	6.1180	0.2529	-0.8473	0.0370	0.0870	0.0249
Bayesian MI	6.1183	0.2591	-0.8472	0.0381	0.0870	0.0255
Bootstrap MI	6.1176	0.2593	-0.8471	0.0377	0.0870	0.0255
PMM (d = 1)	6.1356	0.2925	-0.8471	0.0407	0.0853	0.0287
PMM (d = 3)	6.1465	0.2716	-0.8463	0.0393	0.0842	0.0267
PMM (d = 10)	6.1573	0.2617	-0.8475	0.0381	0.0831	0.0257
PMM (d = 20)	6.1701	0.2558	-0.8463	0.0377	0.0819	0.0252

Table 3.9: Data are MNAR and the proportion missing is 0.3

3.5 Discussion

Overall, the results of the simulations confirm that coefficient estimates are more biased as missingness increases and when the missing data mechanism is not MCAR. The simulations also highlight major differences between when the outcome variable is dichotomous and continuous. When the outcome variable is dichotomous, all methods provide nearly unbiased estimates of the treatment effect (β_1) under MCAR, MAR, and MNAR data, Bayesian MI is the least biased method to analyze the data, and increasing the donors (even up to 20) improves the capabilities of predictive mean matching.

The strong performance of Bayesian MI relative to other methods can be explained by the properties of the other methods that Bayesian MI outperforms. Bootstrapping as a general procedure or technique derives much of its value when parametric assumptions are violated. When the outcome variable Y was dichotomous, logistic regression models were fit. The three conditions for drawing accurate inferences using logistic regression are logit-linearity (i.e., the log(odds) are a linear function of the predictors), randomness, and independence (Cannon et al., 2013). Values of the

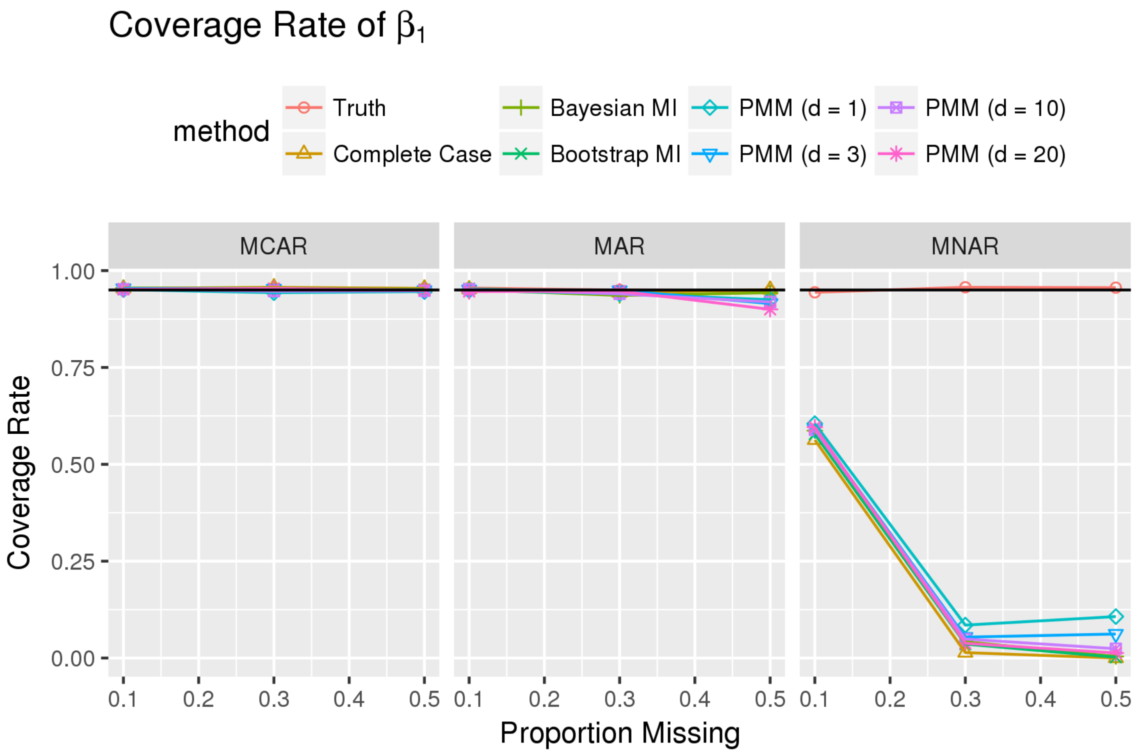

Figure 3.5: Continuous outcome with relationship with auxiliary variable

outcome Y were independently and randomly generated using a logit-linear model, so clearly the simulation data satisfies the three conditions for logistic regression. Since there are no violations of the parametric conditions, bootstrapping gains no advantage as a nonparametric method which may explain why it does not outperform any other method.

Since Y only takes values of 0 or 1, this removes the advantage predictive mean matching has in imputing realistic values which could explain why it does not perform as well. However, this also reduces the likelihood of a bad match when including multiple donors, and since the realization of the value of Y is based on a probabilistic model, predictive mean matching should perform better when the donor pool is larger as it should better reflect this model. For example, consider a scenario where the true probability that $y_i = 1$ when $t_i = -1$ is $\frac{1}{2}$. If the number of donors was just 3, the probability of imputing $y_i = 1$ would be limited to these values: 0, $\frac{1}{3}$, $\frac{2}{3}$, and 1. None of these probabilities are reflective of what the probability of imputing $y_i = 1$ actually should be (should be $\frac{1}{2}$). In contrast, having a large candidate pool of observations with $t_i = -1$ would make it more likely that close to half the candidates would have values $y_i = 1$. When the auxiliary variable provides no information about Y, this

	beta0	se0	beta1	se1	beta2	se2
Truth	5.6975	0.2220	-1.0005	0.0316	0.0003	0.0220
Complete Case	5.6901	0.2658	-1.0002	0.0378	0.0010	0.0263
Bayesian MI	5.6897	0.2726	-1.0005	0.0389	0.0010	0.0270
Bootstrap MI	5.6875	0.2720	-1.0004	0.0388	0.0012	0.0269
PMM (d = 1)	5.6998	0.4352	-1.0005	0.0816	-0.0002	0.0428
PMM (d = 3)	5.6854	0.3418	-1.0006	0.0596	0.0012	0.0337
PMM (d = 10)	5.6904	0.2916	-1.0008	0.0470	0.0009	0.0287
PMM (d = 20)	5.6888	0.2759	-1.0001	0.0428	0.0011	0.0273

Table 3.10: Data are MCAR and the proportion missing is 0.3

	beta0	se0	beta1	se1	beta2	se2
Truth	5.6971	0.2222	-0.9985	0.0316	0.0002	0.0220
Complete Case	5.6987	0.2664	-0.9976	0.0395	0.0001	0.0263
Bayesian MI	5.6972	0.2729	-0.9977	0.0406	0.0003	0.0270
Bootstrap MI	5.7004	0.2734	-0.9974	0.0403	0.0000	0.0271
PMM (d = 1)	5.6963	0.5004	-1.0027	0.0979	-0.0003	0.0488
PMM (d = 3)	5.6961	0.3748	-0.9976	0.0692	0.0003	0.0368
PMM (d = 10)	5.7008	0.3066	-0.9977	0.0522	-0.0001	0.0303
PMM (d = 20)	5.6993	0.2847	-0.9972	0.0460	0.0001	0.0280

Table 3.11: Data are MAR and the proportion missing is 0.3

exacerbates the problem caused by the small number of donors. If X provides no information, then predictive mean matching only uses T to select the donor pool.

For the simulations where Y was continuous, all missing data methods are biased when the missing data mechanism is MNAR and no one method outperforms another. There seemed to be no difference when X provided information about Y and when it did not. This is perhaps because the effect of X on Y was not as strong as the effect of T on Y. The results of this simulation study agree with the assertion made by van Buuren (2012) that when data are only missing in Y and there are no available predictors for Y outside the analysis model, the multiple imputation and the complete case analysis are equivalent. In fact, because complete case is more computationally efficient, in this specific case, it is the preferred method over multiple imputation. If more auxiliary variables outside the analysis model were added to the imputation model, multiple imputation would gain an edge over complete case.

When Y was dichotomous, as long as there were many similar observations, increasing the number of donors improved performance. This is because X and T provided information about the probability that $Y = 1$ and having a large donor pool could better reflect the probability that $Y = 1$ for a given X and T. When Y was

	beta0	se0	beta1	se1	beta2	se2
Truth	5.6878	0.2226	-0.9989	0.0317	0.0011	0.0220
Complete Case	5.9782	0.2513	-0.8454	0.0370	0.0010	0.0248
Bayesian MI	5.9779	0.2584	-0.8451	0.0381	0.0010	0.0255
Bootstrap MI	5.9806	0.2589	-0.8455	0.0377	0.0008	0.0256
PMM (d = 1)	5.9765	0.4566	-0.8463	0.0906	0.0009	0.0447
PMM (d = 3)	5.9701	0.3438	-0.8484	0.0637	0.0015	0.0337
PMM (d = 10)	5.9808	0.2778	-0.8472	0.0472	0.0006	0.0274
PMM (d = 20)	5.9796	0.2616	-0.8470	0.0418	0.0007	0.0258

Table 3.12: Data are MNAR and the proportion missing is 0.3

continuous, X and T provided direct information about the value of Y so increasing the number of donors in this case only increased the likelihood of having a bad match as there is no longer a probability model to account for.

Earlier simulation studies seem to suggest that predictive mean matching should be used when the proportion of observations with missing values is less than 0.5 and the missing data mechanism is not MNAR (Marshall et al., 2010a, 2010b). However, this set of simulation studies provides examples where Bayesian multiple imputation outperforms predictive mean matching.

3.5.1 Limitations and future considerations

There are two main caveats that limit the scope of this study. The first is that the data generated in the set of simulations does not necessarily reflect data that would be seen in a real randomized clinical trial. The continuous outcome simulations were very loosely based on the alcohol intervention trial data, but many of the parameters were still chosen arbitrarily. For the dichotomous outcome simulations, all parameters were chosen arbitrarily. A resampling study using data from a past clinical trial would alleviate this concern and validate the findings.

Secondly, as stated in the methods section, the confidence intervals generated in this study are completely symmetric. According to Heinze, Ploner, & Beyea (2013), under non-normal distributions which occur in logistic regression, the symmetric confidence intervals generated using Rubin's rules may not be valid. This could explain why many of the confidence intervals looked erratic with a dichotomous outcome variable. Creating confidence intervals using profile likelihood could perhaps rectify the potential violation of the normality condition.

There are a few natural extensions to this study. The first would be to generate a continuous Y without using a normal distribution. A log normal distribution is

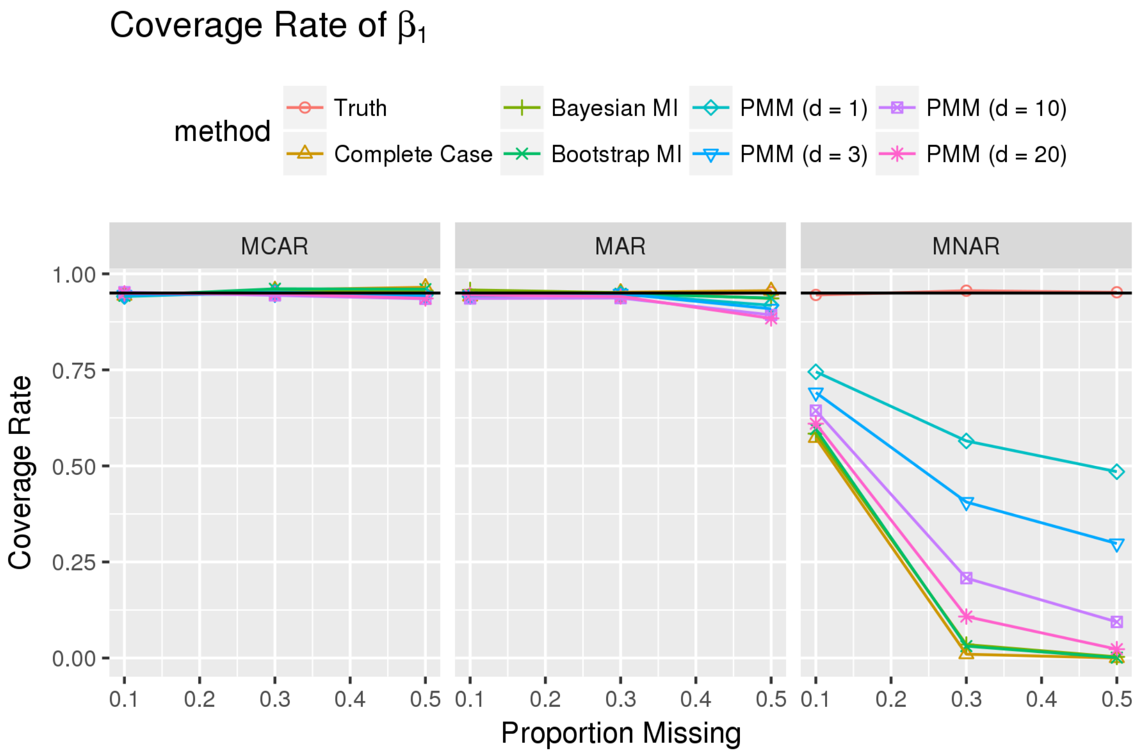

Figure 3.6: Continuous outcome without relationship with auxiliary variable

a prime candidate for this. Such a simulation study would test the robustness of predictive mean matching, Bayesian MI, and bootstrap MI. Another extension would be to include more variables within the study. It could be interesting to see how the methods perform when there is an underlying interaction term. The current simulation study also does not provide any insight when the predictor, not the outcome, is missing. Creating multiple measures study where patients drop out over time would be a way to test this in a clinical trial framework. Lastly, missing data mechanisms are tested only one at a time. It is quite likely that there are multiple reasons as to why data are missing which implies it is likely that there are multiple missing data mechanisms in play. Perhaps implementing multiple mechanisms concurrently or making the data MAR based on an auxiliary variable X (rather than T) could affect the results.

Chapter 4

Conclusion

Performing statistical analysis in the presence of missing data is certainly a daunting task. However, the ramifications of ignoring missing data make attempting to rectify missingness a worthy endeavor. This is especially true in the context of clinical trials, an imperative tool in the medical world. Missing data has and continues to plague and alter the inferences of clinical trials.

In this thesis, we take a look at the missing data problem. The lens of the thesis was not so much the prevention of missing data, but an exploration of what to do when data are inevitably missing. The first half of the thesis was devoted to recovering missing data and sensitivity analysis. We investigated a few simple approaches which are shown to often be inappropriate to the problem at hand. Then attention was turned to multiple imputation, a large improvement over the solutions presented beforehand. Afterwards, we explored two examples highlighting the importance of a principled sensitivity analysis.

The latter half of this thesis presented a simulation study. The simulation design was relatively simple, yet the results of the study uncovered a number of interesting properties. The results of this study demonstrate the potency of Bayesian multiple imputation when there is a missing dichotomous outcome variable. This was a surprise given the results found by Marshall et al. (2010a). In their resampling study, they concluded that multiple imputation via predictive mean matching was the most effective strategy for handling missing data. The study also illustrated how increasing the number of donors can improve predictive mean matching when the outcome variable is dichotomous, but this effect does not hold when the outcome variable is continuous.

In conclusion, missing data is a complex and important issue in statistics especially in the realm of clinical trials. While there is a substantial amount of research concerning missing data methods (e.g. multiple imputation and sensitivity analysis), their implementation in practice has lagged behind. There is no one universal and foolproof panacea to missing data. However, using multiple imputation and sensitivity analysis can certainly go a long way in helping.

Appendix A

Main and background appendix

This first appendix includes all of the R chunks of code in the main Rmd file and the background.Rmd file. The code chunks in the main file were generally to help with readability and setup of the document.

In the main Rmd file:

```
# This chunk ensures that the reedtemplates package is
# installed and loaded. This reedtemplates package includes
# the template files for the thesis and also two functions
# used for labeling and referencing
if(!require(devtools))
  install.packages("devtools", repos = "http://cran.rstudio.com")
```

```
require(mice)
require(mdsr)
require(Ecdat)
require(xtable)
require(tidyr)
require(NHANES)
require(acstats)
options(xtable.comment = FALSE)
```

Code from the background:

Code used for the first example using the alcohol intervention data that compared complete case to a true model.

```
kypri <- read.csv("kypri.csv")
kypri <- kypri %>% select(age, auditCscore, weight, audit1base,
                          audit2base, audit3base, auditCscore,
                          intervention, drink_freqFUP_old,
                          typical_occFUP_old, intervention_group,
                          typical_occFUP, drink_freqFUP)

kypri$intervention_group <- as.factor(kypri$intervention_group)

mod1 <- lm(drink_freqFUP ~ intervention_group + age, data = kypri)
mod1sum <- coef(summary(mod1))[,c("Estimate", "Std. Error")]
xtable(mod1sum,
       caption = "Alcohol Intervention Trial: Model fit with all data",
       label = "KypriTrue")
```

```
set.seed(2017)
# Making variables missing under a MAR mechanism
kypri <- kypri %>% mutate(propMissing = ifelse(age >= 20, .4, 0))
determineMiss <- runif(nrow(kypri))
kypri$determineMiss <- determineMiss
kypri <- kypri %>%
  mutate(r = ifelse(propMissing > determineMiss, 0, 1),
         drink_freqFUP = ifelse(r == 1, drink_freqFUP, NA))

#Fitting Model
mod2 <- lm(drink_freqFUP ~ intervention_group + age, data = kypri)
mod2sum <- coef(summary(mod2))[,c("Estimate", "Std. Error")]

xtable(mod2sum,
       caption = "Alcohol Intervention Trial: Model fit using
       complete case analysis",
       label = "KypriCC")
```

Code used for the second example using the alcohol intervention data that compared complete case and mean imputation to a true model.

```
set.seed(2017)

#Creating misata under a MCAR mechanism
toDelete <- sample(1:nrow(kypri), round(.4 * nrow(kypri)))
kypri$typical_occFUP2 <- kypri$typical_occFUP
kypri$typical_occFUP2[toDelete] <- NA
kypri$typical_occFUP3 <- kypri$typical_occFUP2
kypri$typical_occFUP3[toDelete] <- mean(kypri$typical_occFUP,
                                        na.rm = TRUE)
```

```
kypri_narrow <- kypri %>% select(intervention_group, auditCscore,
                                 Complete_Data = typical_occFUP,
                                 Complete_Case = typical_occFUP2,
                                 Mean_Imputation = typical_occFUP3) %>%
  gather(key = Method, value = typical_occFUP,
         Complete_Case, Mean_Imputation, Complete_Data)

kypri_Plot <- ggplot(data = kypri_narrow,
                     (aes(x=auditCscore, y=typical_occFUP,
                          color = Method))) +
  stat_smooth(method=lm, na.rm = TRUE) +
  geom_point(na.rm = TRUE) +
  labs(title="AUDIT-C Score vs. Typical Occasion Quantity") +
  xlab("AUDIT-C Score") +
  ylab("Typical Occasion Quantity") +
  ylim(0, 25) +
  scale_color_manual(values=c("#59d286", "#3F4145", "#FE6447"))

ggsave("figure/meanimpute.png", plot = kypri_Plot,
       width = 6, height = 4)

label(path = "figure/meanimpute.png",
      caption = "Mean Imputation with data from
      alcohol intervention trial",
      label = "meanimpute", type = "figure",
      scale = 1)
```

```
mod <- lm(typical_occFUP ~ intervention_group + auditCscore,
          data = kypri)
xtable(coef(summary(mod))[,c("Estimate", "Std. Error")],
       caption = "Alcohol Intervention Trial:
       Model fit with complete data",
       label = "KypriTrue2")

ccmod <- lm(typical_occFUP2 ~ intervention_group + auditCscore,
            data = kypri)
xtable(coef(summary(ccmod))[,c("Estimate", "Std. Error")],
       caption = "Alcohol Intervention Trial:
       Model fit with complete case",
       label = "KypriCC2")

meanmod <- lm(typical_occFUP3 ~ intervention_group + auditCscore,
              data = kypri)
xtable(coef(summary(meanmod))[,c("Estimate", "Std. Error")],
       caption = "Alcohol Intervention Trial:
       Model fit with mean imputation",
       label = "KypriMean2")
```

Code used for the examples going through a sensitivity analysis

```
df <- read.csv("df.csv")
ydf <- df %>% mutate(r = ifelse(r==1, "Observed (.8)", "Missing (.2)"),
                     t = ifelse(t==0, "Control", "Treatment")) %>%
  group_by(r, t) %>% summarize(y = mean(y))

total <- df %>% mutate(t = ifelse(t==0, "Control", "Treatment")) %>%
  group_by(t) %>% summarize(y = mean(y))

total <- cbind(r = c("True", "True"), total)
ydf <- rbind(as.data.frame(ydf), total)
ytable <- xtabs(y ~ t + r, ydf)

summarytable <- xtable(ytable, caption = "Summary of MNAR dataset",
                       label = "summaryMNAR1")

align(summarytable) <- "r|rr|r"
summarytable
```

```
true <- lm(y ~ t, data = df)
truesum <- coef(summary(true))["t",c("Estimate", "Std. Error")]

cca <- lm(obsy ~ t, data = df)
ccasum <- coef(summary(cca))["t",c("Estimate", "Std. Error")]

pred <- matrix(c(0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0),
               nrow = 4, ncol = 4, byrow = T)

imp <- mice(df, predictorMatrix = pred, printFlag = FALSE)
mod <- with(imp, lm(obsy ~ t))
info <- summary(pool(mod))[,c("est", "se")]
colnames(info) <- c("Estimate", "Std. Error")
misum <- info["t",]

modsums <- rbind(truesum, ccasum, misum)
rownames(modsums) <- c("True Model", "Complete Case",
                       "Multiple Imputation MAR")
xtable(modsums, caption = "Coefficient of t of models
       fit with MNAR data",
       digits = 2, label = "summaryMNAR2")
```

```
set.seed(6)
post <- imp$post
deltas <- c(0:7)
coefdf <- data.frame()

mod <- lm(y ~ t, data = df)
info <- summary(mod)$coefficients[, c("Estimate", "Std. Error")]
info <- cbind(info, confint.default(mod))
coefdf <- rbind(coefdf, info)
colnames(coefdf) <- c("est", "se", "lo 95", "hi 95")
coefdf <- rownames_to_column(coefdf)

for(k in 1:length(deltas)) {
  d <- deltas[k]
  cmd <- paste("imp[[j]][p$data$t[!r[,j]]==1,i] <-
               imp[[j]][p$data$t[!r[,j]]==1,i] + "
               , d) #command to postprocess imputed values
  post["obsy"] <- cmd #adding command to the post parameter
  imp <- mice(df, predictorMatrix = pred, seed = k * 2017, post = post,
              maxit = 10, printFlag = FALSE)
  mod <- with(imp, lm(obsy ~ t))
  info <- summary(pool(mod))[, c("est", "se", "lo 95", "hi 95")]
  coefdf <- rbind(coefdf,
                  rownames_to_column(as.data.frame(info)))
}

colnames(coefdf) <- c("coefficient", "est", "se", "lo", "hi")
coefdf$model <- rep(c("True", deltas), each = 2)

coefdf$model <- factor(coefdf$model, levels = c("True", 0:7))
beta1coef <- coefdf %>% filter(coefficient == "t")

adjustdelta <- ggplot(data = beta1coef) +
  geom_point(aes(x=model, y=est)) +
  geom_errorbar(aes(ymin = lo, ymax = hi, x = model)) +
  theme(legend.position="none") +
  labs(title = "Sensitivity Analysis using Delta Adjustment") +
  xlab(TeX("Values of $\\delta$")) +
  ylab("Estimate")

ggsave("figure/adjustdelta.png", plot = adjustdelta,
       width = 6, height = 3)
```

```
label(path = "figure/adjustdelta.png",
      caption = "Confidence intervals of the
      coefficient of t using delta adjustment",
      label = "adjust", type = "figure",
      scale = 1, options = "htb")
```

```
label(path = "figure/meanscore.png",
      caption = "Confidence intervals of the
      coefficient of t using the mean score method",
      label = "meanscore", type = "figure",
      scale = .6, options = "htb")
```

Appendix B

Simulation appendix

The second appendix includes all the code used for the simulation study.

B.1 Code used for dichotomous outcome

Below are the functions created for the simulation with dichotomous outcome.

```r
require(mice)
require(mdsr)

generateComplete <- function(numSamp, beta0, beta1,
                             beta2, xmean, xsd,
                             errorsd) {

  x <- rnorm(numSamp, mean = xmean, sd = xsd) #Continuous variable
  t <- c(rep(x = -1, times = numSamp/2), rep(x = 1, times = numSamp/2))

  #Treatment variable. Half assigned to treatment group (1)
  #Half assigned to control group (-1)
  error <- rnorm(numSamp, mean = 0, sd = errorsd) #Error term

  #Relationship between the explanatory variables (x & t)
  #and response variable (y)
  m <- beta0 + beta1 * t + beta2 * x + error

  #Inverse logistic function to get back probability
  sigmam = exp(m)/(1 + exp(m))

  #Randomly determine value based on probability calculated
  y <- (runif(numSamp) < sigmam)

  df <- data.frame(cbind(y = y, x = x, t = t))
  return(df)
}


makeMissing <- function(mechanism, numSamp, propMiss, rr, df) {
  #Regardless of missing data mechanism want to achieve an
  #overall proportion of missing equal to p
  if(mechanism == "mcar") {
    #Idea is to use a logistic regression
    #model to determine missingness
    g0 <- log(propMiss/(1-propMiss))
    g1 <- 0
    g2 <- 0
  }

  else if(mechanism == "mar") {
    z <- 2*propMiss/(1+rr)
    g0 <- log(rr*z^2/((1-z)*(1-rr*z)))/2
    g1 <- log(rr*(1-z)/(1-rr*z))/2
    g2 <- 0
  }

  else if(mechanism == "mnar"){
    w <- prop(~y, df)
    z <- propMiss/(1-w + rr*w)
    g0 <- log(z/(1-z))
    g1 <- 0
    g2 <- log(rr*(1-z)/(1-rr*z))
  }

  else {
    print("Not a valid missing data mechanism")
    return()
  }

  m <- g0 + g1*df$t + g2*df$y
  sigmam <- exp(m) / (1+ exp(m))

  temp <- runif(numSamp)
  r <- ifelse(sigmam > temp, 0, 1)
  df$r <- r
  df <- df %>% mutate(obsy = ifelse(r == 0, NA, y))
  return(df)
}
```

```
generateData <- function(mechanism, propMiss = .1, numSamp = 1000,
                         beta0 = 1, beta1 = 2, beta2 = 1, xmean = 0,
                         xsd = 1, errorsd = .1, rr = 5) {

  df <- generateComplete(numSamp, beta0, beta1, #Creates the data
                         beta2, xmean, xsd,
                         errorsd)

  df <- makeMissing(mechanism = mechanism, numSamp = numSamp, df = df,
                    propMiss = propMiss, rr = rr) #Creates missingness
}

runImputation <- function(df, m = 5, method,
                          interaction = FALSE, donors = 3) {
  if(method == "logreg" || method == "logreg.boot") {
    df$obsy <- as.factor(df$obsy)
  }

  if(interaction == TRUE) {
    df <- df %>% mutate(tx = (t-mean(t)) * (x-mean(x)))
    dummy <- mice(df, method = method, maxit = 0)
    meth <- dummy$meth
    meth["tx"] <- paste("~I(", "(t - mean(t)) * (x - mean(x)))",
                        sep = "")
    pred <- dummy$pred
    pred["obsy",] <- c(0,1,1,0,0,1)
    imp <- mice(df, method = method, m = m, printFlag = FALSE,
                predictorMatrix = pred)
  }

  else {
    pred <- matrix(c(0,0,0,0,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,0,1,1,0,0),
                   nrow = 5, ncol = 5, byrow = T)
    if(method == "pmm") {
      imp <- mice(df, method = method, m = m, printFlag = FALSE,
                  predictorMatrix = pred, donors = donors)
    }
    else {
      imp <- mice(df, method = method, m = m, printFlag = FALSE,
                  predictorMatrix = pred)
    }
  }
  mods <- with(imp, glm(obsy ~ t + x, family = binomial(logit)))
  info <- summary(pool(mods))
  return(as.vector(t(info[, c("est", "se", "lo 95", "hi 95")])))
}


runTrue <- function(df) {
  mod <- glm(y ~ t + x, data = df, family = binomial(logit))
  info <- summary(mod)$coefficients[, c("Estimate", "Std. Error")]
  info <- cbind(info, confint.default(mod))
  return(as.vector(t(info)))
}


runCC <- function(df) {
  mod <- glm(obsy ~ t + x, df, na.action = na.exclude,
             family = binomial(logit))
  info <- summary(mod)$coefficients[, c("Estimate", "Std. Error")]
  info <- cbind(info, confint.default(mod))
  return(as.vector(t(info)))
}


runSimulation <- function(numSim = 1000, m = 5, mechanism = "mcar",
                          propMiss = .1, numSamp = 1000, beta0 = 1,
                          beta1 = 2, beta2 = 1, xmean = 0,
                          xsd = 1, errorsd = .5, rr = 5) {

  tab <- array(NA, dim = c(8, numSim, 12))
  dimnames(tab) <- list(c("Truth", "Complete Case", "Bayesian MI",
                          "Bootstrap MI", "PMM (d = 1)",
                          "PMM (d = 3)", "PMM (d = 10)",
                          "PMM (d = 20)"),
                        as.character(1:numSim),
                        c("beta0", "se0", "lowci0", "highci0",
                          "beta1", "se1", "lowci1", "highci1",
                          "beta2", "se2", "lowci2", "highci2"))

  for (i in 1:numSim){
    df <- generateData(mechanism = mechanism, propMiss = propMiss,
                       numSamp = numSamp, beta0 = beta0, beta1 = beta1,
                       beta2 = beta2, xmean = xmean, xsd = xsd,
                       errorsd = errorsd, rr = rr)
    tab[1, i,] <- runTrue(df = df)
    tab[2, i,] <- runCC(df = df)
    tab[3, i,] <- runImputation(df = df, m = m,
                                method = "logreg")
    tab[4, i,] <- runImputation(df = df, m = m,
                                method = "logreg.boot")
    tab[5, i,] <- runImputation(df = df, m = m,
                                method = "pmm", donors = 1)
    tab[6, i,] <- runImputation(df = df, m = m,
                                method = "pmm", donors = 3)
    tab[7, i,] <- runImputation(df = df, m = m,
                                method = "pmm", donors = 10)
    tab[8, i,] <- runImputation(df = df, m = m,
                                method = "pmm", donors = 20)
  }
  return(tab)
}
```

When $\beta_2 \neq 0$, the simulations were run and saved using code below.

```
sim1 <- runSimulation(numSim = 1000, m = 5, mechanism = "mcar",
                      propMiss = .1, numSamp = 1000, beta0 = 1,
                      beta1 = 2, beta2 = 1, xmean = 0,
                      xsd = 1, errorsd = .1, rr = 5)

sim2 <- runSimulation(numSim = 1000, m = 5, mechanism = "mcar",
                      propMiss = .3, numSamp = 1000, beta0 = 1,
                      beta1 = 2, beta2 = 1, xmean = 0,
                      xsd = 1, errorsd = .1, rr = 5)

sim3 <- runSimulation(numSim = 1000, m = 5, mechanism = "mcar",
                      propMiss = .5, numSamp = 1000, beta0 = 1,
                      beta1 = 2, beta2 = 1, xmean = 0,
                      xsd = 1, errorsd = .1, rr = 5)

sim4 <- runSimulation(numSim = 1000, m = 5, mechanism = "mar",
                      propMiss = .1, numSamp = 1000, beta0 = 1,
                      beta1 = 2, beta2 = 1, xmean = 0,
                      xsd = 1, errorsd = .1, rr = 5)

sim5 <- runSimulation(numSim = 1000, m = 5, mechanism = "mar",
                      propMiss = .3, numSamp = 1000, beta0 = 1,
                      beta1 = 2, beta2 = 1, xmean = 0,
                      xsd = 1, errorsd = .1, rr = 5)
```

```
sim6 <- runSimulation(numSim = 1000, m = 5, mechanism = "mar",
                      propMiss = .5, numSamp = 1000, beta0 = 1,
                      beta1 = 2, beta2 = 1, xmean = 0,
                      xsd = 1, errorsd = .1, rr = 5)

sim7 <- runSimulation(numSim = 1000, m = 5, mechanism = "mnar",
                      propMiss = .1, numSamp = 1000, beta0 = 1,
                      beta1 = 2, beta2 = 1, xmean = 0,
                      xsd = 1, errorsd = .1, rr = 5)

sim8 <- runSimulation(numSim = 1000, m = 5, mechanism = "mnar",
                      propMiss = .3, numSamp = 1000, beta0 = 1,
                      beta1 = 2, beta2 = 1, xmean = 0,
                      xsd = 1, errorsd = .1, rr = 5)

sim9 <- runSimulation(numSim = 1000, m = 5, mechanism = "mnar",
                      propMiss = .5, numSamp = 1000, beta0 = 1,
                      beta1 = 2, beta2 = 1, xmean = 0,
                      xsd = 1, errorsd = .1, rr = 5)

saveRDS(sim1, "binary_sim1.1.rds")
saveRDS(sim2, "binary_sim1.2.rds")
saveRDS(sim3, "binary_sim1.3.rds")
saveRDS(sim4, "binary_sim1.4.rds")
saveRDS(sim5, "binary_sim1.5.rds")
saveRDS(sim6, "binary_sim1.6.rds")
saveRDS(sim7, "binary_sim1.7.rds")
saveRDS(sim8, "binary_sim1.8.rds")
saveRDS(sim9, "binary_sim1.9.rds")
```

When $\beta_2 = 0$, the simulations were run and saved using code below.

```
sim1 <- runSimulation(numSim = 1000, m = 5, mechanism = "mcar",
                      propMiss = .1, numSamp = 1000, beta0 = 1,
                      beta1 = 2, beta2 = 0, xmean = 0,
                      xsd = 1, errorsd = .1, rr = 5)

sim2 <- runSimulation(numSim = 1000, m = 5, mechanism = "mcar",
                      propMiss = .3, numSamp = 1000, beta0 = 1,
                      beta1 = 2, beta2 = 0, xmean = 0,
                      xsd = 1, errorsd = .1, rr = 5)

sim3 <- runSimulation(numSim = 1000, m = 5, mechanism = "mcar",
                      propMiss = .5, numSamp = 1000, beta0 = 1,
                      beta1 = 2, beta2 = 0, xmean = 0,
                      xsd = 1, errorsd = .1, rr = 5)

sim4 <- runSimulation(numSim = 1000, m = 5, mechanism = "mar",
                      propMiss = .1, numSamp = 1000, beta0 = 1,
                      beta1 = 2, beta2 = 0, xmean = 0,
                      xsd = 1, errorsd = .1, rr = 5)

sim5 <- runSimulation(numSim = 1000, m = 5, mechanism = "mar",
                      propMiss = .3, numSamp = 1000, beta0 = 1,
                      beta1 = 2, beta2 = 0, xmean = 0,
                      xsd = 1, errorsd = .1, rr = 5)

sim6 <- runSimulation(numSim = 1000, m = 5, mechanism = "mar",
                      propMiss = .5, numSamp = 1000, beta0 = 1,
                      beta1 = 2, beta2 = 0, xmean = 0,
                      xsd = 1, errorsd = .1, rr = 5)

sim7 <- runSimulation(numSim = 1000, m = 5, mechanism = "mnar",
                      propMiss = .1, numSamp = 1000, beta0 = 1,
                      beta1 = 2, beta2 = 0, xmean = 0,
                      xsd = 1, errorsd = .1, rr = 5)

sim8 <- runSimulation(numSim = 1000, m = 5, mechanism = "mnar",
                      propMiss = .3, numSamp = 1000, beta0 = 1,
                      beta1 = 2, beta2 = 0, xmean = 0,
                      xsd = 1, errorsd = .1, rr = 5)

sim9 <- runSimulation(numSim = 1000, m = 5, mechanism = "mnar",
                      propMiss = .5, numSamp = 1000, beta0 = 1,
                      beta1 = 2, beta2 = 0, xmean = 0,
                      xsd = 1, errorsd = .1, rr = 5)

saveRDS(sim1, "binary_sim2.1.rds")
saveRDS(sim2, "binary_sim2.2.rds")
saveRDS(sim3, "binary_sim2.3.rds")
saveRDS(sim4, "binary_sim2.4.rds")
saveRDS(sim5, "binary_sim2.5.rds")
saveRDS(sim6, "binary_sim2.6.rds")
saveRDS(sim7, "binary_sim2.7.rds")
saveRDS(sim8, "binary_sim2.8.rds")
saveRDS(sim9, "binary_sim2.9.rds")
```

B.2 Code used for continuous outcome

```
require(mice)
require(mdsr)

generateComplete <- function(numSamp, beta0, beta1,
                             beta2, xmean, xsd,
                             errorsd) {

  x <- rnorm(numSamp, mean = xmean, sd = xsd) #Continuous variable
  t <- c(rep(x = -1, times = numSamp/2), rep(x = 1, times = numSamp/2))
  #Treatment variable. Half assigned to treatment group (1)
  #Half assigned to control group (-1)
  error <- rnorm(numSamp, mean = 0, sd = errorsd) #Error term

  y <- beta0 + beta1 * t + beta2 * x + error
  #Relationship between the explanatory variables (x & t)
  #and response variable (y)

  df <- data.frame(cbind(y = y, x = x, t = t))
  return(df)
}


makeMissing <- function(mechanism, numSamp, propMiss,
                        df, rr, beta0, beta1, beta2, xmean) {
  #Regardless of missing data mechanism want to achieve
  #an overall proportion of missing equal to propMiss
  #Idea is to use a logistic regression model
  #to determine missingness

  if(mechanism == "mcar") {
    g0 <- log(propMiss/(1-propMiss))
    g1 <- 0
    g2 <- 0
  }

  else if(mechanism == "mar") {
    z <- 2*propMiss/(1+rr)
    g0 <- log(rr*z^2/((1-z)*(1-rr*z)))/2
    g1 <- log(rr*(1-z)/(1-rr*z))/2
    g2 <- 0
  }
  else if(mechanism == "mnar"){
    z <- propMiss/(.5 + rr*.5)
    g0 <- log(z/(1-z))
    g1 <- 0
    g2 <- log(rr*(1-z)/(1-rr*z))
  }

  else {
    print("Not a valid missing data mechanism")
    return()
  }

  m <- g0 + g1*df$t + g2*((df$y - beta0 + beta1 - beta2*xmean)/
                            (2*beta1))
  sigmam <- exp(m) / (1+ exp(m))

  temp <- runif(numSamp)
  r <- ifelse(sigmam > temp, 0, 1)
  df$r <- r
  df <- df %>% mutate(obsy = ifelse(r == 0, NA, y))
  return(df)
}


generateData <- function(mechanism, propMiss = .1,
                         numSamp = 1000, beta0 = 5, beta1 = -3,
                         beta2 = 2, xmean = 1, xsd = .25,
                         errorsd = 1, rr = 5) {

  df <- generateComplete(numSamp, beta0, beta1, #Creates the data
                         beta2, xmean, xsd,
                         errorsd)

  df <- makeMissing(mechanism = mechanism, numSamp = numSamp,
                    df = df, propMiss = propMiss,
                    rr = rr, beta0 = beta0, beta1 = beta1,
                    beta2 = beta2, xmean = xmean) #Creates missingness
}

runImputation <- function(df, m = 5, method,
                          interaction = FALSE, donors = 3) {
  if(method == "logreg" || method == "logreg.boot") {
    df$obsy <- as.factor(df$obsy)
  }
  if(interaction == TRUE) {
    df <- df %>% mutate(tx = (t-mean(t)) * (x-mean(x)))
    dummy <- mice(df, method = method, maxit = 0)
    meth <- dummy$meth
    meth["tx"] <- paste("~I(", "(t - mean(t)) * (x - mean(x)))",
                        sep = "")
    pred <- dummy$pred
    pred["obsy",] <- c(0,1,1,0,0,1)
    imp <- mice(df, method = method, m = m, printFlag = FALSE,
                predictorMatrix = pred)
  }

  else {
    pred <- matrix(c(0,0,0,0,0,0,0,0,0,0,0,0
                     ,0,0,0,0,0,0,0,0,0,1,1,0,0),
                   nrow = 5, ncol = 5, byrow = T)
    if(method == "pmm") {
      imp <- mice(df, method = method, m = m, printFlag = FALSE,
                  predictorMatrix = pred, donors = donors)
    }
    else {
      imp <- mice(df, method = method, m = m, printFlag = FALSE,
                  predictorMatrix = pred)
    }
  }

  mods <- with(imp, lm(obsy ~ t + x))
  info <- summary(pool(mods))
  return(as.vector(t(info[, c("est", "se", "lo 95", "hi 95")])))
}


runTrue <- function(df) {
  mod <- lm(y ~ t + x, data = df)
  info <- summary(mod)$coefficients[, c("Estimate", "Std. Error")]
  info <- cbind(info, confint.default(mod))
  return(as.vector(t(info)))
}


runCC <- function(df) {
  mod <- lm(obsy ~ t + x, df, na.action = na.exclude)
  info <- summary(mod)$coefficients[, c("Estimate", "Std. Error")]
  info <- cbind(info, confint.default(mod))
  return(as.vector(t(info)))
}


runSimulation <- function(numSim = 1000, m = 5, mechanism = "mcar",
                          propMiss = .1, numSamp = 1000, beta0 = 6,
                          beta1 = -2, beta2 = 1, xmean = 0,
                          xsd = 1, errorsd = .5, rr = 5) {

  tab <- array(NA, dim = c(8, numSim, 12))
  dimnames(tab) <- list(c("Truth", "Complete Case", "Bayesian MI",
                          "Bootstrap MI", "PMM (d = 1)",
                          "PMM (d = 3)", "PMM (d = 10)",
                          "PMM (d = 20)"),
                        as.character(1:numSim),
                        c("beta0", "se0", "lowci0", "highci0",
                          "beta1", "se1", "lowci1", "highci1",
                          "beta2", "se2", "lowci2", "highci2"))

  for (i in 1:numSim){
    df <- generateData(mechanism = mechanism, propMiss = propMiss,
                       numSamp = numSamp, beta0 = beta0, beta1 = beta1,
                       beta2 = beta2, xmean = xmean,
                       xsd = xsd, errorsd = errorsd, rr = rr)
    tab[1, i,] <- runTrue(df = df)
    tab[2, i,] <- runCC(df = df)
    tab[3, i,] <- runImputation(df = df, m = m, method = "norm")
    tab[4, i,] <- runImputation(df = df, m = m, method = "norm.boot")
    tab[5, i,] <- runImputation(df = df, m = m, method = "pmm",
                                donors = 1)
    tab[6, i,] <- runImputation(df = df, m = m, method = "pmm",
                                donors = 3)
    tab[7, i,] <- runImputation(df = df, m = m, method = "pmm",
                                donors = 10)
    tab[8, i,] <- runImputation(df = df, m = m, method = "pmm",
                                donors = 20)
  }
  return(tab)
}
```

When $\beta_2 \neq 0$, the simulations were run and saved using code below.

```
sim1 <- runSimulation(numSim = 1000, m = 5, mechanism = "mcar",
                      propMiss = .1, numSamp = 1000, beta0 = 5.7,
                      beta1 = -1, beta2 = 0.1, xmean = 10,
                      xsd = 1.44, errorsd = 1, rr = 5)

sim2 <- runSimulation(numSim = 1000, m = 5, mechanism = "mcar",
                      propMiss = .3, numSamp = 1000, beta0 = 5.7,
                      beta1 = -1, beta2 = 0.1, xmean = 10,
                      xsd = 1.44, errorsd = 1, rr = 5)

sim3 <- runSimulation(numSim = 1000, m = 5, mechanism = "mcar",
                      propMiss = .5, numSamp = 1000, beta0 = 5.7,
                      beta1 = -1, beta2 = 0.1, xmean = 10,
                      xsd = 1.44, errorsd = 1, rr = 5)

sim4 <- runSimulation(numSim = 1000, m = 5, mechanism = "mar",
                      propMiss = .1, numSamp = 1000, beta0 = 5.7,
                      beta1 = -1, beta2 = 0.1, xmean = 10,
                      xsd = 1.44, errorsd = 1, rr = 5)

sim5 <- runSimulation(numSim = 1000, m = 5, mechanism = "mar",
                      propMiss = .3, numSamp = 1000, beta0 = 5.7,
                      beta1 = -1, beta2 = 0.1, xmean = 10,
                      xsd = 1.44, errorsd = 1, rr = 5)

sim6 <- runSimulation(numSim = 1000, m = 5, mechanism = "mar",
                      propMiss = .5, numSamp = 1000, beta0 = 5.7,
                      beta1 = -1, beta2 = 0.1, xmean = 10,
                      xsd = 1.44, errorsd = 1, rr = 5)

sim7 <- runSimulation(numSim = 1000, m = 5, mechanism = "mnar",
                      propMiss = .1, numSamp = 1000, beta0 = 5.7,
                      beta1 = -1, beta2 = 0.1, xmean = 10,
                      xsd = 1.44, errorsd = 1, rr = 5)

sim8 <- runSimulation(numSim = 1000, m = 5, mechanism = "mnar",
                      propMiss = .3, numSamp = 1000, beta0 = 5.7,
                      beta1 = -1, beta2 = 0.1, xmean = 10,
                      xsd = 1.44, errorsd = 1, rr = 5)

sim9 <- runSimulation(numSim = 1000, m = 5, mechanism = "mnar",
                      propMiss = .5, numSamp = 1000, beta0 = 5.7,
                      beta1 = -1, beta2 = 0.1, xmean = 10,
                      xsd = 1.44, errorsd = 1, rr = 5)

saveRDS(sim1, "continuous_sim1.1.rds")
```

```r
saveRDS(sim2, "continuous_sim1.2.rds")
saveRDS(sim3, "continuous_sim1.3.rds")
saveRDS(sim4, "continuous_sim1.4.rds")
saveRDS(sim5, "continuous_sim1.5.rds")
saveRDS(sim6, "continuous_sim1.6.rds")
saveRDS(sim7, "continuous_sim1.7.rds")
saveRDS(sim8, "continuous_sim1.8.rds")
saveRDS(sim9, "continuous_sim1.9.rds")
```

When $\beta_2 = 0$, the simulations were run and saved using code below.

```r
sim1 <- runSimulation(numSim = 1000, m = 5, mechanism = "mcar",
                      propMiss = .1, numSamp = 1000, beta0 = 5.7,
                      beta1 = -1, beta2 = 0, xmean = 10,
                      xsd = 1.44, errorsd = 1, rr = 5)

sim2 <- runSimulation(numSim = 1000, m = 5, mechanism = "mcar",
                      propMiss = .3, numSamp = 1000, beta0 = 5.7,
                      beta1 = -1, beta2 = 0, xmean = 10,
                      xsd = 1.44, errorsd = 1, rr = 5)

sim3 <- runSimulation(numSim = 1000, m = 5, mechanism = "mcar",
                      propMiss = .5, numSamp = 1000, beta0 = 5.7,
                      beta1 = -1, beta2 = 0, xmean = 10,
                      xsd = 1.44, errorsd = 1, rr = 5)

sim4 <- runSimulation(numSim = 1000, m = 5, mechanism = "mar",
                      propMiss = .1, numSamp = 1000, beta0 = 5.7,
                      beta1 = -1, beta2 = 0, xmean = 10,
                      xsd = 1.44, errorsd = 1, rr = 5)

sim5 <- runSimulation(numSim = 1000, m = 5, mechanism = "mar",
                      propMiss = .3, numSamp = 1000, beta0 = 5.7,
                      beta1 = -1, beta2 = 0, xmean = 10,
                      xsd = 1.44, errorsd = 1, rr = 5)

sim6 <- runSimulation(numSim = 1000, m = 5, mechanism = "mar",
                      propMiss = .5, numSamp = 1000, beta0 = 5.7,
                      beta1 = -1, beta2 = 0, xmean = 10,
                      xsd = 1.44, errorsd = 1, rr = 5)

sim7 <- runSimulation(numSim = 1000, m = 5, mechanism = "mnar",
                      propMiss = .1, numSamp = 1000, beta0 = 5.7,
                      beta1 = -1, beta2 = 0, xmean = 10,
                      xsd = 1.44, errorsd = 1, rr = 5)

sim8 <- runSimulation(numSim = 1000, m = 5, mechanism = "mnar",
                      propMiss = .3, numSamp = 1000, beta0 = 5.7,
                      beta1 = -1, beta2 = 0, xmean = 10,
                      xsd = 1.44, errorsd = 1, rr = 5)

sim9 <- runSimulation(numSim = 1000, m = 5, mechanism = "mnar",
                      propMiss = .5, numSamp = 1000, beta0 = 5.7,
                      beta1 = -1, beta2 = 0, xmean = 10,
                      xsd = 1.44, errorsd = 1, rr = 5)

saveRDS(sim1, "continuous_sim2.1.rds")
saveRDS(sim2, "continuous_sim2.2.rds")
saveRDS(sim3, "continuous_sim2.3.rds")
saveRDS(sim4, "continuous_sim2.4.rds")
saveRDS(sim5, "continuous_sim2.5.rds")
saveRDS(sim6, "continuous_sim2.6.rds")
saveRDS(sim7, "continuous_sim2.7.rds")
saveRDS(sim8, "continuous_sim2.8.rds")
saveRDS(sim9, "continuous_sim2.9.rds")
```

References

Andridge, R. R., & Little, R. J. A. (2010). A review of hot deck imputation for survey non-response. *International Statistical Review*, *78*(1), 40–64.

Barnard, J., & Meng, X.-L. (1999). Applications of multiple imputation in medical studies: From AIDS to NHANES. *Statistical Methods in Medical Research*, *8*(1), 17–36.

Bell, M., Fiero, M., Horton, N., & Hsu, C.-H. (2014). Handling missing data in RCTs; a review of the top medical journals. *BMC Medical Research Methodology*, *14*(1).

Cannon, A. R., Cobb, G. W., Hartlaub, B. A., Legler, J. M., Lock, R. H., Moore, T. L., … Witmer, J. A. (2013). *STAT2: Building models for a world of data*. New York: W.H. Freeman; Company.

Carpenter, J. R., & Kenward, M. G. (2014). *Multiple imputation and its application*. Chichester, United Kingdom: John Wiley & Sons Ltd.

CNStat. (2010). *The prevention and treatment of missing data in clinical trials*. Washington D.C.: National Academies Press.

Collins, L. M., Schafer, J. L., & Kam, C.-M. (2001). A comparison of inclusive and restrictive strategies in modern missing data procedures. *Psychological Methods*, *6*(4), 330–351.

Daniels, M. J., & Hogan, J. W. (2008). *Missing data in longitudinal studies: Strategies for bayesian modeling and sensitivity analysis*. Boca Raton, Florida: CRC Press.

Fay, R. E. (1992). When are inference from multiple imputation valid? *Proceedings-Section on Survey Research Methods American Statistical Association*, 227–232.

Food and Drug Administration, U. (2015). The drug development process. Retrieved from `https://www.fda.gov/ForPatients/Approvals/Drugs/ucm405382.htm`

Food and Drug Administration, U. (2016). What is a biological product? Retrieved from `https://www.fda.gov/aboutfda/transparency/basics/ucm194516.htm`

Graham, J. W., Olchowski, A. E., & Gilreath, T. D. (2007). How many imputations are really needed? Some practical clarifications of multiple imputation theory.

Prevention Science, *8*(3), 206–213.

Groenwold, R. H., White, I. R., Donders, A. R. T., Carpenter, J. R., Altman, D. G., & Moons, K. G. (2012). Missing covariate data in clinical research: When and when not to use the missing-indicator method for analysis. *CMAJ: Canadian Medical Association Journal*, *184*(11), 1265–1269.

Heinze, G., Ploner, M., & Beyea, J. (2013). Confidence intervals after multiple imputation: Combining profile likelihood information from logistic regressions. *Statistics in Medicine*, *32*(29), 5062–5076.

Heitjan, D. F., & Little, R. J. A. (1991). Multiple imputation for the Fatal Accident Reporting System. *Journal of the Royal Statistical Society*, *40*(1), 13–29.

Horton, N. J., & Fitzmaurice, G. M. (2002). Maximum likelihood estimation of bivariate logistic models for incomplete responses with indicators of ignorable and non-ignorable missingness. *Journal of the Royal Statistical Society*, *51*(3), 281–295.

Horton, N. J., & Kleinman, K. P. (2007). Much ado about nothing: A comparison of missing data methods and software to fit incomplete data regression models. *The American Statistician.* American Statistical Association.

Kenward, M. (2013). The prevention and treatment of missing data in clinical trials. *Clinical Investigation*, *3*(3), 241–250.

Kessler, R. C., Sonnega, A., Bromet, E., Hughes, M., & Nelson, C. B. (1995). Posttraumatic stress disorder in the National Comorbidity Survey. *Arch Gen Psychiatry*, *52*(12), 1048–1060.

Kypri, K., Vater, T., Bowe, S. J., Saunders, J. B., Cunningham, J. A., Horton, N. J., & McCambridge, J. (2014). Web-based alcohol screening and brief intervention for university students a randomized trial. *Journal of the American Medical Association*, *311*(12), 1218–1224.

LaVange, L. M., & Permutt, T. (2016). A regulatory perspective on missing data in the aftermath of the NRC report. *Statistics in Medicine*, *35*(17), 2853–2864.

Leacy, F. P., Floyd, S., Yates, T. A., & White, I. R. (2017). Analyses of sensitivity to the missing-at-random assumption using multiple imputation with delta adjustment: Application to a tuberculosis/HIV prevalence survey with incomplete hiv-status data. *American Journal of Epidemiology*, *185*(4), 204–315.

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.

Lesaffre, E. (2008). Superiority, equivalence, and non-inferiority trials. *Bulletin of the NYU Hospital for Joint Diseases*, *66*(2), 150–154.

Little, R. J. A. (1988). Missing-data adjustments in large surveys. *Journal of Business*

& Economic Statistics, *6*(3), 287–296.

Little, R. J. A. (1992). Regression with missing X's. *Journal of the American Statistical Association*, *87*(420), 1227–1237.

Marshall, A., Altman, D. G., & Holder, R. L. (2010a). Comparison of imputation methods for handling missing covariate data when fitting a Cox proportional hazards model: A resampling study. *BMC Medical Research Methedology*, *10*(112).

Marshall, A., Altman, D. G., Royston, P., & Holder, R. L. (2010b). Comparison of techniques for handling missing covariate data within prognostic modelling studies: A simulation study. *BMC Medical Research Methedology*, *10*(7).

Meng, X.-L. (1994). Multiple-imputation inferences with uncongenial sources of input. *Statistical Science*, *9*(4), 538–558.

National Institute of Health. (2017). NIH clinical research trials and you. Retrieved from `https://www.nih.gov/health-information/nih-clinical-research-trials-you/basics#1`

Pepe, M. S., & Reilly, M. (1995). A mean score method for missing and auxiliary covariate data in regression models. *Biometrika*, *82*(2), 299–314.

Pepe, M. S., Reilly, M., & Fleming, T. R. (1994). Auxiliary outcome data and the mean score method. *Journal of Statistical Planning and Inference*, *42*(1–2), 137–160.

Permutt, T. (2016). Sensitivity analysis for missing data in regulatory submissions. *Statistics in Medicine*, *35*(17), 2876–2879.

Raghunathan, T. E., Lepkowski, J. M., Hoewyk, J. V., & Solenberger, P. (2001). A multivariate technique for multiply imputing missing values using a sequence of regression models. *Survey Methedology*, *27*(1), 85–95.

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

Rubin, D. B. (1986). Statistical matching using file concatenation with adjusted weights and multiple imputations. *Journal of Business & Economic Statistics*, *4*(1), 87–94.

Rubin, D. B. (1987). *Multiple imputation for nonresponse in surveys*. New York: John Wiley & Sons Ltd.

Rubin, D. B. (2004). The design of a general and flexible system for handling nonresponse in sample surveys. *The American Statistican*, *58*(4), 298–302.

Saunders, J. B., Aasland, O. G., Babor, T. F., Fuente, J. R. D. L., & Grant, M. (1993). Development of the alcohol use disorders identification test (AUDIT): WHO collaborative project on early detection of persons with harmful alcohol consumption–II. *Addiction*, *88*(6), 791–804.

Schafer, J. L. (1997). *Analysis of incomplete multivariate data.* London, United Kingdom: Chapman & Hall Ltd.

Schafer, J. L., & Graham, J. W. (2002). Missing data: Our view of the state of the art. *Psychological Methods*, 147–177.

Scheuren, F. (2004). History corner: Introduction. *The American Statistician*, *58*(4), 290–291.

Siddique, J., & Belin, T. R. (2008). Multiple imputation using an iterative hot-deck with distance-based donor selection. *Statistics in Medicine*, *27*(1), 83–102.

van Buuren, S. (2012). *Flexible imputation of missing data.* Leiden, The Netherlands: CRC Press.

van Buuren, S., & Groothuis-Oudshoorn, K. (2017). *Mice: Multivariate imputation by chained equations.* Retrieved from `https://cran.r-project.org/web/packages/mice/mice.pdf`

van Buuren, S., Boshuiezen, H., & Knook, D. (1999). Multiple imputation of missing blood pressure covariates in survival analysis. *Statistics in Medicine*, *18*(6), 681–694.

White, I. R., & Thompson, S. G. (2004). Adjusting for partially missing baseline measurements in randomized trials. *Statistics in Medicine*, *24*(7), 993–1007.

White, I. R., Carpenter, J., & Horton, N. J. (2012). Including all individuals is not enough: Lessons for intention-to-treat analysis. *Clinical Trials*, *9*(4), 396–407.

White, I. R., Carpenter, J., & Horton, N. J. (in revision). A mean score method for sensitivity analysis to departures from the missing at random assumption in randomised trials.

White, I. R., Carpenter, J., Evans, S., & Schroter, S. (2007). Eliciting and using expert opinions about dropout bias in randomized controlled trials. *Clinical Trials*, *4*(2), 125–139.

Corrections

When originally submitted, this honors thesis contained some errors which have been corrected in the current version. Here is a list of the errors that were corrected.

Various places in the thesis. Approximately 6 missing periods or commas were added and approximately 19 typos were fixed.

Title page. Changed formatting and added all necessary parts to the title page.

p. 1, l. 19. The phrase "the researchers were unable to collect their data" was changed to "the researchers were unable to collect the relevant data."

p. 4, l. 3. The semicolon was replaced with a period and the word it was replaced with the word randomization

p. 7, l. 32. Removed the word itself from the end of the sentence.

p. 8, l. 7. Removed the word themselves from the end of the sentence.

p. 9-10, 12. Changed 6 instances of complete case to complete case analysis.

p. 10. Simplified the phrase "fit for the purposes of the example" to "fit for this example."

p. 10, l. 27-28. Added "model was" after the word "One" and then added "was" after the word "another."

p. 11, l. 2. Changed font of the word age.

p. 12. In the first paragraph, all verbs in future tense were changed to the past tense.

p. 13, l. 2. The sentence was changed to be in past tense.

p. 13, l. 3. Added the word models after the word these.

p. 15. Revised the sentence "Under most situations, the recommended number of imputed datasets N, is. . ." to "In the past, the recommended number of imputed datasets N, was. . ." Added the sentence "Recently, the number of imputations recommended has been updated to be between 20 and 100 (Graham et al., 2007)."

p. 16, l. 9. Added a hat to the last β.

p. 16. Changed β_0 to 0.

p. 18. Changed "the predictive mean metric of the missing observation of concern" to "zero."

p. 21. In first sentence of the new section, "is as below" was changed to "is:."

p. 23, l. 5. Appended "where T is a treatment variable and X is an auxiliary variable" to the end of the sentence.

p. 25. After the second sentence of the second paragraph, added the sentence "The α value for the hypothesis test was .05, but since there were six outcome variables measured, using a Bonferroni correction, the adjusted *alpha* was $0.008\overline{3}$."

p. 29, l. 16. Changed $\#Patients$ to n.

p. 30. The first sentence of the new section, changed passive voice to active voice. In the same paragraph, changed "are designed" to "were created."

p. 31. Changed $logit(m_i) = \frac{m_i}{1-m_i} = g_0 + g_1 \cdot t_i + g_2 \cdot y_i$ to $logit(m_i) = \log\left(\frac{m_i}{1-m_i}\right) = g_0 + g_1 \cdot t_i + g_2 \cdot y_i$.

p. 33. Added parentheses in the first equation.

p. 34, l. 19. Capitized the p's in the equation of RR.

p. 43. Removed two instances of "again" in the beginning of sentences. Changed "using" to "of" in the last sentence.

p. 49. Changed the phrase "are shown to be inapt solutions to" to "are shown to often be inappropriate to."

p. 72-73. Capitalized 3 proper nouns in the bibliography.

p. 74. Added URL to the `Mice` citation.