Methods for Multiple Imputation by Chained Equations
Accounting for Missingness: A Simulation Study

Nathaniel Wilson
*University of Kentucky*, nathaniel.wilson@uky.edu

Follow this and additional works at: [https://uknowledge.uky.edu/cph_etds](https://uknowledge.uky.edu/cph_etds)

**Part of the** Public Health Commons

**Recommended Citation**
[https://uknowledge.uky.edu/cph_etds/252](https://uknowledge.uky.edu/cph_etds/252)

This Graduate Capstone Project is brought to you for free and open access by the College of Public Health at UKnowledge. It has been accepted for inclusion in Theses and Dissertations--Public Health (M.P.H. & Dr.P.H.) by an authorized administrator of UKnowledge. For more information, please contact UKnowledge@lsv.uky.edu.
STUDENT AGREEMENT:

I represent that my capstone and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained needed written permission statement(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine) which will be submitted to UKnowledge as Additional File.

I hereby grant to The University of Kentucky and its agents the irrevocable, non-exclusive, and royalty-free license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless an embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

REVIEW, APPROVAL AND ACCEPTANCE

The document mentioned above has been reviewed and accepted by the student's advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student's capstone including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Nathaniel Wilson, Student
Dr. Emily Slade, Committee Chair
Dr. Sarah Wackerbarth, Director of Graduate Studies
Methods for Multiple Imputation by Chained Equations Accounting for Missingness: A Simulation Study

Nathaniel Wilson
Masters of Public Health Capstone Project

Advisor: Dr. Emily Slade
Department of Biostatistics
College of Public Health
University of Kentucky
Table of Contents

Abstract.................................................................................................................................1
Introduction ..............................................................................................................................1
Forms of Missingness............................................................................................................2
Imputation Methods and Procedures ..................................................................................2
Simulation Study ..................................................................................................................4
Results .................................................................................................................................6
  Simulation results ................................................................................................................6
  Complete Records Analysis ...............................................................................................7
  Default MICE ......................................................................................................................8
  Y-Interaction MICE ...........................................................................................................8
  All-Interaction MICE .........................................................................................................8
Discussion .............................................................................................................................9
References .............................................................................................................................11
Appendix A: Code for Formulas Method ............................................................................13
Appendix B: Code for Predictor Matrix Method .................................................................26
Appendix C: Code for Graphing .........................................................................................41
Appendix D: Code for Displaying Predictor Matrix ............................................................51
Abstract

Improper treatment of missing data can lead to biased or invalid results. If the data are missing at random (MAR), multiple imputation by chained equations (MICE) is one method utilized to reduce bias. When implementing MICE, the imputation model must be compatible with the final analysis model. We aim to show how to include interaction terms in the imputation model to ensure valid results are obtained. Data were simulated for one continuous outcome originating from two binary predictor variables and their interaction. Missingness was imposed via a MAR mechanism. To handle the missing data, four methods were performed: complete records analysis (CRA) and three variations of MICE, each with different imputation models that vary in their inclusion of interaction effects. We also investigated two different methods for specifying these imputation models in R using different arguments in the mice() function. We utilized a final analysis model consisting of linear regression of the outcome on both main effects of the predictors and their interaction. The analyses performed with MICE including all two-way interactions had the least biased estimates and appropriate coverages. CRA often led to wide confidence intervals which in turn yielded less efficiency. Utilization of the MICE package in R is not entirely intuitive and few resources exist online to assist R users needing to implement MICE in data with interaction effects. There are caveats that must be included when coding for the imputation procedure, and misspecification can lead to inappropriate results.

Introduction

Missing data is a problematic and shared occurrence in many research endeavors, and the most common approach for handling this issue is ignoring it through complete records analysis (CRA) (Kontopantelis, White, Sperrin, & Buchan, 2017; Lee & Carlin, 2011). However, when data are not missing completely at random (MCAR), CRA can lead to biased parameter estimates and inflated standard errors (Kontopantelis et al., 2017; Sullivan, Salter, Ryan, & Lee, 2015; Tilling, Williamson, Spratt, Sterne, & Carpenter, 2016; Wang & Wu, 2011). Standard multiple imputation (MI) is a common procedure applied to missing data to help reduce bias when performing analyses. The process of MI includes imputing missing values a specified number of times, performing the desired analyses on the imputed datasets, and combining the results into a single analysis using Rubin's rules (Rubin, 1987). While this procedure is useful in addressing bias incurred from analyzing missing data, it lacks flexibility in the types of variables it can handle. Multiple imputation by chained equations (MICE), however, allows for more complexities that arise in the data and can handle both continuous and binary variables (Akande, Li, & Reiter, 2017; Sullivan et al., 2015). Additionally, MICE allows for separate, univariate imputations models for each variable to be specified.

When the data are missing at random (MAR), information can be recovered through MICE. To perform multiple imputation by chained equations (MICE), the imputation model needs to be compatible with the analysis model (Collins, Schafer, & Kam, 2001; Tilling et al., 2016). This means all variables and relationships between the variables in the final analysis model must be included in the imputation model.
Forms of Missingness

There are three forms of missingness that data can assume: missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR). Data that is MCAR indicates that there is no relationship between the missingness and observed values, that is, the propensity for a data point to be missing is completely random. MAR means that the tendency for a data point to be missing is related to the data that is observed. When the data are MAR the use of MICE can reduce bias from the missingness (Bhaskaran & Smeeth, 2014). Lastly, data that are MNAR denote a relationship exists between the missingness of the data and the missing values themselves or some other unobserved variable.

Determining the classification of missingness in data can be a difficult task for researchers because it involves a fundamentally untestable assumption (Sterne et al., 2009). In order to determine statistically if data are MAR, a statistician would need to know both the distribution of the observed values and the distribution of the unobserved values. As the latter is impossible to measure (being unobserved) it is the responsibility of the researcher to justify a probable distribution of the unobserved values through literature reviews and discussions with experts (Bhaskaran & Smeeth, 2014; Sterne et al., 2009). After the postulations of the distribution of missing data have been formed, a determination of the type of missingness can be assumed.

When data are MCAR, using either CRA or MI tends to have negligible bias (White & Carlin, 2010); however, when data are MNAR, the underlying relationships between the variables and missingness does not allow for bias correction. For these reasons, data that are MAR will be simulated in the following study. MAR assumes the data can be stratified and that within each stratum the distribution of the unobserved variables is similar to the distribution of observed variables (Bhaskaran & Smeeth, 2014). When this is true, MI can be implemented to help correct for bias that could occur when CRA is used instead.

Imputation Methods and Procedures

Correctly framing the imputation models in MICE is vitally important, and there are many different ways of doing this. The default specification of MICE across programming platforms includes only the main effects in each imputation model. It has been shown that if an interaction is included in the final analysis model, that interaction also needs to be accounted for in the imputation model (Collins et al., 2001; Kenward & Carpenter, 2007; Moons, Donders, Stijnen, & Harrell, 2006; Spratt et al., 2010; Tilling et al., 2016). Simulation studies are exceedingly pragmatic in assessing the value of differing methods; both the access to the full dataset and knowledge of how the missingness was created aid the researcher in better understanding and comparing different processes. In 2016, Tilling et al. demonstrated the need for imputation models to be compatible with analysis models (Tilling et al., 2016). Through a simulation in Stata, the researchers generated missingness in data that mimicked real world missing data situations, specifically data that was MAR They produced five different missing data mechanisms, along with three differing levels of interaction strength for each mechanism. When the appropriate interaction terms were included in the imputation model and compatibility between the imputation model and analysis model was achieved, MICE gave unbiased estimates
and appropriate coverage of the confidence intervals (Tilling et al., 2016). Variables that help estimate incomplete data are frequently referred to as auxiliary variables (Collins et al., 2001). For the imputation model to achieve compatibility with the analysis model, all variables in the analysis model including auxiliary variables must be imputed (Tilling et al., 2016). The results from Tilling et al. (2016) have significant implications for how to correctly manage data with missingness; however, it is unclear how the imputation models should be specified in the presence of interaction effects when using the MICE package in R. One of the aims of the current study is to explore various procedures within the MICE package and determine the most valid approach.

Table 1 describes the various missing data approaches that will be analyzed in the current study and the names we have assigned to each. These procedures include CRA, Default MICE (D-MICE), Y interactions MICE (Y-MICE), and All interactions MICE (A-MICE).

<table>
<thead>
<tr>
<th>Method</th>
<th>Imputation Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRA</td>
<td>N/A</td>
</tr>
<tr>
<td>D-MICE</td>
<td>Y ~ X + Z; X ~ Y + Z; Z ~ X + Y</td>
</tr>
<tr>
<td>Y-MICE</td>
<td>Y ~ X + Z + XZ; where XZ is the interaction between variables X and Z</td>
</tr>
<tr>
<td></td>
<td>X ~ Y + Z; Z ~ X + Y</td>
</tr>
<tr>
<td>A-MICE</td>
<td>Y ~ X + Z + XZ; where XZ is the interaction between variables X and Z</td>
</tr>
<tr>
<td></td>
<td>X ~ Y + Z + YZ; where YZ is the interaction between variables Y and Z</td>
</tr>
<tr>
<td></td>
<td>Z ~ X + Y + XY; where XY is the interaction between variables X and Y</td>
</tr>
</tbody>
</table>

Table 1: Approaches for analysis with missing data

The D-MICE method does not take into consideration any interactions between covariates and each other or covariates and the outcome variable. We include it for comparison because it is the default specification of MICE imputation models in statistical software packages. The Y-MICE method does consider the interactions between the covariates and the outcome variable, as has been previously recommended. A-MICE considers all pairwise interactions.

Passive imputation is often recommended in the MICE algorithm for imputing interaction terms (Tilling et al., 2016; van Buuren & Groothuis-Oudshoorn, 2010). Passive imputation allows the MICE algorithm to utilize the most recent imputations of each main effect to recompute the interaction at each step in the chain. The process of passive imputation is carried out in the mice() function in R through either the predictor matrix or the formulas arguments.

The predictor matrix, made up of 0's and 1's, tells the MICE package which variables are to be included in each imputation model and can be adjusted to match various imputation methods. Depending on the imputation method of choice, the predictor matrix appears in form
as a cross-tabulation of variables (van Buuren & Groothuis-Oudshoorn, 2010). Figures 1a. and 1b. show two examples of how a predictor matrix might be visualized. In addition to the predictor matrix argument, passive imputation of interaction effects requires specification of the methods and blocks arguments as well. The interaction terms in figures 1a and 1b are not being imputed by the predictor matrix specifications, but rather an additional argument in MICE known as methods. For example, specification for these interaction terms in the methods argument may take the following form (when passively imputing):

\[
\text{myMethod } \leftarrow \sim \text{I}(X*Z)
\]

The predictor matrix can be difficult to conceptualize without consistent printing of the data, and when a larger number of variables need to be specified for imputation it is easy to see how keeping track of this matrix of 1's and 0's can be cumbersome. An argument in MICE known as "formulas" is a potential alternative to the traditional predictor matrix argument. To specify how imputations will be carried out, the formulas argument allows the R user to explicitly write each variable's name and the function used to impute them (typically a function of some other covariates in the analysis model). For example, to specify the formula to impute the missing values of y, the following information would be input into the mice() function:

\[
\text{myFormulas}\$y.miss \leftarrow \text{y.miss } \sim \text{x } + \text{z.miss } + \text{x*z.miss}
\]

Saving this formula, along with additional formulas for the other covariates under the object "myFormulas" will give the mice() function instructions on how to impute each individual variable with missing values.

Simulation Study

While knowledge of the MICE package available in R is common among researchers, it is currently unclear which coding procedure is preferable between the predictor matrix argument and the formulas argument for imputing data with interaction effects. Understanding the correct...
method for performing MICE can be a challenge when both of these processes yield different results. The aim of performing this simulation study in R was to (1) assess the functionality of both the predictor matrix and formulas arguments, (2) evaluate the most appropriate method that minimizes bias and has appropriate coverage of regression coefficients, and (3) determine the consequences of inappropriate model matching when specifying both the imputation and analysis models.

We created a simulated dataset that represents real world situations researchers are likely to encounter in data, including missing values in the data. The missingness mechanism was chosen based on likely scenarios that would be encountered in data of this type. Additionally, the selected mechanism represents data that are MAR so multiple imputation can be correctly applied to recover missing information.

Three variables were created within the dataset allowing for the analysis model to take the form of a regression of $Y$ on $X$, $Z$, and their interaction $XZ$. The variables $X$ and $Z$ are both binary with $X \sim \text{Bernoulli}(0.75)$ and the conditional distribution of $Z$ given $X$ is defined by $Z|X = 0 \sim \text{Bernoulli}(0.6)$ and $Z|X = 1 \sim \text{Bernoulli}(0.533)$. The response variable is simulated as:

$$Y = 0.45X + 0.55Z + 0.6XZ + \varepsilon, \quad \varepsilon \sim N(0,1)$$

The missingness mechanism imposed on the generated dataset is:

$$\text{Logit}\{ \Pr(\text{observe } Z) \} = -3 + 1.5X + Y$$
$$\text{Logit}\{ \Pr(\text{observe } Y) \} = -2 + 2.5Z + 2X$$

Additionally, data for $Y$ and $Z$ were first split into two distinct halves with one variable set to be missing within each half of the data to satisfy the MAR assumption.

Once the missingness mechanism was applied to the data, we proceeded with applying the following missing data approaches to determine the preferred method:

1. CRA
2. MICE with no interactions (D-MICE)
3. MICE with the $XZ$ interaction to impute $Y$ (Y-MICE)
4. MICE with all necessary variable interaction terms included in the imputation model (A-MICE)

After imputations were completed, the final analysis model is fit consisting of linear regression of $Y$ on $X$, $Z$, and $XZ$. The results of each method were summarized by calculating mean coefficient estimates, standard deviations of the coefficient estimates, and the estimated coverage of the 95% confidence intervals for each of the methods applied. These results were then compared to the true underlying coefficient from the simulated dataset to assess bias. The comparisons made between the various missing data approaches fall under the assumption that the analysis model is correctly specified to the data generating model. Model misspecification in the final analysis model is outside the scope of this paper.
Results

*Simulation results*

Coefficient estimates from the simulation using the predictor matrix argument can be seen in figures 2a through 2c. Coefficient estimates from the formulas argument can be seen in figures 3a through 3c. Coverage and average 95% confidence interval width (ACIW) for both methods can be seen in Tables 2a through 3c.

![Figure 2a.](image)

![Figure 2b.](image)

![Figure 2c.](image)

Figures 2a. through 2c. show the estimates for the coefficients of X, Z, and the XZ interaction term respectively. The horizontal lines display empirical confidence intervals for each missing data method and the full data analysis. The vertical line in each figure represents the true coefficient for each variable. For X, the true coefficient was 0.45. For Z, the true coefficient was 0.55. Lastly, for XZ interaction, the true coefficient was 0.6.
Complete Records Analysis

As was expected, CRA yielded bias in the coefficient estimates and undercoverage of confidence intervals for X, Z and XZ. For estimates of the coefficient for X, the average 95% confidence interval width (ACIW) was 0.95 (coverage of confidence intervals 92.3%; Table 2a). CRA gave invalid inference and undercoverage for estimates of the coefficient for Z as well (ACIW 1.00, coverage 86.8%; Table 2b). Lastly, for estimates of the coefficient for XZ, undercoverage was again an issue (ACIW 1.04, coverage 92.0%; Table 2c). Additionally, there was loss of efficiency when CRA was implemented (with larger standard deviations compared to the full data analysis).

Figures 3a. through 3c. show the estimates for the coefficients of X, Z, and the XZ interaction term respectively. The horizontal lines display empirical confidence intervals for each missing data method and the full data analysis. The vertical line in each figure represents the true coefficient for each variable. For X, the true coefficient was 0.45. For Z, the true coefficient was 0.55. Lastly, for XZ interaction, the true coefficient was 0.6.
Default MICE

All parameters estimated with D-MICE were biased with poor coverage. In fact, D-MICE was outperformed across all variables by CRA. D-MICE performed especially poorly for the imputation specification of X-Z interaction. As expected, this method leads to underestimation of the XZ interaction due to this effect not being present in the imputation modes (Tables 2a-2c).

Y-Interaction MICE

Y-MICE was also outperformed by CRA on the whole, although the average 95% confidence interval width was smaller compared to CRA. All parameters estimated with Y-MICE were somewhat biased with poor coverage. Y-MICE performed especially poorly for the imputation specification of Z and the X-Z interaction. Once again, the XZ interaction effect is underestimated due to this effect not being fully captured in all of the imputation models. (Tables 2a-2c)

All-Interaction MICE

All-interaction specification of MICE performed as anticipated, surpassing the estimates presented by D-MICE and Y-MICE. There was little bias in the coefficient estimates and high coverage of confidence intervals for X, Z and XZ. For estimates of the coefficient for X, the average 95% confidence interval width was 0.55 (coverage of confidence intervals 94.4%; Table 2a). A-MICE gave valid coverage for estimates of the coefficient for Z as well (ACIW 0.70, coverage 94.2%; Table 2b). Lastly, for estimates of the coefficient for XZ, coverage was again excellent (ACIW 0.73, coverage 94.3%; Table 2c). In the context of average coverage of the confidence intervals, A-MICE performed better than CRA, D-MICE, and Y-MICE. The reported results are based on the values estimated from the predictor matrix process. Comparisons between the predictor matrix and formulas processes indicated predictor matrix performs with better efficiency and coverage overall.

<table>
<thead>
<tr>
<th>Table 2a. Estimates for X</th>
<th>Coverage</th>
<th>Average 95% Confidence Interval Width</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Predictor Matrix</td>
<td>Formulas</td>
</tr>
<tr>
<td>Full Data</td>
<td>95.4%</td>
<td>95.9%</td>
</tr>
<tr>
<td>CRA</td>
<td>92.3%</td>
<td>93.1%</td>
</tr>
<tr>
<td>D-MICE</td>
<td>67.7%</td>
<td>66.7%</td>
</tr>
<tr>
<td>Y-MICE</td>
<td>63.6%</td>
<td>65.0%</td>
</tr>
<tr>
<td>A-MICE</td>
<td>94.4%</td>
<td>89.3%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2b. Estimates for Z</th>
<th>Coverage</th>
<th>Average 95% Confidence Interval Width</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Predictor Matrix</td>
<td>Formulas</td>
</tr>
<tr>
<td>Full Data</td>
<td>95.5%</td>
<td>95.8%</td>
</tr>
<tr>
<td>CRA</td>
<td>86.8%</td>
<td>87.8%</td>
</tr>
<tr>
<td>D-MICE</td>
<td>54.0%</td>
<td>55.3%</td>
</tr>
<tr>
<td>Y-MICE</td>
<td>55.4%</td>
<td>56.0%</td>
</tr>
<tr>
<td>A-MICE</td>
<td>94.2%</td>
<td>89.4%</td>
</tr>
</tbody>
</table>
Discussion

Consistent with the findings from Tilling et al., the imputation model must be compatible with the analysis model to avoid bias during MI (Kenward & Carpenter, 2007; Tilling et al., 2016). We demonstrated the role fully conditional specification has in maintaining efficiency and maximizing coverage. Our imputation models that excluded important effects were biased with undercoverage of the confidence intervals.

The simulation that incurred the least amount of bias while maintaining near 95% coverage was A-MICE. In fact, A-MICE had slightly better coverage than CRA with narrower 95% confidence intervals, demonstrating the better efficiency of A-MICE as compared to CRA. CRA also has the potential to lead to a loss of statistical power when sample sizes are reduced significantly by case-wise deletion methods. Additionally, implementation of MICE without specifying all pairwise interactions in the imputation models will heavily bias results and lead to invalid inference as seen in the D-MICE and Y-MICE methods.

It is intuitive why A-MICE would outperform other methods—the imputation model is developed with the final analysis in mind. The specifications of A-MICE are effectively assuming the associations between Y and X vary within levels of Z and the associations between Y and Z vary within levels of X. It is clear including the XZ interaction term in our imputation model is important while imputing values for Y, as this interaction term is included in the analysis model. However, it is just as important to include both the YZ interaction term and the YX interaction term when imputing X and Z respectively. If we are to assume all variables in our model have some association, failure to include these associations will lead to bias during analysis of our imputed values.

While A-MICE was confirmed to be the best specification of imputation models for data with an interaction effect, there were two ways to specify these models in R that were assessed in this study: predictor matrix argument and formulas argument. Across all applicable imputation methods, simulations specified with the predictor matrix argument had the largest coverage of the confidence interval and the smallest average 95% confidence interval width (indicating the most preserved efficiency). The formulas argument may appear to be the simpler of the two processes to carry out, and the predictor matrix method is not without its challenges; however, correctly specifying the predictor matrix will yield the most accurate and strongest level of inference. We therefore conclude that the formulas argument is not, in fact, an equal alternative to the predictor matrix specification, but rather an inferior method.

<table>
<thead>
<tr>
<th>Table 2c. Estimates for XZ</th>
<th>Coverage</th>
<th>Average 95% Confidence Interval Width</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Predictor Matrix</td>
<td>Formulas</td>
</tr>
<tr>
<td>Full Data</td>
<td>95.6%</td>
<td>95.5%</td>
</tr>
<tr>
<td>CRA</td>
<td>92.0%</td>
<td>92.8%</td>
</tr>
<tr>
<td>D-MICE</td>
<td>55.1%</td>
<td>55.0%</td>
</tr>
<tr>
<td>Y-MICE</td>
<td>53.6%</td>
<td>54.9%</td>
</tr>
<tr>
<td>A-MICE</td>
<td>94.3%</td>
<td>90.7%</td>
</tr>
</tbody>
</table>

Tables 2a - 2c: Results of Coverage of the confidence intervals and average 95% confidence interval widths from the simulations
Additionally, our study is not without limitations. MAR missingness mechanisms can present in a number of ways and this study only examined one possible scenario. The strength of the interaction term in our analysis model was only examined at one level; varying this level in the interaction term may in fact change how poorly or well certain methods performed. Future research will explore a wider range of missingness mechanisms and interaction effect sizes. The model we chose to act as our analysis framework is simplistic as well containing one outcome and two predictors. Real world applications for these methods often involve many more variables.

This study has important implications for users of the mice() package in R. The process of imputing variables in mice() is not the most intuitive, and while the documentation is helpful, there were still questions about which arguments to specify and how to implement them. Results from our simulations show which arguments should be implemented under specific scenarios. Those who implement MI must have a strong understanding of the missingness present in their data before determining if the methods we propose are the solution they need. Lastly, a comprehensive understanding of the relationships between variables in the analysis model is vital to correctly specifying imputations.
References


Appendix A: Code for Formulas Method

### Functions and Libraries ###
library(mice)

expit = function(x){
    return(exp(x)/(1+exp(x)))
}

### Initial Setup for Runs ###
xbeta <- 0.45
zbeta <- 0.55
xzdelta <- 0.6
prx <- 0.75
prz_x0 <- 0.6
prz_x1 <- 0.533
R <- 1000
numsub <- 2000
trials <- 1
coef.fl <- matrix(NA, nrow=R, ncol=3)
SE.fl <- matrix(NA, nrow=R, ncol=3)
lower.fl <- matrix(NA, nrow=R, ncol=3)
upper.fl <- matrix(NA, nrow=R, ncol=3)
coef.cra <- matrix(NA, nrow=R, ncol=3)
SE.cra <- matrix(NA, nrow=R, ncol=3)
lower.cra <- matrix(NA, nrow=R, ncol=3)
upper.cra <- matrix(NA, nrow=R, ncol=3)
coef.st <- matrix(NA, nrow=R, ncol=3)
SE.st <- matrix(NA, nrow=R, ncol=3)
lower.st <- matrix(NA, nrow=R, ncol=3)
upper.st <- matrix(NA, nrow=R, ncol=3)
coef.y <- matrix(NA, nrow=R, ncol=3)
SE.y <- matrix(NA, nrow=R, ncol=3)
lower.y <- matrix(NA, nrow=R, ncol=3)
upper.y <- matrix(NA, nrow=R, ncol=3)
coef.al <- matrix(NA, nrow=R, ncol=3)
SE.al <- matrix(NA, nrow=R, ncol=3)
lower.al <- matrix(NA, nrow=R, ncol=3)
upper.al <- matrix(NA, nrow=R, ncol=3)

### Imputation For Loop Mechanism B ###
for(i in 1:R){
    e <- rnorm(numsub, mean=0, sd=1)

    # Generate x and z
    x <- rbinom(n=numsub, size=trials, prob=prx)
z <- ifelse(x==1, rbinom(n=numsub, size=trials, prob=prz_x1),
          rbinom(n=numsub, size=trials, prob = prz_x0))

## Generate y and full dataframe
y = xbeta*x + zbeta*z + xzdelta*x*z + e
data.full <- data.frame(x,y,z)

## Generate missingness
z.misstop <- z[1:(numsub/2)]
z.missbot <- z[(numsub/2+1):numsub]
z.misstop[rbinom(n=numsub/2, size=trials, prob=expit(-3+1.5*x[1:(numsub/2)])) ==0] <- NA
z.miss <- c(z.misstop, z.missbot)

y.misstop <- y[1:(numsub/2)]
y.missbot <- y[(numsub/2+1):numsub]
y.misstop[rbinom(n=numsub/2, size=trials, prob=expit(-2+2.5*z[((numsub/2)+1):numsub])) ==0] <- NA
y.miss <- c(y.misstop, y.missbot)

## Generate missingness dataset
data.missing <- data.frame(x,y.miss,z.miss)

## Analysis 1 Full data analysis ".fl"
fit.fl = lm(y~x+z+x*z, data=data.full)
est.fl <- summary(fit.fl)
coef.fl[i,1] <- est.fl$coeff[2,1]
coef.fl[i,2] <- est.fl$coeff[3,1]
coef.fl[i,3] <- est.fl$coeff[4,1]

SE.fl[i,1] <- est.fl$coeff[2,2]
SE.fl[i,2] <- est.fl$coeff[3,2]
SE.fl[i,3] <- est.fl$coeff[4,2]

confint.fl = confint(fit.fl)
lower.fl[i,1] <- confint.fl[2,1]
lower.fl[i,2] <- confint.fl[3,1]
lower.fl[i,3] <- confint.fl[4,1]

upper.fl[i,1] <- confint.fl[2,2]
upper.fl[i,2] <- confint.fl[3,2]
upper.fl[i,3] <- confint.fl[4,2]

## Analysis 2 CRA ".cra"
fit.cra <- with(data.missing, lm(y.miss~x+z.miss+x*z.miss))
est.cra <- summary(fit.cra)

coef.cra[i,1] <- est.cra$coeff[2,1]
coef.cra[i,2] <- est.cra$coeff[3,1]
coef.cra[i,3] <- est.cra$coeff[4,1]

SE.cra[i,1] <- est.cra$coeff[2,2]
SE.cra[i,2] <- est.cra$coeff[3,2]
SE.cra[i,3] <- est.cra$coeff[4,2]

confint.cra = confint(fit.cra)
lower.cra[i,1] <- confint.cra[2,1]
lower.cra[i,2] <- confint.cra[3,1]
lower.cra[i,3] <- confint.cra[4,1]

upper.cra[i,1] <- confint.cra[2,2]
upper.cra[i,2] <- confint.cra[3,2]
upper.cra[i,3] <- confint.cra[4,2]

## Analysis 3 Standard No interaction ".st"
imp.st <- mice(data.missing, print=FALSE, m=5, maxit=10)
fit.st <- with(imp.st, lm(y.miss~x*z.miss))
est.st <- pool(fit.st)

coef.st[i,1] <- est.st$pooled$estimate[2]
coef.st[i,2] <- est.st$pooled$estimate[3]
coef.st[i,3] <- est.st$pooled$estimate[4]

SE.st[i,1] <- summary(est.st)[2,2]
SE.st[i,2] <- summary(est.st)[3,2]
SE.st[i,3] <- summary(est.st)[4,2]

lower.st[i,1] <- summary(est.st, conf.int = T)[2,6]
lower.st[i,2] <- summary(est.st, conf.int = T)[3,6]
lower.st[i,3] <- summary(est.st, conf.int = T)[4,6]

upper.st[i,1] <- summary(est.st, conf.int = T)[2,7]
upper.st[i,2] <- summary(est.st, conf.int = T)[3,7]
upper.st[i,3] <- summary(est.st, conf.int = T)[4,7]

## Analysis 4 y interactions ".y"
ini <- mice(data.missing, maxit=0, print=FALSE)
myformulas <-ini$formulas
myformulas$x <- x ~ y.miss + z.miss
myformulas$y.miss <- y.miss ~ x + z.miss + x*z.miss
myformulas$z.miss <- z.miss ~ x + y.miss

imp.y <- mice(data.missing, print = FALSE, m = 5, maxit = 10, formulas = myformulas)

fit.y <- with(imp.y, lm(y.miss~x*z.miss))
est.y <- pool(fit.y)

coef.y[i,1] <- est.y$pooled$estimate[2]
coef.y[i,2] <- est.y$pooled$estimate[3]
coef.y[i,3] <- est.y$pooled$estimate[4]

SE.y[i,1] <- summary(est.y)[2,2]
SE.y[i,2] <- summary(est.y)[3,2]
SE.y[i,3] <- summary(est.y)[4,2]

lower.y[i,1] <- summary(est.y, conf.int = T)[2,6]
lower.y[i,2] <- summary(est.y, conf.int = T)[3,6]
lower.y[i,3] <- summary(est.y, conf.int = T)[4,6]

upper.y[i,1] <- summary(est.y, conf.int = T)[2,7]
upper.y[i,2] <- summary(est.y, conf.int = T)[3,7]
upper.y[i,3] <- summary(est.y, conf.int = T)[4,7]

## Analysis 5 All Interactions ".al"
ini <- mice(data.missing, maxit=0, print=FALSE)

myformulas <-ini$formulas

myformulas$x <- x ~ y.miss + z.miss + y.miss*z.miss
myformulas$y.miss <- y.miss ~ x + z.miss + x*z.miss
myformulas$z.miss <- z.miss ~ x + y.miss + x*y.miss

imp.al <- mice(data.missing, print = FALSE, m = 5, maxit = 10, formulas = myformulas)

fit.al <- with(imp.al, lm(y.miss~x*z.miss))
est.al <- pool(fit.al)

coef.al[i,1] <- est.al$pooled$estimate[2]
coef.al[i,2] <- est.al$pooled$estimate[3]
coef.al[i,3] <- est.al$pooled$estimate[4]

SE.al[i,1] <- summary(est.al)[2,2]
SE.al[i,2] <- summary(est.al)[3,2]
SE.al[i,3] <- summary(est.al)[4,2]

lower.al[i,1] <- summary(est.al, conf.int = T)[2,6]
lower.al[i,2] <- summary(est.al, conf.int = T)[3,6]
lower.al[i,3] <- summary(est.al, conf.int = T)[4,6]

upper.al[i,1] <- summary(est.al, conf.int = T)[2,7]
upper.al[i,2] <- summary(est.al, conf.int = T)[3,7]
upper.al[i,3] <- summary(est.al, conf.int = T)[4,7]

if(i%%(R/20)==0) print(paste0((i/R)*100,"%"))
}

### Saving imputed Mice objects ###

setwd(<insert working directory here>)

save(coef.fl, file = "coef.fl.RData")
save(SE.fl, file = "SE.fl.RData")
save(lower.fl, file = "lower.fl.RData")
save(upper.fl, file = "upper.fl.RData")
save(coef.cra, file = "coef.cra.RData")
save(SE.cra, file = "SE.cra.RData")
save(lower.cra, file = "lower.cra.RData")
save(upper.cra, file = "upper.cra.RData")
save(coef.st, file = "coef.st.RData")
save(SE.st, file = "SE.st.RData")
save(lower.st, file = "lower.st.RData")
save(upper.st, file = "upper.st.RData")
save(coef.y, file = "coef.y.RData")
save(SE.y, file = "SE.y.RData")
save(lower.y, file = "lower.y.RData")
save(upper.y, file = "upper.y.RData")
save(coef.al, file = "coef.al.RData")
save(SE.al, file = "SE.al.RData")
save(lower.al, file = "lower.al.RData")
save(upper.al, file = "upper.al.RData")

### ###
### Imputation For Loop Mechanism C ###
for(i in 1:R){
  e <- rnorm(numsub, mean=0, sd=1)
  # Generate x and z
  x <- rbinom(n=numsub, size=trials, prob=prx)
  z <- ifelse(x==1, rbinom(n=numsub, size=trials, prob=prz_x1),
              rbinom(n=numsub, size=trials, prob=prz_x0))
## Generate y and full dataframe
\[ y = \text{xbeta} \times x + \text{zbeta} \times z + \text{xzdelta} \times x \times z + e \]
data.full <- data.frame(x, y, z)

## Generate missingness
z.misstop <- z[1:(numsub/2)]
z.missbot <- z[(numsub/2+1):numsub]
z.misstop[rbinom(n=numsub/2, size=trials, prob=expit(-3+1.5\times x[1:(numsub/2)] + y[1:(numsub/2)])) == 0] <- NA
z.miss <- c(z.misstop, z.missbot)
y.misstop <- y[1:(numsub/2)]
y.missbot <- y[(numsub/2+1):numsub]
y.misstop[rbinom(n=numsub/2, size=trials, prob=expit(-2+2.5\times z[((numsub/2)+1):numsub] + 2\times x[((numsub/2)+1):numsub])) == 0] <- NA
y.miss <- c(y.misstop, y.missbot)

## Generate missingness dataset
data.missing <- data.frame(x, y.miss, z.miss)

## Analysis 1 Full data analysis ".fl"
fit.fl = lm(y~x+z+x*z, data=data.full)
est.fl <- summary(fit.fl)
coef.fl[i,1] <- est.fl$coeff[2,1]
coef.fl[i,2] <- est.fl$coeff[3,1]
coef.fl[i,3] <- est.fl$coeff[4,1]
SE.fl[i,1] <- est.fl$coeff[2,2]
SE.fl[i,2] <- est.fl$coeff[3,2]
SE.fl[i,3] <- est.fl$coeff[4,2]
confint.fl = confint(fit.fl)
lower.fl[i,1] <- confint.fl[2,1]
lower.fl[i,2] <- confint.fl[3,1]
lower.fl[i,3] <- confint.fl[4,1]
upper.fl[i,1] <- confint.fl[2,2]
upper.fl[i,2] <- confint.fl[3,2]
upper.fl[i,3] <- confint.fl[4,2]

## Analysis 2 CRA ".cra"
fit.cra <- with(data.missing, lm(y.miss~x+z.miss+x*z.miss))
est.cra <- summary(fit.cra)

coef.cra[i,1] <- est.cra$coeff[2,1]
coef.cra[i,2] <- est.cra$coeff[3,1]
coef.cra[i,3] <- est.cra$coeff[4,1]

SE.cra[i,1] <- est.cra$coeff[2,2]
SE.cra[i,2] <- est.cra$coeff[3,2]
SE.cra[i,3] <- est.cra$coeff[4,2]

confint.cra = confint(fit.cra)
lower.cra[i,1] <- confint.cra[2,1]
lower.cra[i,2] <- confint.cra[3,1]
lower.cra[i,3] <- confint.cra[4,1]

upper.cra[i,1] <- confint.cra[2,2]
upper.cra[i,2] <- confint.cra[3,2]
upper.cra[i,3] <- confint.cra[4,2]

## Analysis 3 Standard No interaction ".st"
imp.st <- mice(data.missing, print=FALSE, m=5, maxit=10)
fit.st <- with(imp.st, lm(y.miss~x*z.miss))
est.st <- pool(fit.st)

coef.st[i,1] <- est.st$pooled$estimate[2]
coef.st[i,2] <- est.st$pooled$estimate[3]
coef.st[i,3] <- est.st$pooled$estimate[4]

SE.st[i,1] <- summary(est.st)[2,2]
SE.st[i,2] <- summary(est.st)[3,2]
SE.st[i,3] <- summary(est.st)[4,2]

lower.st[i,1] <- summary(est.st, conf.int = T)[2,6]
lower.st[i,2] <- summary(est.st, conf.int = T)[3,6]
lower.st[i,3] <- summary(est.st, conf.int = T)[4,6]

upper.st[i,1] <- summary(est.st, conf.int = T)[2,7]
upper.st[i,2] <- summary(est.st, conf.int = T)[3,7]
upper.st[i,3] <- summary(est.st, conf.int = T)[4,7]

## Analysis 4 y interactions ".y"
ini <- mice(data.missing, maxit=0, print=FALSE)
myformulas <- ini$formulas
myformulas$x <- x ~ y.miss + z.miss
myformulas$y.miss <- y.miss ~ x + z.miss + x*z.miss
myformulas$z.miss <- z.miss ~ x + y.miss

imp.y <- mice(data.missing, print = FALSE, m = 5, maxit = 10, formulas = myformulas)  # method = myMethod, predictorMatrix=myPredMat

fit.y <- with(imp.y, lm(y.miss~x*z.miss))
est.y <- pool(fit.y)

coef.y[i, 1] <- est.y$pooled$estimate[2]
coef.y[i, 2] <- est.y$pooled$estimate[3]
coef.y[i, 3] <- est.y$pooled$estimate[4]

SE.y[i, 1] <- summary(est.y)[2, 2]
SE.y[i, 2] <- summary(est.y)[3, 2]
SE.y[i, 3] <- summary(est.y)[4, 2]

lower.y[i, 1] <- summary(est.y, conf.int = T)[2, 6]
lower.y[i, 2] <- summary(est.y, conf.int = T)[3, 6]
lower.y[i, 3] <- summary(est.y, conf.int = T)[4, 6]

upper.y[i, 1] <- summary(est.y, conf.int = T)[2, 7]
upper.y[i, 2] <- summary(est.y, conf.int = T)[3, 7]
upper.y[i, 3] <- summary(est.y, conf.int = T)[4, 7]

## Analysis 5 All Interactions ".al"
ini <- mice(data.missing, maxit=0, print=FALSE)

myformulas <- ini$formulas

myformulas$x <- x ~ y.miss + z.miss + y.miss*z.miss
myformulas$y.miss <- y.miss ~ x + z.miss + x*z.miss
myformulas$z.miss <- z.miss ~ x + y.miss + x*y.miss

imp.al <- mice(data.missing, print = FALSE, m = 5, maxit = 10, formulas = myformulas)  # method = myMethod, predictorMatrix=myPredMat

fit.al <- with(imp.al, lm(y.miss~x*z.miss))
est.al <- pool(fit.al)

coef.al[i, 1] <- est.al$pooled$estimate[2]
coef.al[i, 2] <- est.al$pooled$estimate[3]
coef.al[i, 3] <- est.al$pooled$estimate[4]
SE.al[i,1] <- summary(est.al)[2,2]
SE.al[i,2] <- summary(est.al)[3,2]
SE.al[i,3] <- summary(est.al)[4,2]

lower.al[i,1] <- summary(est.al, conf.int = T)[2,6]
lower.al[i,2] <- summary(est.al, conf.int = T)[3,6]
lower.al[i,3] <- summary(est.al, conf.int = T)[4,6]

upper.al[i,1] <- summary(est.al, conf.int = T)[2,7]
upper.al[i,2] <- summary(est.al, conf.int = T)[3,7]
upper.al[i,3] <- summary(est.al, conf.int = T)[4,7]

if(i%%(R/20)==0) print(paste0((i/R)*100, "\%"))
}

### Saving imputed Mice objects; R = 1000 ###

setwd(<insert working directory here>)

save(coef.fl, file = "coef.fl.RData")
save(SE.fl, file = "SE.fl.RData")
save(lower.fl, file = "lower.fl.RData")
save(upper.fl, file = "upper.fl.RData")
save(coef.cra, file = "coef.cra.RData")
save(SE.cra, file = "SE.cra.RData")
save(lower.cra, file = "lower.cra.RData")
save(upper.cra, file = "upper.cra.RData")
save(coef.st, file = "coef.st.RData")
save(SE.st, file = "SE.st.RData")
save(lower.st, file = "lower.st.RData")
save(upper.st, file = "upper.st.RData")
save(coef.y, file = "coef.y.RData")
save(SE.y, file = "SE.y.RData")
save(lower.y, file = "lower.y.RData")
save(upper.y, file = "upper.y.RData")
save(coef.al, file = "coef.al.RData")
save(SE.al, file = "SE.al.RData")
save(lower.al, file = "lower.al.RData")
save(upper.al, file = "upper.al.RData")

### Imputation For Loop Mechanism D ###
for(i in 1:R){
  e <- rnorm(numsub, mean=0, sd=1)
  # Generate x and z
```r
x <- rbinom(n=numsub, size=trials, prob=prx)
z <- ifelse(x==1, rbinom(n=numsub, size=trials, prob=prz_x1), 
          rbinom(n=numsub, size=trials, prob=prz_x0))

## Generate y and full dataframe
y = x*beta + z*beta + x*z*delta + e
data.full <- data.frame(x,y,z)

## Generate missingness
z.misstop <- z[1:(numsub/2)]
z.missbot <- z[(numsub/2+1):numsub]
z.misstop[rbinom(n=numsub/2, size=trials, prob=expit(-3 + 
          1.5*x[1:(numsub/2)] + y[1:(numsub/2)])) == 0] <- NA
z.miss <- c(z.misstop, z.missbot)
x.misstop <- x[1:(numsub/2)]
x.missbot <- x[(numsub/2+1):numsub]
x.missbot[rbinom(n=numsub/2, size=trials, prob=expit(-2.5 + 
          1.3*z[((numsub/2)+1):numsub] + 0.8*y[1:(numsub/2)])) == 0] <- NA
x.miss <- c(x.misstop, x.missbot)

## Generate missingness dataset
data.missing <- data.frame(y,x.miss,z.miss)

## Analysis 1 Full data analysis ".fl"
fit.fl = lm(y~x+z+x*z, data=data.full)
est.fl <- summary(fit.fl)

coef.fl[i,1] <- est.fl$coeff[2,1]
coef.fl[i,2] <- est.fl$coeff[3,1]
coef.fl[i,3] <- est.fl$coeff[4,1]

SE.fl[i,1] <- est.fl$coeff[2,2]
SE.fl[i,2] <- est.fl$coeff[3,2]
SE.fl[i,3] <- est.fl$coeff[4,2]

confint.fl = confint(fit.fl)
lower.fl[i,1] <- confint.fl[2,1]
lower.fl[i,2] <- confint.fl[3,1]
lower.fl[i,3] <- confint.fl[4,1]

upper.fl[i,1] <- confint.fl[2,2]
upper.fl[i,2] <- confint.fl[3,2]
```

22
upper.fl[i,3] <- confint.fl[4,2]

## Analysis 2 CRA ".cra"
fit.cra <- with(data.missing,
  lm(y~x.miss+z.miss+x.miss*z.miss))
est.cra <- summary(fit.cra)

coef.cra[i,1] <- est.cra$coeff[2,1]
coef.cra[i,2] <- est.cra$coeff[3,1]
coef.cra[i,3] <- est.cra$coeff[4,1]

SE.cra[i,1] <- est.cra$coeff[2,2]
SE.cra[i,2] <- est.cra$coeff[3,2]
SE.cra[i,3] <- est.cra$coeff[4,2]

confint.cra = confint(fit.cra)
lower.cra[i,1] <- confint.cra[2,1]
lower.cra[i,2] <- confint.cra[3,1]
lower.cra[i,3] <- confint.cra[4,1]

upper.cra[i,1] <- confint.cra[2,2]
upper.cra[i,2] <- confint.cra[3,2]
upper.cra[i,3] <- confint.cra[4,2]

## Analysis 3 Standard No interaction ".st"
imp.st <- mice(data.missing, print=FALSE, m=5, maxit=10)
fit.st <- with(imp.st, lm(y~x.miss*z.miss))
est.st <- pool(fit.st)

coef.st[i,1] <- est.st$pooled$estimate[2]
coef.st[i,2] <- est.st$pooled$estimate[3]
coef.st[i,3] <- est.st$pooled$estimate[4]

SE.st[i,1] <- summary(est.st)[2,2]
SE.st[i,2] <- summary(est.st)[3,2]
SE.st[i,3] <- summary(est.st)[4,2]

lower.st[i,1] <- summary(est.st, conf.int = T)[2,6]
lower.st[i,2] <- summary(est.st, conf.int = T)[3,6]
lower.st[i,3] <- summary(est.st, conf.int = T)[4,6]

upper.st[i,1] <- summary(est.st, conf.int = T)[2,7]
upper.st[i,2] <- summary(est.st, conf.int = T)[3,7]
upper.st[i,3] <- summary(est.st, conf.int = T)[4,7]

## Analysis 4 y interactions ".y"
ini <- mice(data.missing, maxit=0, print=FALSE)

myformulas <- ini$formulas

myformulas$x.miss <- x.miss ~ y + z.miss
myformulas$y <- y ~ x.miss + z.miss + x.miss*z.miss
myformulas$z.miss <- z.miss ~ x.miss + y

imp.y <- mice(data.missing, print = FALSE, m = 5, maxit = 10, formulas = myformulas)
fit.y <- with(imp.y, lm(y~x.miss*z.miss))
est.y <- pool(fit.y)

coef.y[i,1] <- est.y$poold$estimate[2]
coef.y[i,2] <- est.y$poold$estimate[3]
coef.y[i,3] <- est.y$poold$estimate[4]

SE.y[i,1] <- summary(est.y)[2,2]
SE.y[i,2] <- summary(est.y)[3,2]
SE.y[i,3] <- summary(est.y)[4,2]

lower.y[i,1] <- summary(est.y, conf.int = T)[2,6]
lower.y[i,2] <- summary(est.y, conf.int = T)[3,6]
lower.y[i,3] <- summary(est.y, conf.int = T)[4,6]

upper.y[i,1] <- summary(est.y, conf.int = T)[2,7]
upper.y[i,2] <- summary(est.y, conf.int = T)[3,7]
upper.y[i,3] <- summary(est.y, conf.int = T)[4,7]

## Analysis 5 All Interactions ".al"
ini <- mice(data.missing, maxit=0, print=FALSE)

myformulas <- ini$formulas

myformulas$x.miss <- x.miss ~ y + z.miss + y*z.miss
myformulas$y <- y ~ x.miss + z.miss + x.miss*z.miss
myformulas$z.miss <- z.miss ~ x.miss + y + x.miss*y

imp.al <- mice(data.missing, print = FALSE, m = 5, maxit = 10, formulas = myformulas)

fit.al <- with(imp.al, lm(y~x.miss*z.miss))
est.al <- pool(fit.al)

coef.al[i,1] <- est.al$poold$estimate[2]
coef.al[i,2] <- est.al$poold$estimate[3]
coef.al[i,3] <- est.al$pooled$estimate[4]

SE.al[i,1] <- summary(est.al)[2,2]
SE.al[i,2] <- summary(est.al)[3,2]
SE.al[i,3] <- summary(est.al)[4,2]

lower.al[i,1] <- summary(est.al, conf.int = T)[2,6]
lower.al[i,2] <- summary(est.al, conf.int = T)[3,6]
lower.al[i,3] <- summary(est.al, conf.int = T)[4,6]

upper.al[i,1] <- summary(est.al, conf.int = T)[2,7]
upper.al[i,2] <- summary(est.al, conf.int = T)[3,7]
upper.al[i,3] <- summary(est.al, conf.int = T)[4,7]

if(i%%(R/20)==0) print(paste0((i/R)*100, "%"))
}

### Saving imputed MICE objects ###

setwd(<insert working directory here>)

save(coef.fl, file = "coef.fl.RData")
save(SE.fl, file = "SE.fl.RData")
save(lower.fl, file = "lower.fl.RData")
save(upper.fl, file = "upper.fl.RData")
save(coef.cra, file = "coef.cra.RData")
save(SE.cra, file = "SE.cra.RData")
save(lower.cra, file = "lower.cra.RData")
save(upper.cra, file = "upper.cra.RData")
save(coef.st, file = "coef.st.RData")
save(SE.st, file = "SE.st.RData")
save(lower.st, file = "lower.st.RData")
save(upper.st, file = "upper.st.RData")
save(coef.y, file = "coef.y.RData")
save(SE.y, file = "SE.y.RData")
save(lower.y, file = "lower.y.RData")
save(upper.y, file = "upper.y.RData")
save(coef.al, file = "coef.al.RData")
save(SE.al, file = "SE.al.RData")
save(lower.al, file = "lower.al.RData")
save(upper.al, file = "upper.al.RData")
Appendix B: Code for Predictor Matrix Method

### Functions and Libraries ###
library(mice)

expit = function(x){
  return(exp(x)/(1+exp(x))}
}

CIx = function(est){
  return(c(summary(est)[2,1] - 1.96*summary(est)[2,2],
          summary(est)[2,1] + 1.96*summary(est)[2,2]))
}

CIz = function(est){
  return(c(summary(est)[3,1] - 1.96*summary(est)[3,2],
          summary(est)[3,1] + 1.96*summary(est)[3,2]))
}

CIxz = function(est){
  return(c(summary(est)[4,1] - 1.96*summary(est)[4,2],
          summary(est)[4,1] + 1.96*summary(est)[4,2]))
}

covp = function(coverage){
  return(mean(coverage))
}

### Initial Setup for Runs ###
xbeta <- 0.45
zbeta <- 0.55
xzdelta <- 0.6
prx <- 0.75
prz_x0 <- 0.6
prz_x1 <- 0.533
R <- 1000
numsub <- 2000
trials <- 1
coef.fl <- matrix(NA, nrow=R, ncol=3)
SE.fl <- matrix(NA, nrow=R, ncol=3)
lower.fl <- matrix(NA, nrow=R, ncol=3)
upper.fl <- matrix(NA, nrow=R, ncol=3)
coef.cra <- matrix(NA, nrow=R, ncol=3)
SE.cra <- matrix(NA, nrow=R, ncol=3)
lower.cra <- matrix(NA, nrow=R, ncol=3)
upper.cra <- matrix(NA, nrow=R, ncol=3)
coef.st <- matrix(NA, nrow=R, ncol=3)
SE.st <- matrix(NA, nrow=R, ncol=3)
lower.st <- matrix(NA, nrow=R, ncol=3)
upper.st <- matrix(NA, nrow=R, ncol=3)
coef.y <- matrix(NA, nrow=R, ncol=3)
SE.y <- matrix(NA, nrow=R, ncol=3)
lower.y <- matrix(NA, nrow=R, ncol=3)
upper.y <- matrix(NA, nrow=R, ncol=3)
coef.al <- matrix(NA, nrow=R, ncol=3)
SE.al <- matrix(NA, nrow=R, ncol=3)
lower.al <- matrix(NA, nrow=R, ncol=3)
upper.al <- matrix(NA, nrow=R, ncol=3)

#### Before moving on, select Missingness Mechanism and Begin in that section ####
#### Imputation For Loop Mechanism B ####
for(i in 1:R){
e <- rnorm(numsub, mean=0, sd=1)
  ## Generate x and z
  x <- rbinom(n=numsub, size=trials, prob=prx)
  z <- ifelse(x==1, rbinom(n=numsub, size=trials, prob=prz_x1),
              rbinom(n=numsub, size=trials, prob=prz_x0))

  ## Generate y and full dataframe
  y = xbeta*x + zbeta*z + xzdelta*x*z + e
  data.full <- data.frame(x,y,z)

  ## Generate missingness
  z.misstop <- z[1:(numsub/2)]
  z.missbot <- z[(numsub/2+1):numsub]
  z.misstop[rbinom(n=numsub/2, size=trials, prob=expit(-
             3+1.5*x[1:(numsub/2)])) ==0] <- NA
  z.miss <- c(z.misstop, z.missbot)

  y.misstop <- y[1:(numsub/2)]
  y.missbot <- y[(numsub/2+1):numsub]
  y.missbot[rbinom(n=numsub/2, size=trials, prob=expit(-
             2+2.5*z[((numsub/2)+1):numsub])) ==0] <- NA
  y.miss <- c(y.misstop, y.missbot)

  ## Generate missingness dataset
  data.missing <- data.frame(x,y.miss,z.miss)

  ## Analysis 1 Full data analysis ".fl"
  fit.fl = lm(y~x+z+x*z, data=data.full)
  #summary(fit.fl)
est.fl <- summary(fit.fl)

coef.fl[i,1] <- est.fl$coeff[2,1]
coef.fl[i,2] <- est.fl$coeff[3,1]
coef.fl[i,3] <- est.fl$coeff[4,1]

SE.fl[i,1] <- est.fl$coeff[2,2]
SE.fl[i,2] <- est.fl$coeff[3,2]
SE.fl[i,3] <- est.fl$coeff[4,2]

confint.fl = confint(fit.fl)
lower.fl[i,1] <- confint.fl[2,1]
lower.fl[i,2] <- confint.fl[3,1]
lower.fl[i,3] <- confint.fl[4,1]

upper.fl[i,1] <- confint.fl[2,2]
upper.fl[i,2] <- confint.fl[3,2]
upper.fl[i,3] <- confint.fl[4,2]

## Analysis 2 CRA ".cra"
fit.cra <- with(data.missing, lm(y.miss~x+z.miss+x*z.miss))
est.cra <- summary(fit.cra)

coef.cra[i,1] <- est.cra$coeff[2,1]
coef.cra[i,2] <- est.cra$coeff[3,1]
coef.cra[i,3] <- est.cra$coeff[4,1]

SE.cra[i,1] <- est.cra$coeff[2,2]
SE.cra[i,2] <- est.cra$coeff[3,2]
SE.cra[i,3] <- est.cra$coeff[4,2]

confint.cra = confint(fit.cra)
lower.cra[i,1] <- confint.cra[2,1]
lower.cra[i,2] <- confint.cra[3,1]
lower.cra[i,3] <- confint.cra[4,1]

upper.cra[i,1] <- confint.cra[2,2]
upper.cra[i,2] <- confint.cra[3,2]
upper.cra[i,3] <- confint.cra[4,2]

## Analysis 3 Standard No interaction ".st"
imp.st <- mice(data.missing, print=FALSE, m=5, maxit=10)
fit.st <- with(imp.st, lm(y.miss~x*z.miss))
est.st <- pool(fit.st)
coef.st[i,1] <- est.st$pooled$estimate[2]
coef.st[i,2] <- est.st$pooled$estimate[3]
coef.st[i,3] <- est.st$pooled$estimate[4]

SE.st[i,1] <- summary(est.st)[2,2]
SE.st[i,2] <- summary(est.st)[3,2]
SE.st[i,3] <- summary(est.st)[4,2]

lower.st[i,1] <- summary(est.st, conf.int = T)[2,6]
lower.st[i,2] <- summary(est.st, conf.int = T)[3,6]
lower.st[i,3] <- summary(est.st, conf.int = T)[4,6]

upper.st[i,1] <- summary(est.st, conf.int = T)[2,7]
upper.st[i,2] <- summary(est.st, conf.int = T)[3,7]
upper.st[i,3] <- summary(est.st, conf.int = T)[4,7]

## Analysis 4 y interactions ".y"
data.missing.int = data.frame(data.missing, data.missing$x*data.missing$z.miss)
colnames(data.missing.int)[4] = "xzint"
ini <- mice(data.missing.int, maxit=0, print=FALSE)
myMethod = ini$method
myMethod[4] = "-I(x*z.miss)"
myPredMat = ini$predictorMatrix
myPredMat[2,] = c(1,0,1,1)
myPredMat[3,] = c(1,1,0,0)
myPredMat[c(1,4),] = c(0,0,0,0)

imp.y <- mice(data.missing.int, print = FALSE, m =5, maxit = 10, method = myMethod, predictorMatrix=myPredMat)
fit.y <- with(imp.y, lm(y.miss~x*z.miss))
est.y <- pool(fit.y)

coef.y[i,1] <- est.y$pooled$estimate[2]
coef.y[i,2] <- est.y$pooled$estimate[3]
coef.y[i,3] <- est.y$pooled$estimate[4]

SE.y[i,1] <- summary(est.y)[2,2]
SE.y[i,2] <- summary(est.y)[3,2]
SE.y[i,3] <- summary(est.y)[4,2]

lower.y[i,1] <- summary(est.y, conf.int = T)[2,6]
lower.y[i,2] <- summary(est.y, conf.int = T)[3,6]
lower.y[i,3] <- summary(est.y, conf.int = T)[4,6]

upper.y[i,1] <- summary(est.y, conf.int = T)[2,7]
upper.y[i,2] <- summary(est.y, conf.int = T)[3,7]
upper.y[i,3] <- summary(est.y, conf.int = T)[4,7]

## Analysis 5 All Interactions ".al"
data.missing.int = data.frame(data.missing,
data.missing$x*data.missing$z.miss,
data.missing$y.miss*data.missing$z.miss,
data.missing$x*data.missing$y.miss)
colnames(data.missing.int)[4:6] = c("xzint", "yzint", "xyint")
ini <- mice(data.missing.int, maxit=0, print=FALSE)
myMethod = ini$method
myMethod[4:6] = c("~I(x*z.miss)", "~I(y.miss*z.miss)", "~I(x*y.miss)")

myPredMat = ini$predictorMatrix
myPredMat[2,4] = 1
myPredMat[2,5] = 0
myPredMat[3,4:5] = 0
myPredMat[3,6] = 1
myPredMat

myVisSeq = c("x","y.miss","yzint","xyint")
ini$blocks
myBlocks <-ini$blocks
myBlocks$x <- c("x","xzint", "xyint")
myBlocks$y.miss <- c("y.miss", "xyint", "yzint")
myBlocks$z.miss <- c("z.miss", "xzint", "yzint")

imp.al <- mice(data.missing.int, print = FALSE, m =5, maxit = 10, method = myMethod, predictorMatrix = myPredMat, blocks = myBlocks)

fit.al <- with(imp.al, lm(y.miss~x*z.miss))
est.al <- pool(fit.al)

coef.al[i,1] <- est.al$pooled$estimate[2]
coef.al[i,2] <- est.al$pooled$estimate[3]
coef.al[i,3] <- est.al$pooled$estimate[4]

SE.al[i,1] <- summary(est.al)[2,2]
SE.al[i,2] <- summary(est.al)[3,2]
SE.al[i,3] <- summary(est.al)[4,2]

lower.al[i,1] <- summary(est.al, conf.int = T)[2,6]
lower.al[i,2] <- summary(est.al, conf.int = T)[3,6]
lower.al[i,3] <- summary(est.al, conf.int = T)[4,6]
upper.al[i,1] <- summary(est.al, conf.int = T)[2,7]
upper.al[i,2] <- summary(est.al, conf.int = T)[3,7]
upper.al[i,3] <- summary(est.al, conf.int = T)[4,7]

if(i%%(R/20)==0) print(paste0((i/R)*100,"\%"))
}

### Saving imputed Mice objects 1000 ###

setwd(<insert working directory here>)

save(coef.fl, file = "coef.fl.RData")
save(SE.fl, file = "SE.fl.RData")
save(lower.fl, file = "lower.fl.RData")
save(upper.fl, file = "upper.fl.RData")
save(coef.cra, file = "coef.cra.RData")
save(SE.cra, file = "SE.cra.RData")
save(lower.cra, file = "lower.cra.RData")
save(upper.cra, file = "upper.cra.RData")
save(coef.st, file = "coef.st.RData")
save(SE.st, file = "SE.st.RData")
save(lower.st, file = "lower.st.RData")
save(upper.st, file = "upper.st.RData")
save(coef.y, file = "coef.y.RData")
save(SE.y, file = "SE.y.RData")
save(lower.y, file = "lower.y.RData")
save(upper.y, file = "upper.y.RData")
save(coef.al, file = "coef.al.RData")
save(SE.al, file = "SE.al.RData")
save(lower.al, file = "lower.al.RData")
save(upper.al, file = "upper.al.RData")

### ####
### Imputation For Loop Mechanism C ####

for(i in 1:R){
e <- rnorm(numsub, mean=0, sd=1)

# Generate x and z
x <- rbinom(n=numsub, size=trials, prob=prx)
z <- ifelse(x==1, rbinom(n=numsub, size=trials, prob=prz_x1),
rbinom(n=numsub, size=trials, prob=prz_x0))

# Generate y and full dataframe
y = xbeta*x + zbeta*z + xzdelta*x*z + e
data.full <- data.frame(x,y,z)
## Generate missingness

```r
z.misstop <- z[1:(numsub/2)]
z.missbot <- z[(numsub/2+1):numsub]
z.misstop[rbinom(n=numsub/2, size=trials, prob=expit(-3+1.5*x[1:(numsub/2)] + y[1:(numsub/2)])) == 0] <- NA
z.miss <- c(z.misstop, z.missbot)
```

```r
y.misstop <- y[1:(numsub/2)]
y.missbot <- y[(numsub/2+1):numsub]
y.missbot[rbinom(n=numsub/2, size=trials, prob=expit(-2+2.5*z[((numsub/2)+1):numsub] + 2*x[((numsub/2)+1):numsub])) == 0] <- NA
y.miss <- c(y.misstop, y.missbot)
```

## Generate missingness dataset

```r
data.missing <- data.frame(x, y.miss, z.miss)
```

## Analysis 1 Full data analysis ".fl"

```r
fit.fl = lm(y~x+z+x*z, data=data.full)
#summary(fit.fl)
est.fl <- summary(fit.fl)
```

```r
coef.fl[i,1] <- est.fl$coeff[2,1]
coef.fl[i,2] <- est.fl$coeff[3,1]
coef.fl[i,3] <- est.fl$coeff[4,1]
```

```r
SE.fl[i,1] <- est.fl$coeff[2,2]
SE.fl[i,2] <- est.fl$coeff[3,2]
SE.fl[i,3] <- est.fl$coeff[4,2]
```

```r
confint.fl = confint(fit.fl)
lower.fl[i,1] <- confint.fl[2,1]
lower.fl[i,2] <- confint.fl[3,1]
lower.fl[i,3] <- confint.fl[4,1]
```

```r
upper.fl[i,1] <- confint.fl[2,2]
upper.fl[i,2] <- confint.fl[3,2]
upper.fl[i,3] <- confint.fl[4,2]
```

## Analysis 2 CRA ".cra"

```r
fit.cra <- with(data.missing, lm(y.miss~x+z.miss+x*z.miss))
est.cra <- summary(fit.cra)
```
coef.cra[i,1] <- est.cra$coeff[2,1]
coef.cra[i,2] <- est.cra$coeff[3,1]
coef.cra[i,3] <- est.cra$coeff[4,1]

SE.cra[i,1] <- est.cra$coeff[2,2]
SE.cra[i,2] <- est.cra$coeff[3,2]
SE.cra[i,3] <- est.cra$coeff[4,2]

confint.cra = confint(fit.cra)
lower.cra[i,1] <- confint.cra[2,1]
lower.cra[i,2] <- confint.cra[3,1]
lower.cra[i,3] <- confint.cra[4,1]

upper.cra[i,1] <- confint.cra[2,2]
upper.cra[i,2] <- confint.cra[3,2]
upper.cra[i,3] <- confint.cra[4,2]

## Analysis 3 Standard No interaction ".st"
imp.st <- mice(data.missing, print=FALSE, m=5, maxit=10)
fit.st <- with(imp.st, lm(y.miss~x*z.miss))
est.st <- pool(fit.st)

coef.st[i,1] <- est.st$pooled$estimate[2]
coef.st[i,2] <- est.st$pooled$estimate[3]
coef.st[i,3] <- est.st$pooled$estimate[4]

SE.st[i,1] <- summary(est.st)[2,2]
SE.st[i,2] <- summary(est.st)[3,2]
SE.st[i,3] <- summary(est.st)[4,2]

lower.st[i,1] <- summary(est.st, conf.int = T)[2,6]
lower.st[i,2] <- summary(est.st, conf.int = T)[3,6]
lower.st[i,3] <- summary(est.st, conf.int = T)[4,6]

upper.st[i,1] <- summary(est.st, conf.int = T)[2,7]
upper.st[i,2] <- summary(est.st, conf.int = T)[3,7]
upper.st[i,3] <- summary(est.st, conf.int = T)[4,7]

## Analysis 4 y interactions ".y"
data.missing.int = data.frame(data.missing, 
data.missing$x*data.missing$z.miss)
colnames(data.missing.int)[4] = "xzint"
ini <- mice(data.missing.int, maxit=0, print=FALSE)
myMethod = ini$method
myMethod[4] = "-I(x*z.miss)"
myPredMat = ini$predictorMatrix
myPredMat[2,] = c(1,0,1,1)
myPredMat[3,] = c(1,1,0,0)
myPredMat[c(1,4),] = c(0,0,0,0)

imp.y <- mice(data.missing.int, print = FALSE, m = 5, maxit = 10, method = myMethod, predictorMatrix=myPredMat)
fit.y <- with(imp.y, lm(y.miss~x*z.miss))
est.y <- pool(fit.y)

coef.y[i,1] <- est.y$pooled$estimate[2]
coef.y[i,2] <- est.y$pooled$estimate[3]
coef.y[i,3] <- est.y$pooled$estimate[4]

SE.y[i,1] <- summary(est.y)[2,2]
SE.y[i,2] <- summary(est.y)[3,2]
SE.y[i,3] <- summary(est.y)[4,2]

lower.y[i,1] <- summary(est.y, conf.int = T)[2,6]
lower.y[i,2] <- summary(est.y, conf.int = T)[3,6]
lower.y[i,3] <- summary(est.y, conf.int = T)[4,6]

upper.y[i,1] <- summary(est.y, conf.int = T)[2,7]
upper.y[i,2] <- summary(est.y, conf.int = T)[3,7]
upper.y[i,3] <- summary(est.y, conf.int = T)[4,7]

## Analysis 5 All Interactions ".al"
data.missing.int = data.frame(data.missing, data.missing$x*data.missing$z.miss, data.missing$y.miss*data.missing$z.miss, data.missing$x*data.missing$y.miss)
colnames(data.missing.int)[4:6] = c("xzint", "yzint", "xyint")
ini <- mice(data.missing.int, maxit=0, print=FALSE)
myMethod = ini$method
myMethod[4:6] = c("~I(x*z.miss)", "~I(y.miss*z.miss)", "~I(x*y.miss)")

myPredMat = ini$predictorMatrix
myPredMat[2,4] = 1
myPredMat[2,5] = 0
myPredMat[3,4:5] = 0
myPredMat[3,6] = 1
myPredMat[1,] = 0
myPredMat[2,6] = 0
myPredMat[4:6,] = 0
myPredMat

myVisSeq = c("x","y.miss","yzint","xyint")
ini$blocks
myBlocks <- ini$blocks
myBlocks$x <- c("x", "xzint", "xyint")
myBlocks$y.miss <- c("y.miss", "xyint", "yzint")
myBlocks$z.miss <- c("z.miss", "xzint", "yzint")

imp.al <- mice(data.missing.int, print = FALSE, m = 5, maxit = 10, method = myMethod, predictorMatrix = myPredMat, blocks = myBlocks)

fit.al <- with(imp.al, lm(y.miss ~ x*z.miss))
est.al <- pool(fit.al)

coef.al[i,1] <- est.al$pooled$estimate[2]
coef.al[i,2] <- est.al$pooled$estimate[3]
coef.al[i,3] <- est.al$pooled$estimate[4]

SE.al[i,1] <- summary(est.al)[2,2]
SE.al[i,2] <- summary(est.al)[3,2]
SE.al[i,3] <- summary(est.al)[4,2]

lower.al[i,1] <- summary(est.al, conf.int = T)[2,6]
lower.al[i,2] <- summary(est.al, conf.int = T)[3,6]
lower.al[i,3] <- summary(est.al, conf.int = T)[4,6]

upper.al[i,1] <- summary(est.al, conf.int = T)[2,7]
upper.al[i,2] <- summary(est.al, conf.int = T)[3,7]
upper.al[i,3] <- summary(est.al, conf.int = T)[4,7]

if(i%%(R/20)==0) print(paste0((i/R)*100,"%"))
}

#### Saving imputed Mice objects; R = 1000 ####
setwd(<insert working directory here>)
save(coef.fl, file = "coef.fl.RData")
save(SE.fl, file = "SE.fl.RData")
save(lower.fl, file = "lower.fl.RData")
save(upper.fl, file = "upper.fl.RData")
save(coef.cra, file = "coef.cra.RData")
save(SE.cra, file = "SE.cra.RData")
save(lower.cra, file = "lower.cra.RData")
save(upper.cra, file = "upper.cra.RData")
save(coef.st, file = "coef.st.RData")
save(SE.st, file = "SE.st.RData")
save(lower.st, file = "lower.st.RData")
save(upper.st, file = "upper.st.RData")
save(coef.y, file = "coef.y.RData")
save(SE.y, file = "SE.y.RData")
save(lower.y, file = "lower.y.RData")
save(upper.y, file = "upper.y.RData")
save(coef.al, file = "coef.al.RData")
save(SE.al, file = "SE.al.RData")
save(lower.al, file = "lower.al.RData")
save(upper.al, file = "upper.al.RData")

### Imputation For Loop Mechanism D ###
for(i in 1:R){
e <- rnorm(numsub, mean=0, sd=1)

  # Generate x and z
  x <- rbinom(n=numsub, size=trials, prob=prx)
  z <- ifelse(x==1, rbinom(n=numsub, size=trials, prob=prz_x1),
                rbinom(n=numsub, size=trials, prob=prz_x0))

  # Generate y and full dataframe
  y = xbeta*x + zbeta*z + xzdelta*x*z + e
  data.full <- data.frame(x, y, z)

  # Generate missingness
  z.misstop <- z[1:(numsub/2)]
  z.missbot <- z[(numsub/2+1):numsub]
  z.misstop[rbinom(n=numsub/2, size=trials, prob=expit(-3 + 1.5*x[1:(numsub/2)] + y[1:(numsub/2)])) == 0] <- NA
  z.miss <- c(z.misstop, z.missbot)

  x.misstop <- x[1:(numsub/2)]
  x.missbot <- x[(numsub/2+1):numsub]
  x.missbot[rbinom(n=numsub/2, size=trials, prob=expit(-2.5 + 1.3*z[((numsub/2)+1):numsub] + 0.8*y[((numsub/2)+1):numsub])) == 0] <- NA
  x.miss <- c(x.misstop, x.missbot)

  # Generate missingness dataset
  data.missing <- data.frame(y, x.miss, z.miss)

  # Analysis 1 Full data analysis ",.fl"
  fit.fl = lm(y~x+z+x*z, data=data.full)
est.fl <- summary(fit.fl)

coef.fl[i,1] <- est.fl$coeff[2,1]
coef.fl[i,2] <- est.fl$coeff[3,1]
coef.fl[i,3] <- est.fl$coeff[4,1]

SE.fl[i,1] <- est.fl$coeff[2,2]
SE.fl[i,2] <- est.fl$coeff[3,2]
SE.fl[i,3] <- est.fl$coeff[4,2]

confint.fl = confint(fit.fl)
lower.fl[i,1] <- confint.fl[2,1]
lower.fl[i,2] <- confint.fl[3,1]
lower.fl[i,3] <- confint.fl[4,1]

upper.fl[i,1] <- confint.fl[2,2]
upper.fl[i,2] <- confint.fl[3,2]
upper.fl[i,3] <- confint.fl[4,2]

## Analysis 2 CRA ".cra"
fit.cra <- with(data.missing, lm(y~x.miss+z.miss+x.miss*z.miss))

est.cra <- summary(fit.cra)

coef.cra[i,1] <- est.cra$coeff[2,1]
coef.cra[i,2] <- est.cra$coeff[3,1]
coef.cra[i,3] <- est.cra$coeff[4,1]

SE.cra[i,1] <- est.cra$coeff[2,2]
SE.cra[i,2] <- est.cra$coeff[3,2]
SE.cra[i,3] <- est.cra$coeff[4,2]

confint.cra = confint(fit.cra)
lower.cra[i,1] <- confint.cra[2,1]
lower.cra[i,2] <- confint.cra[3,1]
lower.cra[i,3] <- confint.cra[4,1]

upper.cra[i,1] <- confint.cra[2,2]
upper.cra[i,2] <- confint.cra[3,2]
upper.cra[i,3] <- confint.cra[4,2]

## Analysis 3 Standard No interaction ".st"
imp.st <- mice(data.missing, print=FALSE, m=5, maxit=10)
fit.st <- with(imp.st, lm(y~x.miss*z.miss))
est.st <- pool(fit.st)
```r
coef.st[i,1] <- est.st$pooled$estimate[2]
coef.st[i,2] <- est.st$pooled$estimate[3]
coef.st[i,3] <- est.st$pooled$estimate[4]

SE.st[i,1] <- summary(est.st)[2,2]
SE.st[i,2] <- summary(est.st)[3,2]
SE.st[i,3] <- summary(est.st)[4,2]

lower.st[i,1] <- summary(est.st, conf.int = T)[2,6]
lower.st[i,2] <- summary(est.st, conf.int = T)[3,6]
lower.st[i,3] <- summary(est.st, conf.int = T)[4,6]

upper.st[i,1] <- summary(est.st, conf.int = T)[2,7]
upper.st[i,2] <- summary(est.st, conf.int = T)[3,7]
upper.st[i,3] <- summary(est.st, conf.int = T)[4,7]

## Analysis 4 y interactions ".y"
data.missing.int = data.frame(data.missing,
data.missing$x.miss*data.missing$z.miss)
colnames(data.missing.int)[4] = "xzint"
ini <- mice(data.missing.int, maxit=0, print=FALSE)
myMethod = ini$method
myMethod[4] = "~I(x.miss*z.miss)"
myPredMat = ini$predictorMatrix
myPredMat[2,] = c(1,0,1,1)
myPredMat[3,] = c(1,1,0,0)
myPredMat[c(1,4),] = c(0,0,0,0)

imp.y <- mice(data.missing.int, print = FALSE, m =5, maxit = 10, method = myMethod, predictorMatrix=myPredMat)
fit.y <- with(imp.y, lm(y~x.miss*z.miss))
est.y <- pool(fit.y)

coef.y[i,1] <- est.y$pooled$estimate[2]
coef.y[i,2] <- est.y$pooled$estimate[3]
coef.y[i,3] <- est.y$pooled$estimate[4]

SE.y[i,1] <- summary(est.y)[2,2]
SE.y[i,2] <- summary(est.y)[3,2]
SE.y[i,3] <- summary(est.y)[4,2]

lower.y[i,1] <- summary(est.y, conf.int = T)[2,6]
lower.y[i,2] <- summary(est.y, conf.int = T)[3,6]
lower.y[i,3] <- summary(est.y, conf.int = T)[4,6]

upper.y[i,1] <- summary(est.y, conf.int = T)[2,7]
upper.y[i,2] <- summary(est.y, conf.int = T)[3,7]
```
upper.y[i,3] <- summary(est.y, conf.int = T)[4,7]

## Analysis 5 All Interactions ".al"
data.missing.int = data.frame(data.missing,
data.missing$x.miss*data.missing$z.miss,
data.missing$y*data.missing$z.miss,
data.missing$x.miss*data.missing$y)
colnames(data.missing.int)[4:6] = c("xzint", "yzint", "xyint")
i <- mice(data.missing.int, maxit=0, print=FALSE)
myMethod = ini$method
myMethod[4:6] = c("~I(x.miss*z.miss)", "~I(y*z.miss)", "~I(x.miss*y)"

myPredMat = ini$predictorMatrix
myPredMat[2,4] = 1
myPredMat[2,5] = 0
myPredMat[3,4:5] = 0
myPredMat[3,6] = 1
myPredMat

myVisSeq = c("x.miss","y","yzint","xyint")
ini$blocks
myBlocks <-ini$blocks
myBlocks$x.miss <- c("x.miss", "xzint", "xyint")
myBlocks$y <- c("y", "xyint", "yzint")
myBlocks$z.miss <- c("z.miss", "xzint", "yzint")

imp.al <- mice(data.missing.int, print = FALSE, m =5, maxit = 10, method = myMethod, predictorMatrix = myPredMat, blocks = myBlocks)

fit.al <- with(imp.al, lm(y~x.miss*z.miss))
est.al <- pool(fit.al)
coef.al[i,1] <- est.al$pooled$estimate[2]
coef.al[i,2] <- est.al$pooled$estimate[3]
coef.al[i,3] <- est.al$pooled$estimate[4]

SE.al[i,1] <- summary(est.al)[2,2]
SE.al[i,2] <- summary(est.al)[3,2]
SE.al[i,3] <- summary(est.al)[4,2]

lower.al[i,1] <- summary(est.al, conf.int = T)[2,6]
lower.al[i,2] <- summary(est.al, conf.int = T)[3,6]
lower.al[i,3] <- summary(est.al, conf.int = T)[4,6]
upper.al[i,1] <- summary(est.al, conf.int = T)[2,7]
upper.al[i,2] <- summary(est.al, conf.int = T)[3,7]
upper.al[i,3] <- summary(est.al, conf.int = T)[4,7]

if(i%%(R/20)==0) print(paste0((i/R)*100,"%"))
}

#### Saving imputed Mice objects ####

setwd(<insert working directory here>)
	save(coef.fl, file = "coef.fl.RData")
save(SE.fl, file = "SE.fl.RData")
save(lower.fl, file = "lower.fl.RData")
save(upper.fl, file = "upper.fl.RData")
save(coef.cra, file = "coef.cra.RData")
save(SE.cra, file = "SE.cra.RData")
save(lower.cra, file = "lower.cra.RData")
save(upper.cra, file = "upper.cra.RData")
save(coef.st, file = "coef.st.RData")
save(SE.st, file = "SE.st.RData")
save(lower.st, file = "lower.st.RData")
save(upper.st, file = "upper.st.RData")
save(coef.y, file = "coef.y.RData")
save(SE.y, file = "SE.y.RData")
save(lower.y, file = "lower.y.RData")
save(upper.y, file = "upper.y.RData")
save(coef.al, file = "coef.al.RData")
save(SE.al, file = "SE.al.RData")
save(lower.al, file = "lower.al.RData")
save(upper.al, file = "upper.al.RData")
Appendix C: Code for Graphing

```r
require(ggplot2)

setwd(<insert working directory here>)

## Load all data objects
load("coef.al.RData")
load("coef.y.RData")
load("coef.st.RData")
load("coef.cra.RData")
load("coef.fl.RData")
load("SE.al.RData")
load("SE.y.RData")
load("SE.st.RData")
load("SE.cra.RData")
load("SE.fl.RData")
load("lower.al.RData")
load("lower.y.RData")
load("lower.st.RData")
load("lower.cra.RData")
load("lower.fl.RData")
load("upper.al.RData")
load("upper.y.RData")
load("upper.st.RData")
load("upper.cra.RData")
load("upper.fl.RData")

# Confidence Interval Width X
mean(upper.fl[,1] - lower.fl[,1])
mean(upper.cra[,1] - lower.cra[,1])
mean(upper.st[,1] - lower.st[,1])
mean(upper.y[,1] - lower.y[,1])
mean(upper.al[,1] - lower.al[,1])

# Confidence Interval Width Z
mean(upper.fl[,2] - lower.fl[,2])
mean(upper.cra[,2] - lower.cra[,2])
mean(upper.st[,2] - lower.st[,2])
mean(upper.y[,2] - lower.y[,2])
mean(upper.al[,2] - lower.al[,2])

# Confidence Interval Width XZ
mean(upper.fl[,3] - lower.fl[,3])
mean(upper.cra[,3] - lower.cra[,3])
mean(upper.st[,3] - lower.st[,3])
mean(upper.y[,3] - lower.y[,3])
```
mean(upper.al[,3] - lower.al[,3])

xbeta <- 0.45
zbeta <- 0.55
xzdelta <- 0.6
R = 1000
qmin = function(X){  
  return(quantile(X, .05))
}
qmax = function(X){  
  return(quantile(X, .95))
}
coverage.fl <- matrix(NA, nrow=R, ncol=3)  
coverage.cra <- matrix(NA, nrow=R, ncol=3)  
coverage.st <- matrix(NA, nrow=R, ncol=3)  
coverage.y <- matrix(NA, nrow=R, ncol=3)  
coverage.al <- matrix(NA, nrow=R, ncol=3)

for(i in 1:R){
  coverage.fl[i,1] <- (lower.fl[i,1] < xbeta & upper.fl[i,1] > xbeta)  
  coverage.fl[i,2] <- (lower.fl[i,2] < zbeta & upper.fl[i,2] > zbeta)  
  coverage.fl[i,3] <- (lower.fl[i,3] < xzdelta & upper.fl[i,3] > xzdelta)

  coverage.cra[i,1] <- (lower.cra[i,1] < xbeta & upper.cra[i,1] > xbeta)  
  coverage.cra[i,2] <- (lower.cra[i,2] < zbeta & upper.cra[i,2] > zbeta)  
  coverage.cra[i,3] <- (lower.cra[i,3] < xzdelta & upper.cra[i,3] > xzdelta)

  coverage.st[i,1] <- (lower.st[i,1] < xbeta & upper.st[i,1] > xbeta)  
  coverage.st[i,2] <- (lower.st[i,2] < zbeta & upper.st[i,2] > zbeta)  
  coverage.st[i,3] <- (lower.st[i,3] < xzdelta & upper.st[i,3] > xzdelta)

  coverage.y[i,1] <- (lower.y[i,1] < xbeta & upper.y[i,1] > xbeta)  
  coverage.y[i,2] <- (lower.y[i,2] < zbeta & upper.y[i,2] > zbeta)  
  coverage.y[i,3] <- (lower.y[i,3] < xzdelta & upper.y[i,3] > xzdelta)

  coverage.al[i,1] <- (lower.al[i,1] < xbeta & upper.al[i,1] > xbeta)
coverage.al[i,2] <- (lower.al[i,2] < zbeta & upper.al[i,2] > zbeta)
coverage.al[i,3] <- (lower.al[i,3] < xzdelta & upper.al[i,3] > xzdelta)

if(i%%(R/20)==0) print(paste0((i/R)*100,"%"))
}

# Setup for X data
est.x <- as.data.frame(c(coef.fl[,1], coef.cra[,1], coef.st[,1],
coef.y[,1], coef.al[,1]))
names(est.x) = c("est.x")
summary(est.x)
se.x <- as.data.frame(c(SE.fl[,1], SE.cra[,1], SE.st[,1],
SE.y[,1], SE.al[,1]))
names(se.x) = c("se.x")
summary(se.x)
cov.x <- as.data.frame(c(coverage.fl[,1], coverage.cra[,1],
coverage.st[,1], coverage.y[,1], coverage.al[,1]))
names(cov.x) = c("cov.x")
summary(cov.x)
method.x <- as.data.frame(c(rep("Full Data", R), rep("Complete
Records", R), rep("MI (Standard)", R), rep("MI (y
interactions)", R), rep("MI (all interactions)", R))
names(method.x) <- c("method.x")
summary(method.x)
covp.x <- as.data.frame(c(rep(mean(coverage.fl[,1]),R),
rep(mean(coverage.cra[,1]),R), rep(mean(coverage.st[,1]),R),
rep(mean(coverage.y[,1]),R), rep(mean(coverage.al[,1]),R)))
names(covp.x) <- c("covp.x")
summary(covp.x)

datx <- data.frame(est.x, se.x, cov.x, method.x, covp.x)

### X data
fl <- subset(datx$est.x, datx$method.x == "Full Data")
fl1 <- subset(fl, fl >= qmin(fl) & fl <= qmax(fl))
cra <- subset(datx$est.x, datx$method.x == "Complete
Records")
cral <- subset(cra, cra >= qmin(cra) & cra <= qmax(cra))
st <- subset(datx$est.x, datx$method.x == "MI (Standard)")
stl <- subset(st, st >= qmin(st) & st <= qmax(st))
yi <- subset(datx$est.x, datx$method.x == "MI (y
interactions)")
yil <- subset(yi, yi >= qmin(yi) & yi <= qmax(yi))
al <- subset(datx$est.x, datx$method.x == "MI (all
interactions)")
all <- subset(al, al >= qmin(al) & al <= qmax(al))
meanx <- as.data.frame(c(fl, cra, st, yi, al))
mean(fl)
mean(cra)
mean(st)
mean(yi)
mean(al)
meanx <- c(rep(mean(fl), R*.9), rep(mean(cra), R*.9),
rep(mean(st), R*.9), rep(mean(yi), R*.9), rep(mean(al), R*.9))

plotx <- as.data.frame(c(fl1, cra1, st1, yi1, al1))
names(plotx) <- c("plotx")
mx <- as.data.frame(c(rep("Full Data", R*.9), rep("Complete Records", R*.9), rep("MI (Standard)", R*.9), rep("MI (y interactions)", R*.9), rep("MI (all interactions)", R*.9)))
names(mx) <- c("mx")
cperx <- as.data.frame(c(rep(mean(coverage.fl[,1]),R*.9),
rep(mean(coverage.cra[,1]),R*.9),
rep(mean(coverage.st[,1]),R*.9), rep(mean(coverage.y[,1]),R*.9),
rep(mean(coverage.al[,1]),R*.9)))
names(cperx) <- c("cperx")

plx <- data.frame(plotx, mx, cperx, meanx)

## X Box Plots
p <- ggplot(plx, aes(x = mx, y = plotx))
p +  geom_boxplot(outlier.shape = NA, width=0) +
  geom_hline(yintercept=xbeta) +
  geom_point(aes(x=5, y=meanx[1]), colour="blue", geom="point") +
  geom_point(aes(x=4, y=meanx[901]), colour="blue", geom="point") +
  geom_point(aes(x=3, y=meanx[1801]), colour="blue", geom="point") +
  geom_point(aes(x=2, y=meanx[2701]), colour="blue", geom="point") +
  geom_point(aes(x=1, y=meanx[3601]), colour="blue", geom="point") +
  scale_x_discrete(limits = c("MI (all interactions)", "MI (y interactions)", "MI (Standard)", "Complete Records", "Full Data")) +
  scale_y_discrete(limits = c(-0.25, 0.00, 0.25, 0.50, 0.75, 1.00, 1.25)) +
  coord_flip() +
ggtitle("<Name of Method>") +
  theme_classic() +
theme(plot.title = element_text(hjust=0.5),
    axis.text.x=element_text(angle=45, hjust=1),
    panel.border = element_rect(colour = "black", fill=NA, size=2)) +
  ylab("Estimated Values for X") +
xlab("Missing Data Method") #+
annotate(geom = "text", x=5, y=1.5,
  label=round(mean(coverage.fl[,1])*100), size=4.5) +
annotate(geom = "text", x=4, y=1.5,
  label=round(mean(coverage.cra[,1])*100), size=4.5) +
annotate(geom = "text", x=3, y=1.5,
  label=round(mean(coverage.st[,1])*100), size=4.5) +
annotate(geom = "text", x=2, y=1.5,
  label=round(mean(coverage.y[,1])*100), size=4.5) +
annotate(geom = "text", x=1, y=1.5,
  label=round(mean(coverage.al[,1])*100), size=4.5) +
annotate(geom = "text", x=5, y=1.59, label="%", size=4.5) +
annotate(geom = "text", x=4, y=1.59, label="%", size=4.5) +
annotate(geom = "text", x=3, y=1.59, label="%", size=4.5) +
annotate(geom = "text", x=2, y=1.59, label="%", size=4.5) +
annotate(geom = "text", x=1, y=1.59, label="%", size=4.5) +
geom_vline(xintercept = 4.5, fill = NA, size = .5) +
geom_vline(xintercept = 3.5, fill = NA, size = .5) +
geom_vline(xintercept = 2.5, fill = NA, size = .5) +
geom_vline(xintercept = 1.5, fill = NA, size = .5)

# Setup for z data
est.z <- as.data.frame(c(coef.fl[,2], coef.cra[,2], coef.st[,2],
                        coef.y[,2], coef.al[,2]))
names(est.z) = c("est.z")
summary(est.z)
se.z <- as.data.frame(c(SE.fl[,2], SE.cra[,2], SE.st[,2],
                      SE.y[,2], SE.al[,2]))
names(se.z) = c("se.z")
summary(se.z)
cov.z <- as.data.frame(c(coverage.fl[,2], coverage.cra[,2],
                         coverage.st[,2], coverage.y[,2], coverage.al[,2]))
names(cov.z) = c("cov.z")
summary(cov.z)
method.z <- as.data.frame(c(rep("Full Data", R), rep("Complete Records", R), rep("MI (Standard)", R), rep("MI (y interactions)", R), rep("MI (all interactions)", R)))
names(method.z) <- c("method.z")
summary(method.z)
covp.z <- as.data.frame(c(rep(mean(coverage.fl[,2]),R),
                          rep(mean(coverage.cra[,2]),R), rep(mean(coverage.st[,2]),R),
                          rep(mean(coverage.y[,2]),R), rep(mean(coverage.al[,2]),R)))
names(covp.z) <- c("covp.z")
summary(covp.z)

datz <- data.frame(est.z, se.z, cov.z, method.z, covp.z)

## z data
fl <- subset(datz$est.z, datz$method.z == "Full Data")
fl1 <- subset(fl, fl >= qmin(fl) & fl <= qmax(fl))
cra <- subset(datz$est.z, datz$method.z == "Complete Records")
cral <- subset(cra, cra >= qmin(cra) & cra <= qmax(cra))
st <- subset(datz$est.z, datz$method.z == "MI (Standard)")
st1 <- subset(st, st >= qmin(st) & st <= qmax(st))
fy <- subset(datz$est.z, datz$method.z == "MI (y interactions)")
fy1 <- subset(fy, fy >= qmin(fy) & fy <= qmax(fy))
al <- subset(datz$est.z, datz$method.z == "MI (all interactions)")
all <- subset(al, al >= qmin(al) & al <= qmax(al))

meanz <- as.data.frame(c(fl, cra, st, fy, al))
mean(fl)
mean(cra)
mean(st)
mean(fy)
mean(al)
meanz <- c(rep(mean(fl), R*.9), rep(mean(cra), R*.9),
rep(mean(st), R*.9), rep(mean(fy), R*.9), rep(mean(al), R*.9))

plotz <- as.data.frame(c(fl1, cra1, st1, fy1, all))

mz <- as.data.frame(c(rep("Full Data", R*.9), rep("Complete
Records", R*.9), rep("MI (Standard)", R*.9), rep("MI (y
interactions)", R*.9), rep("MI (all interactions)", R*.9))
names(mz) <- c("mz")
cperz <- as.data.frame(c(rep(mean(coverage.fl[,2]),R*.9),
rep(mean(coverage.cra[,2]),R*.9),
rep(mean(coverage.st[,2]),R*.9), rep(mean(coverage.y[,2]),R*.9),
rep(mean(coverage.al[,2]),R*.9)))
names(cperz) <- c("cperz")

plz <- data.frame(plotz, mz, cperz, meanz)

## Z Box Plots
p <- ggplot(plz, aes(x = mz, y = plotz))
p + geom_boxplot(outlier.shape = NA, width=0) +
  geom_hline(yintercept=zbeta) +
geom_point(aes(x=5, y=meanz[1]), colour="blue", geom="point") +
  geom_point(aes(x=4, y=meanz[901]), colour="blue", geom="point") +
  geom_point(aes(x=3, y=meanz[1801]), colour="blue", geom="point") +
  geom_point(aes(x=2, y=meanz[2701]), colour="blue", geom="point") +
  geom_point(aes(x=1, y=meanz[3601]), colour="blue", geom="point") +
  scale_x_discrete(limits = c("MI (all interactions)", "MI (y interactions)", "MI (Standard)", "Complete Records", "Full Data")) +
  scale_y_discrete(limits = c(-0.25, 0.00, 0.25, 0.50, 0.75, 1.00, 1.25)) +
  coord_flip() +
  ggtitle("<Name of Method>") +
  theme_classic() +
  theme(plot.title = element_text(hjust=0.5),
        axis.text.x=element_text(angle=45, hjust=1),
        panel.border = element_rect(colour = "black", fill=NA,
size=2)) +
  ylab("Estimated Values for Z") +
  xlab("Missing Data Method") #+
  annotate(geom = "text", x=5, y=1.5,
label=round(mean(coverage.fl[,2])*100), size=4.5) +
  annotate(geom = "text", x=4, y=1.5,
label=round(mean(coverage.cra[,2])*100), size=4.5) +
  annotate(geom = "text", x=3, y=1.5,
label=round(mean(coverage.st[,2])*100), size=4.5) +
  annotate(geom = "text", x=2, y=1.5,
label=round(mean(coverage.y[,2])*100), size=4.5) +
  annotate(geom = "text", x=1, y=1.5,
label=round(mean(coverage.al[,2])*100), size=4.5) +
  annotate(geom = "text", x=5, y=1.59, label="%", size=4.5) +
  annotate(geom = "text", x=4, y=1.59, label="%", size=4.5) +
  annotate(geom = "text", x=3, y=1.59, label="%", size=4.5) +
  annotate(geom = "text", x=2, y=1.59, label="%", size=4.5) +
  annotate(geom = "text", x=1, y=1.59, label="%", size=4.5) +
  geom_vline(xintercept = 4.5, fill = NA, size = .5) +
  geom_vline(xintercept = 3.5, fill = NA, size = .5) +
  geom_vline(xintercept = 2.5, fill = NA, size = .5) +
  geom_vline(xintercept = 1.5, fill = NA, size = .5)

# Setup for xz data
est.xz <- as.data.frame(c(coef.fl[,3], coef.cra[,3],
  coef.st[,3], coef.y[,3], coef.al[,3]))
names(est.xz) = c("est.xz")
summary(est.xz)
se.xz <- as.data.frame(c(SE.fl[,3], SE.cra[,3], SE.st[,3],
SE.y[,3], SE.al[,3]))
names(se.xz) = c("se.xz")
summary(se.xz)
cov.xz <- as.data.frame(c(coverage.fl[,3], coverage.cra[,3],
coverage.st[,3], coverage.y[,3], coverage.al[,3]))
names(cov.xz) = c("cov.xz")
summary(cov.xz)
method.xz <- as.data.frame(c(rep("Full Data", R), rep("Complete
Records", R), rep("MI (Standard)", R), rep("MI (y
interactions)", R), rep("MI (all interactions)", R)))
names(method.xz) <- c("method.xz")
summary(method.xz)
covp.xz <- as.data.frame(c(rep(mean(coverage.fl[,3]),R),
rep(mean(coverage.cra[,3]),R), rep(mean(coverage.st[,3]),R),
rep(mean(coverage.y[,3]),R), rep(mean(coverage.al[,3]),R)))
names(covp.xz) <- c("covp.xz")
summary(covp.xz)

datxz <- data.frame(est.xz, se.xz, cov.xz, method.xz, covp.xz)

## xz data
fl <- subset(datxz$est.xz, datxz$method.xz == "Full Data")
fl1 <- subset(fl, fl >= qmin(fl) & fl <= qmax(fl))
cra <- subset(datxz$est.xz, datxz$method.xz == "Complete
Records")
cral <- subset(cra, cra >= qmin(cra) & cra <= qmax(cra))
st <- subset(datxz$est.xz, datxz$method.xz == "MI (Standard)")
stl <- subset(st, st >= qmin(st) & st <= qmax(st))
yi <- subset(datxz$est.xz, datxz$method.xz == "MI (y
interactions)")
yil <- subset(yi, yi >= qmin(yi) & yi <= qmax(yi))
al <- subset(datxz$est.xz, datxz$method.xz == "MI (all
interactions)")
al1 <- subset(al, al >= qmin(al) & al <= qmax(al))

meanxz <- as.data.frame(c(fl, cra, st, yi, al))
mean(fl)
mean(cra)
mean(st)
mean(yi)
mean(al)
meanxz <- c(rep(mean(fl), R*.9), rep(mean(cra), R*.9),
rep(mean(st), R*.9), rep(mean(yi), R*.9), rep(mean(al), R*.9))
plotxz <- as.data.frame(c(fl1, cra1, st1, yi1, al1))
names(plotxz) <- c("plotxz")
mxz <- as.data.frame(c(rep("Full Data", R*.9), rep("Complete Records", R*.9), rep("MI (Standard)", R*.9), rep("MI (y interactions)", R*.9), rep("MI (all interactions)", R*.9)))
names(mxz) <- c("mxz")
cperxz <- as.data.frame(c(rep(mean(coverage.fl[,3]),R*.9), rep(mean(coverage.cra[,3]),R*.9), rep(mean(coverage.st[,3]),R*.9), rep(mean(coverage.y[,3]),R*.9), rep(mean(coverage.al[,3]),R*.9)))
names(cperxz) <- c("cperxz")
plxz <- data.frame(plotxz, mxz, cperxz, meanxz)

## xz Box Plots

p <- ggplot(plxz, aes(x = mxz, y = plotxz))
p + geom_boxplot(outlier.shape = NA, width=0) +
  geom_hline(yintercept=xzdelta) +
  geom_point(aes(x=5, y=meanxz[1]), colour="blue", geom="point") +
  geom_point(aes(x=4, y=meanxz[901]), colour="blue", geom="point") +
  geom_point(aes(x=3, y=meanxz[1801]), colour="blue", geom="point") +
  geom_point(aes(x=2, y=meanxz[2701]), colour="blue", geom="point") +
  geom_point(aes(x=1, y=meanxz[3601]), colour="blue", geom="point") +
  scale_x_discrete(limits = c("MI (all interactions)", "MI (y interactions)", "MI (Standard)", "Complete Records", "Full Data")) +
  scale_y_discrete(limits = c(-0.25, 0.00, 0.25, 0.50, 0.75, 1.00, 1.25)) +
  coord_flip() +
  ggtitle("<Name of Method>") +
  theme_classic() +
  theme(plot.title = element_text(hjust=0.5),
        axis.text.x=element_text(angle=45, hjust=1),
        panel.border = element_rect(colour = "black", fill=NA, size=2)) +
  ylab("Estimated Values for xz") +
  xlab("Missing Data Method") #+
  annotate(geom = "text", x=5, y=1.5, label=round(mean(coverage.fl[,3])*100), size=4.5) +
annotate(geom = "text", x=4, y=1.5, label=round(mean(coverage.cra[,3]) * 100), size=4.5) + 
annotate(geom = "text", x=3, y=1.5, label=round(mean(coverage.st[,3]) * 100), size=4.5) + 
annotate(geom = "text", x=2, y=1.5, label=round(mean(coverage.y[,3]) * 100), size=4.5) + 
annotate(geom = "text", x=1, y=1.5, label=round(mean(coverage.al[,3]) * 100), size=4.5) + 
annotate(geom = "text", x=5, y=1.59, label="\%", size=4.5) + 
annotate(geom = "text", x=4, y=1.59, label="\%", size=4.5) + 
annotate(geom = "text", x=3, y=1.59, label="\%", size=4.5) + 
annotate(geom = "text", x=2, y=1.59, label="\%", size=4.5) + 
annotate(geom = "text", x=1, y=1.59, label="\%", size=4.5) + 
geom_vline(xintercept = 4.5, fill = NA, size = .5) + 
geom_vline(xintercept = 4.5, fill = NA, size = .5) + 
geom_vline(xintercept = 3.5, fill = NA, size = .5) + 
geom_vline(xintercept = 2.5, fill = NA, size = .5) + 
geom_vline(xintercept = 1.5, fill = NA, size = .5)
Appendix D: Code for Displaying Predictor Matrix

```r
library(reshape2)
library(ggplot2)

myPredMat1 <- myPredMat

cor.dat <- melt(myPredMat1)
cor.dat <- data.frame(cor.dat)
cor.dat <- cor.dat[complete.cases(cor.dat),]

cor.dat$Var1 <- with(cor.dat, ifelse(Var1 == "x", "X",
                                ifelse(Var1 == "y.miss",
                                        ifelse(Var1 == "z.miss",
                                                "Z.miss",
                                                "X*Z")),
                                "Y.miss"))

cor.dat$Var2 <- with(cor.dat, ifelse(Var2 == "x", "X",
                                ifelse(Var2 == "y.miss",
                                        ifelse(Var2 == "z.miss",
                                                "Z.miss",
                                                "X*Z")),
                                "Y.miss"))

ggplot(cor.dat, aes(Var2, Var1, fill = value)) +
  geom_tile(colour="white", size=1.5, stat="identity") +
  geom_text(data=cor.dat, aes(Var2, Var1, label = value),
            color="black", size=rel(4.5)) +
  scale_fill_gradient(low = "gray90", high = "dodgerblue", space = "Lab",
                     na.value = "gray90", guide = "colourbar") +
  scale_x_discrete(expand = c(0, 0), position = "top", limits =
                   c("Y.miss", "X", "Z.miss", "X*Z")) +
  scale_y_discrete(expand = c(0, 0), limits = c("X*Z",
                   "Z.miss","X", "Y.miss")) +
  xlab("") +
  ylab("") +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_rect(fill=NA,color="white",
                                   size=0.5, linetype="solid"),
        axis.line = element_blank(),
        axis.ticks = element_blank(),
        panel.background = element_rect(fill="white"),
        plot.background = element_rect(fill="white"),
        legend.position = "none",
        axis.text = element_text(color="black", size=14) )
```
library(reshape2)
library(ggplot2)

myPredMat2 <- myPredMat

cor.dat <- melt(myPredMat2)
cor.dat <- data.frame(cor.dat)
cor.dat <- cor.dat[complete.cases(cor.dat),]

cor.dat$Var1 <- with(cor.dat, ifelse(Var1 == "x", "X",
                                        ifelse(Var1 == "y.miss",
                                                "Y.miss",
                                                ifelse(Var1 == "z.miss", "Z.miss",
                                                        ifelse(Var1 == "xzint", "X*Z", ifelse(Var1 == "yzint", "Y*Z", "X*Y"))))))

cor.dat$Var2 <- with(cor.dat, ifelse(Var2 == "x", "X",
                                        ifelse(Var2 == "y.miss",
                                                "Y.miss",
                                                ifelse(Var2 == "z.miss", "Z.miss",
                                                        ifelse(Var2 == "xzint", "X*Z", ifelse(Var2 == "yzint", "Y*Z", "X*Y"))))))

ggplot(cor.dat, aes(Var2, Var1, fill = value)) +
  geom_tile(colour="white", size=1.5, stat="identity") +
  geom_text(data=cor.dat, aes(Var2, Var1, label = value),
             color="black", size=rel(4.5)) +
  scale_fill_gradient(low = "gray90", high = "dodgerblue", space = "Lab", na.value = "gray90", guide = "colourbar") +
  scale_x_discrete(expand = c(0, 0), position = "top", limits = c("Y.miss", "X", "Z.miss", "X*Z", "Y*Z", "X*Y")) +
  scale_y_discrete(expand = c(0, 0), limits = c("X*Y","Y*Z","X*Z","Z.miss","X","Y.miss")) +
  xlab("") +
  ylab("") +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_rect(fill=NA,color="white", size=0.5, linetype="solid"),
        axis.line = element_blank(),
        axis.ticks = element_blank(),
   ...)
panel.background = element_rect(fill="white"),
plot.background = element_rect(fill="white"),
legend.position = "none",
axis.text = element_text(color="black", size=14) )