UNIVERSITY OF LEEDS

You must sign this (digitally signing with your name is acceptable) and include it with each piece of work you submit.

I am aware that the University defines plagiarism as presenting someone else's work, in whole or in part, as your own. Work means any intellectual output, and typically includes text, data, images, sound or performance.

I promise that in the attached submission I have not presented anyone else's work, in whole or in part, as my own and I have not colluded with others in the preparation of this work. Where I have taken advantage of the work of others, I have given full acknowledgement. I have not resubmitted my own work or part thereof without specific written permission to do so from the University staff concerned when any of this work has been or is being submitted for marks or credits even if in a different module or for a different qualification or completed prior to entry to the University. I have read and understood the University's published rules on plagiarism and also any more detailed rules specified at School or module level. I know that if I commit plagiarism I can be expelled from the University and that it is my responsibility to be aware of the University's regulations on plagiarism and their importance.

I re-confirm my consent to the University copying and distributing any or all of my work in any form and using third parties (who may be based outside the UK) to monitor breaches of regulations, to verify whether my work contains plagiarised material, and for quality assurance purposes.

I confirm that I have declared all mitigating circumstances that may be relevant to the assessment of this piece of work and that I wish to have taken into account. I am aware of the University's policy on mitigation and the School's procedures for the submission of statements and evidence of mitigation. I am aware of the penalties imposed for the late submission of coursework.

Student Signature Mohammed Fahaidur Rahman Date 12/05/2023
Student Name Mohammed Fahaidur Rahman Student Number 201306687

Statistical Methods for Missing Data

Mohammed Fahaidur Rahman (201306687)

May 12, 2023

Contents

1 Introduction

The following section has been inspired by Graham[1] and also Little and Rubin[2]

In many aspects of research, missing data is frequently common. What do we mean by missing data, well missing data is as intuitive as it sounds. Any data that was intended to be *observed* but hasn't is classified as missing data. Researchers have relied on a variety of ad-hoc methods for decades in an effort to either complete the data by eliminating invalid cases (cases with any missing data) or adding predictions for the missing values so the data analysis could be carried out. However, these methods have their downsides, albeit they are convenient due to the methods having to rely on assumptions on how the missing data came about. In this report, we will discuss the problem that missing data brings and the different *types* of missing data. Then some of these methods that have been used originally will be discussed, looking at some results of using these methods and then discussing why these methods don't actually work. Then one method, *multiple imputation* will be discussed which is a widely popular method that was developed to tackle missing data by creating feasible replacements so that a non-bias analysis can take place.

There are 2 types of missing data which can occur in a dataset, *item nonresponse* and *wave nonresponse*. Item nonresponse is the idea that a research participant has only some amount of data but not the intended amount. This generally occurs in cross-sectional studies which are studies where data is taken in at one point in time only. Typically, the variables are only measured once in each case. For example a ballot for general elections in the UK. The voter only votes once for their preferred party and so all the data was only taken at one time. There are many trivial reasons why item nonresponse could happen example in survey research this could be due to the fact the respondent forgot to answer a question on the survey. item nonresponse could also be due to the fact that data was lost during data collection, in data storage or there could have been a problem with equipment while gathering data.

Wave nonresponse only applies to longitudinal research. longitudinal research is research where data where the same respondents are measured more than once. An example of longitudinal research could be medical research testing for the side effects of a new drug. Here, they would check up on the medical participants at set intervals to see if anything has changed. So wave nonresponse generally means that a respondent has failed to complete the whole survey by not being able to give more than one data entry. This could be due to the respondent being absent to fill in the data. As longitudinal research is taken multiple times the individual that has missed one survey may be measured in the next, or the respondent never returns which is called *attrition*. Longitudinal research has it's own set of problems with missing data that needs handling but will not be further looked at in this report.

2 Missing Data Theory and Mechanisms

The following section is inspired by Enders [3].

2.1 Missing data patterns

This section is inspired by Enders' Applied missing data [3]

Firstly, we define Y to be a $n \times p$ matrix. Y_j is the j^{th} column in Y and Y_{-j} is the complement of Y_j which means every column of Y except of Y_j. The *Missing data pattern* is represented by the missing response matrix R which is an $n \times k$ matrix as well which has been already introduced in Section 2.1.

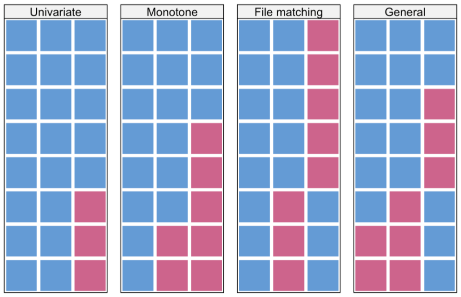


Figure 1: This diagram shows the different types of missing data patterns that can occur.

Firstly we should distinguish the difference between a missing data pattern and a missing data mechanism. A missing data pattern displays where the observed and missing data lie on a data set, so shows where the holes are in the data but does not explain why the data is missing while a missing data mechanism describes the link between the measured variables and the probability of missing data. Figure 1 shows us some missing data patterns. The first one on the left is a *Univariate* missing data pattern which we have already discussed but it simply means when only one missing data pattern is said to be univariate. The 2^{nd} missing data pattern represents the *monotone* typically occurs in a longitudinal study as the study progresses participants drop out so they cause missingness through each round of the study after they leave. This would mean if the variables Y_j are missing then the variables can be ordered in the missing data pattern such that all variables after j are also missing. If the missing data pattern is not monotone it can be called non-monotone or *general.* Another type of missing data pattern is *connected* which is when any observed data point can be reached from any other point that has been observed by only moving vertically or horizontally. All the patterns in Figure 1 are connected however if the 3rd pattern did not have the first column it would not be connected anymore.

Before we start on the theory and mechanisms of missing data, we should consider an example of a data set (Table 1) to use as an example for the following section. This data set illustrates a scenario where an employer has made potential employees take a test for the application process for a job opening and a psychological well-being questionnaire during their interview. The company then hired the employees who had test results in the top half of the 20 applicants and then performed a job performance review 1 year later. Here we should realise that the job performance scores that are missing are due to the applicants who scored in the bottom half of the test results and were never hired therefore having no performance rating. In addition, 3 randomly chosen well-being scores were deleted to imitate a situation where the data may have been lost or not recorded properly. Table 2 has the Test Scores and Job performance ratings with the complete, MCAR, MAR and MNAR values. These terms will be explained in the following section. The complete data column may be confusing as it has values not in the real data set but this is the data that would have been collected if all the applicants had been hired and if no data was lost. So this is a hypothetical data set with some real values but some values that would have been true. Note this table similar to the example in section 1 in *Enders [3]*, but instead IQ scores are replaced by application test scores and the values of each were slightly changed.

Table 1: Employee Selection Data Set

Test Score	**Psychological well-being**	**Job Peformance**
56	12	-
60	8	-
60	9	-
61	9	-
62	-	-
65	2	-
66	10	-
67	3	-
67	14	-
69	-	-
71	5	8
75	13	9
75	13	11
76	10	15
77	-	10
80	11	10
81	13	12
82	13	13
84	15	17
96	9	12

The following segment has been inspired by Little and Rubin [2], Enders [3], Allison [4], and Van Buuren [5].

Here we'll introduce a classification system for missing data scenarios which is the general way used in literature today. There are 3 missing data mechanisms that are standard in all of the missing data literature which are mechanisms

Table 2: Job Performance Ratings with MCAR,MAR and MNAR missing values

Job Performance ratings				
Test score	Complete	MCAR	MAR	MNAR
56	9	9	-	9
60	13	13	-	13
60	10	-	-	10
61	8	-	-	-
62	7	7	-	-
65	7	7	7	-
66	9	9	9	9
67	9	9	9	9
67	11	-	11	11
69	7	7	7	-
71	8	8	8	-
75	9	9	9	9
75	11	-	11	11
76	15	15	15	15
77	10	10	10	10
80	10	10	10	10
81	12	12	12	12
82	13	13	13	13
84	17	17	17	17
96	12	12	12	12

which relate the probability of missing values to the data, if it can be related. These mechanisms were first introduced by Rubin [6] and we will go through a conceptual description of each with some examples then will formalise these mechanisms with mathematical notation and expressions. These mechanisms will be widely used in the rest of the report so they are very important ideas to understand.

There are 3 categories of mechanisms of missingness. These are *Missing Completely at Random* (MCAR), *Missing At Random* (MAR) and *Missing Not At Random* (MNAR). Every data point has some likelihood of being missing. The way of governing probabilities is known as a missing data mechanism or response mechanism, and it can be described using a missing data model or response model.

2.2 Missing Completely at Random

Definition 2.1. The *missing completely at random* (MCAR) mechanism is what we would think of as pure random missingness. Formally it is defined as the probability of missing data on a variable Y having no relation to the other measured variables and having no relation to the values of Y itself. In other words, we can say that the observed data is a random sample of the data that would have been observed in the completed data.

Looking at Table 2 The missing values in the MCAR column are missing at random without any bias to the remaining data or the data that is missing. So the missing data has no correlation with the employee test score and job performance, therefore here missingness is unrelated to the data and therefore the 16 data entries in that column actually represent the observed data. This should indicate that there shouldn't be much of a difference between the mean of the missing data and the observed data. So if we calculate the mean of the observed data of the MCAR column, the mean is 10.56 while the mean for the missing data (looking at the Complete column) is calculated to be 10. So there is only a small difference but still suggests the 2 groups are randomly equivalent and provides evidence that the job performance score can be classified as MCAR.

Now we can formalise the definition with mathematical notation but first, we shall reintroduce some terms first.

Definition 2.2. Let complete data (complete data meaning this is all the data we have, not that every unit is present), $Y = (y_{ij})$ which is an $(n \cdot k)$ data set (matrix) without any missing units with the i^{th} row $y_i \;= (y_{i1}, ..., y_{ik})$ where the $_{ij}$ is the value in Y_j for subject i. so each column of the dataset represents a variable.

Definition 2.3. Let the missing-data indicator matrix $R = (r_{ij})$ where each element of the matrix is either 1 or 0 such that if $r_{ij} = 0$ then Y_{ij} is missing (i.e. there is no value) and if $r_{ij} = 1$ then y_{ij} is not missing(i.e. there is a value). The matrix R represents the missing data patterns. Let's consider the data matrix $Y =$

Example 2.1.

$$\begin{Bmatrix} 2 & N/A \\ N/A & 6 \end{Bmatrix}.$$

The following missing indicator matrix for this data set is $R =$

$$\begin{Bmatrix} 1 & 0 \\ 0 & 1 \end{Bmatrix}.$$

The mechanism of missing data is characterized by the conditional distribution of R given b Y, say $f(R \mid Y, \theta)$ where θ denotes the parameters of the missing data model. If the probability of missing data on Y is not related to the value of Y itself or to values of any other variables in the data set then we can assume that the data is missing completely at random (MCAR) or in notation:

$$Pr(R = 0, \theta) = Pr(R = 1 \mid \theta) \sim \text{ for all } Y, \theta \tag{1}$$

Where θ are the measured parameters. An example of when data are MCAR is if a set of lab results is missing as a batch of lab samples was not processed correctly. In these instances the missing data does reduce the population of data which can be analyzed however, no bias is introduced. Another note for

MCAR is that it does allow for the possibility that the probability of missing data on Y is related to the probability of missing data on another separate variable X. For example, even if the same people who refused to disclose their ethnicity are always the same people who refuse to reveal their age, there is still a possibility that their data could be missing completely at random. The MCAR assumption is broken when the people who did not reveal their ethnicity were older on average than the people who were younger.

2.3 Missing at Random (MAR)

Here we introduce a weaker assumption which is that the data is *missing at random* (MAR). Data that is considered to be MAR when the probability of missing data on a variable X is related to some other measured variable (or variables) in the model but not the data of X itself. In other words, there is no connection between the inclination of missing data on Y and the values of Y after separating other variables of X. The term *missing at random* can be misleading as it suggests that the missing data has occurred in a completely arbitrary way (similarly to MCAR).

However, with MAR actually, there is a relationship between the probability of missing data and one or more of the measured variables. For example, we can take a look at our scenario of a company hiring employees after the applicants take a test and then the subsequent employees that had been hired had a Job performance rating after a year. There could be a case where the employed had only hired 15 employees with the highest test scores (so 5 applicants would not be hired) then one can see that the job performance ratings that are missing under the MAR column are from the applicants that had scored the lowest in the test. As a result, the probability of a missing job performance rating is merely due to the applicant's test scores and not the job performance ratings themselves. If we look at the averages of the test scores in the MAR column of the observed data and then the missing data we can see the averages are 75.4 and 59.6 so it is clear with the much larger difference in test scores compared to the MCAR missing and observed test scores averages. This then suggests that this column of data goes against the MCAR mechanism. To put this in mathematical notation we would need to introduce some new notation.

Definition 2.4. So let Y_{obs} represent the observed entries or data of Y, and Y_{miss} the missing entries such that when combined $Y = (Y_{miss}, Y_{obs})$. where Y_{obs} and Y_{miss} are the same dimensions of Y but where there was observed data in Y_{obs} is represented as 0 in Y_{miss} and where data in Y_{miss} is shown as 0 in Y_{obs}. An assumption less restrictive than MCAR is that the probability of missing data depends only on the components in Y_{obs} of Y, i.e. the observed data only and not on the data that is missing. So if the data is seen as *missing at random* then,

$$Pr(R = 1 \mid Y, \theta) = Pr(R = 1 \mid Y_{obs}, \theta) \quad \text{for all } Y, \theta \tag{2}$$

Equation (2) in words says that the probability of missingness is dependent on the observed data via some parameters θ that are related to Y_{obs}. If we link this to our example the mechanism is MAR if the probability of a job performance rating being missing is due to the applicant's test score and not the job performance ratings themselves. If we look at panel (B) of Figure 2 we can see there is no arrow between the R and the job performance scores, but there is a link between R and the test scores. This could be an explicit link for example the test scores were the result of the applicants being hired but also R and the test scores could have a mutual correlation between X, the unmeasured variables. For example, an interview may have taken place also influencing if the applicants were hired alongside the test scores so those who hadn't impressed in the interview may also not have been hired alongside their under performing test scores but still, this does not have any relation with their job performance rating as they had not been hired.

A real-life example could be whether or not someone has answered question 10 on a questionnaire has nothing to do with the values that are missing but if females are more likely to miss out on question 10 than males then the probability of data being missing is based on sex which is a different variable. It is impossible to test whether data is MAR or not as we obviously do not have the missing data so we cannot compare the values of those with and without missing data to see if they differ consistently on that variable. This proposes a problem when doing an analysis of missing data as 2 currently used methods, multiple imputation and maximum likelihood assume that the data missing is due to a MAR mechanism.

2.4 Missing Not at Random

Finally, data are missing not at random (MNAR) when the probability of missing data on a variable Y is related to the values of Y itself, even after having controlled variables. If we look at the MCAR column in Table 2 we can see that the missing job performance ratings are the lowest rating in the completed column. Some context to this could be that although the applicants were hired before the performance ratings were carried out, these employees were sacked or had left themselves on the basis they were struggling at the job. This relation shows that the missing data could be due to the data itself. If we look at Figure 2 (A) we can see that the missing indicator, R has links to the test scores (TS), the other unmeasured variables (X) and the job performance ratings themselves (JP). The last link is why the data goes against the MAR mechanism. In another example, we think of a scenario where a cancer trial with patients being analysed takes a new treatment throughout the course of time. Some data may be missing as time goes on due to severe conditions of the patients' health so are unable to attend the later stages or may even have passed away due to the illness they were being treated for in the trial. In this case, the missing data is MNAR as the data that was supposed to be collected i.e. the missing data, has a direct link to the variable that caused the missingness. Like the MAR mechanism, it is impossible to know for sure if the missing data follows an MCAR mechanism without knowing the missing variables. Lastly, the data is said to be missing not at random (MNAR) if:

$$Pr(R = 1 \mid Y_{obs}, Y_{miss}, \theta) \sim \text{ for all } Y, \theta, \qquad (3)$$

does not simplify, so here the probability to be missing also depends on unobserved information Y_{miss} itself. If we relate this back to our example, this suggests that the probability of missing data is due to both the applicant's test score, job performance or both. The box X in Figure 2 (A) holds the unknown parameters, like the ones we have discussed but X itself not needed to be in Equation 3 as the mechanisms are only considering R and the data Y.

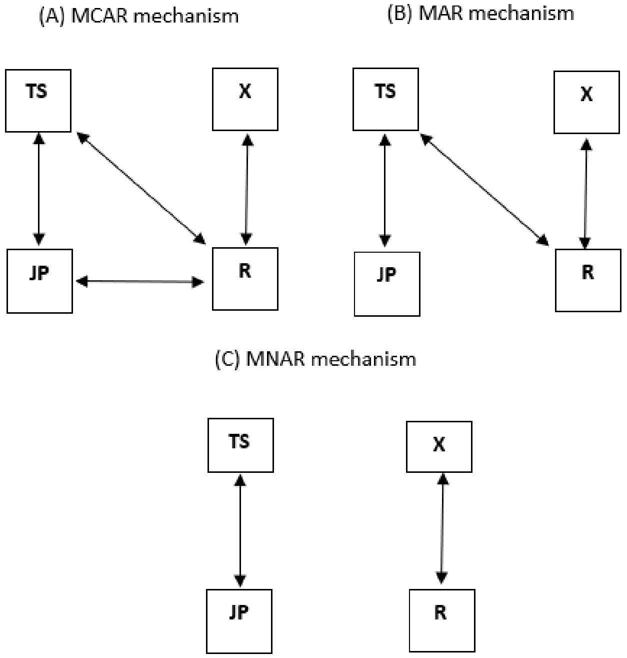


Figure 2: A figurative representation of the missing data mechanism on the test score and job performance rating example the report has been going through. The figure illustrates a bivariate scenario in which test scores (TS) are completely observed and the job performance ratings (JP) have some missing for some applicants. The arrows exhibit statistical relation and θ is a parameter that governs the probability of scoring a 1 or 0 on the missing indicator R. The boxes labelled with Y represent the unmeasured variables that affect the missing data indicator R.

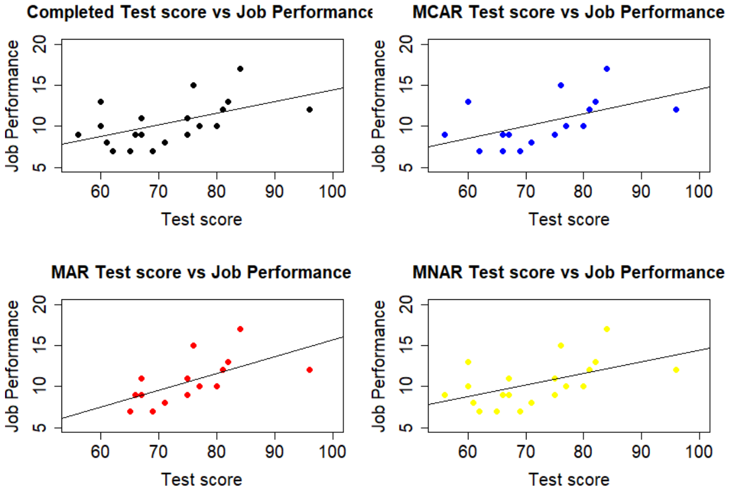


Figure 3: Plots of Test Scores Vs Job Performance for completed data, MCAR, MAR and MNAR with a linear regression fit

Table 3: Analysis Values of the completed data , MCAR, MAR and MNAR

Missing Mechanism	Mean of TS	Gradient
Complete	71.50	0.142
MCAR	73.00	0.149
MAR	75.40	0.205
MNAR	71.50	0.142

2.5 The effects of Missing Data

If we look at an example of a complete dataset, for example we have the numbers 1, 2 and 3, the mean can be calculated from this which is 2. However, if we knew there are 3 data entries and we have the data 1,2 and NA for the last data entry then we can either not calculate a true mean or we can ignore the last entry and calculate a mean of only 1 and 2 which is 1.5. This is obviously different from our true mean which may cause problems to the statistical inference. A problem also occurs when an opinion survey some individuals may not be able to express a preference for one option over another on a question. In the previous example the missing value can be treated as such and analysis can be done however in this example It is less clear as we have a survey candidate who has just refused to respond so it is not natural to treat it as a missing value. If anything the lack of an answer could suggest that it is an answer in itself and can be taken as ”no preference” or ”don’t know” out of the choices available in that question. Many software packages have lines of code which allow the non-respondents for entries of the data which are missing or not observed. Also, the software can exclude units with missing value codes for any of the variables involved in the analysis and this is called ”complete-case analysis” but it is generally unsuitable as the statistician is normally interested in making inferences about the entire population of respondents instead of just the respondents who have provided responses on all relevant variables (questions).

3 Traditional Methods

Table 4: Table containing the complete data of the data set with variables A,B,Y and then variable A when after amputation of each type of missingness, MCAR, MAR and MNAR.

Complete			MCAR	MAR	MNAR
Y	A	B	A	A	A
61.25	4.08	15.43	NA	4.08	4.08
131.96	16.31	30.45	NA	16.31	16.31
65.33	6.17	15.09	NA	6.17	6.17
61.57	6.32	14.18	6.32	6.32	NA
45.48	2.01	11.26	2.01	2.01	2.01
79.34	8.91	17.94	8.91	NA	NA
85.25	8.74	19.29	8.74	NA	8.74
66.26	7.49	16.53	7.49	7.49	7.49
90.60	9.57	21.17	9.57	NA	NA
90.82	11.71	19.70	NA	11.71	11.71
70.02	6.89	17.10	6.89	6.89	6.89
53.69	4.82	12.51	4.82	NA	4.82
73.02	6.88	18.04	NA	NA	6.88
76.30	10.05	16.86	10.05	10.05	10.05
80.89	9.39	18.77	9.39	9.39	9.39
73.36	7.19	18.09	7.19	7.19	7.19
101.37	14.76	22.80	NA	14.76	NA
93.18	11.36	21.74	11.36	NA	11.36
101.56	12.03	23.49	NA	12.03	12.03
76.82	8.83	17.84	8.83	8.83	NA
90.23	10.89	20.81	NA	NA	10.89
122.92	18.03	27.80	18.03	18.03	18.03
108.58	14.05	24.96	14.05	NA	14.05
94.40	8.79	22.83	NA	NA	8.79
59.14	5.90	13.80	5.90	5.90	5.90
86.85	8.93	20.40	8.93	NA	NA
84.90	9.20	20.28	9.20	9.20	9.20
82.96	10.52	17.90	10.52	10.52	10.52
86.33	10.58	20.08	10.58	NA	10.58
93.20	11.45	22.08	11.45	11.45	11.45

In this section, we will talk about some traditional methods which were used before more complex (and accurate) methods such as multiple imputation, we will look at their advantages, disadvantages and see how they perform with our previous example with linear regression. These methods can be found in many research articles or in statistical software packages and although were very popular at one point due to more technical approaches to imputation [2][7] exploring these methods can show us why they failed and why other techniques were needed to be developed. Firstly, we need to define some key terms such as *imputing* which simply means filling in a missing value before analysis is carried out. Also *ampute* means the opposite, so data is taken out of the dataset.

We will look at these traditional methods' using the data set that is shown in Table 3. The data was randomly generated, A is randomly generated from a normal distribution with a mean of 10 and variance of 4. $B = 10 + A + \epsilon$ where ϵ is randomly generated from a normal distribution with mean 0 and variance 4. Lastly $Y = 6 + 2A + 3B + \epsilon$. What this section of the report involves is that we will fill in the missing data using the methods that will be mentioned soon then a brief analysis will be carried out on the effects of A on Y. The reason that 3 variables were generated was in order to create missingness or to *ampute* data using the "mice" package in R, 2 variables are needed for this to occur and as we are not looking to cause any missingness in Y another variable was needed to be created in the data set hence variable B was generated.

To create missingness in the variables of A and B , the package MICE had ampute functions which creates MCAR , MAR and MNAR missingness.

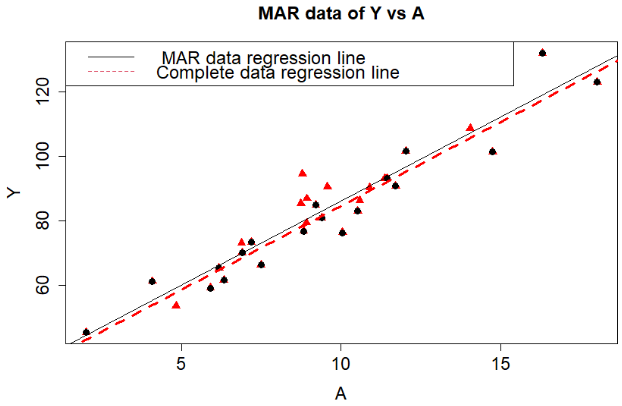


Figure 4: A plot of the Y vs A where the red points are the missing data and the black points are the observed data. The black regression line shows the regression if the data was complete and the red regression line shows the data when doing complete-case analysis on the MAR data set

3.1 Complete Case Analysis

The first traditional method that will be discussed is a fairly straightforward one and one which has already been mentioned called *complete case analysis*. Complete case analysis, also known as listwise deletion, is deleting all cases with any missing values on any of the analysis variables. So in our example, the applicants with missing data in the A column would be removed and then we would take the data set as complete and do any analysis that was planned on that data set. There are 2 advantages of complete case analysis. Firstly, it can be used for any statistical analysis and secondly no special computational methods are needed making it very convenient. If the data is MCAR then using complete case analysis should not produce any biased mean, variance and regression weights however the standard error and significance levels, that will be correct for the subset of data, are generally larger than the complete data.

A disadvantage of complete case analysis is that it can be quite wasteful. It is common in real life that more than half of the original sample is lost especially if the number of variables increases.

The main problem with complete case analysis is that if the data are MAR then analysis of the data available can cause distorted estimation parameters. This can be seen in Figure 4, although the regression lines are not much different. You can see when computing the mean of the MAR example, the means are too high as already discussed.

Complete case analysis can create unbiased estimators of regression slopes under that the missingness is not related a predictor variable and not the outcome variable, due to a fact a dataset that has missingness that MCAR represents the completed dataset in an unbiased manner as it is total random missingness. This specific situation is the only situation when complete case analysis is likely to outperform maximum likelihood estimation and multiple imputation with MNAR. If the data are not MCAR, complete case analysis can have a big impact on means, regression coefficients and correlations.

3.2 Mean imputation

The next traditional method that will be discussed is *mean imputation.* Mean imputation is a type of single imputation method, methods that create one replacement value as opposed to multiple imputation where multiple replacement values are computed. Imputation in general has the benefit of yielding a complete data set and single imputation is also convenient to do the ease of the method. So mean imputation simply is to replace any missing data value with the mean of that variable which calculated using the observed data. So formally, this looks like. If Y_{obs} which is a $i \times j$ matrix has missingess in j , we impute the missing data in j with the mean of the observed data in j.

In mean imputation the mean of the available data is calculated , so for the example in our data set that we are looking at in Table 4 the mean of A under MCAR is 9.24 (2 decimal places) , for MAR 9.6 and for MNAR it is 9.92. When these values are imputed for the missing values this causes skew in the data set as shown in Figure 5-7.Also the imputed values now show no correlation between Y and A which is incorrect as we can see on the completed data.

The distribution of the dataset is skewed from mean imputation, for example the variance of completed A is 12.31 while the variance for A when the data is MCAR, MAR and MNAR is 11.53, 16.40 and 13.98 but after mean amputation the variance becomes 7.16, 9.61, and 9.16 showing that mean imputation causes an underestimate in the variance. Mean imputation should generally not be used in most cases and maybe only used for a quick fix when only a few values are missing out of a large sample size.

3.3 Regression Imputation

The next traditional methods that we will discuss is called *regression imputation* which replaces the missing values by predicting these values using a regression equation. This method uses the correlation of the available data to predict missing data and using information from available data is a strategy that maximum likelihood and multiple imputation use. So mathematically this looks like: If Y_{obs} has missingness in column j then the missing data will be imputed based on column k which is complete. A model has to be fitted $Y_j \sim Y_k$ using all rows with Y_j

$$\hat{Y}_j = \hat{\beta}_0 + \hat{\beta}_1 \theta_k \tag{4}$$

so regression imputation is as follows: impute $Y_{ij} = \hat{\gamma}_0 + \hat{\gamma}_1 x_{ik}$ for each i where Y_j has some missing values

So the first step is to compute some regression equations that can be used to compute the missing values. Using complete case analysis this can be done.

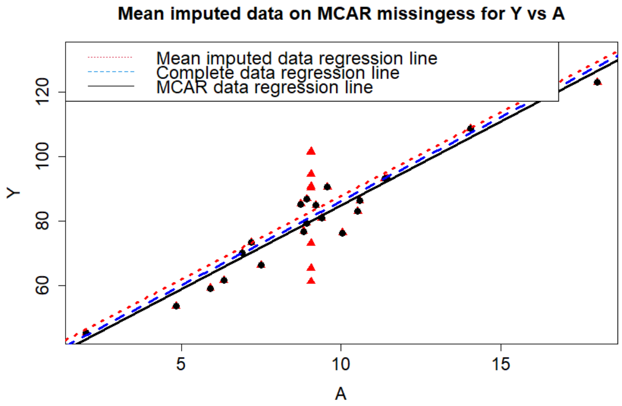


Figure 5: Plot of Y vs A where the red points are the mean imputed data and the black points are the observed data. The black regression line shows the regression using complete case analysis on the dataset with MCAR missingess, the red regression line is the regression when including the mean imputed data and the blue line is the regression line of the complete data , i.e. if no missingness had occurred.

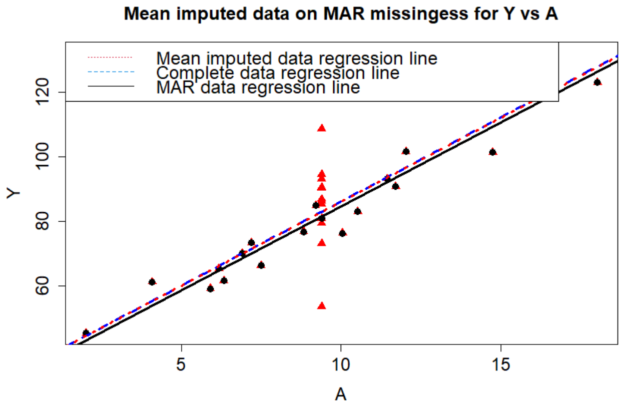


Figure 6: A plot of the Y vs A where the red points are the mean imputed data and the black points are the observed data. The black regression line shows the regression using complete case analysis on MAR dataset, the red regression line is the regression when including the mean imputed data and the blue line is the regression line of the complete data , i.e. if no missingness had occurred.

The second step is to compute the predicted values with the regression equation which then complete the data. So here we will be using the variable B will be used as the dependent variable and A as the independent variable so this looks

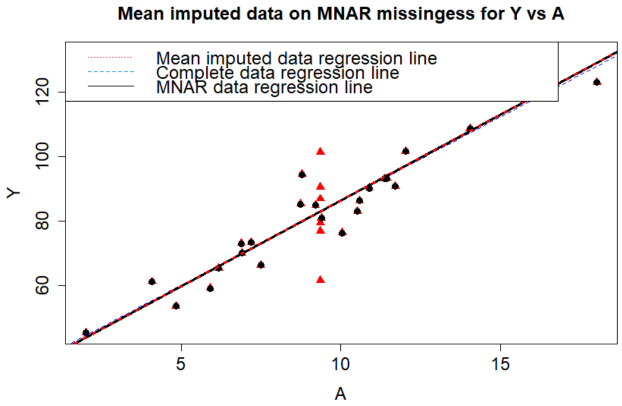


Figure 7: A plot of the Y vs A where the red points are the mean imputed data and the black points are the observed data. The black regression line shows the regression using complete case analysis on MNAR dataset, the red regression line is the regression when including the mean imputed data and the blue line is the regression line of the complete data , i.e. if no missingness had occurred.

like,

$$A_i^* = \hat{\beta}_0 + \hat{\beta}_1(B_i). \tag{5}$$

So this was done for all the MCAR, MAR and MNAR dataset then Y vs A was plotted to see how the imputations of A had done for each type of missingness. These can be seen in Figures 8-10.

Van Buuren [5] states that when the data is under MCAR, regression imputation produces unbiased mean, similarly to mean imputation and regression parameters can be unbiased if the explanatory variables have no missingness. This makes sense as if MCAR the data is an unbiased subset of the hypothetical complete data. So if the variables are linearly correlated then the imputations will follow follow this giving data that should be similar to the unobserved data. For data under MAR if the factors that affect missingness are part of the regression model then the parameters estimated are unbiased, so if B and A are linearly correlated. However, regression imputation can cause correlation to have a positive bias and the variability of the data imputed will be underestimated as generally in real life nothing is ever perfectly linear, regardless if the data at a glance is linear. Also, it hard to be sure if the hypothetical data would have been linearly related at those exact data points. The explained variance and the proportion of missing data affects the degree of underestimation. So overall regression imputation can cause unrealistic imputations and by maintaining the relationships between the variables, we can be misled into thinking the imputation model is appropriate however , actually regression imputation is artificially increasing these relationships in the data making it stronger when in reality this false relationship may have shown to disappear. As correlations have

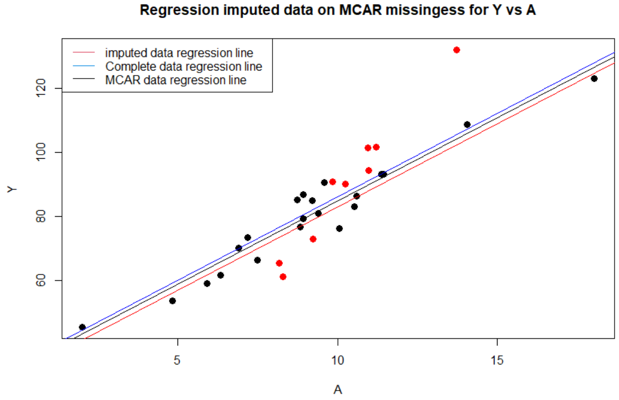


Figure 8: Plot of Y vs A where the red points are the mean imputed data and the black points are the observed data. The black regression line shows the regression using complete case analysis on the dataset with MCAR missingess, the red regression line is the regression when including the regression imputed data and the blue line is the regression line of the complete data , i.e. if no missingness had occurred.

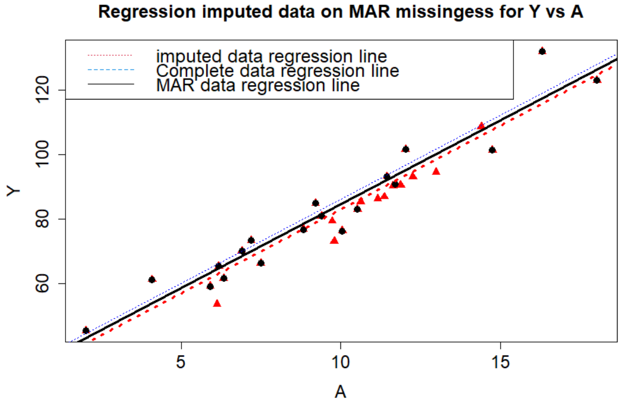


Figure 9: A plot of the Y vs A where the red points are the mean imputed data and the black points are the observed data. The black regression line shows the regression using complete case analysis on MAR dataset, the red regression line is the regression when including the regression imputed data and the blue line is the regression line of the complete data , i.e. if no missingness had occurred.

positive bias and variability isn't as strong, these imputations are unrealistic.

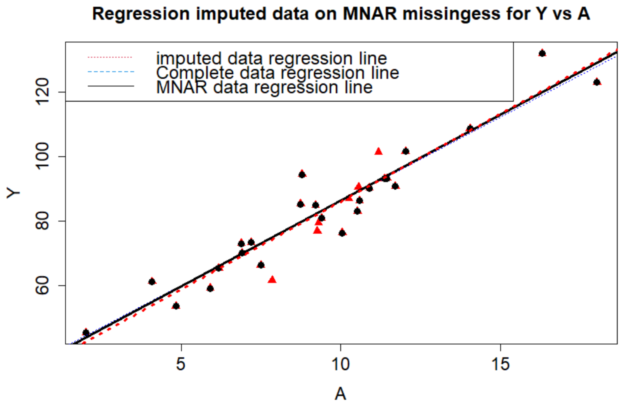


Figure 10: A plot of the Y vs A where the red points are the mean imputed data and the black points are the observed data. The black regression line shows the regression using complete case analysis on MNAR dataset, the red regression line is the regression when including the regression imputed data and the blue line is the regression line of the complete data , i.e. if no missingness had occurred.

3.4 Stochastic Regression Imputation

The last traditional method, stochastic regression imputation, that will be discussed is the most important as it links to another method, that will be discussed later called multiple imputation. Regression imputation although can seem like an attractive methods can cause to biases from the data available as discussed before. Stochastic regression uses the same regression equations to predict the missing data from the data that is available but an extra step of adding a residual from a normal distribution. This step of adding noise to the regression imputed terms removes correlation bias and adds randomness of real data back into the data set and therefore this traditional method is the only method discussed so far that does not create biased estimator parameters (e.g. regression coefficients). Mathematically this looks like: If Y_{obs} has missingness in column j then the missing data will be imputed based on column k which is complete. A model has to be fitted Y_j Y_k using all rows with Y_j . So imputing the missing values Y_{ij} can be written as

$$Y_{ij} \sim N(\hat{\beta}_0 + \hat{\beta}_1 y_{ik}, \hat{\sigma}_j^2) \tag{6}$$

Stochastic regression begins the same as regression imputation, so a complete case analysis is used to estimate the regression equations that predict the incomplete variables from the complete variables. The imputed values are predicted from the regression equations and then the final step is to add a residual term from a normal distribution with variance that is equal to the residual variance of the regression equations, to each predicted score.

The first example we will look at will be similar to the other traditional methods. Here we will look at the same dataset shown in Table 4, with variables A, B and Y and where missingness has been caused in A and B and then stochastic imputation has been done using the MICE package in R. Then A has been regressed against Y for MCAR,MAR and MNAR missingess. Figures 11-13 show the results of the stochastic impuation and regression.

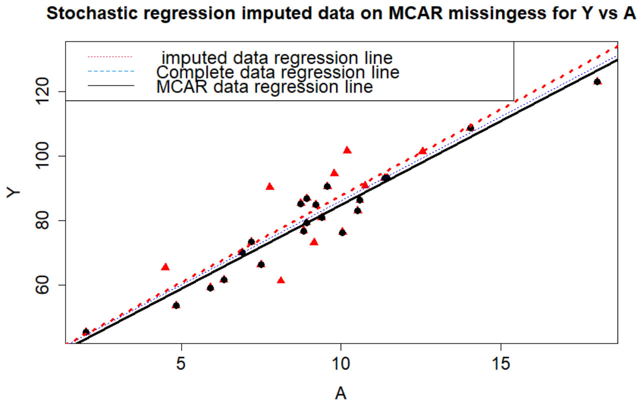


Figure 11: Plot of Y vs A where the red points are the mean imputed data and the black points are the observed data. The black regression line shows the regression using complete case analysis on the dataset with MCAR missingess, the red regression line is the regression when including the stochastic imputed data and the blue line is the regression line of the complete data, i.e. if no missingness had occurred.

Now to further look at stochastic imputation we will look at another situation , here there will be 4 variables A, B, C and Y where the data of A is generated by a normal distribution with mean 10 and standard deviation 4, Then the variables B and C are generated as a linear function of where $B = 10 + A + N(0, 2)$ and where $C = 3 + A + N(0, 2)$ Then Y is a function of A, B and C , $Y = 6 + 2A + 3B + 4C + \epsilon$ where ϵ $Norm(0, 3)$ which can also be seen as white noise. This has been done so the variables have a covariance to each other i.e. the variables link to each other. In this example we are trying analyse Y which can be taken as the dependent variable where A, B and C are the independent variables.

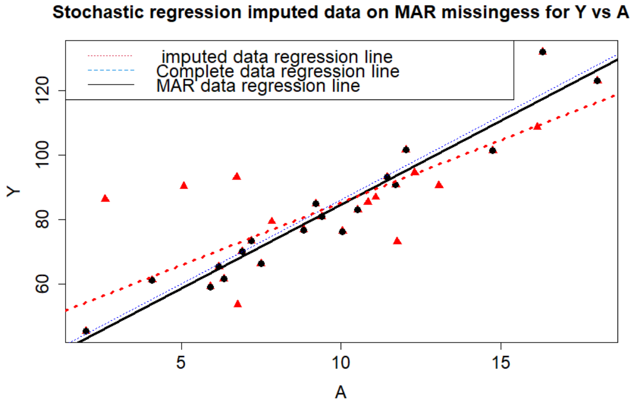


Figure 12: Plot of Y vs A where the red points are the mean imputed data and the black points are the observed data. The black regression line shows the regression using complete case analysis on the dataset with MAR missingess, the red regression line is the regression when including the stochastic imputed data and the blue line is the regression line of the complete data, i.e. if no missingness had occurred.

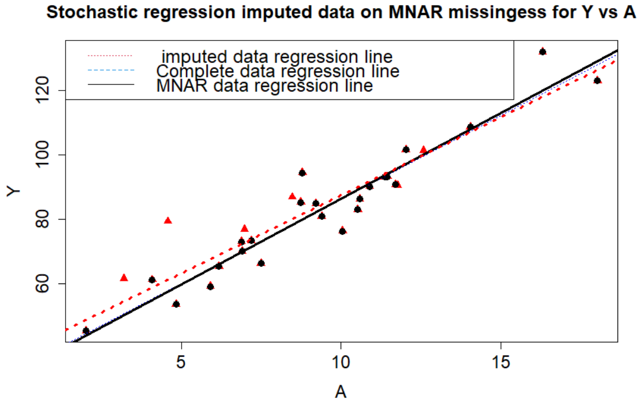


Figure 13: Plot of Y vs A where the red points are the mean imputed data and the black points are the observed data. The black regression line shows the regression using complete case analysis on the dataset with MNAR missingess, the red regression line is the regression when including the stochastic imputed data and the blue line is the regression line of the complete data, i.e. if no missingness had occurred.

3.4.1 1000 Repeats

The next step is that the variables A, B and C undergo MAR, MCAR and MNAR missingness and this was done using the MICE package similarly as before. The dataset was amputed differently 1000 times for each type of missingness and the complete data was to obtain 1000 values of each parameter for each type of missingness. Table 5 shows the real coefficient values of Y before any linear regression or missingness occurs. In this situation we can see how accurate the predicted parameters are after imputing over 1000 datasets with the same parameters. Note, this is not the same as multiple imputation and this will be made clear in the next section. Figures 14-18 show the parameters estimated for the 10000 imputations in box plots and the results show when the dataset has missingness due to MCAR, the spread of the estimated parameter is the least from the true value, which can be seen clearly in Figure 14, 17 and 18. When the data missingness is due to MAR the data looks to have the biggest spread from the true parameter values which can be seen in all the boxplots.

Table 5: Values of the real linear regression parameters, where the values under A, B and C are the coefficients of the linear regression equation.

A	B	C	Intercept	standard deviation
2	3	4	6	3

To further analyse this example, scatter plots were created showing the value estimated for the parameter every time the imputation code ran with their confidence intervals. This was done for every missingness type with the complete datasets too, and for each parameter. We will only focus on the A parameter but all the other parameters had yielded similar results. In Figures 20, 21 and 23 the blue points and lines are "Rejected" which means that the true value of the A coefficient did not lie in the confidence interval of that estimation and red is for "Accepted" which indicates that the true coefficient value does lie in the confidence interval of that estimation. To help analyse these plots Table 6 shows the number of "Accepted" and "Rejected" cases for each type of missingness and also the complete datasets. So when the dataset had missingness due to MCAR there were only 79 out of 1000 cases that were rejected compared to the MAR which had 170. This is similar to our results on the boxplots. This measure of analysis is called "coverage" and will be seen later on in the report. We would like the percentage of accepted cases to be as close to 95% as possible as these are 95% confidence intervals.

Overall stochastic regression imputation is a significant advancement compared to the other methods and the idea of spoiling the regression prediction by adding the random noise gives stochastic regression imputation its strength. Van Buuren [5] states that "a well-executed stochastic regression imputation retains not only the regression weights, but also the correlation between variables".

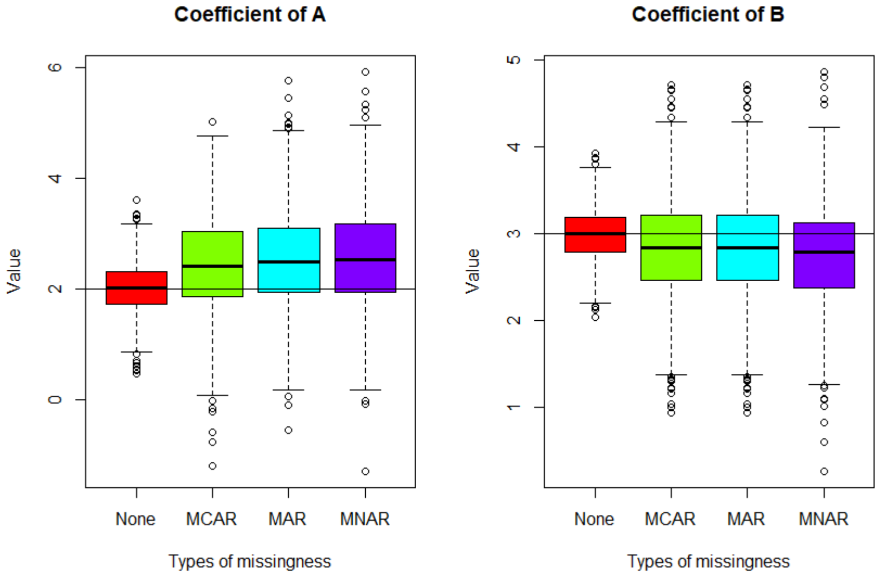


Figure 14: Boxplot of the estimated values for the coefficient of A (left-hand side) and B (right-hand side) for data sets with different types of missingess mechanisms

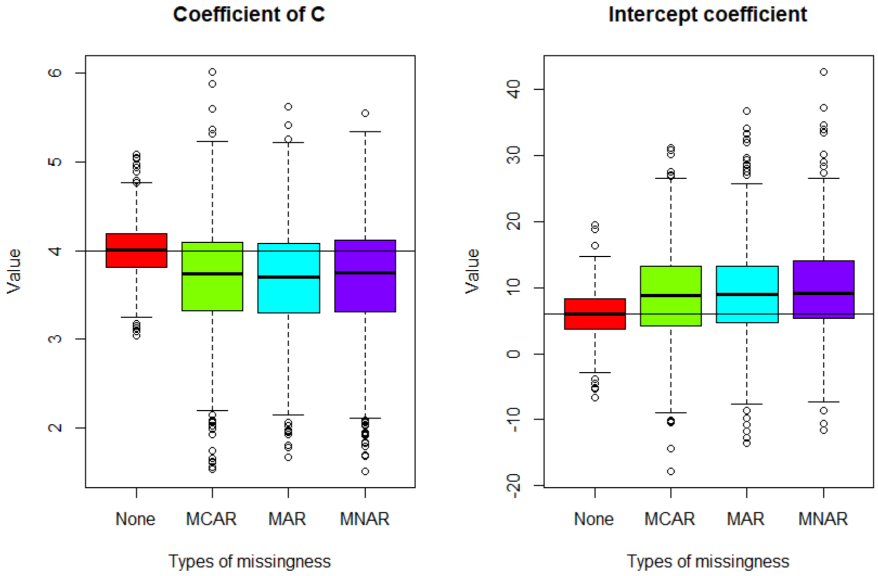


Figure 15: Box plot of the estimated values for the coefficient of C (left-hand side) and the intercept coefficient (right-hand side) for data sets with different types of missingess mechanisms

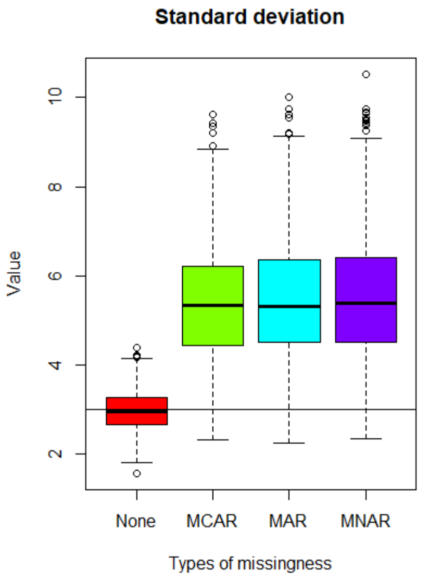


Figure 16: Boxplot of the values for the estimated standard deviation of the regression for data sets with different types of missingess mechanisms

3.5 More Missingness

In Section 3.5 we looked at stochastic regression imputation on a dataset where there was 30% missingness in the dataset. In this small subsection, we will see how increasing the missingness affects the imputation and what the estimated parameters look like with their confidence intervals. Firstly, we'll look at increasing the missingness to 50% and see how imputing using stochastic regression affects the estimated regression parameters. Only the plots showing the confidence intervals for the coefficient of A are shown and the box plots for the values estimated for the coefficient of A and the standard deviation are shown, this is due to the fact the other resulting plots were very similar. Figures 20-22 show the results for 50% missingness in the datasets. In the boxplots, we can see there is more of a spread for the estimated value as in Figure 20 we see the maximum value of A is around 6 while the left-hand boxplot in Figure 15, which shows the values of A from datasets of 30% missingness, we see the max value is around 5, so we see the estimated coefficient value get further away from the true value of 2 as the missingness increases.

This is also shown in the standard deviation, in Figure 16 the amputation on the 30% missingness datasets gives a median of around 5 while the median shown on the right-hand side of Figure 19 shows when the missingness is increased to 50% the median increases to around 7. Also, the spread increases too as the values range from 3 up until just over 10 when missingness is increased to

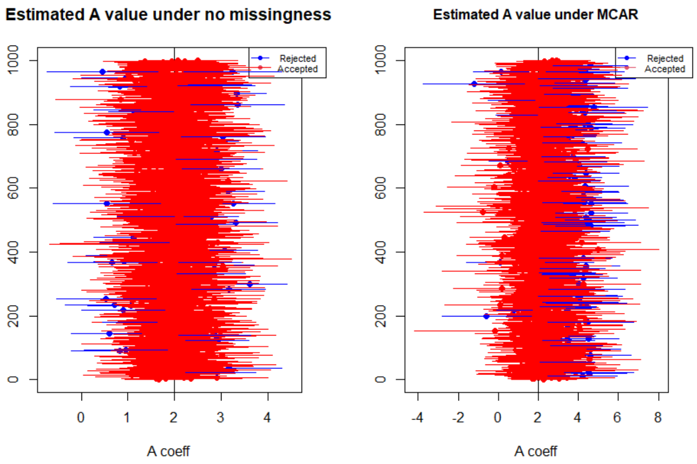


Figure 17: Plot showing the A coefficient estimated using regression on the complete datasets (left-hand side) and when the datasets are under MAR missingess (right-hand side) with the lines showing the confidence interval of that estimation. Blue points and lines are for rejected which means that the true value of the A coefficient did not lie in the confidence interval of that estimation and red for accepted indicates that the true coefficient value does lie in the confidence interval of that estimation

50% while the range of the standard deviation estimated when missingess is 30% is only from 3-9. The plots showing each A coefficient that was with their 95% confidence intervals show more spread in both values estimated and the confidence intervals. Looking at Table 6 there is a clear positive trend of ”rejected” coefficients as missingness increases for all types of missing mechanisms. As you can see in Figures 23 and 24 values the graphs have broader values for the estimated values of the A coefficient compared to Figures 20 and 21 (except for the graph showing the datasets with no missingness of course). Another note is also the confidence intervals have some upper and lower values that are further away from the real value (2) and the confidence intervals themselves are wider when the missingness in the datasets has been increased to 50%. This shows the standard deviation of the estimates is larger when the number of missingness increases. Figures 22-24 display when there is 80% of missingness in the datasets which further back the claims mentioned when the amount of missingness was increased from 30% to 50%. The confidence intervals are even wider in Figures 23 and 24 with more estimated values predicted being further away from the true value of 2 which can also be seen on the boxplots in Figure 22, with the standard deviation values estimated being wider than when missingness was 50% and even more at 30%. These findings suggest that estimating regression parameters become less accurate as the amount of missingness increases as can

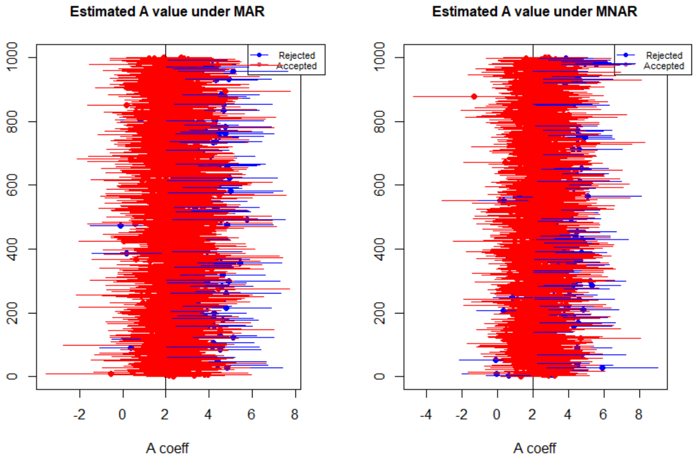


Figure 18: Plot showing the *A* coefficient estimated using regression on the MAR datasets (left-hand side) and the MNAR datasets (right-hand side) with the lines showing the confidence interval of that estimation. Blue points and lines are for rejected which means that the true value of the *A* coefficient did not lie in the confidence interval of that estimation and red for accepted indicates that the true coefficient value does lie in the confidence interval of that estimation

be shown in the decrease in coverage.

Table 6: Table showing the number of ”Accepted” and ”Rejected” cases for the estimated coefficient of *A* for each type of missingness and the complete tables when the datasets under the missing mechanisms have 30%, 50% and 80% missingness. Under the column ”None” there is no missingness at all which is why the numbers are very similar through the different amounts of missingness.

% of missingness		None	MCAR	MAR	MNAR
30%	Rejected	49	98	98	105
	Accepted	951	902	902	895
50%	Rejected	56	114	130	125
	Accepted	944	886	870	875
80%	Rejected	50	176	171	172
	Accepted	950	824	829	828

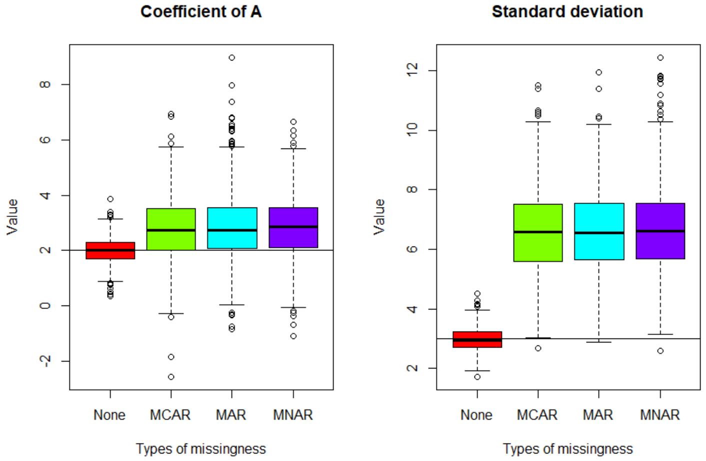


Figure 19: Box plot of the estimated values for the coefficient of A (left-hand side) and the standard deviation of the regression (right-hand side) for data sets with different types of missingess mechanisms and missingness of 50% .

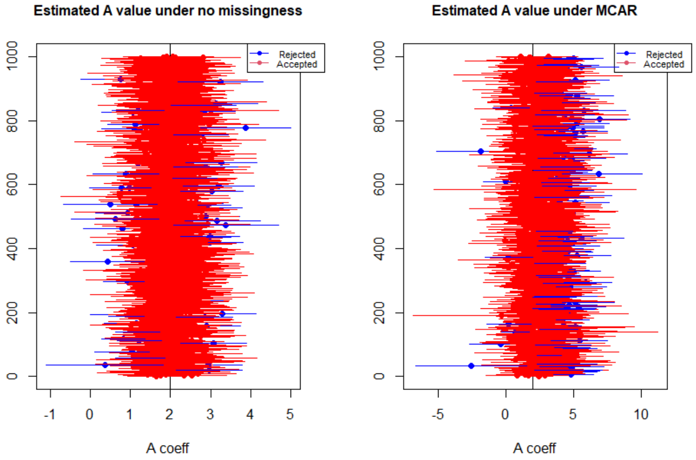


Figure 20: Plot showing the A coefficient estimated using regression on the complete datasets (left-hand side) and when the datasets are under MAR missingess (right-hand side) with the lines showing the confidence interval of that estimation. The datasets with MAR missingness had 50% missingness in their datasets. Blue points and lines are for rejected which means that the true value of the A coefficient did not lie in the confidence interval of that estimation and red for accepted indicates that the true coefficient value does lie in the confidence interval of that estimation.

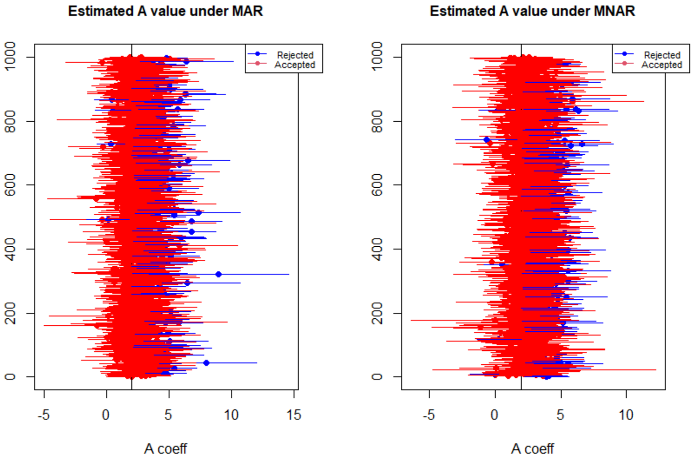


Figure 21: Plot showing the A coefficient estimated using regression on the MAR datasets (left-hand side) and the MNAR data sets (right-hand side) with the lines showing the confidence interval of that estimation.The datasets with MAR missingness had 50% missingness in their datasets. Blue points and lines are for rejected which means that the true value of the A coefficient did not lie in the confidence interval of that estimation and red for accepted indicates that the true coefficient value does lie in the confidence interval of that estimation.

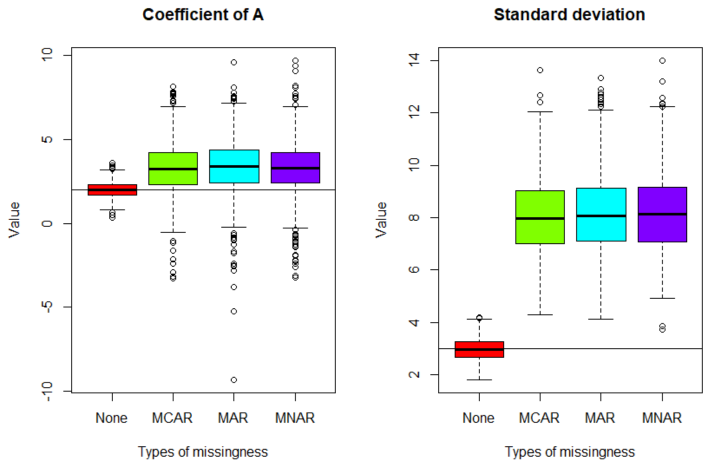


Figure 22: Box plot of the estimated values for the coefficient of A (left-hand side) and the standard deviation of the regression (right-hand side) for data sets with different types of missingess mechanisms and missingness of 80% .

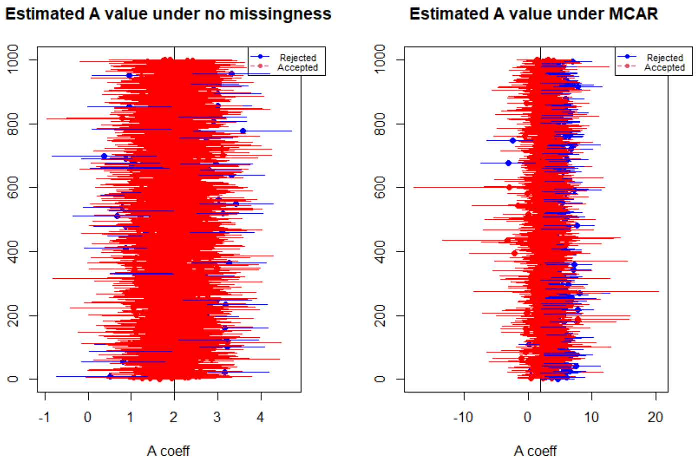


Figure 23: Plot showing the A coefficient estimated using regression on the complete datasets (left-hand side) and when the datasets are under MAR missingess (right-hand side) with the lines showing the confidence interval of that estimation. The datasets with MAR missingness had 80% missingness in their datasets. Blue points and lines are for rejected which means that the true value of the A coefficient did not lie in the confidence interval of that estimation and red for accepted indicates that the true coefficient value does lie in the confidence interval of that estimation.

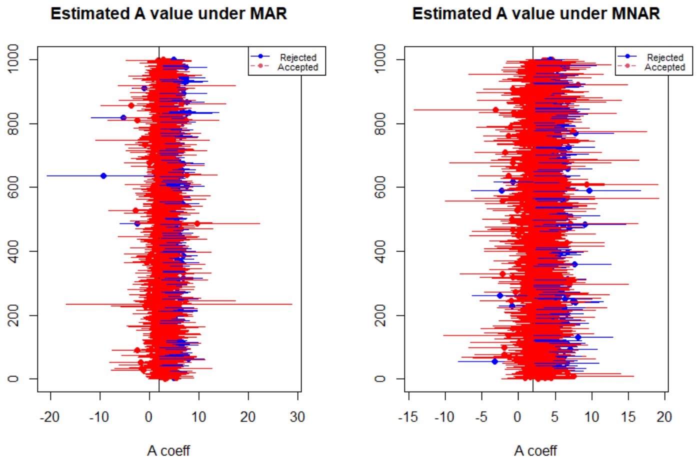


Figure 24: Plot showing the A coefficient estimated using regression on the MAR datasets (left-hand side) and the MNAR data sets (right-hand side) with the lines showing the confidence interval of that estimation.The datasets with MAR missingness had 80% missingness in their datasets. Blue points and lines are for rejected which means that the true value of the A coefficient did not lie in the confidence interval of that estimation and red for accepted indicates that the true coefficient value does lie in the confidence interval of that estimation.

4 Multiple Imputation

This section is based on Van Buuren [5]

First proposed by Rubin [8], multiple imputation is a technique that substitutes each missing value with a chosen number of imputed values. Rubin noticed that single imputation could not, in most cases, be accurate. He observed that even for a given model, it was impossible to calculate the imputed values with complete precision. He needed a model to connect the unobserved data to the observed data. Creating several imputations that reflect the uncertainty of the missing data was his massive breakthrough. Also, multiple imputation deals with the uncertainty of the imputed values, with Rubins rules as we will see later on, which is not the case with the single imputation methods we have already seen. Deciding the models and determining the imputations are both covered in the 1977 report and even a low amount of imputations like five imputations can be deemed enough to create a set of "realistic" imputations. Several versions of the data being created must have seemed odd at the time. Instead of estimating the "best" value, drawing imputations from a distribution constituted a major shift from what was being done at the time. The method's methodological and statistical foundation was presented by Rubin [9]. Although there have been a number of advancements since the book was published in 1987, it was truly ahead of its time and covered the fundamentals of current imputation technology. It also defines the conditions under which statistical inference under multiple imputations will be valid and provides the formulas required to combine the repeated complete-data estimates (now known as Rubin's rules).

4.1 Purpose of multiple imputation

The following subsection is inspired by Van Buuren [5].

A scientific estimand θ is something that an analyst would be interested in that we could hypothetically calculate if the full population of data was available. For instance, we would be curious about the population's average height. The population data can be used to express θ as a known function in general. θ will be a vector if we are interested in many variables. θ is a characteristic of the population, such as regression coefficient or the mean of population, hence it is important to keep in mind that it is independent of the method of multiple imputation.

θ can only be calculated if we have the population data however, this is very unlikely in most situations. So the goal of multiple imputation (or imputation in general) is to estimate $\hat{\theta}$ that is *unbiased* and *confidence valid* [10].

Unbiasedness is defined as the mean of $\hat{\theta}$ over all possibilities that might be drawn from the population being equal to θ. This can be written as

$$E(\hat{\theta}|Y) = \theta. \tag{7}$$

Let U represent the estimated variance-covariance matrix of θ. If the average of U across all different cases is equal to or greater than the variance of θ, then this estimate is *confidence valid*. This can be written as

$$E(U|Y) \geq V(\hat{\theta}|Y) \tag{8}$$

where the function $V(\hat{\theta}|Y)$ signifies the variance caused by the sampling process. When the null hypothesis is true, a statistical test with a chosen rejection rate of 5% should reject it in no more than 5% of the scenarios. If this is true, the approach is said to be confidence valid. In conclusion, the purpose of multiple imputation is to obtain population estimates of the scientific estimand, θ. This estimate should, on average, match the population parameter's value. Also, at least the given chosen value should be achieved via the associated confidence intervals and hypothesis tests.

The imputed data should be such that it contains specific qualities and these imputations are called *proper*[9]. Equations (7) and (8) state the requirements for the estimate of θ calculated from a hypothetical complete sample, to be valid so a vital theory is that if the imputation is proper and the complete data model is valid as to the condition from Equation (7) and (8) then the whole procedure is valid. There are 3 requirements that make imputation proper, and these state that for imputation to be *confidence proper* for complete data statistics $(\hat{\theta}, U)$, $\hat{\theta}$ is the under a higher number of imputations i.e. a large m all the requirements must be met which are:

$$E(\bar{\theta}|Y) = \hat{\theta} \tag{9}$$

$$E(\bar{U}|Y) = U \tag{10}$$

$$\left(1 + \frac{1}{m}\right) E(B|Y) \geq V(\bar{\theta}) \tag{11}$$

where $\hat{\theta}$ is the estimand of θ and similarly $\hat{U}$ is an estimand of $\bar{U}$. Equations (10) and (11) state that the multiple imputation estimates $(\bar{\theta}, \bar{U})$ are unbiased estimates of the statistics $(\bar{\theta}, U)$ of the hypothetical complete dataset. While the last condition is due to the extra uncertainty about θ due to missing data being correctly reflected.

Equation (9) and (10) states that $\bar{\theta}$ amd $\bar{U}$ should be an unbiased estimate of $\hat{\theta}$ and U and Equation (11) suggests that the extra uncertainty about $\hat{\theta}$ is taken into account .

4.2 Rubin's Rules

As already mentioned the goal of multiple imputation is to arrive at a prediction or estimate of θ which can be labelled as θ. The level of uncertainty of $\hat{\theta}$ depends on the missing data Y_{miss} and as one will not be able to create the missing data perfectly so we need to summarize the distribution of θ under Y_{miss}. The posterior distribution $Pr(\theta \mid Y_{obs})$ contains the possible values of θ given the data from Y_{obs} and this can be separated into 2 parts in the following:

$$Pr(\theta|Y_{\text{obs}}) = \int Pr(\theta|Y_{\text{obs}}, Y_{\text{miss}}) Pr(Y_{\text{miss}}|Y_{\text{obs}}) dY_{\text{miss}} \tag{12}$$

We want to know $Pr(\theta|Y_{\text{obs}})$. $Pr(\theta|Y_{\text{obs}}, Y_{\text{miss}})$ is the posterior distribution of the hypothetical complete data and $Pr(Y_{\text{miss}}|Y_{\text{obs}})$ is the posterior distribution of the missing data given the observed data.

This then leads to the mean of $Pr(\theta|Y_{\text{obs}})$ is equal to

$$E(\theta|Y_{\text{obs}}) = E(E[\theta|Y_{\text{obs}}, Y_{\text{miss}}]|Y_{\text{obs}}) \tag{13}$$

which is the average of the θ's calculated from the number of imputations from the observed data. So this is basically the mean of all the parameters calculated. This leads to Rubin's Rules.

Rubin's Rules state for the estimate for the parameter vector $\hat{\theta}$ containing k parameters, the multiple imputation (MI) estimators of are θ

$$\overline{\theta} = \frac{1}{m}\sum_{\ell=1}^{m}\hat{\theta}_{\ell}, \tag{14}$$

where $\hat{\theta}_l$ is the estimate of the l^{th} imputation which contains k parameters and is represented as $k \cdot 1$ column vector. The total variance estimator is written as

$$\hat{V_{MI}} = \hat{U} + (1 + \frac{1}{m})\hat{B} \tag{15}$$

where

$$\bar{U} = \frac{1}{m}\sum_{\ell=1}^{m}\bar{U}_{\ell} \tag{16}$$

which is the average of the complete-data variances, $\bar{U}_{\ell}$ is the variance-covariance matrix of $\hat{\theta}_{\ell}$ obtained for the ℓ^{th} imputation.

$$B = \frac{1}{m-1}\sum_{\ell=1}^{m}(\hat{\theta}_{\ell} - \bar{\theta})(\hat{\theta}_{\ell} - \bar{\theta})' \tag{17}$$

Which is the standard unbiased estimate of the variance between the m complete-data estimates and $(\hat{\theta}_{\ell} - \bar{\theta})'$ is the complex conjugate of $(\hat{\theta}_{\ell} - \bar{\theta})$ The total variance V_{MI} is from 3 sources:

1. $\bar{U}$, the variance brought on by the fact that we are only looking at a sample and not the entire population and can be called *within-imputation variance*[3]. As you can see this is the usual formula for the sample mean that acts as an average of each imputation's variance so U estimates the sample variance if there was no missingness.

2. B, the additional variance brought on by the sample's missing values which can be called the *between-imputation variance* and is caused by the range of different estimated parameter values from m imputations which estimates the extra spread of variance from the missing data [3] .

3. B/m, the extra simulation variance caused by the fact that $\bar{\theta}$ itself is estimated for finite m and acts as the correction factor [3] .

The latter term must be added to make multiple imputations operate for low values of m. without this, the p-values or confidence intervals would be too low and then common values for m are $m = 3$, $m = 5$, and $m = 10$ could be used.

Table 7: Simple example of Multiple imputation; imputed values in bold

Row no.	B	B_0	Values imputed in imputation (bold)			
			1	2	3	4
1	15.43	15.43	15.43	15.43	15.43	15.43
2	30.45	NA	**26.12**	**26.49**	**22.42**	**26.18**
...						
14	16.86	NA	**18.04**	**18.78**	**19.56**	**19.55**
...						
18	21.74	NA	**24.01**	**21.74**	**21.74**	**21.74**
19	23.49	NA	**21.74**	**23.49**	**23.49**	**23.49**
20	17.84	NA	**23.49**	**17.84**	**17.84**	**17.84**
...						
23	24.96	NA	**24.96**	**24.96**	**24.96**	**24.96**
...						
28	17.90	NA	**20.00**	**18.57**	**22.01**	**19.84**
...						
30	22.07	NA	**18.90**	**20.39**	**21.04**	**20.27**
Mean	19.31	19.01	19.07	19.27	18.97	19.23
SE	4.24	4.34	3.79	3.86	3.74	3.85

Example 4.1. Let us look at some variables of a data set that we have already looked at. From Chapter 3 where the variables A_i, where $i = 1, ..., 30$, $A_i \sim N(10, 16)$ and generate B (not to be confused to B from Rubin's rules)

$$B_i = 10 + A_i + \epsilon_i, \quad \epsilon_i \overset{\text{iid}}{\sim} N(0, 4).$$

Suppose we are trying to estimate the mean of B , but B is MCAR dependent on A so that around 30% of the observation on B are missing. This gives 7 missing observations that need to be imputed which can be seen in Table 7. To impute the missing values , the imputation model will be a stochastic regression model of B on A,

$$B_i = \alpha_0 + \alpha_1 A_i + \epsilon_i, \quad \epsilon_i \overset{\text{iid}}{\sim} N(0, \sigma^2). \tag{18}$$

From Equation (14) we get the marginal mean of B to be

$$\hat{\bar{B}}_M I = (19.07 + 19.27 + 18.97 + 19.23)/4 = 19.125$$

and to work out the standard error we first calculate

$$\hat{U} = (3.79^2 + 3.86^2 + 3.74^2 + 3.85^2)/4 = 14.51845 \tag{19}$$

and

$$B = \frac{1}{m-1}\sum_{\ell=1}^{m}(\hat{\theta}_\ell - \bar{\theta})^2 \tag{20}$$

due to there only being one independent variable in the regression equation, so

$$\begin{aligned} B = ((19.07 - 19.125)^2 + (19.27 - 19.125)^2 + (18.97 - 19.125)^2 \\ + (18.97 - 19.125)^2 + (19.23 - 19.125)^2)/3 = 0.0277083 \end{aligned} \tag{21}$$

so this gives the standard error as

$$\sqrt{V_{MI}} = \sqrt{14.151845 + (1 + \frac{1}{4}) \cdot 0.0277083} = 3.76 \quad 2 \text{ d.p..} \tag{22}$$

Here multiple imputation has decreased the standard error from the observed values and calculated a mean that is closer to the actual mean.

4.3 Variance Ratios

This section is from Van Buuren [5] For the single estimand θ , the ratio

$$\lambda = \frac{B + B/m}{T} \tag{23}$$

is the ratio of the change caused by the missingness. When 0 this means the missing data does cause any deviation to the sampling variance, however, this should not happen or may only happen when the missingness is unrealistically replaced with values that would have been the same as their hypothetical original values while 1 means that the whole sample variance is from the missingness in the data which is also just as extraordinary. Higher values of λ, such as $\lambda > 0.5$ suggest that the imputations are actually more important to the analysis of the completed dataset compared to th eactual observed data.

Rubin states the *relative increase in variance due to nonresponse* which can be written as

$$r = \frac{B + B/m}{\bar{U}} \tag{24}$$

is related to λ by $r = \lambda/(1-\lambda)$.

Another measure related to λ however for a finite number of imputations is γ which can be shown as

$$\gamma = \frac{r + 2/(\nu + 3)}{1 + r} \tag{25}$$

Where ν is the degrees of freedom which will be discussed soon. γ can also be written as

$$\gamma = \frac{\nu + 1}{\nu + 3}\lambda + \frac{2}{\nu + 3} \tag{26}$$

As one may be looking at more than estimand such that θ would be a vector we can calculate a single $\hat{\lambda}$ for all estimands in $\hat{\theta}$ as

$$\bar{\lambda} = \left(1 + \frac{1}{m}\right) \mathrm{tr}(BT^{-1})/k \tag{27}$$

where k is the number of estimands in θ and where B and T are $k \sim k$ matrices and r can be written as

$$\bar{r} = \left(1 + \frac{1}{m}\right) \mathrm{tr}(B\bar{U}^{-1})/k \tag{28}$$

which is the mean of the increase in variance.

These measures discussed show the gravity of the missing data problem for example missingness of up to 0.2 is classed as small, 0.3 is somewhat extensive and 0.5 is seen as large according to Li, Ragathnatahn and Rubin. The bigger the values the harder the problem of how the final statistical analysis is due to the missingness is dealt with as it is unrealistic to estimate good imputed data with such little information.

4.3.1 Degrees of freedom

In statistics we take the degrees of freedom as the number of independent pieces (data) used to calculate a statistic so if data is missing then the degrees of freedom for the complete data has to be greater than the observed data. Rubin first wrote the degrees of freedoms to be used for testing in multiple imputation as

$$\begin{aligned}\nu_{\text{old}} &= (m-1)\left(1 + \frac{1}{r^2}\right) \\ &= \frac{m-1}{\lambda^2}.\end{aligned} \tag{29}$$

Here the lowest possible is $\nu_{\text{old}} = m-1$ which is when the variance is due to the missing data. The flaw with this equation is that $\nu_{\text{old}} = \infty$ is possible which is when the variation is due to the observed data only which cannot be a reality. So an alteration was made. For this we let ν_{com} be the degrees of freedom of $\bar{\theta}$ in the hypothetical complete data. If there are k parameters to fit with a sample size of n then $\nu_{\text{com}} = n - k$. The estimated observed data degrees of freedom that acknowledge the missingness is

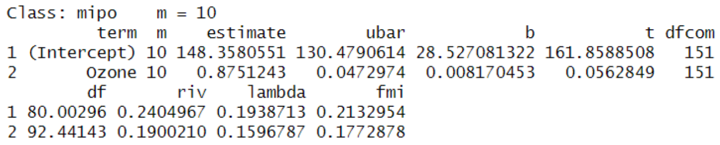
$$\nu_{\text{obs}} = \frac{\nu_{\text{com}} + 1}{\nu_{\text{com}} + 3} \nu_{\text{com}}(1 - \lambda) \tag{30}$$

making the degrees of freedom that will be used when the multiple imputation procedure is tested as

$$\nu = \frac{\nu_{\text{old}} \nu_{\text{obs}}}{\nu_{\text{old}} + \nu_{\text{obs}}}. \tag{31}$$

This makes sure that $\nu \leq \nu_{\text{com}}$ and if $\nu_{\text{com}} = \infty$ then Equation (31) becomes Equation (29). If ν is less than 1 than the imputation methods should no tested as there isn't enough information.

Example 4.2. If we look at the earlier situation in Example 4.1 we can look at these measures that have been introduced in the last few sections. If we do a linear regression of B on A and use the *pool()* function from *mice* fits the model and pools the results which can be shown in Figure 25. The column estimate is the value of $\bar{\theta}$ as defined in Equation(14) columns ubar, b and t are the estimates for the variance from Equations(15-17). The dfcom column shows the values of ν_{com}, the df column are the values for ν and the last 3 columns hold the values for $r, \lambda,$ and γ.



```
Class: mipo    m = 10
         term  m    estimate        ubar           b           t dfcom
1 (Intercept) 10 148.3580551 130.4790614 28.527081322 161.8588508   151
2       Ozone 10   0.8751243   0.0472974  0.008170453   0.0562849   151
        df       riv    lambda       fmi
1 80.00296 0.2404967 0.1938713 0.2132954
2 92.44143 0.1900210 0.1596787 0.1772878
```

Figure 25: Output from R showing the linear model results

4.4 Scalar Inference

The aim of multiple imputation is to give good statistical inferences from the observed data that contains missingness. For a scalar θ one could easily obtain the confidence intervals and p-values from the multiple imputed data. However, if θ contains multiple parameters there are 2 methods for evaluating the data. First, is to calculate the confidence intervals and the p-values which is suitable if each parameter is independent of the other while the other option is to use one statistical test involving all the parameters in θ simultaneously which is more suitable when all the parameters are being analysed together such as a regression equation containing all the parameters. The both methods take $\hat{\theta}$ to be normally distributed around the population value θ as

$$\hat{\theta} \sim N(\theta, U) \tag{32}$$

where U is the variance-covariance matrix of $(\theta - \hat{\theta})$.

Scalar parameter inference is used when there is only 1 parameter. $\bar{\theta}$ follows a t-distribution rather than a normal distribution as V_{MI} , the total variance, is unknown. The univariate tests are based on the approximation

$$\frac{\theta - \bar{\theta}}{\sqrt{V_{MI}}} \sim t_{\nu} \tag{33}$$

where the t_{ν} is from the t-distribution with ν degrees of freedom where ν was stated in Equation (31). The chosen $(100 - \alpha)\%$ confidence interval is $\bar{\theta}$ is

$$\bar{\theta} \pm t_{\nu,1-\alpha/2}\sqrt{V_{MI}}$$

where $t_{\nu,1-\alpha/2}$ is the quantile corresponding to probability $1 - \alpha/2$.

If the null hypothesis $\theta = \theta_0$ for a estimated value of θ_0. The p-value for the null would be

$$P_s = \Pr\left[F_{1,\nu} > \frac{(\theta_0 - \bar{\theta})^2}{V_{MI}}\right]$$

where $F_{1,\nu}$ is an F-distribution with 1 and ν degrees of freedom.

4.5 How many imputations?

We need to discuss the number of imputations needed for a 'good' process. Rubin [9] states that as few as 2-10 are needed for the value of m i.e. the number of imputations and this involves *the fraction of missing information* γ_0. We know the total variance of the estimated parameter from multiple imputation is $\hat{V}_{MI} = \hat{U} + (1 + \frac{1}{m}\hat{B}$. If m was set to ∞ this would get rid of the error from $\frac{\hat{B}}{m}$) simulation error and this total variance would be V_∞ and so what would the V_{MI} , the total variance of m imputations ,be compared to this. Rubin showed

$$V_{MI} = (1 + \frac{\gamma_0}{m})V_\infty \tag{34}$$

γ_0 can be described as the amount of information missing of Y is a single variable and is less than this when dealing with multiple variables. If we have $\gamma_0 = 0.4$ (so if Y is a single variable with 40% of its values not observed) and $m = 5$ this would $V_{MI} = (1 + \frac{0.4}{5})V_\infty = 1.08V_\infty$ so our total variance for this procedure would be 8% larger than variance that we would like. The confidence interval of this would be $\sqrt{1.08} = 1.04$ times wider than the ideal confidence interval if we had taken m as ∞. If we compare this if we increase m to 10 or 20 the confidence interval would only be just under 2% less wide making the increase of imputations negligible. However, increasing m would mean the computer would have to store more information and if m was 20 instead of 5 this would mean storing 4 times the amount of data for only a negligible improvement to the process therefore making the claim keeping m low is ideal.

The following paragraph is based on White, Royston and Wood's paper[11].

However, This alone isn't sufficient enough to choose the value of m as we would want to be sure that an analysis of data could be produced, giving the same conclusions. Therefore, we should consider the Monte Carlo error which can be defined as the standard deviation over all the simulations for the same imputation approach using the same data. The Monte Carlo error is for the estimates including their p-values, their confidence intervals and standard errors and can be easily calculated if Y has one variable as $\sqrt{B/m}$. Generally, a rule of thumb is used by Von Hippel[12]:*the number of imputations should be similar to the percentage of cases that are incomplete.* For a parameter β if m is such that $m \approx 100\gamma$ then:

1. The Monte Carlo error of $\hat{\beta}$ is approximately 10 per cent of its standard error.

2. The Monte Carlo error of the test statistic $\hat{\beta}/se(\hat{\beta})$ is approximately 0.1 (se is the standard error).

3. The Monte Carlo error of the p-value is approximately 0.01 when the true p-value is 0.05, and 0.02 when the true p-value is 0.1.

White, Royston and Wood [11] believe that these conditions make sure that results can be reproduced to a good enough standard so much so that it has become used throughout research. However, knowing the percentage of missing data can be unknown making it hard to estimate the Monte Carlo effect. Overall, it is logical to want to use a higher value of m however, due to the computational power and storage used it may not be wise in the situation. So for cases where a high m is suitable like when there is only one estimand to estimate but if one is not interested in the standard errors and p-values then a lower amount of imputations, say 5-20 is enough.

5 Imputation methods

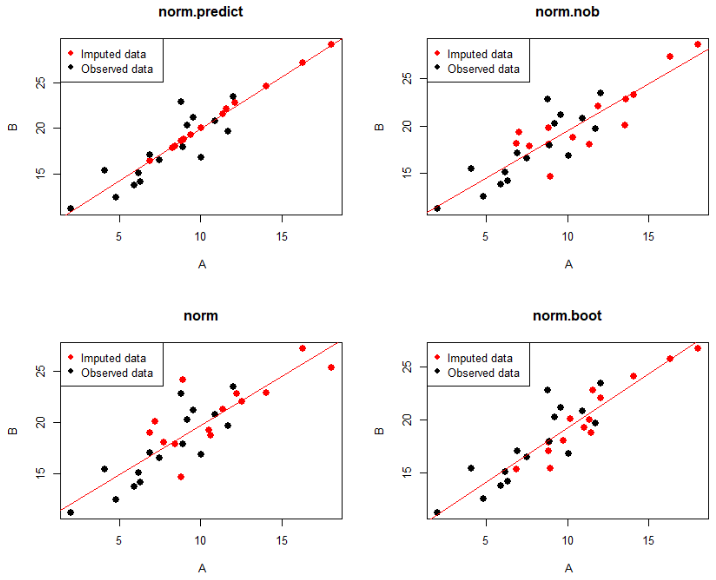


Figure 26: Figure showing the 4 types of imputation methods in mice package in R, norm.predict which is the regression imputation, norm

This section is based upon Van Buuren [5]

In this section, we will begin discussing which methods of imputation that can be used in the multiple imputation process. To simplify these methods for understanding we will be restricting our dataset as univariate missing data so we will have a univariate vector Y which is the variable with missing data which is also called the *target variable.* Y has n entries where n_1 entries have observations and n_0 entries have no observation i.e. missing, such that $n_1 + n_0 = n$. The predictors for the imputation will be in X which can be divided to $(X_{\text{obs}}, X_{\text{miss}})$ where X_{obs} depicts the entries of Y that are observed (with n_1 accounts) and X_{miss} accounts for the missingness in Y and is also the complement of $_{\text{obs}}$.

Here we will discuss 4 different of imputation methods all under the normal linear model. These are:

1. *Predict.* This has already been discussed in Section 2.3 and was called *regression imputation.* So here we predict $\dot{Y} = \hat{\beta}_0 + \hat{\beta}_1 X_{\text{miss}}$, where $\hat{\beta}_1, \hat{\beta}_0$ are least squares estimates from the observed data. In the package "mice" in R this method is called "norm.predict"

2. *Predict + noise.* This method has also been discussed in Section 2.4 and can be called *stochastic imputation.* Here $\dot{Y} = \hat{\beta}_0 + \hat{\beta}_1 X_{\text{miss}} + \dot{\epsilon}$, where $\hat{\beta}_1, \hat{\beta}_0$ are least squares estimates from the observed data and $\dot{\epsilon}$ is drawn from the normal distribution as $\dot{\epsilon} \sim N(0, \hat{\sigma}^2)$ where $\hat{\sigma}^2$ is the standard deviation of the observed data. This method in R is available as "norm" in the "mice" package.

3. *Bayesian multiple imputation.* This method has yet to be introduced and unlike the first 2 methods, the parameters $\hat{\beta}_0, \hat{\beta}_1 and \hat{\sigma}$ are not known so are taken from their posterior distribution from the data. The reason for this change is if were to do a complete case analysis on the observed data , the estimated parameters may be different if we had a different observed sample from the same population. Also , the variance is related to the number of observations with an increase of variability if the number of observations is smaller .So the predicted $\dot{Y} = \dot{\beta}_0 + X_{\text{miss}}\dot{\beta}_1 + \dot{\epsilon}$ where $\dot{\epsilon} \sim N(0, \dot{\sigma}^2)$ and $\dot{\beta}_0$ and $\dot{\beta}_1$ and $\dot{\sigma}$ are picked randomly from their prior distribution. This method is known as "norm" in the "mice" package in R.

4. *Boostrap multiple imputation.* Here $\dot{Y} = \dot{\beta}_0 + X_{\text{miss}}\dot{\beta}_1 + \dot{\epsilon}$ where here $\dot{\epsilon} \sim N(0, \dot{\sigma}^2)$ and $\dot{\beta}_0$ and $\dot{\beta}_1$ and $\dot{\sigma}$ are the least squared estimates calculated from a bootstrap sample taken from the data that has been observed as mentioned in Van Buuren Section 3.2 [5] and this method in "mice" is "norm.boot".

Figure 26 shows an example of all these types of imputation and how they look when graphed. The dataset is taken from Example 4.1 however instead there is 50% of missingness.

5.1 Methodology

Now we will discuss in detail the methods mentioned above. The first 2 have already been described in Sections 2.3 and 2.4 and are quite simple to do. We will start with the Bayesian imputation method but before that should state the inputs needed for these methods. These are:

- Y_{obs}, the $n_1 \times 1$ vector containing the observed data in the variable Y
- X_{obs}, the $n_1 \times q$ matrix of the independent variables of rows with the observed data in Y
- X_{miss}, the $n_0 \times q$ matrix of independent variables of rows with the missing data in Y.

Here the algorithm assumes there is no missingness in X_{obs} and X_{miss}.

Now this algorithm that is to be described is taken from Rubin[9]. Here this algorithm predicts the univariate Y from some independent variables X from a normal linear regression model so

$$Y \sim N(X_\beta, \sigma^2)$$

Where β is a vector of q components and σ is a fixed scalar. We also assume $n_1 > q$. Gep and Tiao [13] explained the Bayesian theory behind the normal linear model.

Algorithm 1 (Bayesian imputation under the linear model)

1. Calculate $S = X'_{\text{obs}} X_{\text{obs}}$.
2. Calculate $V + (S + diag(S)\kappa)^{-}1$, with some small ridge parameter κ.
3. Calculate the regression weights $\hat{\beta} = V X'_{obs} Y_{obs}$.
4. let $g = \chi^2_{n_1 - q}$ and draw a random variable from this distribution.
5. Calculate $\dot{\sigma}^2 = (Y_{\text{obs}} - X_{\text{obs}}\hat{\beta})'(Y_{\text{obs}} - X_{\text{obs}}\hat{\beta})/\dot{g}$ where $(Y_{\text{obs}} - X_{\text{obs}}\hat{\beta}')$ is the matrix transpose of $Y_{\text{obs}} - X_{\text{obs}}\hat{\beta})$.
6. create a vector Z_1 with q elements by drawing from independent $N(0,1)$ distribution and let
$$\dot{\beta} = \hat{\beta} + \dot{\sigma} Z_1 V^{1/2}$$
where $V^{1/2}$ is calculated from Cholesky decomposition.
7. Draw the n_0 values of Y_{miss} from independent $N(0,1)$ and store in vector Z_2
8. Finally calculate the values to replace the n_0 values of Y_{miss} with
$$Y_{\text{imp}} = X_{\text{miss}}\dot{\beta} + \dot{Z}_2\dot{\sigma}.$$

This algorithm is adapted from Rubin by Van Buuren [5] and is the algorithm used in the method "norm". The ridge parameter κ is so there isn't a singular matrix in V in step 3 and generally this value should be a positive close number to 0 such as 0.0000001. If we are imputing Y_{miss} m times then m draws of the parameter $\dot{\sigma}^2$ are needed so this step needs to be repeated m times.

Algorithm 2 (Imputation with bootstrapping under the linear model) Bootstrapping is a technique that produces repeated samples with replacement from a single original sample. These samples are used to calculate estimates of estimands of interests and were introduced by Bradley Efron in his 1979 paper "Bootstrap Methods: Another Look at the Jackknife"[14]. The fundamental notion underlying bootstrapping is to produce a distribution of sample statistics by repeatedly sampling from the original data, which can then be used to estimate the distribution of the population parameter of interest. Van Buuren[5] gives an algorithm for the imputation with bootstrap in Section 3.2 as follows:

1. Draw a bootstrap sample $(\dot{Y}_{\text{obs}}, \dot{X}_{\text{obs}})$ of size n_1 from $(Y_{\text{obs}}, X_{\text{obs}})$
2. Calculate $S = X\prime_{\text{obs}} X_{\text{obs}}$.
3. Calculate $V + (S + diag(S)\kappa)^{-}1$, with some small ridge parameter . κ.
4. Calculate the regression weights $\hat{\beta} = VX\prime_{obs}Y_{obs}$.
5. Calculate $\dot{\sigma}^2 = (\dot{y}_{\text{obs}} - \dot{X}_{\text{obs}}\dot{\beta})\prime(\dot{y}_{\text{obs}} - \dot{X}_{\text{obs}}\dot{\beta})/(n_1 - q - 1)$
6. Finally calculate the values to replace the n_0 values of Y_{miss} with
$$Y_{\text{imp}} = X_{\text{miss}}\dot{\beta} + \dot{Z}_2\dot{\sigma}.$$

The difference between the 2 algorithms is that there is no need to use a square root of V by Cholesky factorization and we don't use $\dot{g}$ from the χ^2 distribution.

5.1.1 Performance of the Algorithms

To test these imputation methods we will 3 measures to tell us about the test validity of a procedure used. These are:

1. *Raw bias (RB)* and *Percentage bias (PB)*. If one is interested in the parameter θ, the RB is $100 \times (E\hat{\theta} - \theta/SE(\hat{\theta})$, ($SE$ is the standard error. PB is defined as $100 \times \left| E(\hat{\theta}) - \theta/SE(\hat{\theta}) \right|$.
2. *Coverage Rate (CR).* This is the percentage of times that the actual (true) parameter value is in the confidence interval that has been calculated. A good process gives a coverage rate close to the percentage used for the confidence interval. Of course we do not know the true value for the population.
3. *Average width of confidence interval (AW).* The gap between the average upper and lower limits across 1000 confidence intervals. The narrower this is the better the process with a high CR.

Table 8: Properties of β_1 under imputation of missing y by 5 methods of normal linear model (1000 simulations)

Method	Bias	/% Bias	Coverage	CI width
Regression	0.0000	0.0	0.652	0.114
Stochastic	-0.0001	0.0	0.908	0.226
Bayesian	-0.0001	0.0	0.951	0.314
Bootstrapping	-0.0001	0.0	0.941	0.299
Complete-case analysis	0.0001	0.0	0.946	0.251

Example 5.1. To show how each imputation method performs in relation to the measures discussed above we will look at an example that can be found in Van Buuren [5] in Section 3.1. The dataset that was investigated represents the average external temperature (in °C) and the average gas consumption for the week (in 1000 cubic metres) for 26 weeks before and 30 weeks after cavity-wall insulation had been installed. The thermostat had a controlled temperature set at 20°C. Let y be the gas consumption in 1000 cubic feet and x be the temperature recorded in °C then the linear model on the data would mean $y = \beta_0 + \beta_1 x$ giving $\beta_0 = 5.49$ and $\beta_1 = -0.29$. The other parameter that will be needed to calculate the above measures is the standard deviation, $\sigma = 0.86$. So to keep close to the data these values are taken to be the true data. Then some random missingness is caused on the datasets then using each of the 4 imputation methods mentioned in Section 5.1 Raw Bias, Percentage Bias, Coverage rate and average Confidence interval width were calculated for each method.

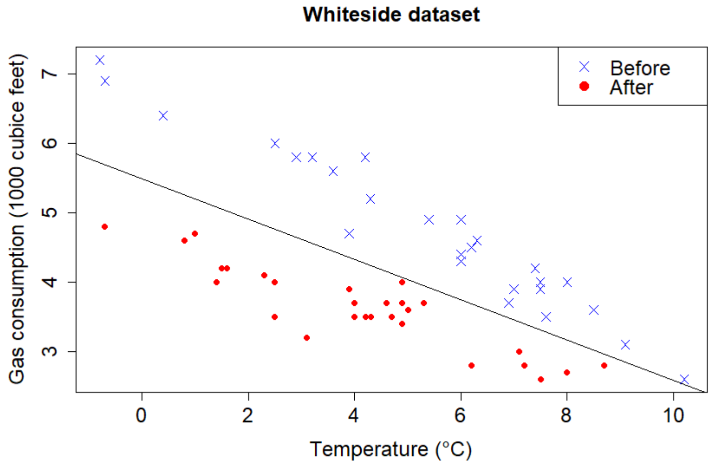


Figure 27: Plot showing the data Whiteside had recorded, the Gas consumption and Temperature before and after the cavity wall insulation had been inserted into his house.

Table 9: Properties of β_1 under imputation of missing x by five methods for the normal linear model.

Method	Bias	% Bias	Coverage	CI Width
Regression	-0.1007	34.7	0.359	0.160
Stochastic	0.0006	0.2	0.924	0.202
Bayesian	0.0075	2.6	0.955	0.254
Bootstrapping	-0.0014	0.5	0.946	0.238
Complete Case	-0.0001	0.0	0.946	0.251

Table 8 summarizes the result when there is 50% of random missing data in y and the number of imputations, $m = 5$. For the coefficient of x, β_1 the results all show no bias. If we look at the coverage we see the coverage for imputation using regression is very low, showing this is due to the much smaller average confidence interval width which is a general result of using regression imputation. Stochastic regression shows better results in these aspects however the coverage is still undesirable for 95% confidence intervals the hope is that the coverage is as close to this number as possible. The other 3 methods Bayesian, bootstrapping and complete case analysis produces results in which all 3 methods can be deemed suitable. Complete case analysis is actually deemed to be the best method [2] and would be the most efficient due to its shortest average confidence interval. As mentioned in Section 3 we know this is a rare occurrence and in more complicated circumstances the multiple imputation methods will show their strength in producing more quality results.

An alternative approach to the above was also taken, here instead of there being missingness in y, there was missingness in x and the results of this can be seen in Table 9. Imputation has shown an increase in bias. The coverage is still below the desired percentage, for stochastic, due to the confidence interval being too short. The other 3 methods show very little bias so could all be deemed as appropriate methods so due to efficiency complete case analysis would be the best choice.

6 Multivarite imputation

The following section is incomplete due to time constraints

6.1 Summary Statistics

The missing data pattern, from section 2, can show and determine how much information can be linked to the measured variables. Imputation is more accurate if there is observed data for the other variables for cases where the variable has non-observed data. A variable has more influence if it has observed data in a row where the other variables do not have observed data. There are measures for the missing data pattern.

In a paper by Van Buuren, Boschuizen and Knook (1999)[15] *the proportion of usable cases* is defined as

$$I_{jk} = \frac{\sum_i^n (1 - r_{ij}) r_{ik}}{\sum_i^n 1 - r_{ij}} \tag{35}$$

for when imputing the variable Y_j from Y_k. This equation can be seen as the number of pairs (Y_j, Y_k) with Y_j missing and Y_k observed divided by the amount of missing cases in Y_k. If the value is 1 then it means that for any missing data point in Y_j there was an observed data point in Y_k. Low values of I_{jk} are not desirable as was shown by Van Buuren, Boschuizen and Knook who had imputed values for a dataset for a study to see the impact of blood pressure on survival for people who were classed as very old, the paper had dismissed variables with I_{jk} values of less than 0.5.

6.1.1 Influx and outflux statistics

The *influx statsitic* I_j for the j^{th} variable shows how linked the missing data in that variable is with the observed data in other variables and can be defined as :

$$I_j = \frac{\sum_j^p \sum_k^p \sum_i^n (1 - r_{ij}) r_{ik}}{\sum_k^p \sum_i^n r_{ik}} \tag{36}$$

I_j if that variable has no missingness and vice versa where it is equal to 1 when that variable has no data at all. The higher the influx the easier it may be to impute that variable.

The *outflux statistic* O_j shows how the observed data in variable j is linked to the missing data of all the other variables and can be defined as:

$$O_j = \frac{\sum_j^p \sum_k^p \sum_i^n r_{ij}(1 - r_{ik})}{\sum_k^p \sum_i^n 1 - r_{ij}} \tag{37}$$

Unlike influx, outflux O_j is 1 when the variable has no missingness and is 0 when there are no data entries for that variable. The higher the outflux the easier it may be to impute that variable.

6.2 Imputation on Multivariate missing data

The imputation methods discussed in Section 5 are imputation methods for when there is only missingness in one variable or column in a dataset. In reality, it is very common for there to be missingness in more than one if not all columns or variables in a dataset. The imputation models discussed so far for Y_j use the rest of the variables Y_{-j} as the predictor variables so the correlations between the rest of the variables and the variable with missing data can be continued with the imputed data. This way of imputing has problems such as when there are non-observed data in the in Y_{-j}; there is a dependence from another variable Y_k that has missingness on Y_j which also has missingness and vice versa;another problem is that variables can be of different types such as true-false statements or rankings from 1-10, so using a model assumes a distribution which is normal would not make sense. In this section, 2 general methodologies will be discussed that handle imputing multivariate data. Note that these are not the only 2 methods but just ones that will be touched in this report.

6.3 Monotone data imputation

The first method is the monotone data imputation which handles datasets with a monotone missing data pattern that uses a sequence of univariate methods and then draws observations from each method. Rubin [9] outlines the procedure. Let Y be ordered , $Y_1, ..., Y_p$ such that it is a monotone data pattern which ahs been explained in Section 2.1. The first column Y_1 is imputed from the covariates X ignoring $Y_2, .., Y_p$; then Y_2 is imputed using (X, Y_1) ignoring the rest of Y ; then Y_3 is imputed using (X, Y_1, Y_2) and this continues until Y_p is imputed from $(X, Y_1, ..., Y_{p-1})$. The univariate imputation methods discussed in Section 5 are suitable to use for imputation in each of the described step.

Algorithm 3 (Monotone data imputation of multivariate missing data[5]

1. Sort the observed data Y_j from $j = 1, , .p$ in order of amount of missingness.
2. Draw $\dot{\phi}_1 \sim Pr(Y_1^{\text{obs}}|X)$
3. Impute $\dot{Y}_1 \sim Pr(Y_1^{\text{miss}}|X, \dot{\phi}_1)$
4. Draw $\dot{\phi}_2 \sim Pr(Y_2^{\text{obs}}|X, \dot{Y}_1)$
5. Impute $\dot{Y}_2 \sim Pr(Y_1^{\text{miss}}|X, \dot{Y}_1, \dot{\phi}_2)$
6. ...
7. Draw $\dot{\phi}_p \sim Pr(Y_p^{\text{obs}}|X, \dot{Y}_1, \ldots, \dot{Y}_{p-1})$
8. Impute $\dot{Y}_p \sim Pr(Y_p^{\text{miss}}|X, \dot{Y}_1, \ldots, \dot{Y}_{p-1}, \dot{\phi}_p)$

6.4 Fully Conditional specification (FCS)

The following subsection is based on the paper by Van Buuren et al (2006) [16].

The FCS method involves setting an imputation model for each variable in order to impute the data variable by variable. FCS methods imputes the non observed data by using the observed values to model the conditional distribution of the values that need to be imputed. FCS aim to define $Pr(Y, X, R|\theta)$ using a series of conditional densities $Pr(Y_j|X, Y_{-j}, R, \phi_j)$ for each Y_{-j}. Y_{-j}^{miss} is imputed using these densities. Imputation under FCS isperformed by repeatedly running these imputation models starting from simple estimated values. The methods mentioned already can be used as the initial steps and FCS can be seen as concluding the univariate imputation methods for univariate imputation.

Rubin (1987)[9] separated the procedure of imputation into 3 tasks.

1. *The modelling task.*The task involves establishing a statistical model based on the data on hand to fill in what was missing. The model must be flexible enough to account for the uniqueness of the data as well as the link between the observed data and the missing data. The imputation model's underlying assumptions and the possible effects of model misspecification on the imputation results must be carefully taken into account for this job.

2. *The estimation task.* This task involves using the imputed data to estimate the parameters of interest. This task involves applying standard statistical methods to the imputed data (such as calculating regression coefficient or mean), but with appropriate adjustment for the uncertainty introduced by the imputation process .

3. *The imputation task.* Here, the relevant parameters using the model prescribed by the previous step. Here Y_{miss} is extracted from its posterior distribution from the model chosen in the previous steps. First a value of θ is drawn from $Pr(\theta|X, Y_{obs})$, call this θ^*. Then Y_miss is drawn from the conditional posterior distribution of $Pr(Y_{miss}|X, Y_{obs}, \theta = \theta^*)$, repeating this m times.

6.4.1 MICE algorithm

One algorithm that implements imputation is underspecified model. As we will see in the Appendix MICE is used to ampute the data MICE is also used to impute the data. MICE stands for multiple imputation by chained equations and is featured heavily in Van Buuren (2018) [5]. A paper by Azur et al ([17] describes what MICE is and how it works. So in brief the chain equation process works in 6 steps that Azur et al [17] mentions.

Algorithm 4 (MICE algorithm)

1. For every missing value in the dataset a simple imputation is carried out, for e.g. mean imputation. In the paper, these imputed values are called "place holders".

2. Imputation for one of the variables (this variable can be called "var") is reversed back to missing.

3. The observed values in "var" in Step 2 are regressed on the other variables in the imputation model which may or may not include every one of the variables so here "var" is the dependent variable and all the other variables are the independent variables in the regression model. These models function in the same way that one would make when performing linear, logistic or Poison regression models outside the context if imputing data.

4. The predictions (imputations) from the regression model from Step 3 are used to fill in the missing values in "var". Both the observed and these imputed values will be used when "var" is used as an independent variable.

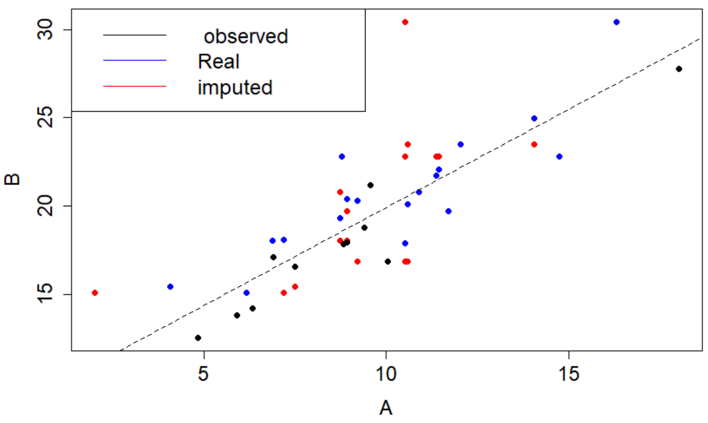


Figure 28: Plot showing MICE imputation values, with the real values.

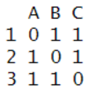
5. For every variable that has missing data, steps 2-4 are repeated. One iteration or "cycle" is the process of going over each variable. When a cycle finishes all of the missing values have been replaced with predictions from regressions that correspond to the relationships that are observed in the data.

6. Steps 2-4 are repeated for a set number of cycles and the imputations are updated at each cycle

Step 6 is where the multiple imputation part of MICE takes part but the following traditional methods are all single imputation ,if they involve imputation, so although MICE is used to impute the data in R the number of cycles has been set to 1. To create dataset MICE has an ampute function where one can create missingness of each type. So here 3 new datasets were created, one with MCAR,MAR and MNAR missingess, and the amount of data set to be amputed was 40%. These 3 datasets were used to be imputed using the following methods.

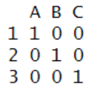
Example 6.1 (h!)**.** This is a simple example of how the MICE algorithm has an effect on a small dataset with 2 variables A and B where there is missingness in both variables. Here $A \sim N(10, 16)$ and $B = 10 + A + N(0, 4)$. There are 30 cases of data, so $n = 30$. With around 40% MCAR missingness, the number of imputations is set to 5. Figure 28 shows the results. Here, the estimated intercept was 9.78 and the gradient was estimated to be 1.10.

So, how does MICE ampute the MAR and MNAR data then? This is done by using weights. We generate a weighted sum score for all of the data rows for the MAR and MNAR mechanisms. This algorithm is based on pre-determined



```
  A B C
1 0 1 1
2 1 0 1
3 1 1 0
```

Figure 29: MAR weight matrix



```
  A B C
1 1 0 0
2 0 1 0
3 0 0 1
```

Figure 30: MNAR weight matrix

weights that the user can adjust using a matrix. The weights matrix contains the patterns on the rows and the variables on the columns (n x k). A weighted sum score is the result of a linear regression equation with weight matrix values as coefficients. The weighted sum score for data row i, when it is a candidate for pattern m, is as follows:

$$wss_i = w_{n,1} \cdot x_{1,i} + w_{n,2} \cdot x_{2,i} + ... + w_{n,k} \cdot x_{k,i}, \tag{38}$$

where $x_{1,i}, x_{2,i}, ..., x_{k,i}$ is the set of variable causes of case i and $w_{m,1}, w_{m,2}, ..., w_{m,K}$ are the pre-specified weights on row n on the weights matrix. Here, in our example $k = 3$ and $n \in \{1, 2, 3\}$ because there are 3 variables and 3 missing data patterns.

Larger weights typically result in greater sum scores than lower weights. For example if variables A and B have weights 6 and 3 respectively, the influence from A would be twice as much than that of B. Also, the influence of the weights is relative so; so weight values of 60 and 30 would have the same effect. Another note is that negative weights decrease the weighted sum and positive weights increase it. Also keep in mind that each pattern is given its own weights and that weight comparisons only occur within patterns, not between them.

The default weight matrices were used for MAR and MNAR which can be seen in Figures 28 and 29. The weights matrix gives variables that will remain completely in a specific pattern a value of 1, and variables that will become incomplete a value of 0. As a result, a MAR mechanism with equal influence for each variable is present in the default weights matrix. It seems that the function impute employs a weights matrix that displays 1 for those variables that, according to the patterns matrix, should become incomplete (0 in the patterns matrix).

6.5 The Analysis and Pooling phase of multiple imputation

This subsection is based on Chapter 8 of Enders' Applied Missing Data [3] So far we have discussed the imputation phase of multiple imputation. There are 2 other steps of multiple imputation, the analysis and pooling phase. The analysis phase aims to analyse each m datasets that have been imputed and then the pooling phase aims to combine all these estimates into one set of results. We have already discussed *Rubin's Rules* in Section 4.2 which pool's each of the m estimated parameters.

6.5.1 The Analysis phase

The analysis phase is the most straightforward aspect of multiple imputation. Each m imputed dataset would go through the statistical analysis planned for that dataset if it had no missingness. For example, linear regression could be the method of analysis. Most software packages can do this step easily, so if the number of imputations was high it would not be a tedious job as it would have been before software advancements.

6.5.2 The Pooling phase

If we refer back to Rubin's rules Equation (14), which describes the multiple imputation parameter estimate $\bar{\theta}$ as the average of all m estimates. This formula assumes that the parameter estimates are in very large samples normally distributed. This would not be the case for all parameters especially when working with smaller samples. So how can we combine such non-normal distributions such as correlation coefficients? A transformation is required to push the non-normal distribution to normality. This can be done using the Fisher z transformation for normalizing Pearson's correlation coefficient.

$$z_\ell = \frac{1}{2} \ln \frac{1+\rho_\ell}{1-\rho_\ell} \tag{39}$$

where ρ_ℓ is the correlation for $\ell = 1, ..., m$ and z_ℓ is the corresponding transformed coefficient. For large samples, the distribution of z is normal with variance $\sigma^2 = 1/(n-3)$ where n is the sample size. Substituting the transformed correlations into Equation (14) gives the average correlation. The back transformation can reverse the result by

$$\bar{\rho} = \frac{e^{2\bar{z}} - 1}{e^{2\bar{z}} + 1}. \tag{40}$$

The confidence interval of $\bar{\rho}$ is calculated in the z-scale then changed back using Equation (40). There are other transformations that would be needed for different test statistics for example Hazard ratio would need a logarithmic transformation as shown by Marshall, Billingham and Bryan [18]

Pooling standard errors have been discussed in Section 4.2 and are shown in Equations (15-17).

6.6 Multi-parameter inference

Investigating the notion that the parameters that have been estimated are notably away from 0 has been looked into. For example, in multiple regression analysis, one may want to see if the gradient of more than one line is notably nigger or smaller than 0. Multiple imputation has a set of measures to test this called *multiparamter inference*. It is to be noted that there isn't much information to see if these perform well. Schafer [19] recognised 3 test statistics in multi-parameter testing D_1 (multivariate Wald test),D_2 (combining test statistics) and D_3 (likelihood ratio test).

6.6.1 D_1 Multivariate Wald test

The approach for scalar values as explained in Section 4.4 is extended to the multivariate Wald test. Here, $\theta = \theta_0$, where θ_0 is a k-vector values under the null hypothesis. Under the assumption that $(Q_0 - \bar{\theta})$ is close to to a normal distribution

$$D_1 = (\bar{\theta} - \theta_0)' \tilde{V}_{MI}^{-1} (\bar{\theta} - \theta_0)/k \tag{41}$$

where Li, Raghunathan, and Rubin [20] put forward $\tilde{V}_{MI} = (1 + r_1)\bar{U}$ which proved to be a more reliable estimation of the total variance and where $r_1 = \hat{r}$ from Equation (28) is the average fraction of missing information. The test statistic D_1 also follows an F-distribution, F_{k,ν_w} with k and ν_1 degrees of freedom, where

$$\nu_1 = \begin{cases} 4 + (t-4)[1 + (1 - 2t^{-1})r_1^{-1}]^2 & \text{if} \quad t = k(m-1) > 4 \\ t(1 + k^{-1})(1 + r_1^{-1})^2/2 & \text{otherwise} \end{cases} \tag{42}$$

The p-value for D_1 is

$$P_1 = \Pr[F_{k,\nu_1} > D_1] \tag{43}$$

6.6.2 D_2 Combining test statistics

This second approach pools significance tests from the analysis phase. D_2 is used when the number of parameters in θ, k becomes large. Let d_ℓ is the test statsitsic from the analysis of the ℓ^{th} imputed dataset Y_ℓ, $\ell = 1, ..., m$. Then the average test statistic is $\bar{d} = m^{-1} \sum_\ell d_\ell$. Then the statistic for the combined test D_2 is

$$D_2 = \frac{\bar{d}k^{-1} - (m+1)(m-1)^{-1} r_2}{1 + r_2} \tag{44}$$

where the relative increase in variance r_2 is given by

$$r_2 = \left(1 + \frac{1}{m}\right) \frac{1}{m-1} \sum_{\ell=1}^{m} \left(\sqrt{d_\ell} - \overline{\sqrt{d}}\right)^2 \tag{45}$$

with $\overline{\sqrt{d}} = m^{-1} \sum_\ell \sqrt{d_\ell}$, so that r_2 is the sample variance of $\sqrt{d_1}$, $\sqrt{d_2}$,...,$\sqrt{d_m}$ multiplied by $(1 + 1/m)$. The p-value for testing the null hypothesis is

$$P_2 = \Pr[F_{k,\nu_2} > D_2] \tag{46}$$

where

$$\nu_2 = k^{-3/m}(m-1)(1+r_2^{-1})^2. \tag{47}$$

D_2 could be seen as a test statistic that is statistically significant and shows that the parameter estimations vary from their hypothesised values.

6.6.3 D_3 Likelihood ratio test

The last multiparameter significance test is to combine likelihood ratio test statistics. The likelihood ratio test [21] is used when there is no covariance matrix of the complete data estimates. This could be due to, a large number of parameters in θ which can happen for contingency tables. For a large sample, this procedure is the same as the multivariate Walds test.

For θ, the parameters of interest, the test is for the hypothesis $\theta = \theta_0$ for some given θ_0. For the likelihood ratio test, the first step is to calculate $\bar{\theta}$, from the newly fiiled in datasets after imputation, and $\hat{\theta}_0$, from the observed data, from the m number of datasets (using Rubin's rules). We then want to calculate the log-likelihood functions $L(\hat{\theta}_\ell)$ (for the complete datasets) and $L(\hat{\theta}_{0,\ell})$ for the observed data, then use these to work out the log-likelihood ratio statistic which works out the average likelihood ratio test across the m datasets, i.e.,

$$\hat{d} = m^{-1}\sum_{\ell} -2(l(\hat{\theta}_{0,\ell}) - l(\hat{\theta}_\ell)). \tag{48}$$

Then we take an average of the m likelihood ratio statistics, ad follows

$$\bar{d} = m^{-1}\sum_{\ell} -2(l(\bar{\theta}_{0,\ell}) - l(\bar{\theta}_\ell)). \tag{49}$$

The test statistic D_3 then can be shown as

$$D_3 = \frac{\bar{d}}{k(1+r_3)} \tag{50}$$

where

$$r_3 = \frac{m+1}{k(m-1)}(\hat{d} - \bar{d}) \tag{51}$$

estimates the average relative increase in variance due to missingness. Here, r_3 is equivalent to $\bar{r}$ from Equation (28). The p-value for D_3 is

$$P_3 = \Pr[F_{k,\nu_3} > D_3] \tag{52}$$

where $\nu_3 = \nu_1$. A positive of the likelihood ratio test is that it does not require normality.

7 Conclusion

In this report, the problem of missing data has been introduced. This involves types of missing data, with some reasons why this may occur. We then introduced missing data patterns, the different types of patterns and what this could mean for imputation. Then an important aspect of missing data analysis was introduced, the missing data mechanisms. These are crucial to the rest of the report, and MCAR, MAR and MNAR were explained thoroughly with examples. Then the majority of the report explained all the different types of imputation. We first looked at the traditional methods in Section 3. These methods including, complete case analysis, mean imputation, regression imputation and stochastic imputation were discussed with an example. The weaknesses of these "simple" methods were then considered showing why more complex methods needed to be developed to create better imputation so non-biased analysis can be carried out on the completed datasets. This carries on to the next sections of the report which mainly discussed multiple imputations which is a method that was developed to not have to deal with the weaknesses of the traditional methods.

Multiple imputation is a process where first a chosen number of imputations occur on the non-observed data, then the analysis is carried out on all the imputed datasets then finally all the estimated number of parameters are pooled. We introduced Rubin's rules which gave us a method of pooling all of the numbers of imputed estimated parameters with their variances. Another point to consider regarding multiple imputation is the method of imputation, so multiple imputation involves imputing many types, but what else needs to be considered is the method of imputation. 4 types of imputation methods were discussed in Section 5 with then a test was considered, where some measures were used to see if these methods created suitable imputations. Lastly, we looked at multivariate imputation as this poses another problem which univariate imputation doesn't have. We discussed some methods on how to get around this and also discussed the pooling phase of multiple imputation. Lastly, a little bit about multi-parameter inference was discussed.

To conclude this report, I would like to devote this final part to what could have been improved and what then my next steps would be if more time was allocated. Section 6 needs to be incomplete due to time constraints. To finish this section off, the methods proposed in this section would have been tested with dataset, preferably generated by myself so the parameters that would have been investigated later would have known actual values. To further take this report forward, the practical issues of multiple imputation could be looked at. Van Buuren [5] has a chapter on this and I feel like this would have been an important subject to discuss. Also, with more research another method to handle missing data that improves upon the traditional methods could be looked at, this can be then compared to multiple imputation and maybe a conclusion on which performs favourably and in what situations.

References

[1] Graham, JW. *Missing Data: Analysis and Design.* New York: Springer-Velag, 2012.

[2] Little, R J , and Rubin, D B. *Statistical Analysis with Missing Data* Hoboken, Nj, John Wiley Sons, Inc,2020.

[3] Enders, CK. *Applied Missing Data Analysis (Methodology in the social sciences).* Guilford Publications,2010.

[4] Allison, PD. *Missing Data. Quantitative Applications in the Social Sciences.* Thousand Oaks, CA: SAGE Publications, Inc,2002.

[5] Van Buuren,S . *Flexibe Imputation of Missing Data* .2nd Edition. CRC Press,2018.

[6] Rubin, D.B. *Inference and missing data.* Biometrika. 1976.

[7] Wilkinson, L. *Statistical methods in psychology journals: Guidelines and explanations.* American psychologist. 1999, 54(8), p.594.

[8] Rubin, D.B. *Multiple imputations in sample surveys-a phenomenological Bayesian approach to nonresponse.* In: Proceedings of the survey research methods section of the American Statistical Association: American Statistical Association Alexandria, VA, USA, 1978, pp.20-34.

[9] Rubin, D. B.*Multiple Imputation for Nonresponse in Surveys.*New York: John Wiley Sons. 1987

[10] Rubin, D.B. *Multiple imputation after 18+ years.* Journal of the American statistical Association. 1996, 91(434), pp.473-489.

[11] White, I.R., Royston, P. and Wood, A.M. *Multiple imputation using chained equations: issues and guidance for practice.* Statistics in medicine. 2011, 30(4), pp.377-399.

[12] Von Hippel, P.T. 8. *How to impute interactions, squares, and other transformed variables.* Sociological methodology. 2009, 39(1), pp.265-291.

[13] Gep, B. and Tiao, G. *Bayesian inference in statistical analysis.* Reading: Addison-Wesley. 1973.

[14] Efron, B. Bootstrap methods: another look at the jackknife. Springer, 1979

[15] Van Buuren, S., Boshuizen, H.C. and Knook, D.L.*Multiple imputation of missing blood pressure covariates in survival analysis.* Statistics in medicine. 1999, 18(6), pp.681-694.

[16] Van Buuren, S. *Multiple imputation of discrete and continuous data by fully conditional specification.* Statistical methods in medical research. 2007, 16(3), pp.219-242.

[17] Azur, M.J., Stuart, E.A., Frangakis, C. and Leaf, P.J. *Multiple imputation by chained equations: what is it and how does it work?* International journal of methods in psychiatric research. 2011, 20(1), pp.40-49.

[18] Marshall, A., Billingham, L.J. and Bryan, S. *Can we afford to ignore missing data in cost-effectiveness analyses?* : Springer. 2009, 10, pp.1-3.

[19] Schafer, J.L. *Analysis of incomplete multivariate data.* CRC press, 1997.

[20] Li, K.H., Meng, X.L., Raghunathan, T.E. and Rubin, D.B. *Significance levels from repeated p-values with multiply-imputed data.* Statistica Sinica. 1991, pp.65-92.

[21] Meng, X.-L. and Rubin, D.B. *Performing likelihood ratio tests with multiply-imputed data sets. Biometrika.* 1992, 79(1), pp.103-111.

A Appendix

In this section, the code that was used to generate some of the examples throughout the report will be shown and explained. All code was created on R.

A.1 Missing Data mechanism example

In Section 2, Table 2 shows a table with Test scores for some applicants for a job and their corresponding Job performance ratings. There are 4 columns in the graph showing the complete, MCAR,MAR and MNAR missingness of the job performance ratings. Table 3 shows the mean and gradient of Job performance against Test scores for each type of missing mechanism and the complete data. Figure 3 shows plots of each type of missing mechanism with a linear regression line. The following code in R in this subsection will show how all this was completed.

```
    par(mfrow=c(2,2))
TSC= c(56,60,60,61,62,65,66,67,67,69,71,75,75,76,77,80,81,82,84,96)
COM= c(9,13,10,8,7,7,9,9,11,7,8,9,11,15,10,10,12,13,17,12)
plot(TSC,COM,xlim=c(55,100),ylim=c(5,20), pch = 16,cex=1.2,cex.axis=1.5,
     cex.lab=1.5,cex.main=1.5, xlab = "Test score",
     ylab = "Job Performance",
     main = "Completed Test score vs Job Performance")
fit1= lm(COM~TSC)
fit1
abline(lm(COM~TSC))
```

The above code starts with par(mfrow=c(2,2)) which outputs 4 plots in a 2·2 arrangement . Variable TSC is the vector that holds the complete Test scores, COM is the vector holding the corresponding values for Job performance. The plot function plots the complete data set which can be seen in Figure 3 (top left). Finally, fit 1 creates the linear regression line which is fitted using the abline function.

```
    TSMD=c(71,75,75,76,77,80,81,82,84,96)
MD= c(8,9,11,15,10,10,12,13,17,12)
missing_fit=lm(MD~TSMD)

TSMCAR = c(56,60,62,66,66,67,69,71,75,76,77,80,81,82,84,96)
missiMCAR = c(9,13,7,7,9,9,7,8,9,15,10,10,12,13,17,12)
plot(TSMCAR,missiMCAR,xlim=c(55,100),ylim=c(5,20) ,pch = 16, cex=1.2,
     cex.axis=1.5, cex.lab=1.5,cex.main=1.5,col ="blue",xlab =
       "Test score", ylab = "Job Performance",
       main = "MCAR Test score vs Job Performance")
fit2 = lm(missiMCAR~TSMCAR)
fit2
abline(lm(missiMCAR~TSMCAR))
```

Here similarly to the previous code, this section of code starts with the variable TMSCAR, which is the vector that holds the MCAR test scores and missiMCAR holds the corresponding Job performance ratings. Then the plot function plots these 2 variables with x and y labels the same as in A.1 then finally, fit2 holds the linear regression followed by the abline function to put the linear regression on the plot. This plot can be seen at the top right of Figure 3.

```
    TSMAR= c(65,66,67,67,69,71,75,75,76,77,80,81,82,84,96)
MAR= c(7,9,9,11,7,8,9,11,15,10,10,12,13,17,12)
plot(TSMAR,MAR,xlim=c(55,100),ylim=c(5,20) , pch = 16, cex=1.2,cex.axis=1.5,
     cex.lab=1.5,cex.main=1.5,col="red",xlab = "Test score",
     ylab = "Job Performance", main = "MAR Test score vs Job Performance")
fit3= lm(MAR~TSMAR)
fit3
abline(lm(MAR~TSMAR))

TSMNAR= c(56,60,60,61,62,65,66,67,67,69,71,75,75,76,77,80,81,82,84,96)
MNAR= c(9,13,10,8,7,7,9,9,11,7,8,9,11,15,10,10,12,13,17,12)
plot(TSMNAR,MNAR,xlim=c(55,100),ylim=c(5,20) , pch = 16, cex=1.2,cex.axis=1.5,
     cex.lab=1.5,cex.main=1.5,col="yellow",
     xlab = "Test score", ylab = "Job Performance",
     main = "MNAR Test score vs Job Performance")
fit4 = lm(MNAR~TSMNAR)
fit4
abline(lm(COM~TSC))
```

These 2 sections are nearly identical to the first 2 sections of code however vectors TSMAR and TSMNAR are the MAR and MNAR test scores and MAR and MNAR are the corresponding Job performance ratings.

A.2 Traditional methods

In this subsection, the code we will be looking at is the code that produced graphs 5-14 in Sections 3.3-3.5.

```
    library(ggplot2)
library(mice)
library(broom)

set.seed(12)
A<-rnorm(30, mean = 10,sd=4)
B<- 10+A+rnorm(30,mean=0, sd=2)

Y<- 6+2*A+rnorm(30,mean=0, sd=2)+3*B
comp_fit=lm(Y~A+B)
summary(comp_fit)
dataset<-data.frame(A,B)
```

```
df<-data.frame(Y,A,B)

head(amp_frame$data)

lengths(amp_frame$amp)

mar_df<-ampute(dataset, prop=0.3,mech = "MAR", bycases= FALSE)
mcar_df<-ampute(dataset, prop=0.3,mech = "MCAR", bycases= FALSE)
mnar_df<-ampute(dataset, prop=0.3,mech = "MNAR", bycases= FALSE)
}
```

To start off, this section of code begins with calling the libraries ggplot2, mice and broom. Variables A,B and Y were created using the ”rnorm” function which draws values from the normal distribution given the values for the standard deviation and each one of their means. ”Comp_fit” creates the linear regression of Y as the independent variable and A and B as the dependent variables. The variable ”dataset” is the dataframe containing A and B which will go through amputation for each type of missing mechanism. This occurs in lines 19-21 using the ampute function with ”prop” the proportion of data that would be missing, in this case, 0.3 or 30%, ”mech” describes the type of missingness so MAR, MCAR and MNAR were used and ”bycases=FALSE” is needed to specify the proportion of missing cells.

```
    #Mean imputation
#create duplicated of df’s to impute
MI_mar<- mar_df$amp
for (i in 1:2) {
  MI_mar[ ,i][is.na(MI_mar[ ,i])] <- mean(MI_mar[ , i], na.rm= TRUE)

}
MI_mar
MI_mcar<-mcar_df$amp
for (i in 1:2) {
  MI_mcar[ ,i][is.na(MI_mcar[ ,i])] <- mean(MI_mcar[ , i], na.rm= TRUE)

}

MI_mcar
MI_mnar<-mnar_df$amp
for (i in 1:2) {
  MI_mnar[ ,i][is.na(MI_mnar[ ,i])] <- mean(MI_mnar[ , i], na.rm= TRUE)
}
```

In this section of code, the imputation starts with first using mean imputation as the method. For each dataframe containing each type of missingness both variables A and B’s missing values are replaced with their mean of the observed data. A ”for” loop is created to run over each variable in the dataframes and the line after the ”for” loop states to replace any N/A values with the mean of that column which is from the ”na.rm=TRUE” function. This is done for each type of missingness. These plots can be seen in Figures 5-7.

```
      #mean imputation graphs

plot(MI_mar$A,Y,pch = 17,col="red", cex = 1.3, xlab = "A", ylab = "Y"
     , main = "Mean imputed data on MAR missingess for Y vs A",cex.lab=1.5,
     cex.axis=1.5, cex.main=1.5, cex.sub=1.5)
points(mar_df$amp$A,Y,pch = 16, cex = 1.3)
abline(lm(Y~A), col="blue" , lty=2, lwd=3 )
abline(lm(Y~mar_df$amp$A),lwd=3)
abline(lm(Y~MI_mar$A),lwd=3, col="red" , lty=3)
legend(x="topleft", legend=c("Mean imputed data regression line",
                "Complete data regression line ",
                  "MAR data regression line"),
                    lty=c(3,2,1),col=c(2,4,1), cex=1.5)

plot(MI_mcar$A,Y,pch = 17,col="red", cex = 1.3, xlab = "A", ylab = "Y",
     main = "Mean imputed data on MCAR missingess for Y vs A",
     cex.lab=1.5, cex.axis=1.5, cex.main=1.5, cex.sub=1.5)
points(mcar_df$amp$A,Y,pch = 16, cex = 1.3)
abline(lm(Y~A), col="blue" , lty=2, lwd=3)
abline(lm(Y~mcar_df$amp$A),lwd=3)
abline(lm(Y~MI_mcar$A),lwd=3, col="red" , lty=3)
legend(x="topleft", legend=c("Mean imputed data regression line",
                             "Complete data regression line ",
                             "MCAR data regression line"),
         lty=c(3,2,1),col=c(2,4,1),cex=1.5 )
plot(MI_mnar$A,Y,pch = 17,col="red", cex = 1.3,
xlab = "A", ylab = "Y",
     main = "Mean imputed data on MNAR missingess for Y vs A",
     cex.lab=1.5, cex.axis=1.5, cex.main=1.5, cex.sub=1.5)
points(mnar_df$amp$A,Y,pch = 16, cex = 1.3)
abline(lm(Y~A), col="blue" , lty=2)
abline(lm(Y~mnar_df$amp$A),lwd=3)
abline(lm(Y~MI_mnar$A),lwd=3, col="red" , lty=3)
legend(x="topleft", legend=c("Mean imputed data regression line",
                             "Complete data regression line ",
                             "MNAR data regression line"),
         lty=c(3,2,1),col=c(2,4,1) ,cex=1.5)
```

Next is to create the graphs for these imputations. First, the plots were created using the ”plot” function for the imputation using the mean for each type of missingness . These points were coloured red and then the ”observed” data were plotted on top of these points as black, using the ”points” function so the red points left are the ones that were imputed and the black points show the observed data.

```
    # Regression imputation and plots
#MAR
RI_mar<- mar_df$amp
temp_RI_mar<- mice(RI_mar,method = "norm.predict", m=1,maxit=1, seed = 1243)
complete_RI_mar<- complete(temp_RI_mar,1)
RI_mar_fit<-lm(Y~complete_RI_mar$A+complete_RI_mar$B)
plot(complete_RI_mar$A,Y,pch = 17,col="red", cex = 1.3, xlab = "A", ylab = "Y",
     main = "Regression imputed data on MAR missingess for Y vs A",
     cex.lab=1.5, cex.axis=1.5, cex.main=1.5, cex.sub=1.5)
points(mar_df$amp$A,Y,pch = 16, cex = 1.3)
abline(lm(Y~A), col="blue" , lty=3)
abline(lm(Y~mar_df$amp$A),lwd=3)
abline(lm(Y~complete_RI_mar$A),lwd=3, col="red" , lty=3)
legend(x="topleft", legend=c("imputed data regression line",
                             "Complete data regression line ",
                             "MAR data regression line"),
       lty=c(3,2,1),col=c(2,4,1),cex=1.5 )
#MCAR
RI_mcar<- mcar_df$amp
temp_RI_mcar<- mice(RI_mcar,method = "norm.predict", m=1,maxit=1, seed= 1279)
complete_RI_mcar<- complete(temp_RI_mcar,1)
RI_mcar_fit<-lm(Y~complete_RI_mcar$A+complete_RI_mcar$B)
plot(complete_RI_mcar$A,Y,pch = 17,col="red", cex = 1.3, xlab = "A", ylab = "Y",
     main = "Regression imputed data on MCAR missingess for Y vs A"
     ,cex.lab=1.5, cex.axis=1.5, cex.main=1.5, cex.sub=1.5)
points(mcar_df$amp$A,Y,pch = 16, cex = 1.3)
abline(lm(Y~A), col="blue" , lty=3)
abline(lm(Y~mcar_df$amp$A),lwd=3)
abline(lm(Y~complete_RI_mar$A),lwd=3, col="red" , lty=3)
legend(x="topleft", legend=c("imputed data regression line",
"Complete data regression line ",
                             "MCAR data regression line"),
       lty=c(3,2,1),col=c(2,4,1),cex=1.5 )
#MNAR
RI_mnar<- mnar_df$amp
temp_RI_mnar<- mice(RI_mnar,method = "norm.predict", m=1,maxit=1, seed=12)
complete_RI_mnar<- complete(temp_RI_mnar,1)
RI_mnar_fit<-lm(Y~complete_RI_mnar$A+complete_RI_mnar$B)
plot(complete_RI_mnar$A,Y,pch = 17,col="red", cex = 1.3, xlab = "A", ylab = "Y",
     main = "Regression imputed data on MNAR missingess for Y vs A",
     cex.lab=1.5, cex.axis=1.5, cex.main=1.5, cex.sub=1.5)
points(mnar_df$amp$A,Y,pch = 16, cex = 1.3)
```

```
abline(lm(Y~A), col="blue" , lty=3)
abline(lm(Y~mnar_df$amp$A),lwd=3)
abline(lm(Y~complete_RI_mnar$A),lwd=3, col="red" , lty=3)
legend(x="topleft", legend=c("imputed data regression line",
"Complete data regression line ","MNAR data regression line"),
       lty=c(3,2,1),col=c(2,4,1) ,cex=1.5)
```

So the above piece of code is very similar as the previous 2 sections of code. If we look at this first part, Identical data frames for each type of missingness are called ”RI_mar”, ”RI_mcar” and ”RI_mnar” are. This data frame then goes through regression imputation using the ”mice” function with ”m=1” and ”maxit=1” to make sure that only one iteration of imputed values is created as this is a single imputation method. The ”complete” function is used to obtain completed data sets after imputation has been performed on a dataset using the mice function. Then plots are created showing the imputed values as red while the observed data as black. Then 3 linear regression lines are made, one for the observed data, the real ”complete” data and the imputed data.

The next 2 parts of this piece of code (MAR and MNAR) carry out the same function but instead of the MAR dataset, the MCAR and MNAR datasets are imputed and then plotted. The plots produced in this section of code can be seen in Figures 8-10.

```
    #Stochastic regression
#MAR
SI_mar<- mar_df$amp
temp_SI_mar<- mice(SI_mar,method = "norm.nob", m=1,maxit=1, seed = 232)
complete_SI_mar<- complete(temp_SI_mar,1)

plot(complete_SI_mar$A,Y,pch = 17,col="red", cex = 1.3, xlab = "A", ylab = "Y",
     main = "Stochastic regression imputed data on MAR missingess for Y vs A",
     cex.lab=1.5, cex.axis=1.5, cex.main=1.5, cex.sub=1.5)
points(mar_df$amp$A,Y,pch = 16, cex = 1.3)
abline(lm(Y~A), col="blue" , lty=3 )
abline(lm(Y~mar_df$amp$A),lwd=3)
abline(lm(Y~complete_SI_mar$A),lwd=3, col="red" , lty=3)
legend(x="topleft", legend=c(" imputed data regression line",
"Complete data regression line ","MAR data regression line"),
       lty=c(3,2,1),col=c(2,4,1),cex=1.5 )
#MCAR
SI_mcar<- mcar_df$amp
temp_SI_mcar<- mice(SI_mcar,method = "norm.nob", m=1,maxit=1, seed= 15)
complete_SI_mcar<- complete(temp_SI_mcar,1)
plot(complete_SI_mcar$A,Y,pch = 17,col="red", cex = 1.3, xlab = "A", ylab = "Y",
     main = "Stochastic regression imputed data on MCAR missingess for Y vs A",
     cex.lab=1.5, cex.axis=1.5, cex.main=1.5, cex.sub=1.5)
points(mcar_df$amp$A,Y,pch = 16, cex = 1.3)
```

```
abline(lm(Y~A), col="blue" , lty=3)
abline(lm(Y~mcar_df$amp$A),lwd=3)
abline(lm(Y~complete_SI_mcar$A),lwd=3, col="red" , lty=3)
legend(x="topleft", legend=c(" imputed data regression line",
"Complete data regression line ",
                             "MCAR data regression line"),
       lty=c(3,2,1),col=c(2,4,1),cex=1.5 )
#MNAR
SI_mnar<- mnar_df$amp
temp_SI_mnar<- mice(SI_mnar,method = "norm.nob", m=1,maxit=1, seed=1248)
complete_SI_mnar<- complete(temp_SI_mnar,1)

plot(complete_SI_mnar$A,Y,pch = 17,col="red", cex = 1.3, xlab = "A", ylab = "Y",
     main = "Stochastic regression imputed data on MNAR missingess for Y vs A"
     ,cex.lab=1.5, cex.axis=1.5, cex.main=1.5, cex.sub=1.5)
points(mnar_df$amp$A,Y,pch = 16, cex = 1.3)
abline(lm(Y~A), col="blue" , lty=3)
abline(lm(Y~mnar_df$amp$A),lwd=3)
abline(lm(Y~complete_SI_mnar$A),lwd=3, col="red" , lty=3)
legend(x="topleft", legend=c(" imputed data regression line",
                           "Complete data regression line ",
                             "MNAR data regression line"),
       lty=c(3,2,1),col=c(2,4,1) ,cex=1.5)
```

The code shown above carries out the same function as the code before this but creates identical data frames for each type of missingness called ”SI_mar”, ”SI_mcar” and ”SI_mnar” so stochastic regression imputation takes place on these datasets instead. The ”mice” function is used again, however, ”mech=”norm.nob”” which makes imputations using the stochastic regression method. The plots are made identically as before. The plots produced here are seen in Figures 11-13.

A.3 1000 repeats

This section will show and explain the code required to create the 1000 repeated stochastic imputations.

```
library(missMethods)
library(ggplot2)
library(mice)
library(broom)
library(Rmisc)
library(gmodels)

Acoef<- rep(NA,1000)
Bcoef<- rep(NA,1000)
Ccoef<-rep(NA,1000)
```

```
cept<-rep(NA,1000)
sigma<-rep(NA,1000)
lower_cept <- rep(NA, 1000)
upper_cept <- rep(NA, 1000)
lower_A <- rep(NA, 1000)
upper_A <- rep(NA, 1000)
lower_B <- rep(NA, 1000)
upper_B <- rep(NA, 1000)
lower_C <- rep(NA, 1000)
upper_C <- rep(NA, 1000)


A2coef<- rep(NA,1000)
B2coef<- rep(NA,1000)
C2coef<-rep(NA,1000)
cept2<-rep(NA,1000)
sigma2<-rep(NA,1000)
lower_cept2 <- rep(NA, 1000)
upper_cept2 <- rep(NA, 1000)
lower_A2 <- rep(NA, 1000)
upper_A2 <- rep(NA, 1000)
lower_B2 <- rep(NA, 1000)
upper_B2 <- rep(NA, 1000)
lower_C2 <- rep(NA, 1000)
upper_C2 <- rep(NA, 1000)
lower_sigma2 <- rep(NA, 1000)
upper_sigma2 <- rep(NA, 1000)

A3coef<- rep(NA,1000)
B3coef<- rep(NA,1000)
C3coef<-rep(NA,1000)
cept3<-rep(NA,1000)
sigma3<-rep(NA,1000)
lower_cept3 <- rep(NA, 1000)
upper_cept3 <- rep(NA, 1000)
lower_A3 <- rep(NA, 1000)
upper_A3 <- rep(NA, 1000)
lower_B3 <- rep(NA, 1000)
upper_B3 <- rep(NA, 1000)
lower_C3 <- rep(NA, 1000)
upper_C3 <- rep(NA, 1000)
lower_sigma3 <- rep(NA, 1000)
upper_sigma3 <- rep(NA, 1000)


A4coef<- rep(NA,1000)
```

```
B4coef<- rep(NA,1000)
C4coef<-rep(NA,1000)
cept4<-rep(NA,1000)
sigma4<-rep(NA,1000)
lower_cept4 <- rep(NA, 1000)
upper_cept4 <- rep(NA, 1000)
lower_A4 <- rep(NA, 1000)
upper_A4 <- rep(NA, 1000)
lower_B4 <- rep(NA, 1000)
upper_B4 <- rep(NA, 1000)
lower_C4 <- rep(NA, 1000)
upper_C4 <- rep(NA, 1000)

set.seed(12)
```

Figures 14-16 show the estimated A coefficients estimated after the stochastic regression imputation of each dataset with each type of missingness and the complete dataset. The above code starts by creating an array with a length of 1000 and each element is NA which will be filled with the estimates later on. This is done for every parameter so $A, B, C, cept$ and $sigma$. Also, arrays are made for the lower and upper values of intervals hence the variables that start with lower_and upper_. There are 4 sets of these coefficients, the first set is for the values estimates from the complete dataset, and the variables that have a 2 in them such as ”A2coef”, ”cept2” and so on are for the datasets with MAR missingness, the variables that have a 3 in them are for the datasets with MCAR missingness and if there's a 4 in the variable name then there is MNAR missingness.

```
#Complete data
A<-rnorm(30, mean = 10,sd=4)
B<- 10+A+rnorm(30,mean=0, sd=2)

C<-3+A+rnorm(30,mean=0,sd=2)

eps<-rnorm(30, mean=0,sd=3)
Y<-6+2*A+3*B+4*C+eps

ds_mar=ampute(ds_comp, prop=0.3,mech="MAR")
ds_mcar=ampute(ds_comp, prop=0.3,mech="MCAR")
ds_mnar=ampute(ds_comp, prop=0.3,mech="MNAR")
for (i in 1:1000){
  Y
  pheta_c<-lm(Y~A+B+C)
  Acoef[i]<- pheta_c$coefficients[2]
  Bcoef[i]<- pheta_c$coefficients[3]
  Ccoef[i]<- pheta_c$coefficients[4]
  cept[i]<-pheta_c$coefficients[1]
```

```
sigma[i]<- summary(pheta_c)$sigma
lower_cept[i]<-confint(pheta_c)[1]
upper_cept[i]<-confint(pheta_c)[5]
lower_A[i]<-confint(pheta_c)[2]
upper_A[i]<-confint(pheta_c)[6]
lower_B[i]<-confint(pheta_c)[3]
upper_B[i]<-confint(pheta_c)[7]
lower_C[i]<-confint(pheta_c)[4]
upper_C[i]<-confint(pheta_c)[8]
ds_comp=data.frame(A,B,C)
#MAR

ds_mar$B
imp1 <- mice(ds_mar$amp,method = "norm.nob", m=1)
comp_mar<-complete(imp1,1)
comp_mar
pheta_mar=lm(Y~comp_mar$A+comp_mar$B+comp_mar$C)
A2coef[i]<- pheta_mar$coefficients[2]
B2coef[i]<- pheta_mar$coefficients[3]
C2coef[i]<- pheta_mar$coefficients[4]
cept2[i]<-pheta_mar$coefficients[1]
sigma2[i]<- summary(pheta_mar)$sigma
lower_cept2[i]<-confint(pheta_mar)[1]
upper_cept2[i]<-confint(pheta_mar)[5]
lower_A2[i]<-confint(pheta_mar)[2]
upper_A2[i]<-confint(pheta_mar)[6]
lower_B2[i]<-confint(pheta_mar)[3]
upper_B2[i]<-confint(pheta_mar)[7]
lower_C2[i]<-confint(pheta_mar)[4]
upper_C2[i]<-confint(pheta_mar)[8]
```

Now the next part of the code, shown above, is to create a loop that creates a linear model such that $Y = \beta_0 + \beta_1 A + \beta_2 B + \beta_3 C$. Firstly, the variables A, B, C and Y are created with their ”true” parameters. Then the linear model is created, and then the following lines store the predicted values in their associated variables from the previous section of code from the linear model called ”pheta_c”. Then, the function extracts the lower and upper bounds of the confidence interval for each parameter from the linear model using the ”confint” function. In the next section (MAR) we see that a similar ”for” loop is created however MAR missingness is created using the ”ampute” function with 30% missingness and then imputation occurs using the impute function as seen before. Then the linear model is created and the parameter estimates with their corresponding upper or lower confidence values are extracted similarly to the first part of this code.

```
  #MCAR
ds_mcar$B
```

```
  imp2<- mice(ds_mcar$amp,method = "norm.nob", m=1)
  comp_mcar<-complete(imp2,1)
  pheta_mcar=lm(Y~comp_mcar$A+comp_mcar$B+comp_mcar$C)
  A3coef[i]<- pheta_mcar$coefficients[2]
  B3coef[i]<- pheta_mcar$coefficients[3]
  C3coef[i]<- pheta_mcar$coefficients[4]
  cept3[i]<-pheta_mcar$coefficients[1]
  sigma3[i]<- summary(pheta_mcar)$sigma
  lower_cept3[i]<-confint(pheta_mcar)[1]
  upper_cept3[i]<-confint(pheta_mcar)[5]
  lower_A3[i]<-confint(pheta_mcar)[2]
  upper_A3[i]<-confint(pheta_mcar)[6]
  lower_B3[i]<-confint(pheta_mcar)[3]
  upper_B3[i]<-confint(pheta_mcar)[7]
  lower_C3[i]<-confint(pheta_mcar)[4]
  upper_C3[i]<-confint(pheta_mcar)[8]

  #MNAR


  ds_mnar$B
  imp3<- mice(ds_mnar$amp,method = "norm.nob", m=1)
  comp_mnar<-complete(imp3,1)
  pheta_mnar=lm(Y~comp_mnar$A+comp_mnar$B+comp_mnar$C)
  A4coef[i]<- pheta_mnar$coefficients[2]
  B4coef[i]<- pheta_mnar$coefficients[3]
  C4coef[i]<- pheta_mnar$coefficients[4]
  cept4[i]<-pheta_mnar$coefficients[1]
  sigma4[i]<- summary(pheta_mnar)$sigma
  lower_cept4[i]<-confint(pheta_mnar)[1]
  upper_cept4[i]<-confint(pheta_mnar)[5]
  lower_A4[i]<-confint(pheta_mnar)[2]
  upper_A4[i]<-confint(pheta_mnar)[6]
  lower_B4[i]<-confint(pheta_mnar)[3]
  upper_B4[i]<-confint(pheta_mnar)[7]
  lower_C4[i]<-confint(pheta_mnar)[4]
  upper_C4[i]<-confint(pheta_mnar)[8]

}
```

The code above does exactly the same as the previous section but creates MCAR and MNAR missingness and then imputes the data using stochastic regression.

```
par(mfrow=c(1,2))
accepted_CI_A<- (lower_A<2) & (2< upper_A)
```

```
table(accepted_CI_A)
colour<- ifelse(accepted_CI_A,"red","blue")
plot(Acoef,1:1000,pch=16,cex.main=1.0,col=colour, ylab="",
xlab="A coeff",main="Estimated A value under no
     missingness",xlim=range(c(lower_A,upper_A)))
abline(v=2)
legend("topleft",c(" Rejected", "Accepted "),
       lty=c(1,2),col=c("blue",2), pch = c(16,16), cex=0.7 ,
       inset=c(0.8, 0), xpd = TRUE)



for(i in 1:1000){

  lines(c(lower_A[i],upper_A[i]), c(i,i), col=colour[i])
}
```

Now we need to create the graphs showing each of the 1000 estimates with their confidence intervals as horizontal lines on each point. The R code above is performing a simulation analysis and visualization to assess the performance of a statistical method for estimating a parameter A under no missingness.

The code first computes the confidence interval (CI) for the estimated value of parameter A using lower and upper bounds ”lower_A” and ”upper_A”. It then creates a logical vector ”accepted_CI_A” to indicate whether each CI contains the true value of A, which is assumed to be 2. The ”table” function is then used to display the number of CIs that are accepted and rejected.

The ”ifelse” function is then used to create a vector ‘colour‘ that assigns the colour ”red” to CIs that are rejected and ”blue” to CIs that are accepted. The ”plot” function is used to create a scatterplot of the estimated A values versus the index ”1:1000”, where each dot represents an estimated value of A. The colour of each dot is determined by the ”colour” vector, with ”red” dots indicating rejected CIs and ‘blue‘ dots indicating accepted CIs.

The ”abline” function is used to draw a vertical line at the true value of A, which is assumed to be 2. The ‘legend‘ function is used to add a legend to the plot indicating the colour scheme.

Finally, the ”lines” function is used in a loop to draw horizontal lines for each CI, with the position of the line determined by the index of the corresponding estimated A value. The colour of each line is determined by the ”colour” vector, with ”red” lines indicating rejected CIs and ”blue” lines indicating accepted CIs.

Overall, this code performs a simulation analysis to assess the performance of a method for estimating a parameter A under no missingness and produces a visual representation of the results. This was done for each coefficient and not just A however the code is nearly identical but the variables with the A values and their lower and upper confidence values were swapped with the other parameters. Also in the report, the plots for A were only shown.

```
#Create dataframes for each paramter
```

```
par(mfrow=c(1,1))
df_A<-(data.frame(Acoef,A3coef,A2coef,A4coef))
stackedA<- stack(df_A)
df_B<-(data.frame(Bcoef,B3coef,B3coef,B4coef))
stackedB<- stack(df_B)
df_C<-(data.frame(Ccoef,C3coef,C2coef,C4coef))
stackedC<- stack(df_C)
df_cept<-(data.frame(cept,cept3,cept2,cept4))
stacked_cept<- stack(df_cept)
df_sigma<-(data.frame(sigma,sigma3,sigma2,sigma4))
stacked_sigma<- stack(df_sigma)

#create boxplots for each parameter showing each type of missingness
boxplot(stackedA$values~stackedA$ind,xlab =
"Types of missingness", ylab="Value", main="Coefficient of A"
,col=(rainbow((ncol(df_A)))), names=c("None","MCAR","MAR","MNAR"))
abline(h=2)
boxplot(stackedB$values~stackedB$ind,xlab = "Types of missingness"
, ylab="Value",main="Coefficient of B",
    col=(rainbow((ncol(df_B)))), names=c("None","MCAR","MAR","MNAR"))
abline(h=3)
boxplot(stackedC$values~stackedC$ind,xlab = "Types of missingness",
ylab="Value",main="Coefficient of C",
    col=(rainbow((ncol(df_C)))), names=c("None","MCAR","MAR","MNAR"))
abline(h=4)
boxplot(stacked_cept$values~stacked_cept$ind,xlab =
"Types of missingness", ylab="Value",
        main="Intercept coefficient",col=(rainbow((ncol(df_cept)))),
        names=c("None","MCAR","MAR","MNAR"))
abline(h=6)
par(mfrow=c(1,1))
boxplot(stacked_sigma$values~stacked_sigma$ind,xlab =
"Types of missingness", ylab="Value",
        main="Standard deviation",col=(rainbow((ncol(df_sigma)))),
        names=c("None","MCAR","MAR","MNAR"))
abline(h=3)
```

The above R code creates five data frames ”df_A”, ”df”, ”df”, ”df”, and ”df”, each containing the estimated coefficients for a different parameter ($A, B, C, intercept$ and σ) under different missingness scenarios (MCAR, MAR and MNAR).

Each of the five data frames is created by combining the estimated coefficients for each parameter and missingness scenario into a single data frame using the ”data.frame” function. The resulting data frames have four columns corresponding to the estimated coefficients under each of the four missingness scenarios.

The ”stack” function is then used to convert each data frame into a stacked data frame, where the estimated coefficients for each missingness scenario are stacked on top of each other in a single column. This makes it easier to visualize and analyze the estimated coefficients across different missingness scenarios.

The ”par” function is used to set the plot layout to a single row and column, which will be used in subsequent code to create a set of histograms to visualize the distribution of the estimated coefficients for each parameter under each missingness scenario.

```
    # variables that store confidence intervals
ACI<-sapply(df_A,CI)
BCI<-sapply(df_B, CI)
CCI<-sapply(df_C, CI)
ceptCI<-sapply(df_cept, CI)
sigmaCI<-sapply(df_sigma, CI)
df_CI<-data.frame(ACI,BCI,CCI,ceptCI,sigmaCI)
write.csv(df_CI, "C:\\Users\\fahed\\OneDrive\\Documents\\MATH5004M (Project)
\\0.3 amputation\\CI.csv",row.names=FALSE)

# No of Coefficients CI that include the real coefficients
df_accpeted_A<-data.frame(table(accepted_CI_A),table(accepted_CI_A2),
table(accepted_CI_A3),table(accepted_CI_A4))
df_accpeted_B<-data.frame(table(accepted_CI_B),table(accepted_CI_B2)
,table(accepted_CI_B3),table(accepted_CI_B4))
df_accpeted_C<-data.frame(table(accepted_CI_C),table(accepted_CI_C2),
table(accepted_CI_C3),table(accepted_CI_C4))
df_accpeted_cept<-data.frame(table(accepted_CI_cept),table(accepted_CI_cept2),
table(accepted_CI_cept3),table(accepted_CI_cept4))

write.csv(df_accpeted_A, "C:\\Users\\fahed\\OneDrive\\Documents\\
MATH5004M (Project)\\0.3 amputation\\accepted_Af.csv",row.names=FALSE)

write.csv(df_accpeted_B, "C:\\Users\\fahed\\OneDrive\\Documents\\
MATH5004M (Project)\\0.3 amputation\\accepted_B.csv",row.names=FALSE)

write.csv(df_accpeted_C, "C:\\Users\\fahed\\OneDrive\\Documents\\
MATH5004M (Project)\\0.3 amputation\\accepted_C.csv",row.names=FALSE)

write.csv(df_accpeted_cept, "C:\\Users\\fahed\\OneDrive\\Documents\\
MATH5004M (Project)\\0.3 amputation\\accepted_cept.csv",row.names=FALSE)
```

The above code writes the results of the confidence interval analysis to CSV files using the ‘write.csv‘ function.

The first set of code calculates the confidence intervals for each parameter under each missingness scenario and stores the results in the variables ”ACI”,

”BCI”, ”CCI”, ”ceptCI”, and ”sigmaCI”, respectively. These variables are then combined into a single data frame ”df” using the ”data.frame” function.

The ”write.csv” function is then used to write the ”df_CI” data frame to a CSV file at the specified file path, with the option ”row.names=FALSE” used to exclude row names from the output file.

The second set of code calculates the number of coefficients whose confidence intervals include the real coefficients for each parameter under each missingness scenario and stores the results in the data frames ”df_accpeted”, ”df_accpeted”, ”df_C”, and ”df_accpeted_cept”, respectively. These data frames are then written to separate CSV files using the ”write.csv” function, with the option ”row.names=FALSE” used to exclude row names from the output files.