UPPSALA
UNIVERSITET

Raoul: An R-Package for Handling Missing Data

By David Randahl

Master Thesis

Department of Statistics

Uppsala University

Supervisor: Anna Bornefalk Hermansson

Assistant Supervisor: Erik Lampa

2016

Abstract

This paper introduces the Raoul package for handling missing data in R through multiple imputation by iterated sequential regression. The Raoul package uses a computationally efficient algorithm to generate imputations and allows for the imputation of categorical and count variables without relying on the Multivariate Normal Distribution or Markov Chain Monte Carlo simulations. A simulation study is conducted to compare the performance of the Raoul package with the performance of the mice, Amelia II, and NORM packages, and Listwise Deletion. Simulations are made on data Missing Completely at Random, Missing at Random, and Not Missing at Random, and at missingness levels of 10%, 20%, and 40%. The simulation study shows that the Raoul package is computationally faster than its competitors, and that its performance is roughly on par with these competitors for all types of missing data at the 10% and 20% level of missingness, but that it fails to compete at the 40% missingness level.

Keywords: Missing Data, Maximum Likelihood, Multiple Imputation, Iterated Regression

Contents

List of Tables

List of Figures

List of Algorithms

1 Introduction

Missing data is a common problem in almost all fields of research which rely on human gathered data. Missing data generally refers to the situation which arises when some cases in a data set have incomplete data i.e. when information is missing for some, but not all, variables, in some of the cases included in the data set. There may be several reasons for why the information is missing; a respondent may have declined or forgotten to answer a specific question in a survey, a country may not have the capacity to collect data for a specific variable, or existing data in a data set may have been found to be fraudulent or misrecorded. Missing data is problematic because most of the commonly used statistical methods for data analysis require data to be *complete*, i.e. to not have any missing data, to work properly. Ignoring the presence of missing data will therefore affect the results of any analysis made using that data, and may lead to biased results and false conclusions.[1] Therefore, it is highly important to use statistically sound methods for handling missing data in order to ensure the validity of the analyses made.

There are already several different statistical tools and packages for handling missing data, such as the Amelia II (Honaker et. al 2011), mice (van Buuren and Groothuis-Oudshoorn 2011), and NORM packages for R (Novo and Schafer 2013), and Proc MI for SAS (SAS Institute 2015), which are readily available in statistical software. These tools and packages rely on different types of imputation methods, i.e. methods which replace the missing data with new values, thereby allowing for the use of complete data statistical methods in the analysis. These existing packages and tools do, however, have several weaknesses. One of the primary weaknesses is that these existing packages and tools are often difficult to use and difficult to comprehend without extensive prior statistical knowledge. Additionally, many of these packages and tools are computationally slow and/or do not allow for handling of categorical and ordinal variables without certain assumptions which are not always reasonable.

The Raoul package for R, named after the Swedish Diplomat and Humanitarian Raoul Wallenberg, who disappeared in January 1945 and whose fate is still unknown, aims to address these weaknesses in the existing statistical tools and packages for handling missing data. The Raoul package allows for *Multiple Imputation* through *Sequential Iterated Regression.* The Raoul package is both computationally fast, easy to use, and allows for handling of categorical, ordinal, and count without having to edit the data set and without having to rely on the any

[1] It is important to emphasize that Missing Data is different from non-response in data, since Missing Data requires that *at least some* variables are observed for each case. The errors introduced by missing data are therefore fundamentally different from errors which may occur due to the non-response bias.

joint distributional assumptions for the variables.

1.1 Aim and Outline

The aim of this paper is to introduce the Raoul package for R, and to compare its performance in handling missing data to the performance of some of the existing packages for handling of missing data in R and to conventional missing data methods. More specifically, the performance of the Raoul package is compared to the performance of the Amelia II (Honaker et al. 2011), mice (van Buuren and Groothuis-Oudshoorn 2011), and NORM packages in R (Novo and Schafer 2013), and to the conventional method for handling missing data, Listwise Deletion.[2] These packages, and Listwise Deletion, are chosen as they are the most commonly used tools for handling missing data in R. The comparison is done by through evaluating the performance of these different methods on simulated missing data from a complete data set under different types of missing data.[3]

The remainder of this paper is divided into six sections. In the first two sections, different types of missing data, as well as different methods for handling missing data are discussed. In Section 4, the Raoul package is introduced and its theoretical motivation and algorithms discussed. Section 5 introduces the methodology and the data for the simulation study, which is followed by a section presenting and analyzing the results of this simulation study. Lastly, conclusions about the effectiveness of the Raoul package are drawn and implications discussed.

[2] See Section 3 for more detail on these various methods for handling missing data.

[3] See Section 2 for more details on types of missing data

2 Types of Missing Data[4]

Missing data is a phenomenon which is often accepted as a natural part of data collection, as it is not uncommon that certain information about some of the cases in the data set cannot be obtained. However,just as any other values in a data set, missing data can be though of as having been caused by some sort of missing data mechanism which determines whether or not a specific datum will be observed or missing. The process through which this missing data mechanism operates will then affect the effectiveness of different techniques for handling the missing data (see for instance King et. al. 2001). In order to discuss and evaluate different methods for handling missing data, some general assumptions for how these different missing data mechanisms may operate must therefore be discussed, as well as their implications for different techniques for handling missing data (Allison 2012).

Missing data, or missing data mechanisms, can generally be divided into three categories; data which are *Missing Completely at Random*, data which are *Missing at Random*, and data which are *Not Missing at Random*. The first two of these types of missing data are known as missing data caused by an *ignorable* missing data mechanism, while the last one is caused by a *non-ignorable* missing data mechanism (Schafer 1997, Allison 2009). Missing data generated from an ignorable mechanism allows for valid parameter estimates to be estimated without having to model the missing data mechanism. Missing data generated by a non-ignorable missing data mechanism, on the other hand, generally requires a model of the mechanism itself in order for valid parameter estimates to be obtained. Understanding, or making assumptions about, the mechanism which causes the missingness in the data is therefore important for choosing the correct method of handling the missing data. While there are statistical tools and methods for handling missing data with a non-ignorable missing data mechanism, the aim of this paper is to discuss different methods for handling data from ignorable missing data mechanisms. As will be seen below the assumption of ignorability is under many circumstances not an unreasonable assumption, as non-ignorability can often be remedied by adding auxiliary variables to achieve near-ignorability.

2.1 Data Missing Completely at Random

The strongest, and often the most unrealistic, assumption which can be made about a missing data mechanism is that the mechanism causes the missing data to be *missing completely at*

[4] Parts of this section is based on Randahl (2015)

random (MCaR), i.e. that all datum have an equal probability of being missing. This indicates that the probability of a datum being missing is independent of both the variable itself, as well as of all other variables included in the data set (King et. al. 2001, Allison 2009).

Some notation must be introduced to show this mathematically. Let Y be a variable with missing data, and let M_{Y_i} be a dummy variable which takes the value zero if Y_i, i.e. the value of Y for individual i, is observed and take the value one if Y_i is unobserved. Then, let $\mathbf{X}$ be a set of other variables which are included in the data set. The probability that a specific datum is missing given the value of itself and all other variables in the data set under the MCaR assumption, i.e. $Pr(M_Y = 1|\mathbf{X}, Y)$, can then be thought of as

$$Pr(M_Y = 1|\mathbf{X}, Y) = P(M_Y = 1) \tag{1}$$

This means that the probability of a specific value of Y being missing, conditional on both the values of $\mathbf{X}$ and of Y itself is the same as the unconditional probability that the same specific value of Y being missing. It is, however, important to emphasize that the missingness of the data need only be independent of the values of $\mathbf{X}$ and Y for the data to be MCaR. If there is another set of variables, say $\mathbf{Z}$, not included in the data set (i.e. entirely unobserved) which affect the missingness of the data and which are *uncorrelated* with $\mathbf{X}$ and Y such that

$$\begin{gathered} P(M_Y = 1|\mathbf{X}, Y, \mathbf{Z}) = P(M_Y = 1|\mathbf{Z}) \neq P(M_Y = 1), \text{ and} \\ Cov(Y, \mathbf{Z}) = Cov(\mathbf{X},\mathbf{Z}) = 0 \end{gathered} \tag{2}$$

then the data are still considered MCaR, since no variable(s) in the data set $Y, \mathbf{X}$ can be used to predict the missingness and the variables which affect the missingness, $\mathbf{Z}$, are uncorrelated with Y and $\mathbf{X}$.

While the assumption that data are missing completely at random is an incredibly strong assumption to make, it has the advantage of being an assumption which is partially testable. Assuming that the values of $\mathbf{X}$ are observed, it is possible to test if

$$Pr(M_Y = 1|\mathbf{X}) = Pr(M_Y = 1) \tag{3}$$

by, for instance, a logistic regression of M_Y on $\mathbf{X}$ and reject the MCaR assumption if the F-statistic for the model is significant. However, since the value of Y itself is always unobserved, it is never possible to test

$$Pr(M_Y = 1|Y) = Pr(M_Y = 1) \text{ or } Pr(M_Y = 1|\mathbf{X}, Y) = Pr(M_Y = 1) \tag{4}$$

i.e. it is not possible to test whether or not the missingness is dependent on the values of Y itself. As will be seen in Section 3, missing completely at random is the least severe form of missing data with regards to effectiveness of different methods for handling missing data.

2.2 Data Missing at Random

A less strong, and under most circumstances more reasonable, assumption on the missing data mechanism is that the mechanism causes data to be *missing at random* (MaR). Under MaR assumptions, the probability that a specific datum is missing or observed on the variable with missing data is independent of the value of the datum itself, but may depend on other variables which are included in the data set (Allison 2009, Schafer 1997). Using the same notation as above, this can be expressed as

$$Pr(M_Y = 1|\mathbf{X}, Y) = Pr(M_Y = 1|\mathbf{X}) \tag{5}$$

i.e. the missingness of Y is independent of Y itself when conditioned on $\mathbf{X}$. For example, if in a survey the sample is divided into different groups (the $\mathbf{X}$-variable(s)) and the probability that the individuals in the survey answer a specific question (the Y-variable) varies between the groups, but does not vary (with regards to Y) within the groups, then the missing data mechanism would generate data missing at random on Y (Allison 2009, Graham 2009). From the expression (5) above, it is easy to see that MCaR is a special case of MaR, since if the missingness of Y in expression (5) is independent of $\mathbf{X}$ the expression reduces to

$$Pr(M_Y = 1|\mathbf{X}, Y) = Pr(M_Y = 1|\mathbf{X}) = Pr(M_Y = 1) \tag{6}$$

Unlike MCaR the MaR assumption is not testable since the true values of Y are unknown and therefore it is not possible to test expression (4) above.

While MaR is still a relatively strong assumption, the MaR assumption is in many cases relatively reasonable since if $\mathbf{X}$ and Y are correlated, conditioning the missingness of Y on $\mathbf{X}$ may account for a pattern of missingness which may otherwise have been attributed to the value of Y itself. For instance, if as above, the missingness of Y is dependent on some set of variables $\mathbf{Z}$

which are not included in the data set and independent of $\mathbf{X}$, which are observed, but Y and $\mathbf{Z}$ are *correlated*, then

$$Pr(M_Y = 1|\mathbf{X}, Y) = Pr(M_Y = 1|Y) \neq Pr(M_Y = 1) \tag{7}$$

However, if the set of variables $\mathbf{Z}$ were to be included and observed in the data set then the situation above would reduce to

$$Pr(M_Y = 1|\mathbf{X}, Y, \mathbf{Z}) = Pr(M_Y = 1|Y, \mathbf{Z}) = Pr(M_Y = 1|\mathbf{Z}) \tag{8}$$

which means that the missingness of Y is MaR when $\mathbf{Z}$ is included in the data. This indicates that if the data are not missing at random (see below), it may be possible to correct this by adding more variables to the data set (See for instance Graham 2009, Graham and Donaldson 1993). Since it is possible to model the mechanism which generates the missing data through the other variables included in the data set, missing data which are MaR do not require the missingness mechanism to be modeled separately for valid parameter estimates to be obtained.

2.3 Data Not Missing at Random

If none of the assumptions above can be assumed to hold, then the missing data mechanism is considered to cause data to be *not missing at random* (NMaR)[5]. In essence this means that whether a specific value for Y is missing or observed is dependent on the value of Y itself, and that it is not possible to predict this value from the values of the observed set of variables $\mathbf{X}$ (Allison 2009, King et. al. 2001). In this case the equality in expression (5) does not hold.

There are several different methods for dealing with missing data which are NMaR, for instance methods similar to those used to handle selection bias (King et. al. 2001)[6]. These methods do, however, require analysis of the missing data mechanism since each NMaR situation is unique, and it is therefore difficult to create generalizable approaches to NMaR data. Yet, as seen above in section 2.2, it is often possible to improve the NMaR situation by including more variables in the data set. This may not necessarily turn the NMaR missing data into MaR missing data, but it may reduce the severity of the NMaR problem such that

[5] Also known as data missing not at random (MNaR), or simply as non-ignorable (NI) missing data

[6] For more information on methods for handling NMaR data, see for instance Little (1993, 1994, 1995), Heckman (1979).

$$Pr(M_Y = 1|\mathbf{X}, Y, \mathbf{Z}) \approx Pr(M_Y = 1|\mathbf{X}, \mathbf{Z}) \tag{9}$$

which may allow for an approximate MaR assumption under mild NMaR conditions (Allison 2012). It is, however, important to remember that the methods discussed in this paper, and the methods used by the Raoul-package, on a theoretical level requires the missing data mechanism to generate ignorable missing data, i.e. data which are MCaR or MaR.

3 Methods for Handling Missing Data

As noticed in the introduction, missing data is a problem which occurs in one form or another in most fields of research. Yet, despite the need for handling missing data, the missing data problem is not always discussed, and it is sometimes unclear what methods have been used to handle this problem[7] (van Buuren 2012). Not discussing how the missing data has been handled may be tempting, as it may make it possible to avoid questions about the quality of the data and the quality of the study. Avoiding this discussion does, however, not solve the underlying problem and avoiding the discussion may rather lead to the use of unsuitable methods for handling missing data when more appropriate methods are readily accessible.

There are three general theoretical criteria with which methods for handling missing data can be evaluated. First, the missing data method should minimize parameter bias in the analysis, i.e. when the statistical analysis is conducted on the treated data, the missing data method should ensure that the parameters of interest are as close as possible to the values which would have been obtained if the missing values had been observed (van Buuren 2012). Secondly, the missing data method should ensure that the maximum amount of the available data is used, and that as little as possible of the data is discarded. This is a natural criteria to include, since data collection is often time consuming, expensive, or both. Lastly, the missing data method needs to yield good estimates of the variability of the data, i.e. produce standard errors, p-values, and confidence intervals which are neither too large nor too small. This last criteria is important for the inference drawn from the analysis, as overestimating standard errors, and thereby also p-values and the length of confidence intervals, will lead to lower efficiency in the analysis, while underestimating the standard errors and the length of the confidence intervals will lead to an increased risk of Type I errors and thereby render the p-values unusable (Allison 2009).

As the problem of missing data is so prevalent in research it is not only important that the methods used for handling missing data are evaluated on theoretical performance, but also on their ease of use and their computational efficiency. These two dimensions are important since if the missing data methods are not easy and quick to use, researchers with limited statistical know-how may opt for using other, less complicated, methods for handling the missing data. The section below will discuss a number of different methods for handling missing data and how these different methods fare with regards to the three theoretical, as well as the two practical, evaluation criteria for data which are MCaR and MaR.

[7] Whether or not missing data is discussed varies greatly between different disciplines where statistical data is used.

3.1 Conventional Methods for Handling Missing Data

The most commonly used methods for dealing with missing data are methods which either remove the missing values or entire observations with missing data, or simpler forms of imputation methods, i.e. methods where the missing values are replaced by other values to enable complete data analysis. As will be shown below, these "conventional" methods for dealing with missing data do not perform well on the three theoretical criteria above, especially when the data is not MCaR.

3.1.1 Listwise Deletion

The standard approach for dealing with missing data in most statistical software, and probably also in most research (van Buuren 2012), is to simply delete all observations which have missing data. This approach of dealing with missing data is called *Listwise Deletion* or *Complete Case Analysis*, as it only uses the observations, or cases, in which all variables are observed (King et. al 2001, Allison 2009). Listwise Deletion is a very simple method for handling the missing data as it does not require any calculations or manipulations of the data, and because most statistical analysis software have default options of excluding observations with missing data, allowing for a "quick-fix" of the missing data problem.

Under the MCaR assumption, Listwise Deletion yields unbiased parameter estimates for means, variances, as well as in a regression setting. However, as the sample size decreases when observations are removed, the standard errors produced from regressions made using Listwise Deletion will be larger than standard errors produced if the missing data had been observed, since the standard errors are a function of the sample size (van Buuren 2012). Listwise Deletion is also a very wasteful missing data procedure, as it discards all information on an observation if a single variable for that observation has missing data which may lead to high levels of wasted data. For instance, King et al. (2001) claims that in political science surveying, an average of 50 % of all observations have at least one missing value, while in some cases it may be as high as 90%. Discarding all cases with missing values can therefore leave researchers with very little data.

If the data is MaR but not MCaR, Listwise Deletion will no longer yield unbiased parameter estimates as the correlation between the variables which the missingness is dependent upon and the variable in which the missingness occurs will cause a bias in the parameters. For instance, if the parameter of interest is the mean of some variable, and the variable is measured in two groups where the mean of the first group is twice the mean of the second group. Then, if the first group is more likely to answer a question about the variable than the second group, this would create an upward bias for the mean if Listwise Deletion is used to calculate the mean (Allison 2009).

In summary, Listwise Deletion does not fare well on any of the three criteria for handling missing data, as it fails to produce unbiased parameter estimates (unless data is MCaR), it wastes much gathered information, and it does not provide accurate estimates of uncertainty in the regression setting. Listwise Deletion is very easy to use and requires no additional computations, which is most likely why it is still the standard methods for dealing with missing data, despite its theoretical deficiencies.

3.1.2 Pairwise Deletion

An alternative to removing all cases which contain missing values is to use that most parameters of interest in statistical analysis can be expressed as functions of the means and covariance matrix. The means and covariance matrix can then, in turn, be estimated using the available data for each case. Thus, means and variances for individual variables are estimated using the observed data for that variable, and covariances between variables are estimated using all observations with data on both variables. This approach of handling missing data is called *Pairwise Deletion* or *Available Case Analysis*, and is an alternative to Listwise Deletion (Allison 2009).

The parameters yielded by Pairwise Deletion are consistent under the MCaR assumption, which means that they are asymptotically unbiased. When the data are MaR and not MCaR, Pairwise Deletion may, just as Listwise Deletion, yield biased estimates. Estimating the standard errors in a regression setting with Pairwise Deletion is, however, very complicated, as each covariance pattern may, theoretically, be estimated with a different number of observations. Since the standard errors are a function of the sample size it is therefore impossible to properly estimate these errors. Yet an additional complication with Pairwise Deletion is that the covariance and correlation matrices yielded using this missing data method are not guaranteed to be positive definite, which is a requirement for usage in most statistical analysis methods (Graham 2009, Allison 2009).

Pairwise Deletion is a slight improvement over Listwise Deletion on the theoretical level as it makes use of all available information. On the two other theoretical evaluation criteria Pairwise Deletion does, however, not fare much better than Listwise Deletion. In addition, although it does not require any computationally intensive calculations, Pariwise Deletion is a complicated method to use for most users with limited specialized statistical know-how, especially with regards to the calculation of standard errors, why it cannot be considered very user friendly.

3.1.3 Mean Imputation

Another simple method for handling missing data is *Mean Imputation*, whereby the missing values for a variable are simply replaced by the *unconditional* mean for the variable. Imputing the unconditional mean of the variable makes it possible to analyze the data using conventional complete data analysis methods, while still making use of all the observed data. As with both Listwise and Pairwise Deletion, Mean Imputation, will yield unbiased estimates of the mean under an MCaR assumption. Mean Imputation does, however, lower the variability of the data since the same value is imputed for all missing values. This creates a downward bias for the variance, and it also disturbs the covariances with the other variables in the set. Mean Imputation will therefore yield biased estimates for almost all other parameters except the mean, as well as standard errors which are too low due to the decreased variability. If the data is not MCaR, even the estimate of the mean may be biased (van Buuren 2012).

Thus, while Mean Imputation does make greater use of the available data, it does not fare well on any of the two other theoretical evaluation criteria. It is, however, easy to use and does not require any complicated calculations. Researchers with limited statistical know-how may therefore be tempted to use Mean Imputation rather than Listwise Deletion if they are not aware of the shortcomings of Mean Imputation.

3.2 Regression Imputation

The three conventional methods for handling missing data discussed above all have in common that they do not require any complicated calculations or manipulations of the data (except for the calculation of standard errors in Pairwise Deletion), and are easily implemented in the data analysis. In essence, they are statistically weak, but easy to use and easy to understand and may therefore be preferred by many researchers. The last of these conventional methods, Mean Imputation, does, however, take an important step in the right direction by replacing the missing values by other values which are, in some sense, plausible.

One way of improving upon Mean Imputation is to impute the *conditional* mean for the variable, given the observed values of the other values. This approach is commonly known as *Regression*

Imputation, as it simply replaces the missing values with the fitted values from a (linear) regression of the variable with missing values on the observed variables (Schafer and Graham 2002, Allison 2009). Unlike the conventional missing data methods, Regression Imputation yields unbiased parameter estimates under both MCaR and MaR, under the condition that the variables which affect the missingness in MaR are included in the statistical models used. In addition, Regression Imputation does not only allow for the use of the complete data set, it also uses the available data to improve the imputations for the missing data, thereby maximizing the use of the available data. The downside with regression imputation is that imputing the conditional means for the missing values will, naturally, strengthen the relationships between the variables since all values are on the regression line. This will lead to an underestimation of the variability of the data, as well as an underestimation of standard errors in a regression setting (Lee et al. 1994, van Buuren 2012).

Regression Imputation is a step forward compared to Mean Imputation. It remains both relatively user friendly, and does not require complex calculations or computationally intensive algorithms. The problem with the standard errors does, however, remain, but can be remedied by adding uncertainty to the imputations in the regression.

3.2.1 Regression Imputation with Uncertainty

The main problem with Regression Imputation is that imputations from the fitted values of the regression results in a deterministic imputation which always yield the same (fitted) value. The estimation of a regression line is, however, always associated with some residual errors around the regression line. A natural way of handling the problem of deterministic imputations is therefore be to add a random error term from the normal distribution to each imputed value, where the standard deviation for the normal distribution is taken from the residual standard error of the regression. This method of creating random imputations from the conditional mean improves the situation slightly by increasing the variance of the variables and producing larger standard errors in a regression setting. Yet, the variances and standard errors will still be too small (Schafer and Schenker 2000).

The reason that the standard errors and variances remain too small is that in a regression setting, the regression uncertainty is not dependent on the residuals, but also dependent on the uncertainty in the estimated parameters. As the regression parameters by the Central Limit Theorem (CLT) follow an approximate normal distribution under the condition that the sample size is large, or an exact normal distribution if the residuals are normally distributed, it is possible to take a random draw of the parameters of the regression and use these random regression parameters, as well as a normal error term, to create imputations. This recreates the uncertainty associated with the regression and thereby minimizes the problem of underestimating variances and standard errors (van Buuren 2012, Lee et al 1994).

One downside with this approach is that the estimation of parameters is not efficient due to the added noise. This problem can be remedied by using this method in a multiple imputation setting, which will be discussed in section 3.4. In a multiple imputation setting, this method has been shown to work very well for univariate missing data, i.e. when data are missing only on a single variable (van Buuren 2012), but the approach can easily be generalized to cases with multivariate missing data, i.e. when data are missing on several variables. The method of replacing the missing values using Regression Imputation with parameter uncertainty and error term is more complicated than regular Regression Imputation, but still computationally fast and a relatively intuitive procedure.

The easy generalization to the multivariate case, as well as the statistically sound properties, the relative ease of use, and the computationally fast algorithm makes Regression Imputation with parameter uncertainty and error term quite a suitable imputation method. In fact, as will be seen in section 4, the imputation algorithm of the Raoul package is based on a multivariate version of this method, combined with multiple imputation.

3.3 Predictive Mean Matching

An additional way of handling missing data more appropriately than the conventional methods are through *Predictive Mean Matching*, or *Hot Deck Imputation*. Predictive Mean Matching uses the theoretical assumption that the missing value on a variable for an individual should be similar to the observed value for that variable in an individual similar to the one with a missing value. In practice, Predictive Mean Matching scans the data, for each missing value, to gather a set of individuals similar to the individual with the missing value, so called donor candidates. Then, the observed value of one of these candidates is imputed to replace the missing value, creating a full data set (van Buuren 2012, de Waal et al. 2011).

The donor candidates can be selected through a number of different criteria, often involving their statistical distance from the individual with the missing value, i.e. the recipient. The specific donor can also be selected in a number of different ways, both deterministicly and randomly. For instance, the donor can deterministically be selected as the candidate with the shortest statistical distance to the recipient, or a pool of d candidates can be sampled based on their distance to the recipient, and then one of the d candidates are selected randomly. Often, the value of d vary between one and ten. The probability of each candidate can also be weighted to the statistical distance between the donors and the recipient, increasing the likelihood of picking a donor close to the recipient (van Buuren 2012, de Waal et al. 2011).

One strong advantage with Predictive Mean Matching over the Regression Imputation methods is that the imputed values are taken from the existing data, ensuring that no imputations are outside the data range. Predictive Mean Matching is also more robust to misspecification and interaction effects than Regression Imputation, as the model is implicit and therefore does not require an explicitly specified model from which imputations are generated. Predictive Mean Matching also does well on the three theoretical criteria for missing data methods, as it produces approximately unbiased parameters, makes use of all available data, and does not have a minimizing effect on variability as long as the data is MCaR or MaR (van Buuren 2012).

Just as with Regression Imputation with error term and/or parameter uncertainty, extending Predictive Mean Matching to multiple imputation (see section 3.4 below) is relatively straightforward, as it is possible to re-sample the Predictive Mean Matching and thereby make it possible to impute several different values for each missing value, increasing the efficiency of the method (van Buuren 2012, de Waal et al. 2011). For univariate missing data, the method is also computationally fast, and relatively easy to use. For multivariate missing data, the method is more complicated as sequential imputation of the data may lead to oversampling from individual donors.

3.4 Multiple Imputation

A general downside with the missing data methods discussed in section 3.2.1 and 3.3 is that when imputing a single, random, value instead of the missing value, the value would be unbiased but not generally correct, and that the statistical methods used for the analysis cannot differentiate between the imputed and the observed values and therefore treat the imputed values as if they were observed (Rubin 1987, van Buuren 2012). As a solution to this problem, Rubin (1987) suggested that instead of imputing a single value for each missing data point, m different values should be imputed to create m different complete data sets. Each complete data set is then analyzed separately using standard statistical methods, and the results for the parameters are combined. This procedure is known as *Multiple Imputation* (MI) and is described in Figure 3.1[8] below.

[8] Figure based on van Buuren (2012)

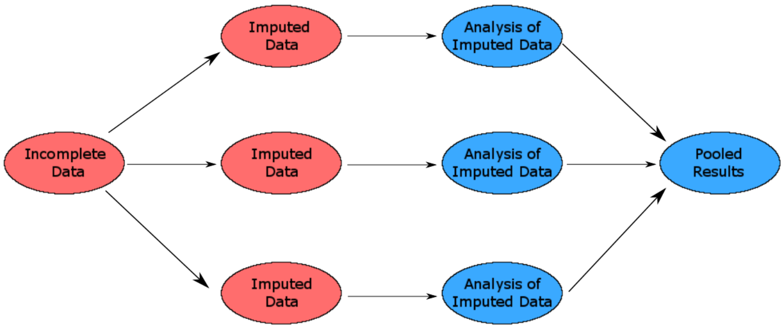

Figure 3.1: The Concept of Multiple Imputation

Multiple Imputation retains the advantages of the methods discussed in 3.2.1 and 3.3, but allows for the differentiation of the imputed values and the observed values, as the observed values remains the same across the m different data sets. This will counter the problem of the imputations not being generally correct, as each imputation will create an additional unbiased data point with which to estimate the parameters. Thus, as m increases, the efficiency of the parameter estimation will increase. This efficiency in combination with the variability which is introduced through the multiple imputations allows for proper inference to be made directly from the pooled results (Rubin 1987).

As was mentioned in the sections above, both Regression Imputation methods with added uncertainty and Predictive Mean Matching can be used in the context of Multiple Imputation. Adding multiple imputations is straightforward as the methods allows for repeating m independent draws from the distribution of regression coefficients and error terms, in the case of Regression Imputation, or independent draws among donor candidates, in the case of Predictive Mean Matching, in order to create the m complete data sets needed for Multiple Imputation.

An alternative to these methods are Multiple Imputation methods based on drawing imputations from Markov Chain Monte Carlo (MCMC) simulation to generate the m complete data sets. This can be done by either assuming a join multivariate distribution for the variables, or through specifying a set of conditional distributions. Another alternative is to use bootstrapped samples from the observed data and apply the Expectation-Maximization algorithm to generate independent imputations of the data. These methods generate good imputations for cases with multivariate missing data which occurs anywhere in the data, and can be seen as the ”standard approach” for multivariate missing data as they are readily available in different statistical software (Honaker et. al. 2011, van Buuren and Groothuis-Oudshoorn 2011, SAS Institute 2015).

These methods for generating Multiple Imputations are discussed in the sections below.

3.4.1 Imputation by Data Augmentation

The general idea of imputation through *Data Augmentation* is to use Markov Chain Monte Carlo simulations to draw random imputations from a posterior multivariate distribution. Drawing the values direct from the multivariate distribution is, however, difficult due to the missingness. Instead of drawing directly from the distribution, the process can be simplified by *augmenting* the data by filling in the missing values (King et al. 2001). By updating the distribution after each draw and generating a sequence, or chain, of these imputations, the actual multivariate distribution can be approximated and used to draw imputations (Schafer 1997). In essence, this procedure involves first specifying a multivariate distribution which the data is assumed to follow, usually the multivariate normal distribution, and a set of starting parameters $\theta^{[0]}$. Then the data is updated through in the following two steps

$$\text{Draw } \dot{\mathbf{Y}}_{mis}^{[t]} \sim P(\mathbf{Y_{mis}}|\mathbf{Y_{obs}}, \dot{\theta}^{[t-1]}) \tag{10}$$

$$\text{Draw } \dot{\theta}^{[t]} \sim P(\theta|\dot{\mathbf{Y}}_{\mathbf{mis}}^{[\mathbf{t}]}, \mathbf{Y_{obs}}) \tag{11}$$

Where $\mathbf{Y}_{mis}$ are the missing data, $\dot{\mathbf{Y}}_{mis}^{[t]}$ are the imputed or *augmented* data for $\mathbf{Y}_{mis}$ at point t in the sequence, and $\dot{\theta}^{[t]}$ are the drawn distribution parameters at time t (Schafer 1997, van Buuren 2012). The first step is known as the *imputation step* and the second step is known as the *posterior step*, why the procedure is sometimes known as the *Imputation-Posterior* (I-P) method (Schafer 1997, King et al. 2001). The two steps of the procedure are then repeated until the distribution converges, after which the final iteration is imputed for the missing data to create the complete data set. After each (final) imputation, the procedure is repeated to create m different complete data sets. In order for the imputations to be independent, a number of iterations must be made in between the imputations to ensure that the time dependency of the Markov Chain is eliminated (SAS Institute 2015).

The advantage of imputing the data trough Data Augmentation is that the random draws of the imputations are taken from the (asymptotically) exact distribution, why it yields imputations which are statistically very sound. The downside is that there is no standard method to assess whether or not the Markov Chain has converged, and therefore it is essentially arbitrary how many iterations need to be used before imputations are made (King et al. 2001). This use of a large number iterations before imputations, so called "burn-in" iterations, also makes the algorithm slow to use when the data set is large (Takahashi and Ito 2013). Another downside with the method is that it requires an explicit multivariate distribution to be specified for the data, with the multivariate normal distribution being the standard choice. However, data sets often consist of mixed data of which some variables may have characteristics which make them highly non-normal, for instance if they are categorical data. The method have been shown to be quite robust to non-normality, but there is nonetheless a theoretical problem (Schafer 1997, Schafer and Graham 2002, Graham 2009).

Data Augmentation is the standard method for handling (continuous) missing data in the statistical software SAS, where the standard setting is to use 200 ”burn-in” iterations before the first imputation, and then 100 iterations between each imputation (SAS institute 2015). Data Augmentation is also available through the NORM package in R (Novo and Schafer 2013). For both of these programs, the Expectation Maximization algorithm[9] is used to obtain the starting values for $\theta^{[0]}$, after which the I-P steps are applied for the pre-specified number of iterations.

In summary, Data Augmentation exhibit excellent statistical properties with regards to the three theoretical criteria under MCaR and MaR, i.e. it provides unbiased parameter estimates, makes use of all available data, and provides good estimates of the variability of the data, as it draws the imputations from the asymptotically exact distribution. The main theoretical drawback of the Data Augmentation method is the assumption of the multivariate normal distribution, which may make some researchers uneasy. Data Augmentation does, however, not do as well on the practical evaluation criteria as Data Augmentation involves several decisions which require extensive statistical know-how, for instance to estimate how many iterations should be used before imputations are made, and whether or not a multivariate normal distribution for the data is a realistic assumption. Many researchers in fields outside statistics may also be deterred from using Data Augmentation by a lack of understanding of how MCMC methods work mathematically. In addition, the Data Augmentation method is computationally slow, which may be problematic in cases where the data sets are large.

3.4.2 Fully Conditional Specification Imputation

An alternative to the traditional Data Augmentation method for creating multiple imputations is the *Fully Conditional Specification* method for creating multiple imputations. In contrast to the Data Augmentation method, the Fully Conditional Specification does not require an

[9] For more information on the Expectation-Maximization algorithm see section 3.4.3, Schafer (1997)

explicit multivariate distribution to be specified for the data. Instead, the Fully Conditional Specification models the joint distribution of the data implicitly through a set of conditional distributions where each individual variable is *conditioned* on all other variables. This allows for different conditional distributions being specified for each variable, which, in turn, means that appropriate distributions can be specified for each type of data, regardless of whether it is continuous, count, ordinal, or categorical (van Buuren 2012).

The perhaps most commonly used version of Fully Conditional Specification is Multiple Imputation through Chained Equations (MICE), which is an Markov Chain Monte Carlo method for imputing data through the conditional distributions of all variables. In many ways the MICE algorithm resembles the Data Augmentation algorithm, except that the MICE algorithm only makes draws for one variable at a time, conditioned on all other variables, and repeats the procedure across all variables. Specifically, the MICE algorithm begins by drawing imputations for the missing values for variable j, $\dot{Y}_j^{[0]}$, through a random draw from the observed values, then it repeats the following two steps

$$\text{Draw } \dot{\theta}_j^{[t]} \sim P(\theta_j^{[t]} | Y_j^{obs}, \bar{\mathbf{Y}}_{-j}^{[t]}) \tag{12}$$

$$\text{Draw } \dot{Y}_j^{[t]} \sim P(Y_j^{mis} | Y_j^{obs}, \bar{\mathbf{Y}}_{-j}^{[t]}, \dot{\theta}_j^{[t]}) \tag{13}$$

where Y_j^{obs} are the observed values for variable j, Y_j^{mis} are the missing values for variable j, and $\bar{\mathbf{Y}}_{-j}^{[t]}$ are the complete data (observed and imputed) except for variable j. These two steps are iterated for all variables with missing data, for a predefined number of iterations, which can usually be low (van Buuren 2012). MICE is implemented in the aptly named R-package mice, where the default number of iterations are 5 (van Buuren et al. 2015).

The MICE algorithm retains the statistically sound properties of the Data Augmentation method for data which are MCaR or MaR, but does not rely on the problematic assumption of the (usually) multivariate normal distribution. MICE is, however, still a method depending on MCMC simulations which makes it computationally slow. Not assuming a predefined explicit distribution also makes some computational shortcuts often implemented in Data Augmentation algorithms impossible, why there are no computational gains compared to Data Augmentation despite the much lower number of iterations needed (van Buuren 2012, Takahashi and Ito 2013). Rather, the lack of these short-cuts render the MICE algorithm even slower than the Data Augmentation method.

In summary, Fully Conditional Specification Imputation with the MICE algorithm is a theoretical step forward compared to Data Augmentation when missing data which cannot be assumed to follow a multivariate normal distribution is imputed. MICE is computationally slow, and not easy to use for users without extensive statistical know-how.

3.4.3 Expectation-Maximization Bootstrapping Imputation

The problem of relying on Markov Chain Monte Carlo simulations in multiple imputation was noted by King et al. (2001), who highlighted the computational difficulties, the difficulties associated with determining when convergence has occurred, and the extensive statistical know-how needed to make these decisions. As an alternative, King et. al suggested using methods based on the Expectation-Maximization (EM) algorithm for data under the multivariate normal distribution, and using different methods of resampling values after the EM algorithm had been run. The EM algorithm was developed as an iterative method for computing maximum-likelihood estimates for parameters in distributions where these parameters could not be estimated directly due to missing data (Graham 2009, Allison 2009, Shafer 1997). The main advantage of using the EM algorithm is that it converges deterministically, why it is simple to determine when the algorithm has converged and pre-specified bounds can be used to terminate the algorithm (King et. al 2001, Honaker and King 2010). The Expectation-Maximization essentially reduces to the following steps, begin by estimating $\theta^{[0]}$ [10] by, for instance, Listwise Deletion. Then repeat the following two steps until convergence;

$$\text{The E-step: Impute } \dot{\mathbf{Y}}^{[\mathbf{t}]} \text{ by } \dot{\mathbf{Y}}^{[\mathbf{t}]} = E[\mathbf{Y^{mis}}|\mathbf{Y^{obs}}, \dot{\theta}^{[t-1]}] \tag{14}$$

$$\text{The M-step: Re-estimate } \dot{\theta}^{[t]} \text{ by maximizing } Q(\theta|\bar{\mathbf{Y}}^{[\mathbf{t}]}) = E[\log L(\theta|\bar{\mathbf{Y}}^{[\mathbf{t}]})] \tag{15}$$

This procedure is very similar to the *Imputation-Posterior* algorithm in section 3.4.1 above, but instead of being based on an MCMC chain, the EM-algorithm converges deterministically to the maximum likelihood values for θ (Allison 2009, Honaker and King 2010, Schafer 1997). Under the multivariate normal model, this procedure is essentially the same as imputing the fitted values from linear regressions of the variables with missing values on all other observed and imputed variables, re-estimating the parameters ($\theta = \{\mu, \mathbf{\Sigma}\}$), with a correction factor for $\mathbf{\Sigma}$, after each imputation, and iterating the procedure until convergence (Allison 2009).

[10] For the multivariate normal distribution $\theta = \{\mu, \mathbf{\Sigma}\}$

The problem is that variability is needed to create multiple imputations and to make valid inference. To solve this problem, Honaker and King (2010) proposes to combine the Expectation-Maximization algorithm with bootstrapping, which introduces variability into the algorithm, making it possible to create multiple imputed data sets. The extension of adding a bootstrapping step to the EM-algorithm is simple: by drawing a bootstrapped sample from the observed data, i.e. a sample with replacement, and then running the EM algorithm, fundamental uncertainty is introduced in the estimation of θ which is then used to impute data to create a complete data set. By repeating this procedure m times, m distinct complete data sets are created which can be used to analyze the data (Honaker and King 2010, Honaker et al. 2011).

Compared to the methods based on MCMC simulations, the EM-bootstrapping algorithm is a computationally fast and relatively simple method for obtaining multiple imputations in the face of missing data. The user is not required to have extensive statistical know-how, as all computations are relatively simple and the algorithm converges deterministically. The main theoretical drawback with the algorithm is, as in the case with the Data Augmentation method, that a multivariate normal distribution is assumed for the variables in the data. Simulation studies have found that the EM-bootstrapping algorithm performs well on the three theoretical evaluation criteria for data which are MCaR and MaR, although the results are slightly worse than the results from Data Augmentation or the MICE algorithm (Takahashi and Ito 2013).

3.5 Summary of Existing Methods

Section 3 has outlined some of the most commonly used methods for handling missing data, and how some of these methods have been implemented in statistical software. From this discussion it seems clear that the conventional methods often employed to deal with missing data do not yield satisfactory results, and neither do the methods which impute single values due to their inefficient estimation of the parameters.

The methods based on Multiple Imputation produce results which satisfactory on the theoretical evaluation criteria, but fare worse on practical evaluation criteria as the Data Augmentation and MICE methods require extensive statistical know-how and are computationally slow. The EM-bootstrapping method is easier to use, but does not perform as well as the Data Augmentation and MICE methods, and is also theoretically problematic due to the assumption that the data follow a multivariate normal distribution. The theoretical and practical performance of these different methods are summarized in Table 3.1.

Method	Bias	Data Use	Variability	Speed	Ease-of-use
Listwise Deletion	Unbiased for MCaR	Minimal	Too Large	Instant	Very Easy
Pairwise Deletion	Unbiased for MCaR	Medium	Unclear	Instant	Difficult
Mean Imputation	Biased for all but μ_{MCaR}	Medium	Too Small	Instant	Very Easy
Regression Imputation	Unbiased for MaR	High	Too Small	Fast	Easy
Regression Imputation w. uncertainty	Unbiased for MaR	Maximal	Appropriate	Fast	Medium
Predictive Mean Matching	Unbiased for MaR	Maximal	Appropriate	Fast	Medium
Data Augmentation	Unbiased for MaR	Maximal	Appropriate	Slow	Difficult
MICE	Unbiased for MaR	Maximal	Appropriate	Very Slow	Difficult
EM Bootstrapping	Unbiased for MaR	Maximal	Appropriate	Medium	Medium

Table 3.1: Summary of Existing Missing Data Methods

For users with limited statistical know-how, it may seem as if the EM-bootstrapping method would be the most convenient choice. However, while the EM-bootstrapping algorithm is relatively fast compared to the Data Augmentation and MICE algorithms (see Takahashi and Ito 2013), it may still require substantial computational power in larger datasets. In addition, the reliance on the multivariate normal distribution may be problematic for many types of data.

4 Introducing the Raoul Package

The Raoul package for R aims to address some of the weaknesses in the current methods for handling missing data by introducing an algorithm which is faster and easier to use than the EM-bootstrapping algorithm, and with built in support for directly obtaining pooled regression results from the imputed data. In essence, the Raoul algorithm is a method of the Fully Conditional Specification class of imputation methods, whereby imputations are made on a variable-by-variable basis, based on the conditional distribution of the variable on all other variables in the dataset. Unlike the MICE algorithm, the Raoul algorithm is a two step procedure where the Maximum Likelihood values are first imputed in the data through iterated regressions, after which uncertainty is re-introduced by drawing parameter estimates and adding an error term from the imputed data. This two-stage approach is illustrated in Figure 4.1 below.

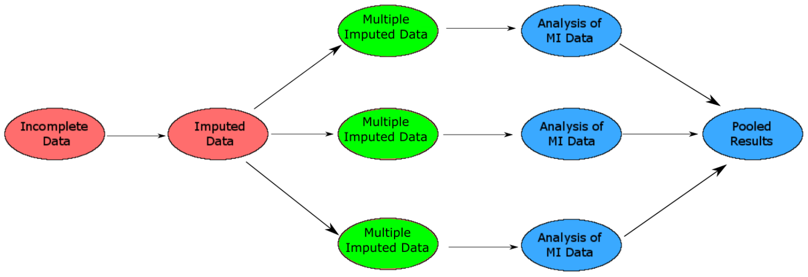

Figure 4.1: The Raoul Approach to Multiple Imputation

The Raoul approach has several advantages compared to the methods in section 3.4.1-3.4.3. First, compared to the methods based on MCMC simulations, the Raoul algorithm converges deterministically to the maximum likelihood estimates for the imputations which neutralizes the issue of convergence. Secondly, avoiding the MCMC simulations also makes the Raoul algorithm, just as the EM-bootstrapping algorithm, computationally more efficient than these methods. Unlike the EM-bootstrapping algorithm, which draws a bootstrapped sample for each of the m complete data sets and thus iterates the algorithm m times, the Raoul algorithm first converges and then creates m complete data sets. This strategy of first letting the algorithm converge and then recreating the uncertainty reduces the number of iterations to 1. Lastly, by employing imputation on a variable-by-variable basis based on all conditional distributions, rather than imputing based on the joint distribution, the Raoul algorithm avoids the reliance on the multivariate normal distribution.

4.1 Handling of Different Data Types

The main reason that a reliance on the multivariate normal distribution is problematic is that categorical and count variables cannot, by definition, follow such a distribution. Even more problematic is the fact that imputation methods based on the multivariate normal distribution will always impute continuous variables, which may not be suitable if missingness occurs on the dependent variable and specific analysis methods for count or categorical data are being used. To counter this problem, the Raoul algorithm allows for special imputation methods for count and categorical variables. More specifically, the Raoul algorithm makes imputations for count variables based on Poisson regression estimates, and imputations for categorical variables based on the predicted probabilities from logistic regression estimates.

It has been suggested that imputing continuous values for ordinal and count variables may, in many cases, be more sensible in an analysis setting than rounding these variables to their closest integer value (Honaker et al. 2011). Therefore, the Raoul package allows users to specify whether they want the algorithm to return integer values or continuous values for count variables. For categorical variables with r categories, the Raoul package will automatically create $r-1$ dummy variables containing observed values, and in the imputation phase it will impute the predicted probabilities for each category. As with the count variables, the user is then allowed to specify whether or not the data should be returned with predicted probabilities for each category, or if the data should be returned with proper categories.

4.2 The Raoul Algorithm

The Raoul algorithm is a two-step algorithm where first the most likely values are imputed, and then multiple imputations are created from this imputed dataset. The algorithm can therefore be divided into two parts where the first deals with how to reach the most likely imputations, while the second deals with how to generate multiple imputations from this data set. The first step is described in Algorithm 4.1 below, where $\mathbf{X}$ is the set of fully observed variables[11], $\mathbf{Y}$ is the set of k variables containing missing values, Y_j^{obs} are the observed values for variable j of $\mathbf{Y}$, Y_j^{mis} are the missing values for variable j of Y, $\dot{Y}^{[t]}$ are the imputations of Y at the t:th iteration of the algorithm, and $\dot{\beta}_j^{[t]}$ are the beta parameters from appropriate regressions of Y_j on the available data at iteration t.[12].

[11] If all variables contain missing values $\mathbf{X}$ is simply a constant

[12] See section 4.1 for details on different data types

```
1. Estimate $\dot{\beta}_j^{[1]}$ by $\dot{\beta}_j^{[1]} = E[\beta|Y_j^{obs}, \mathbf{X}]$ from appropriate regressions
2. Impute $\dot{Y}_j^{[1]}$ by $\dot{Y}_j^{[1]} = E[Y_j^{mis}|\dot{\beta}^{[1]}, \mathbf{X}]$
3. Repeat 1-2. for all $j = 1, ..., k$
4. Estimate $\dot{\beta}_j^{[t]}$ by $\dot{\beta}_j^{[t]} = E[\beta|\mathbf{X}, \bar{\mathbf{Y}}^{[t-1]}], t = 2, 3, ...$
   from appropriate regressions.
5. Impute $\dot{Y}_j^{[t]}$ by $\dot{Y}_j^{[t]} = E[Y_j^{mis}|\dot{\beta}^{[t]}, \mathbf{X}, \bar{\mathbf{Y}}_{-j}^{[t-1]}]$
6. Repeat 4-5. for all $j = 1, ..., k$
7. Repeat 4-6 until convergence
```

Algorithm 4.1: Step 1 of the Raoul algorithm

When Algorithm 4.1 has converged, multiple imputations are created by assuming that the estimated regression coefficients for each variable with missing values follow a multivariate normal distribution[13], i.e. that

$$\beta_j \sim N(\beta_j^{[T]}, \mathbf{\Sigma}_{\beta_j^{[T]}}) \tag{16}$$

where $\beta_j^{[T]}$ are the final estimates of $\dot{\beta}_j$ from Algorithm 4.1. The problem with this specification is the estimation of $\mathbf{\Sigma}_{\beta_j}^{[T]}$.In a regular OLS-setting, $\mathbf{\Sigma}_{\beta^{[T]}}$ would be estimated by $\hat{\sigma}_j^2(\mathbf{Z_{-j}}^{\mathrm{T}}\mathbf{Z_{-j}})^{-1}$, where $\mathbf{Z}$ is the full data set, i.e. $\mathbf{Z} = \{\mathbf{X}, \mathbf{Y}\}$, and σ_j^2 is the variance of the j:th variable in $\mathbf{Y}$ (Greene 2012). However, as $\mathbf{Y}$ contains missing values $\mathbf{Z}$ also contains missing values, and therefore $(\mathbf{Z_{-j}}^{\mathrm{T}}\mathbf{Z_{-j}})^{-1}$ cannot be calculated.

One solution to this problem would be to calculate $(\mathbf{Z_{-j}}^{obs\ \mathrm{T}}\mathbf{Z_{-j}}^{obs})^{-1}$, i.e. using Listwise Deletion, and only use the fully observed observations to estimate the covariance matrix of $\boldsymbol{\beta}_j$. This would, however, not make use of all available information. In order to make use of all information, the imputed data must be used in calculating the covariance matrix. However, as the imputed values will by their design increase the correlations between the variables, it is reasonable to weight the observations by the amount of missingness each observation experience.

$\mathbf{\Sigma}_{\beta_j}^{[T]}$ will therefore be estimated by $\hat{\sigma}_j^2(\bar{\mathbf{Z}}_{-j}^{\mathrm{T}}\mathbf{W}\bar{\mathbf{Z}}_{\mathbf{-j}})^{-1}$, where $\mathbf{W}$ is a diagonal matrix of weights, proportional to the rate of observed variables for each observation, and $\bar{\mathbf{Z}}$ is the complete data

[13] If the sample size is reasonably large, the Central Limit Theorem will ensure that this assumption approximately holds. For more details on the Central Limit Theorem and the normality of β-coefficients, see Greene (2012)

set with both observed and imputed values. In order to further avoid problems of decreased variability due to imputations, $\hat{\sigma}_j^2$ are estimated using only Y_j^{obs}. It is also used that

$$\frac{(n-1)\hat{\sigma}_j^2}{\sigma_j^2} \sim \chi^2_{(n_{j,obs}-p)}$$

why a random draw from $\chi^2_{(n_{j,obs}-p)}$, where $n_{j,obs}$ is the number of observed values for Y_j and p is the number of parameters, can be used to draw a random value for $\dot{\sigma}^2$.

These considerations form the foundation for the second stage of the Raoul algorithm, where the multiple imputations are created from the maximum likelihood imputed data. The details for this stage are found in Algorithm 4.2 below, where $M_{Y_{j,i}}$ is a binary variable taking the value 1 if observation i of variable Y_j is missing, $\mathbf{M}_{Y_j}$ denotes a vector of binary variables which indicate the missingness pattern of variable j from $\mathbf{Y}$, $\mathbf{1_n}$ denotes a vector of n 1's, and p denotes the total number of variables in $\mathbf{Z}$, n the total number of observations, $n_{j,mis}$ is the number of missing observations for variable Y_j, m the number of desired imputations, and $\bar{\mathbf{Z}}^{obs_j}$ are all observations in $\bar{\mathbf{Z}}$ which have observed values on variable Y_j.

```
1. Calculate $\mathbf{W} = \text{diag}\left[\frac{n}{\Sigma_{j=1}^{k}\Sigma_{i=1}^{n} M_{Y_{j,i}}/k}\left(\mathbf{1_n} - \Sigma_{j=1}^{k}\mathbf{M}_{Y_j}/k\right)\right]$
2. Start from $j = 1$
3. Calculate $\mathbf{V}_j = (\bar{\mathbf{Z}}_{-j}^{\mathrm{T}}\mathbf{W}\bar{\mathbf{Z}}_{\mathbf{-j}})^{-1}$
4. Calculate $\mathbf{V}_j^{1/2}$ by Cholesky decomposition
5. Calculate $\hat{\sigma}_j^2$ by $\hat{\sigma}_j^2 = (Y_j^{obs} - \bar{\mathbf{Z}}_{-j}^{obs_j}\dot{\boldsymbol{\beta}}_j^{[T]})^{\mathrm{T}}(Y_j^{obs} - \bar{\mathbf{Z}}_{-j}^{obs_j}\dot{\boldsymbol{\beta}}_j^{[T]})$
6. Draw a random variable $\dot{g}_j \sim \chi^2_{n_{j,obs}-p}$
7. Calculate $\dot{\sigma}_j^2 = \hat{\sigma}_j^2/\dot{g}_j$
8. Draw $k-1$ independent $N(0,1)$ variables in the vector $\dot{Q}_2$
9. Calculate $\dot{\boldsymbol{\beta}}_{j,s} = \dot{\boldsymbol{\beta}}_j^{[T]} + \dot{\sigma}_j^2\dot{Q}_1\mathbf{V}_j^{1/2}$
10. Draw $n_{j,mis}$ independent $N(0,1)$ variables in the vector $\dot{Q}_2$
11.
    (a) For continuous variables, impute
        $\dot{Y}_{j,s} = \bar{\mathbf{Z}}_{-j}\dot{\beta}_{j,s} + \dot{\sigma}_j^2\dot{Q}_2$
    (b) For binary categorical variables^14 impute
        $\dot{Y}_{j,s} = 1/\left(1 + \exp\left(-\bar{\mathbf{Z}}_{-j}\dot{\beta}_{j,s} - \dot{\sigma}_j^2\dot{Q}_2\right)\right)$
    (c) For count variables, impute
        $\dot{Y}_{j,s} = \exp(\bar{\mathbf{Z}}_{-j}\dot{\beta}_{j,r} + \dot{\sigma}_j^2\dot{Q}_2)$
12. Repeat 3-12 for $j = 1, ..., p$
13. *Optionally Algorithm 4.3
14. Repeat 2-13 for $s = 1, ..., m$
```

Algorithm 4.2: Step 2 of the Raoul algorithm

As discussed in section 4.1, the Raoul package allows users to decide whether or not they want their categorical and count data variables to be returned in their original form, or if they want their data to be returned with continuous values. If continuous values are selected, these values are based on, as can be seen in Algorithm 4.2 above, the fitted values from the Poisson and logistic regressions respectively. If the user desires the data to be returned in its original form, Algorithm 4.3 below is executed in order to return the data to its original state.

[14] For categorical variables with more than two categories, the predicted probabilities of the categories are imputed simultaneously to ensure that no nonsensical values are imputed.

```
Optional steps for categorical and count variables:
1. (a) If $Y_j, Y_{j+1}, ..., Y_{j+r-1}$ are a set of variables corresponding to the predicted probabilities of one categorical variable with $r$ categories in the original data
        i. Draw $n_{j,mis}$ independent $U(0,1)$ variables in the vector $R$
        ii. Define the predicted probabilities for each category as $p_q = \dot{Y}_{j+q-1,s}$ for $q = 1, ..., r-1$, $p_r = 1 - \Sigma_{q=1}^{r-1} p_q$, where $r$ is the reference category, and define $p_0 = 0$
        iii. For observation $i$, find $q_i$ such that
             $\sum_{h=0}^{q_i - 1} p_h < R_i < \sum_{h=0}^{q_i} p_h$ for all $i$ in $Y_j^{mis}$
        iv. Replace the set $Y_j, Y_{j+1}, ..., Y_{j+r-1}$ with an imputation of the single variable $\dot{Y}_{j,s}$, where each element in $\dot{Y}_{j,s}$, $\dot{Y}_{i,j,s} = q_i$.
   (b) If $Y_j$ is a count variable
        i. Replace each element in $\dot{Y}_j$ with a random draw such that $\overset{\vdots}{Y}_{i,j,s} \sim Po(\dot{Y}_{i,j,s})$
```

Algorithm 4.3: Optional step of the Raoul algorithm

4.3 Additional Functions in the Raoul Package

Apart to being a fast and easy package for creating multiply imputed data, the Raoul package also aims to make it easier to analyze results from multiply imputed data. The Raoul package therefore contains a set of functions which allows for pooling of results automatically, and thereby allows the user to directly estimate models and get interpretable results without having to go through several different stages or resort to third party packages.

More specifically, the Raoul package allows users to calculate summary statistics as well as estimate linear models and generalized linear models directly in R through the, `raoul.lm()` and `raoul.glm()` functions. The `raoul.lm()` and `raoul.glm()` call the R functions `lm()` and `glm()` respectively and runs the specified model on all multiply imputed data sets, and then pools the results. This pooling of the results is done through the so called *Rubin Rules*, which stipulates that the β coefficients should be estimated through the means of the m sets of estimated β coefficients(van Buuren 2012), while the standard errors of the estimated β coefficients are estimated through the following formula

$$s_b = \sqrt{\frac{1}{m}\sum_{k=1}^{M} s_{b,k}^2 + \left(1 + \frac{1}{m}\right)\left(\frac{1}{m-1}\right)\sum_{k=1}^{M}\left(b_k - \bar{b}\right)^2} \quad (17)$$

where m is the number of data sets, $s_{b,k}$ is the estimated standard error for β in data set k, b_k

is the estimated β value in data set k, and $\bar{b}$ is the mean of the estimated β coefficients (Allison 2009). The t-statistics and the p-values are then calculated based on these estimated β coefficients and their standard errors. The R-code can be downloaded from GitHub at `https://github.com/Airfixer/Raoul`, and examples of how to use the Raoul package can be found in Appendix B.

5 Comparative Simulation Study

In order to test the performance of the Raoul algorithm compared to existing missing data methods, a comparative simulation study is conducted. This simulation study compares the performance of the mice, Amelia, Raoul, and NORM packages for R on simulated missing data from two data sets with fully observed data. As a comparison to the conventional missing data methods, these more advanced methods are also be compared to Listwise Deletion.

More specifically, the performance of methods is tested on data which were MCaR, MaR, and NMaR, with the amount of missingness being set to 10%, 20%, and 40%, creating a total of 9 different specifications. While the methods used to replace missing data theoretically require the assumption of MaR to be fulfilled to work, it is nonetheless interesting to try simulations under NMaR circumstances. As noted in section 2.3, data which are NMaR can often be remedied by the inclusion of auxiliary variables which may make the data *near* MaR. The MaR and NMaR data are generated using different probabilities of missingness for different quantiles of an auxiliary variable for the data generated as MaR and different probabilities of missingness for different quantiles of the variable itself for the data which generated as NMaR.

The methods are evaluated on their performance in a regression setting with one dependent variable and three independent variables, with missingness simulated on all three independent varaibles. All evaluations are be made against the beta coefficients estimated with the complete data set. The procedure is repeated 1000 times, and the results averaged. Three criteria are used to evaluate the missing data methods:

1. Average Relative RMSE across the four β parameters. This value should be as low as possible.

2. Average β parameter mis-rate, i.e. average the proportion of cases where the complete data β parameter is outside the 95% confidence interval of the estimated β parameter of the method. Ideally, this value should be below but close to 0.05

3. Average Relative size of β standard error across the four β parameters. This value should be above but close to 1 under the condition that the mis-rate is below 0.05.

van Buuren (2012) argues that the methods should be evaluated using bias, coverage[15], and width of confidence intervals as evaluation criteria. This paper uses RMSE instead of bias as it provides a more comprehensive measurement of accuracy than bias. The relative size of the

[15] Coverage is the complement to the mis-rate, i.e. the proportion of cases where the complete data β parameter is *inside* the 95% confidence interval of the estimated β parameter

standard errors are used instead of the width of the confidence intervals as the width of the confidence intervals is simply a function of the size of the standard errors. In addition to these tests on the accuracy of the methods, all computer intensive methods, are also compared on their computational efficiency. This is done by measuring the time required to run the algorithm the 1000 iterations of the simulation.

Two data sets are used for the simulations. One larger set with 4177 observations on 8 variables[16] for collected Abalones gathered from the UCL Machine Learning Data Repository (Lichman 2013). This data set allows for testing the performance of the methods under the presence of a a larger number of fully observed auxiliary variables, and the large sample properties of the methods. The second data set is a data set on different irises with 149 observations on 5 variables, taken from the datasets of Applied Multivariate Statistical Analysis by Johnson and Wichern (2014). This data set allows for testing the performance in a smaller sample setting, with only one fully observed auxiliary variable. All variables with missing data were numeric variables. For clarity, only the results relating to the Abalone data are presented in the main text, while the results relating to the iris data can be found in Appendix A.

[16] 9 variable in the original data set, but the first variable was dropped.

6 Results

6.1 Results for Data which are MCaR

Figures 6.1-6.3 below shows the average relative RMSE, the average mis-rate, and the average relative standard error size for the different methods under different levels of simulated MCaR data. These results indicate that with regards to RMSE and mis-rate, the conventional method Listwise Deletion outperforms all of the more advanced methods for handling missing data. The four R-packages for handling missing data seem to perform relatively evenly for the two lower levels of missingness. For the simulations with 40% MCaR, however, the Raoul package performs substantially worse than its competitors. One reason for the poor performance of the Raoul package at 40% MCaR may be seen in the results on the average relative standard error size, as it has substantially lower standard errors compared to its competitors at 40% MCaR, while its mis-rate is very high. This indicates that the Raoul package produces too little variability in the data at higher levels of missing data.

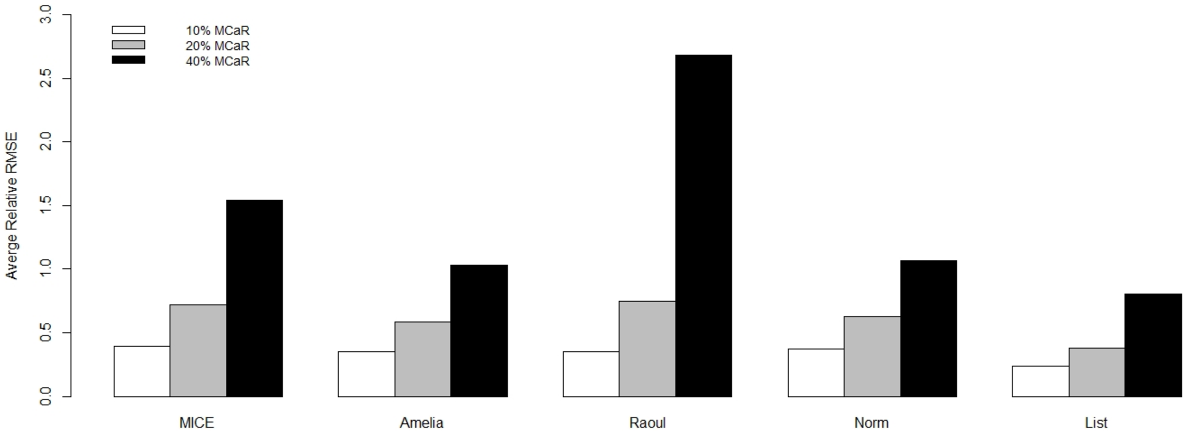

Figure 6.1: Average Relative RMSE of Beta Coefficients for different methods under MCaR in Abalone data.

From an inference perspective, it is interesting to note that while Listwise Deletion performs well on RMSE and mis-rate, it does produce very inflated standard errors under MCaR, which would make inference more difficult from the data. The exact numeric results corresponding to Figures 6.1-6.3 can be found in Tables A.1-A.3 in Appendix A. Figures A.1-A.3 in Appendix A, with corresponding numeric results in Tables A.10-A.12, show the same results for the Iris data, with roughly the same conclusions, except that the more advanced methods perform on par with Listwise Deletion for RMSE and mis-rate.

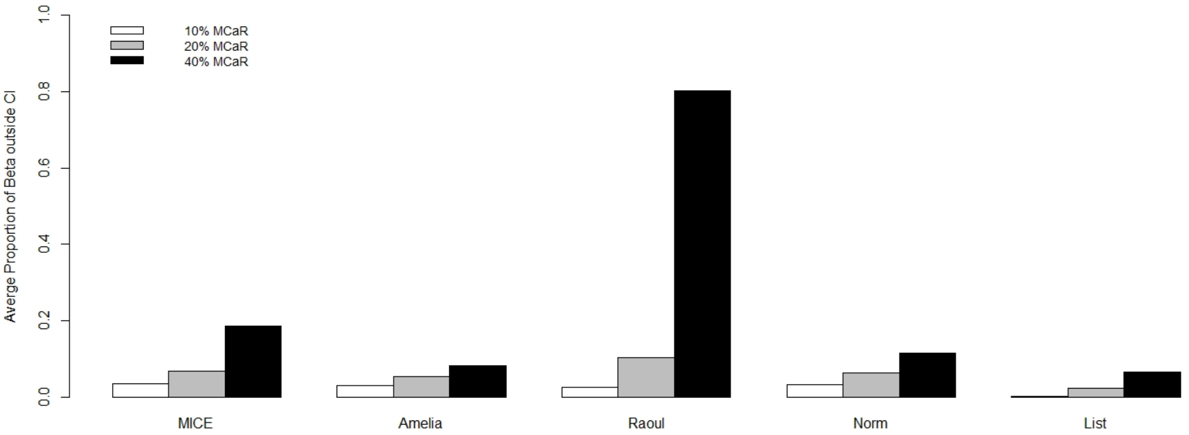

Figure 6.2: Average Relative mis-rate of Beta Coefficients for different methods under MCaR in Abalone data.

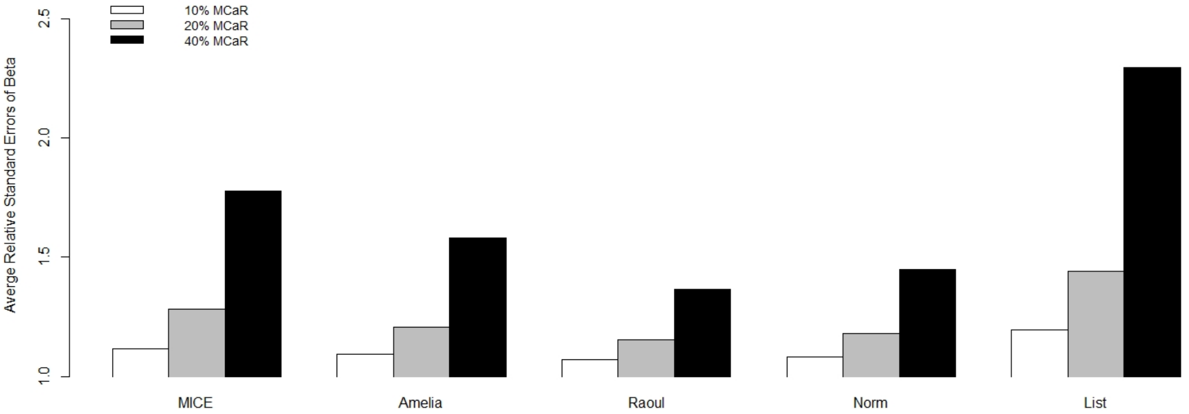

Figure 6.3: Average Relative Standard Error Size of Beta Coefficients for different methods under MCaR in Abalone data.

6.2 Results for Data which are MaR

The results for the data which are MaR, found in Figures 6.4-6.6. below, are similar to those in which the data was MCaR for the more advanced methods of handling missing data. The most substantial difference is that Listwise Deletion no longer fares well on any of the three evaluation criteria, as it has a high average relative RMSE, a high mis-rate, and still greatly overestimates the standard errors of the β coefficients. Among the four R-packages, the Raoul-packages has the lowest average relative RMSE values and relative size of standard errors while still maintaining a low mis-rate for the lower levels of missingness. Just as with the MCaR case, the Raoul package performs very poorly at 40% missingness. Among the three other packages, it is clear that the Amelia package and the NORM package outperform mice on all three criteria at all levels of missingness, and that their results are very similar. The results for the Iris data can be found in Figures A.4-A.6 in the appendix. Worth noting is that Listwise Deletion for the Iris data performs much better than for the Abalone data, and that the mice package outperforms both Amelia and NORM with regards to the relative standard error size, but not RMSE and mis-rate. The correponding numerical results are found in Tables A.4-A.6 for the abalone data, and in Tables A.13-A.15 for the Iris data.

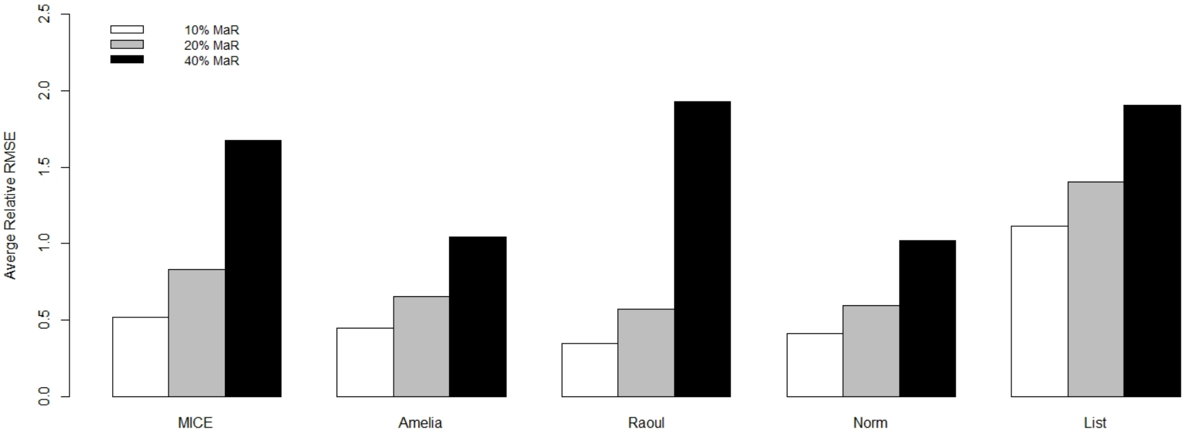

Figure 6.4: Average Relative RMSE of Beta Coefficients for different methods under MaR in Abalone data.

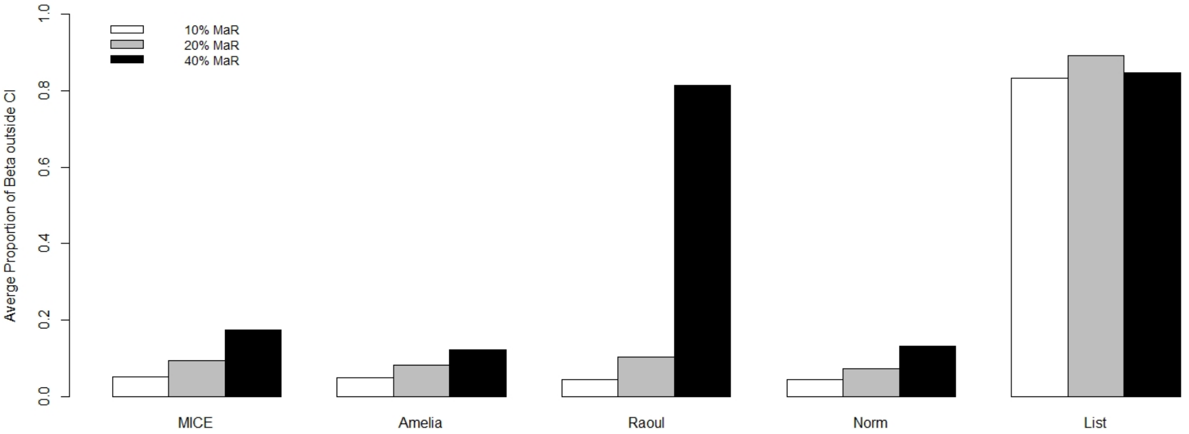

Figure 6.5: Average Relative mis-rate of Beta Coefficients for different methods under MaR in Abalone data.

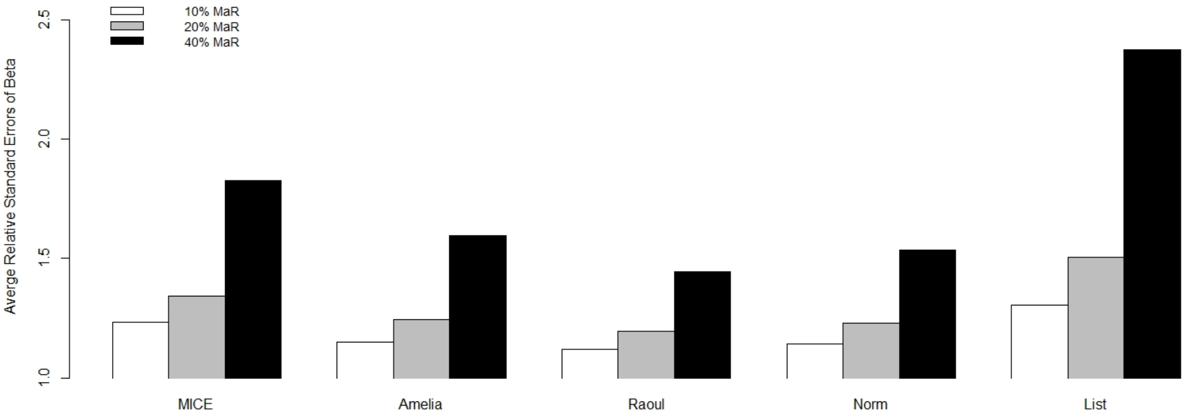

Figure 6.6: Average Relative Standard Error Size of Beta Coefficients for different methods under MaR in Abalone data.

6.3 Results for Data which are NMaR

The final set of simulations are conducted on data which are NMaR, the results of which can be found in Figures 6.7-6.9 below. Surprisingly, these results indicate that all of the methods seem to fare better under the NMaR data than under the MaR data, and that the general result structure of the MaR data remains, i.e. that Listwise Deletion fares worst, and Amelia and NORM fare best. As with the two earlier cases, the Raoul package performs well on the lower levels of missingness, but performs poorly at the higher level of missingness.

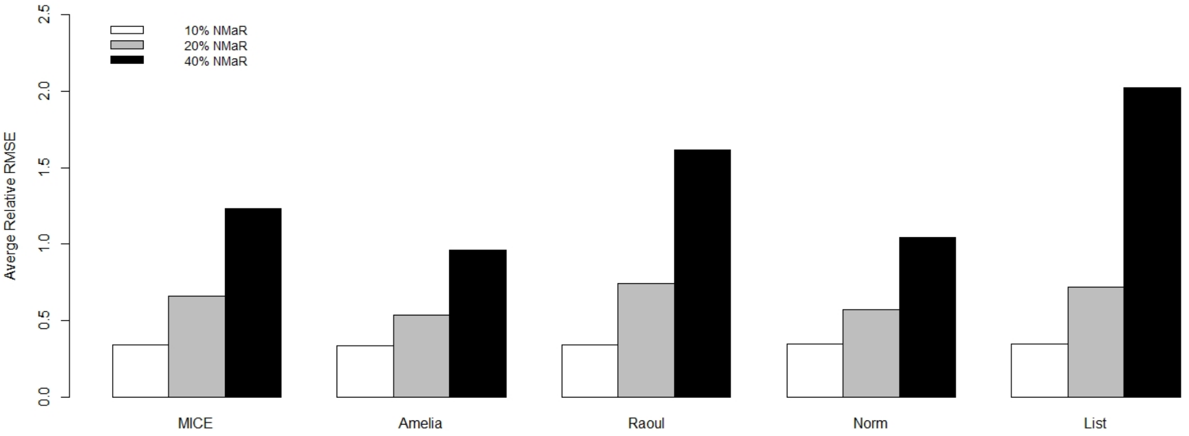

Figure 6.7: Average Relative RMSE of Beta Coefficients for different methods under NMaR in Abalone data.

These surprising results are even more clear for the Iris data, found in Tables A.7-A.9 in the Appendix, where Listwise Deletion performs on par with the more advanced methods on all theoretical criteria, although its relative standard errors are somewhat higher. The corresponding numerical results for the NMaR data can be found in Tables A.7-A.9 for the abalone data, and Tables A.16-A.18 for the Iris data.

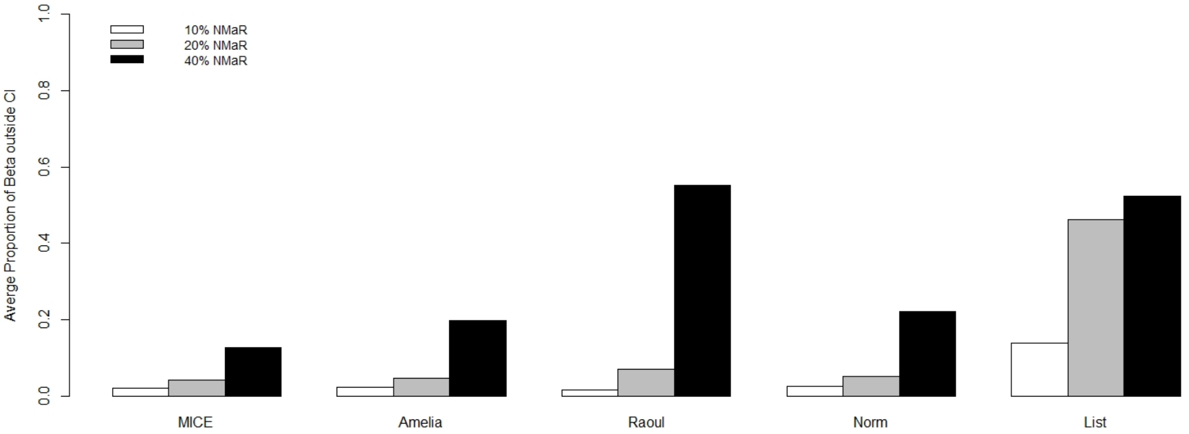

Figure 6.8: Average Relative mis-rate of Beta Coefficients for different methods under NMaR in Abalone data.

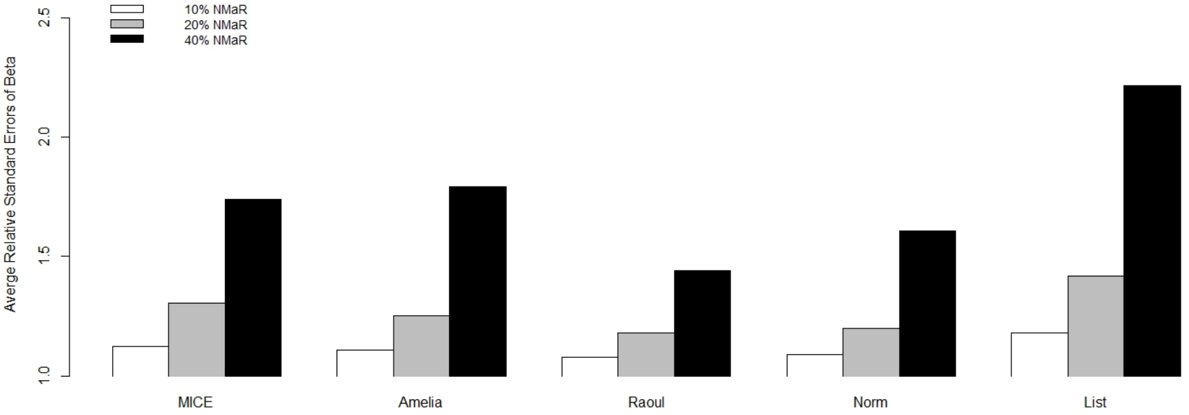

Figure 6.9: Average Relative Standard Error Size of Beta Coefficients for different methods under NMaR in Abalone data.

6.4 Computational Efficiency

Lastly, all methods were evaluated on their computational efficiency by measuring the time it took the four packages to iterate the procedure 1000 times for the different assumptions and amounts of missingness. These results are found in Table 6.1 below. As Listwise Deletion is an instant method for handling missing data, only the results from the four R-packages are reported.

Table 6.1: Time in Minutes to complete 1000 iterations in the abalone data for different levels and types of missingness

		mice	Amelia	Raoul	NORM
MCaR	10%	114.06	17.97	7.910	13.70
MCaR	20%	173.00	18.30	8.06	18.18
MCaR	40%	240.34	19.11	11.21	26.45
MaR	10%	112.56	18.36	7.25	13.70
MaR	20%	160.07	18.64	7.81	16.97
MaR	40%	241.94	19.64	9.16	26.34
NMaR	10%	117.18	18.24	7.95	14.00
NMaR	20%	176.70	18.58	8.42	18.44
NMaR	40%	241.22	20.06	11.86	26.92

These results clearly show that the Raoul package is by far the most computationally efficient package at all levels of missingness. It outperforms the Amelia and NORM packages by roughly a factor of 2-2.5 depending on the type and amount of missingess. The NORM package seems to be the second fastest method for handling the data when the amount of missingess is small, but is outperformed by the Amelia package at a 40% missingness level. The mice package was, by far, the computationally slowest, being outperformed by the Raoul package by a factor of roughly 14-24 depending on the type and amount of missingness, and by a factor of 6-12 by the Amelia and NORM packages.

7 Discussion

The results in the section above are highly interesting from several point of view. Firstly, they show that the Raoul package is a viable alternative to the existing packages for handling lower rates of missing data with a missingness of upwards of 20% and which are MCaR or MaR. The results also show that the Raoul package, in its current form, should *not* be used at higher levels of missingess. At these levels of missingness it is more advisable to use some of the already established packages for handling of missing data. Why this problem occurs at the higher levels of missingness is unclear, but it may be related to the Raoul algorithm's use of only the fully observed variables in the very first imputation step, i.e. step 2 in Algorithm 4.1. This use of only the fully observed variables may, if these variables have low explanatory power in the variables with missingness, create imputations in this first step which biases the subsequent imputations and thereby also the values which the algorithm converge to. This suspicion is strengthened by observation that the problem seems more severe in the Iris data, which only contained one auxiliary variable, than in the Abalone data, which contained several auxiliary variables.

Another interesting result is that the three established R packages perform relatively evenly with regards to their relative RMSE, mis-rate, and relative size of standard errors, with different packages claiming the title of ”best” for different types and amounts of missingness. However, even though the the established packages for handling missing data in most cases perform better than the Raoul package and Listwise Deletion, it is noteworthy that the mis-rate in many cases, especially for the abalone data, greatly exceed the 0.05 threshold set by van Buuren (2012). This indicates that while these packages perform better than the conventional methods, their relative standard error sizes may still be too low compared to what is advisable according to van Buuren (2012), as their mis-rate exceed the threshold. This is especially true for the higher levels of missingness, and the more ”severe” types of missingness, i.e. MaR and NMaR. With regards to computational efficiency it is, however, clear that the mice package is very slow compared to its competitors.

Lastly, the results show that the packages for handling missing data also perform relatively well under NMaR conditions, despite their theoretical need for at least MaR conditions. This is probably, as Allison (2012) argues, due to the inclusion of auxiliary variables correlated with the variables with missingness which may therefore bring the data close to MaR conditions even under NMaR. Perhaps most surprisingly is that even Listwise Deletion seems to be faring relatively well under NMaR conditions, especially when the amounts of missingness is relatively low.

8 Conclusions

This paper has introduced the Raoul package for handling of missing data under MCaR and MaR conditions, as well as compared its performance to existing methods for handling missing data in R. The results show that the Raoul package is a computationally fast method for handling missing data, which performs roughly on par with the existing packages, the mice package, the Amelia package, and the NORM package, under the condition that the amount of missingness is relatively low. At higher levels of missingness in the data, the Raoul package fails to generate usable imputations, while the existing packages continue to perform satisfactorily or near satisfactorily. The results also showed that the performance with regards to accuracy is relatively even among the existing packages for handling missing data. Regarding computational efficiency the Amelia and NORM packages outperform the mice package by roughly a factor of 6-12, and the Amelia and NORM packages are outperformed by the Raoul package by a factor of roughly 2-2.5.

Future work on the Raoul package may improve its ability to handle data with higher levels of missingness. One possible solution to the problem discussed in section 7 could be to sort the missing data by missingness patterns and then use Listwise Deletion, in the very first imputation step (step 2 in Algorithm 4.1), rather than only using the fully observed variables in this step.

Future studies should also focus on the performance of the existing missing data packages in different types and amounts of missingness, and for different types of data sets. For while the results between the two data sets used in this study are relatively consistent with one another, preliminary simulations on other data sets indicate that the performance of the different packages vary greatly with different types of data. The unsatisfactory mis-rate values for the existing missing data packages should also be further investigated, in order to establish whether or not this is a consistent result which appear in data with high amounts of missingness. If so, further work on the existing packages may be needed to address this problem.

9 References

Allison, Paul D. 2009. Missing Data. In *The SAGE Handbook of Quantitative Methods in Psychology*, edited by Roger E. Millsap and Alberto Maydeu-Olivares. Sage Publications.

Allison, Paul D. 2012. Handling Missing Data by Maximum Likelihood. SAS Global Forum.

Buuren, Stef van, and Karin Groothuis-Oudshoorn. 2011. Mice: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software* 45 (3): 167.

Buuren, Stef van. 2012. *Flexible Imputation of Missing Data.* CRC press.

de Waal, Ton, Jeroen Pannekoek, and Sander Scholtus. 2011. *Handbook of Statistical Data Editing and Imputation.* Wiley Handbooks in Survey Methodology. Hoboken, N.J: Wiley.

Greene, William H. 2012. *Econometric Analysis. 7th ed.* Boston: Prentice Hall.

Graham, John W. 2009. Missing Data Analysis: Making It Work in the Real World. *Annual Review of Psychology* 60 (1): 54976.

Heckman, J.J. (1979) Sample selection bias as a specification error, *Econometrica* 47: 153161.

Honaker, James, and Gary King. 2010. What to Do about Missing Values in Time-Series Cross-Section Data. *American Journal of Political Science* 54 (2): 56181.

Honaker, James, Gary King, Matthew Blackwell, and others. 2011. Amelia II: A Program for Missing Data. *Journal of Statistical Software* 45 (7): 147.

King, Gary, James Honaker, Anne Joseph, and Kenneth Scheve. 2001. Analyzing Incomplete Political Science Data: An Alternative Algorithm for Multiple Imputation. *American Political Science Association* 95 (1): 4969.

Lee, Hyunshik, Eric Rancourt, and Carl E. Srndal. 1994. Experiments with Variance Estimation from Survey Data with Imputed Values. *Journal of Official Statistics* 10: 231231.

Lichman, M. 2013. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Little, R. 1993. Pattern-mixture models for multivariate incomplete data. *J. Am. Stat. Assoc.* 88:125-34

Little, R. 1994. A class of pattern-mixture models for normal incomplete data. *Biometrika* 81:471-83

Little, R. 1995. Modeling the drop-out mechanism in repeated-measures studies. *J. Am. Stat. Assoc.* 90:1112-1121

Novo, Alvaro A. (Ported to R) and Joseph L. Schafer (Original). 2013. NORM: Analysis of multivariate normal datasets with missing values.

Randahl, David. 2015. Dealing with Missing Data. Term Paper, Uppsala: Uppsala University.

Rubin, Donald B. 1987. *Multiple Imputation for Nonresponse in Surveys.* Wiley Series in Probability and Mathematical Statistics. New York: Wiley.

SAS Institute Inc. 2015. *SAS/STAT 14.1 Users Guide.* Cary, NC: SAS Institute Inc.

Schafer, Joseph L. 1997. *Analysis of Incomplete Multivariate Data.* CRC press.

Schafer, Joseph L., and John W. Graham. 2002. Missing Data: Our View of the State of the Art. *Psychological Methods* 7 (2): 14777.

Schafer, Joseph L., and Nathaniel Schenker. 2000. Inference with Imputed Conditional Means. *Journal of the American Statistical Association* 95 (449): 14454.

Takahashi, Masayoshi, and Takayuki Ito. 2013. Multiple Imputation of Missing Values in Economic Surveys: Comparison of Competing Algorithms. In *Proceedings of the 59th World Statistics Congress of the International Statistical Institute*, 32403245.

Appendix A. Extended Results

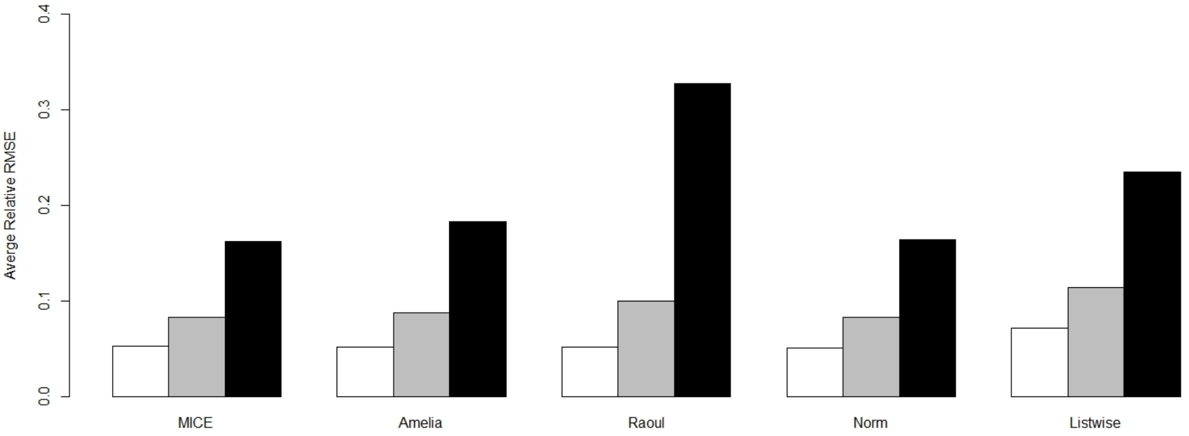

Figure A.1: Average Relative RMSE of Beta Coefficients for different methods under MCaR in Iris data.

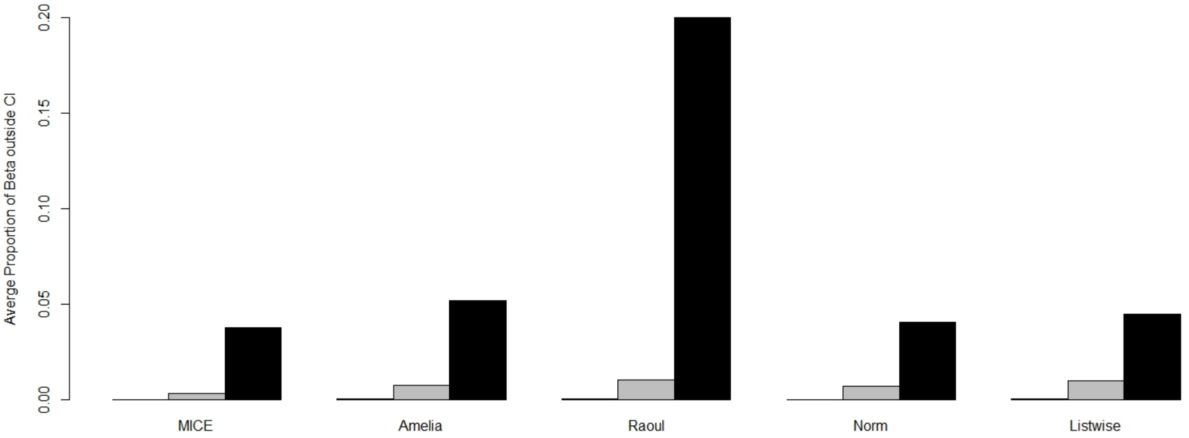

Figure A.2: Average Relative mis-rate of Beta Coefficients for different methods under MCaR in Iris data. Values cropped at 0.2, for actual values see Table A.2.

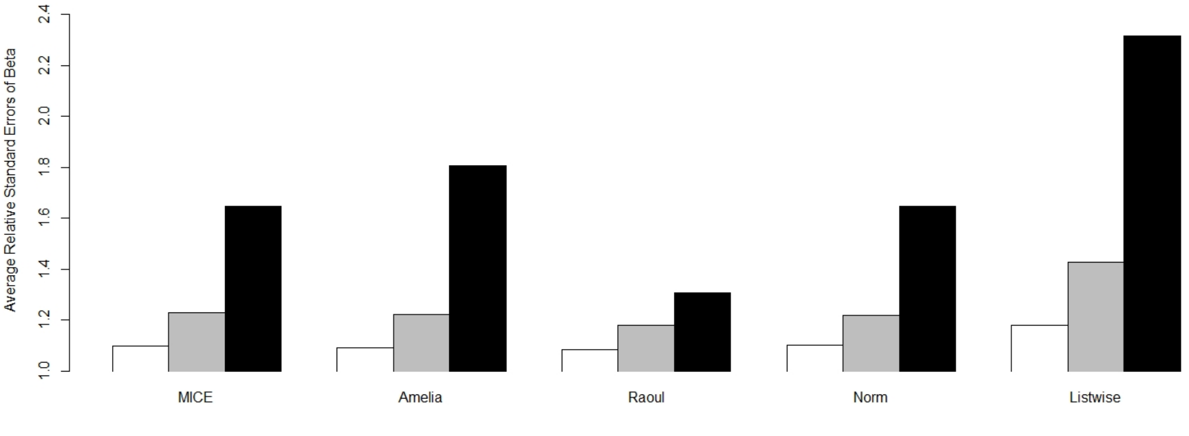

Figure A.3: Average Relative Standard Error Size of Beta Coefficients for different methods under MCaR in Iris data.

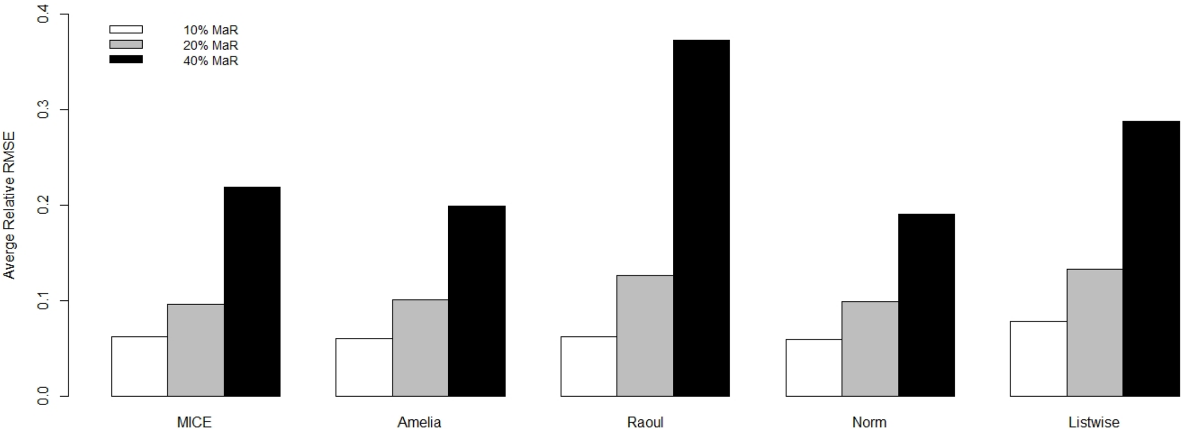

Figure A.4: Average Relative RMSE of Beta Coefficients for different methods under MaR in Iris data.

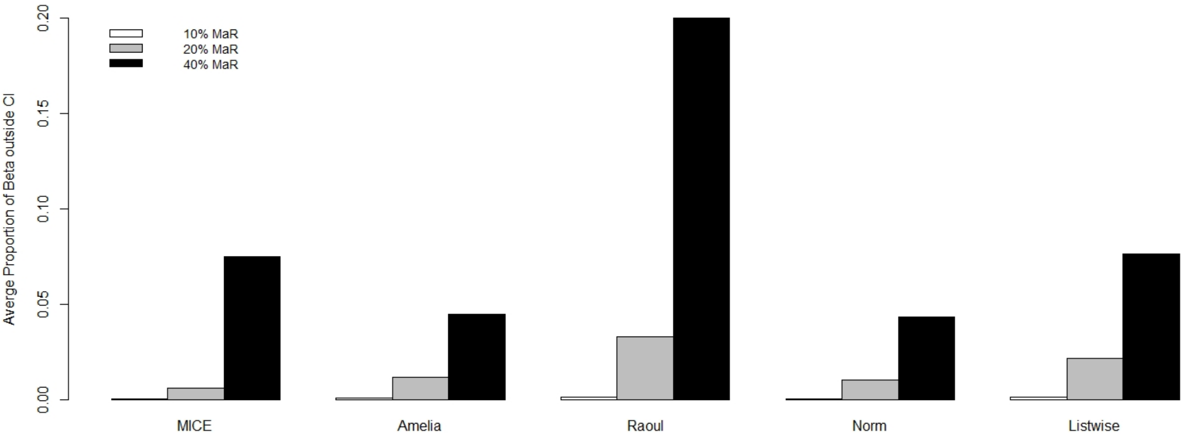

Figure A.5: Average Relative mis-rate of Beta Coefficients for different methods under MaR in Iris data. Values cropped at 0.2, for actual values see Table A.5.

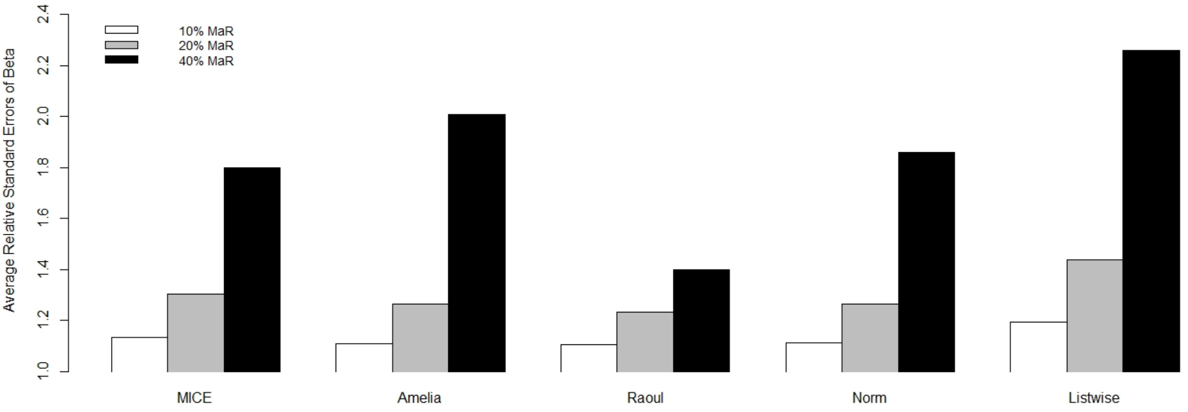

Figure A.6: Average Relative Standard Error Size of Beta Coefficients for different methods under MaR in Iris data.

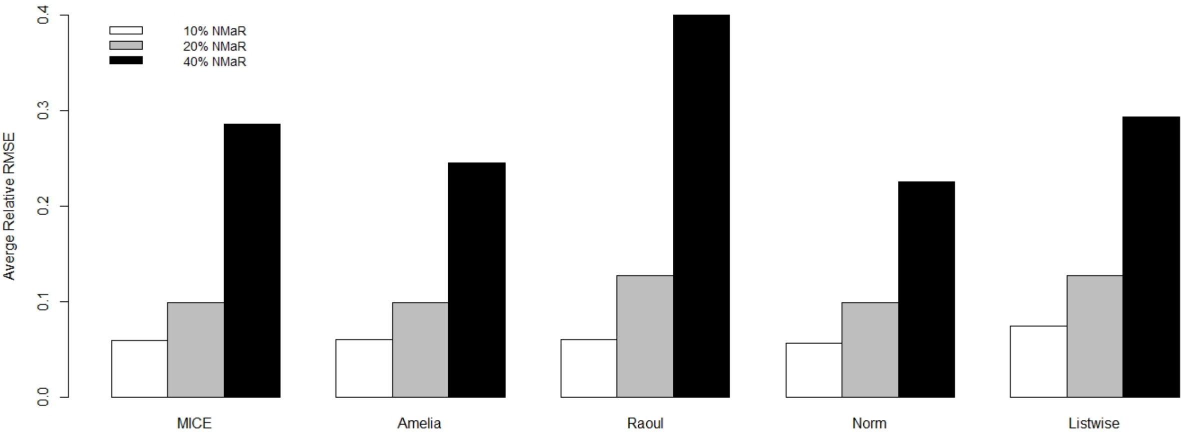

Figure A.7: Average Relative RMSE of Beta Coefficients for different methods under NMaR in Iris data.

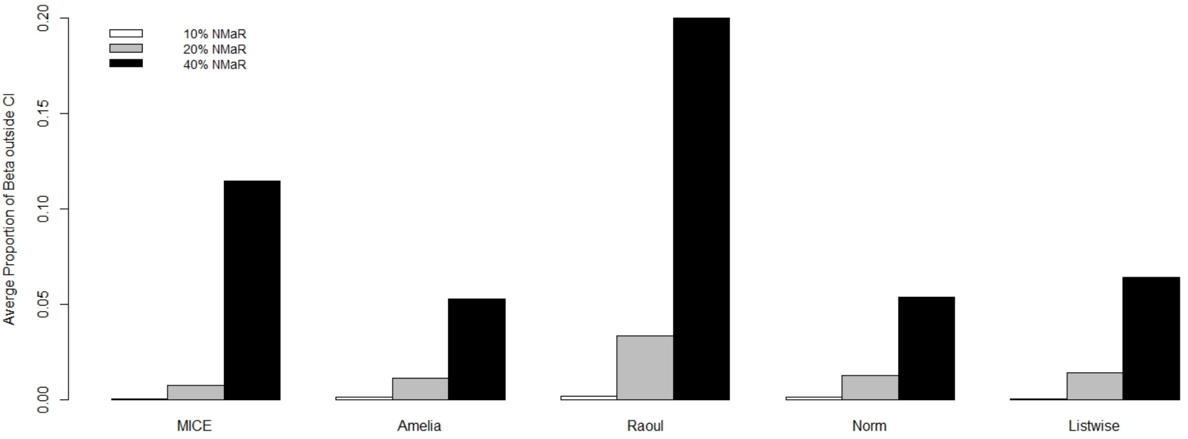

Figure A.8: Average Relative mis-rate of Beta Coefficients for different methods under NMaR in Iris data. Values cropped at 0.2, for actual values see Table A.8.

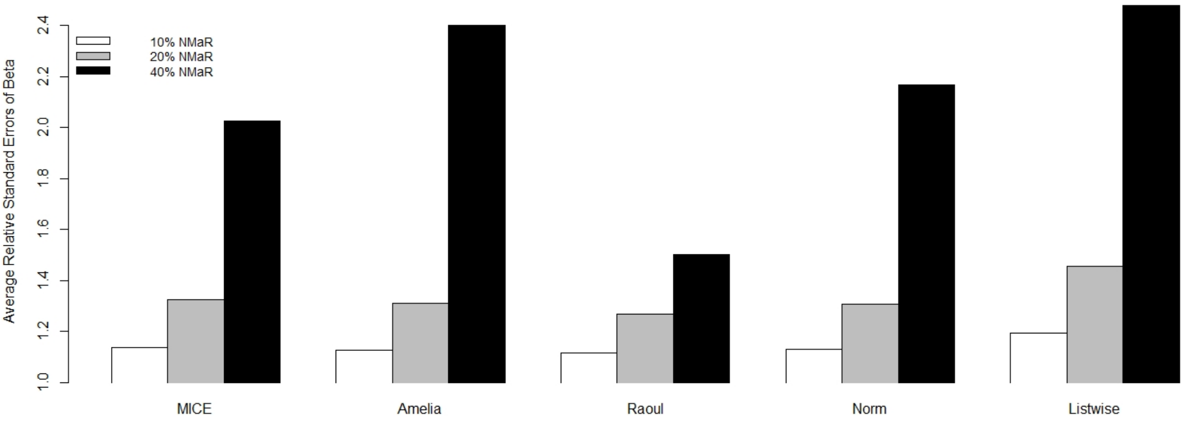

Figure A.9: Average Relative Standard Error Size of Beta Coefficients for different methods under MaR in Iris data.

Table A.1: Average Relative RMSE of Beta Coefficients under MCaR in Abalone data.

	10% MCaR	20% MCaR	40% MCaR
mice	0.394	0.722	1.543
Amelia	0.349	0.588	1.032
Raoul	0.350	0.745	2.682
NORM	0.374	0.630	1.064
List	0.236	0.383	0.802

Table A.2: Average Relative mis-rate of Beta Coefficients under MCaR in Abalone data.

	10% MCaR	20% MCaR	40% MCaR
mice	0.036	0.067	0.185
Amelia	0.030	0.053	0.082
Raoul	0.025	0.104	0.802
NORM	0.032	0.063	0.134
List	0.002	0.022	0.064

Table A.3: Average Relative Standard Error Size for Beta Coefficients under MCaR in Abalone data.

	10% MCaR	20% MCaR	40% MCaR
mice	1.117	1.282	1.777
Amelia	1.092	1.208	1.582
Raoul	1.072	1.154	1.364
NORM	1.081	1.180	1.450
Listwise	1.194	1.441	2.295

Table A.4: Average Relative RMSE of Beta Coefficients under MaR in Abalone data.

	10% MaR	20% MaR	40% MaR
mice	0.516	0.832	1.677
Amelia	0.446	0.654	1.045
Raoul	0.348	0.573	1.931
NORM	0.410	0.596	1.020
List	1.113	1.403	1.904

Table A.5: Average Relative mis-rate of Beta Coefficients under MaR in Abalone data.

	10% MaR	20% MaR	40% MaR
mice	0.052	0.093	0.174
Amelia	0.049	0.080	0.121
Raoul	0.044	0.103	0.834
NORM	0.044	0.073	0.132
List	0.833	0.892	0.846

Table A.6: Average Relative Standard Error Size for Beta Coefficients under MaR in Abalone data.

	10% MaR	20% MaR	40% MaR
mice	1.232	1.344	1.825
Amelia	1.151	1.243	1.596
Raoul	1.121	1.196	1.446
NORM	1.144	1.229	1.533
List	1.304	1.506	2.374

Table A.7: Average Relative RMSE of Beta Coefficients under NMaR in Abalone data.

	10% NMaR	20% NMaR	40% NMaR
mice	0.342	0.659	1.229
Amelia	0.335	0.533	0.959
Raoul	0.342	0.741	1.615
NORM	0.344	0.569	1.042
List	0.343	0.720	2.020

Table A.8: Average Relative mis-rate of Beta Coefficients under NMaR in Abalone data.

	10% NMaR	20% NMaR	40% NMaR
mice	0.020	0.041	0.126
Amelia	0.023	0.046	0.196
Raoul	0.0.016	0.070	0.552
NORM	0.025	0.052	0.221
List	0.138	0.462	0.524

Table A.9: Average Relative Standard Error Size for Beta Coefficients under NMaR in Abalone data

	10% NMaR	20% NMaR	40% NMaR
mice	1.124	1.304	1.739
Amelia	1.109	1.250	1.792
Raoul	1.079	1.179	1.440
NORM	1.090	1.198	1.606
List	1.179	1.417	2.217

Table A.10: Average Relative RMSE of Beta Coefficients under MCaR in Iris data.

	10% MCaR	20% MCaR	40% MCaR
mice	0.052	0.083	0.162
Amelia	0.051	0.087	0.183
Raoul	0.052	0.100	0.328
NORM	0.051	0.083	0.164
Listwise	0.071	0.114	0.235

Table A.11: Average Relative mis-rate of Beta Coefficients under MCaR in Iris data.

	10% MCaR	20% MCaR	40% MCaR
mice	0	0.003	0.038
Amelia	0	0.007	0.052
Raoul	0	0.010	0.452
NORM	0	0.007	0.040
Listwise	0	0.010	0.045

Table A.12: Average Relative Standard Error Size for Beta Coefficients under MCaR in Iris data

mice	1.097	1.229	1.647
Amelia	1.092	1.221	1.808
Raoul	1.083	1.178	1.308
NORM	1.100	1.220	1.646
Listwise	1.178	1.426	2.315

Table A.13: Average Relative RMSE of Beta Coefficients under MaR in Iris data.

	10% MaR	20% MaR	40% MaR
mice	0.062	0.096	0.218
Amelia	0.060	0.100	0.199
Raoul	0.062	0.126	0.373
NORM	0.059	0.098	0.190
Listwise	0.078	0.133	0.288

Table A.14: Average Relative mis-rate of Beta Coefficients under MaR in Iris data.

	10% MaR	20% MaR	40% MaR
mice	0	0.006	0.075
Amelia	0.001	0.012	0.045
Raoul	0.001	0.033	0.480
NORM	0	0.010	0.957
Listwise	0.001	0.022	0.076

Table A.15: Average Relative Standard Error Size for Beta Coefficients under MaR in Iris data.

	10% MaR	20% MaR	40% MaR
mice	1.132	1.304	1.798
Amelia	1.110	1.264	2.008
Raoul	1.105	1.233	1.399
NORM	1.112	1.266	1.861
Listwise	1.193	1.438	2.261

Table A.16: Average Relative RMSE of Beta Coefficients under NMaR in Iris data.

	10% NMaR	20% NMaR	40% NMaR
mice	0.059	0.099	0.286
Amelia	0.060	0.099	0.245
Raoul	0.060	0.127	0.414
NORM	0.056	0.099	0.225
Listwise	0.075	0.127	0.293

Table A.17: Average Relative mis-rate of Beta Coefficients under NMaR in Iris data.

	10% NMaR	20% NMaR	40% NMaR
mice	0	0.007	0.115
Amelia	0.001	0.011	0.053
Raoul	0.002	0.034	0.533
NORM	0.001	0.013	0.054
Listwise	0	0.014	0.064

Table A.18: Average Relative Standard Error Size for Beta Coefficients under NMaR in Iris data

	10% NMaR	20% NMaR	40% NMaR
mice	1.138	1.324	2.025
Amelia	1.128	1.310	2.403
Raoul	1.115	1.267	1.503
NORM	1.129	1.306	2.167
Listwise	1.192	1.457	2.479

Appendix B. Examples for Raoul in R

```
## Install Raoul
library(devtools)
install_github("Airfixer/Raoul",force=TRUE)

## Access data files
library(mice)

## Simple example
set.seed(1)
x1 <- runif(10)
x2 <- runif(10)
x1[1] <- NA
d <- data.frame(x1, x2)
rd <- Raoul::raoul(d)

## Simple example
data(nhanes)
rn<-Raoul::raoul(nhanes)

## Example with two factors
data(selfreport)
dat <- selfreport[, c("age", "sex", "hm", "wm", "hr", "wr", "edu")]
rs <- Raoul::raoul(dat, facs = c(2, 7))

## Example with two factors
data(selfreport)
dat <- selfreport[, c("age", "sex", "hm", "wm", "hr", "wr", "edu")]
rs <- Raoul::raoul(dat, facs = c(2, 7), returncat=TRUE)

## Example with one factor
data(nhanes)
nhanes$hyp <- as.factor(nhanes$hyp)
```

```
rn <- Raoul::raoul(nhanes, facs = 3,returncat=TRUE)

## Example with two factors
data(nhanes)
nhanes$hyp <- as.factor(nhanes$hyp)
nhanes$fubar <- as.factor(sample(c("foo", "bar", "baz"), nrow(nhanes
    ), replace = TRUE))
nhanes$fubar[8:9] <- NA
rn <- Raoul::raoul(nhanes, facs = c(3, 5))

## Example with count
data(nhanes)
rn <- Raoul::raoul(nhanes, counts = 3)

## Example with count and factor
data(nhanes)
nhanes$fubar <- as.factor(sample(c("foo", "bar", "baz"), nrow(nhanes
    ), replace = TRUE))
nhanes$fubar[8:9] <- NA
rn <- Raoul::raoul(nhanes, counts = 3, facs= 5)

## LM Regression example
data(nhanes)
d<-nhanes
rlm<-Raoul::raoul.lm(bmi~age+hyp+chl,raoul=d)

### GLM Regression example

rglm<-Raoul::raoul.glm(hyp~bmi+chl,raoul=d,fam="binomial",facs=3)
```