Erasmus University Rotterdam
Erasmus School of Economics
Master Thesis Business Analytics & Quantitative Marketing

'Garbage in means garbage out': An analysis of imputation techniques under four missingness mechanisms using direct and indirect evaluation combined with computation time

Sjoerd van Velzen (456903)

Deloitte.

Supervisor:	prof. dr. Richard Paap
Deloitte supervisor:	Evert Roobeek
Second assessor:	Name of your second assessor
Date final version:	23rd March 2024

Abstract

This thesis investigates the performance of various (single and multiple) imputation algorithms under four missingness mechanisms: MCAR (Missing Completely At Random), MAR (Missing At Random), and MNAR (Missing Not At Random) Type 1 and 2. Using six different datasets, we evaluate the methods using direct evaluation with RMSE (Root Mean Squared Error) and PCP (Percentage of Correct Predictions), as well as indirect evaluation (post-imputation classification accuracy). Furthermore, computation time is taken into account when determining which methods are most suitable for business use. We find that no single method outperformed all other methods in all evaluation criteria. The goal for why imputation is required heavily influences which method is best, as well as data type and size. For categorical data under MCAR, MAR and MNAR Type 1, Predictive Mean Matching is superior to all other algorithms based on the PCP metric. Under Type 2 MNAR missingness, Linear Bayesian Regression obtained the highest PCP values. For continuous data, non-parametric methods such as k-Nearest Neighbours and Random Forest show great accuracy and post-imputation classification accuracy under all missingness mechanisms, but also become very computationally intensive for large datasets, highlighting the importance of practical constraints in business applications. Under Type 2 MNAR, post-imputation classification accuracy for categorical variables showed that not imputing the values at all yields better results compared to imputing them.

Keywords: Multiple Imputation, Direct Evaluation, Classification, Type 2 MNAR

Contents

1. Introduction

Total data creation is estimated to reach a staggering 180 zettabytes in 2025 (Statista, 2022). With this rise in data creation, data quality management becomes equally important. The availability and utilization of these vast amounts of data have revolutionized various fields: from business to healthcare, and from social sciences to engineering. However, the quality of that data is as, if not more, important than the amount of data available in order to get reliable and valid results. As IBM computer scientist and instructor George Fuechsel famously coined: *"Garbage in means garbage out"* (Fuechsel, 1960s).

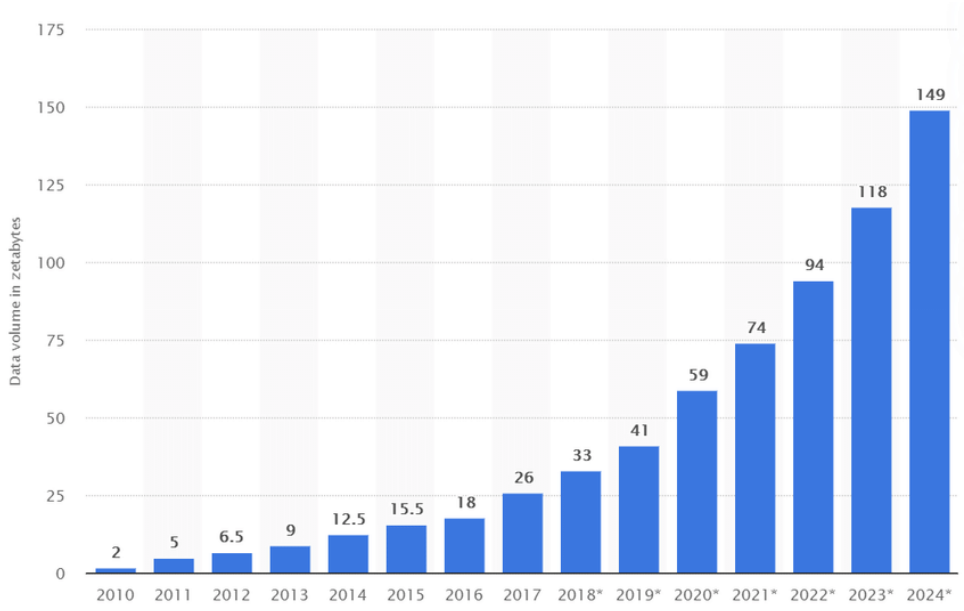

Figure 1.1: Total data volume through the years. (Statista, 2022)

One critical aspect of data quality management is the way in which missing values are handled. Missing values are inevitable in real-world datasets, and how they are dealt with can significantly impact the efficiency, outcome and reliability of data mining techniques (R. J. Little & Rubin, 2019; Jadhav, Pramod & Ramanathan, 2019). Furthermore, for machine learning applications, high data quality is crucial for robust predictions and automated decision-making (Jäger, Allhorn & Bießmann, 2021). As such, the process of filling in or estimating missing values in a dataset, known as data imputation, has emerged as a vital tool in the arsenal of data scientists and analysts. The rise in popularity of data imputation can be seen in Figure A.1 in Appendix A, where the number of publications regarding data imputation has increased

drastically from approximately a dozen a year a decade ago to more than 50 a year in the past couple of years.

Initially, approaches to data imputation were relatively straightforward, often involving the deletion of cases with missing values (complete-case analysis) or filling them with mean or median values (mean/median imputation). However, these techniques come with some inherent flaws, such as the distortion of the data distribution and the loss of information (R. J. Little & Rubin, 2019). Over time, more sophisticated and advanced techniques have been introduced in order to tackle this problem (Schafer & Olsen, 1998).

The existing research regarding missing values considers multiple different mechanisms for missing data. These are called missingness mechanisms. The three main mechanisms are Missing Completely At Random (MCAR), Missing At Random (MAR) and Missing Not At Random (MNAR) (Rubin, 1976). This paper will consider all three and will aim to investigate the effects of the different missingness mechanisms on the performance of imputation methods and downstream data mining tasks. Additionally, this paper investigates the Type 2 MNAR mechanism using a novel approach to introducing missing values.

In the literature, a significant amount of work has been done regarding the various statistical and machine learning techniques that can be implemented for data imputation. A majority of this literature, however, predominantly focuses on the mathematical and computational aspects (Lin & Tsai, 2020). A less explored avenue is understanding and addressing the problem of missing data from a business or company perspective. Here is where this thesis finds its niche, aiming to bridge this gap in the literature by considering the company's viewpoint on data imputation. It delves into how firms can navigate the complexities of missing data, assessing the practicality and time-effectiveness of different imputation methods in a real-world business context when underlying missingness mechanisms are often unknown. Thus, the main research question that this thesis will aim to answer is:

"Which imputation method yields most reliable and robust results under different missingness mechanisms, while being feasible for implementation in a business setting?"

To confidently address the main research question, we have to consider multiple factors. First off, the main factor to include is imputation accuracy, or direct evaluation. This is the most researched part of the imputation literature, and is also important for businesses (Lin & Tsai, 2020). Therefore, the first sub-question that needs to be answered is as follows:

"Which imputation method most accurately estimates missing values?"

In order to obtain an answer to this question, a direct evaluation of imputation accuracy, the Root Mean Squared Error (RMSE), will be used. Another important factor to consider, one which might be more important to companies in certain scenarios than direct performance, is indirect performance. Indirect performance refers to the efficacy of data mining tasks

post-imputation. In the research by Lin and Tsai (2020), only 8 out of 111 analyzed papers investigated both direct and indirect performance, showing that more research is needed on that front. In business analytics, imputation techniques that are effective at generating data that can be used for actionable insights are of paramount importance. For this reason, classification tasks are a good method to measure this effectiveness, as classification tasks are often used by businesses in real life for various objectives, such as predicting customer churn, customer segmentation or fraud detection. Therefore, a classification task in this setting is in line with industry practice. Colleagues at Deloitte, a global consultancy firm, have shared that these classification tasks are common in their client projects. Aside from that, another reason we use a classification task to measure indirect performance is that if an imputation method preserves classification accuracy well, it is a good indicator that the imputation technique has not ruined the internal structures and relationships of the data. Therefore, the sub-question that will be answered in regards to this problem is as follows:

"Which imputation method best preserves classification accuracy post-imputation?"

In a business setting it might not be feasible to spend lots of time setting up an algorithm, and tuning it to perfection. Furthermore, when dealing with large datasets, some methods, especially machine learning methods, can become very computationally intensive. This is not desirable, especially in instances where a company wants to set up a pipeline that continuously imputes the missing values. Again, Lin and Tsai (2020) showed that only 11 out of 111 investigated papers considered some metric for computational efficiency. Therefore, the last sub-question is as follows:

"Which imputation methods are computationally attractive and feasible for business applications?"

To answer that question, computational time will be taken into consideration, as well as the amount of tuning parameters and simplicity of setting the algorithm up.

Lastly, this paper will feature an extension where the MNAR missingness mechanism will be explored in more detail. More precisely, the Type 2 MNAR mechanism will be explored where missing probabilities depend on factors outside of the dataset. As this missing mechanism is often found in real life, it is important to know how to deal with this adequately. Therefore, a final additional sub-question is:

"Which method is most suitable under Type 2 MNAR missingness?"

This thesis is organized as follows: to start, Chapter 2 will offer a summary of the existing literature. Next, Chapter 3 will discuss the data used or generated in this research, detailing any transformations and related aspects. In Chapter 4, a thorough overview of the statistical and machine learning methods will be presented, including the assumptions, strengths, and weaknesses of each method. This chapter will also provide a comprehensive explanation of all

the different missingness mechanisms. Then, Chapter 5 will reveal the main results of this research. After that, Chapter 6 will cover the results of the extension of this thesis regarding Type 2 MNAR missingness. To wrap up, Chapter 7 will further discuss all results, leading to the final conclusions.

2. Related Work

In this section, we aim to shed light on the existing body of literature surrounding missing value imputation (MVI). The goal is to provide a clear overview of the current state of research in this area, highlighting both the strengths and the potential gaps in current knowledge. We will outline the key findings of several significant papers, offering a straightforward summary of their results. Additionally, areas where current research might be lacking will be identified, pointing towards opportunities for further exploration and investigation in this domain.

A big downside often found in the body of literature surrounding missing value imputation is that they mainly rely on smaller machine learning datasets. These usually have a few features, no more than 100, and not a lot of data samples, generally only going from a couple of hundred to a few thousand (Lin & Tsai, 2020). However, there have been some exceptions to this trend. For instance, the studies by Folino and Pisani (2016) and Farhangfar, Kurgan and Pedrycz (2007) made use of much larger datasets. These encompassed 216 feature dimensions and included a massive number of data samples, with counts of 581,012 and 256,000 samples respectively. However, the study by Folino and Pisani (2016) did not study the actual imputation of values, but rather a meta-ensemble method of classifiers for handling missing values. Additionally, Folino and Pisani (2016) only investigated the MAR mechanism, and Farhangfar et al. (2007) only investigated MCAR. Farhangfar et al. (2007) found that their unsupervised imputation methods were more stable compared to the supervised ones, indicating that the imputation accuracy gets less worse than other methods when increasing the amount of missing values. Specifically, the unsupervised methods seemed less sensitive to the amount of missing values.

In a paper by Jäger et al. (2021), a comprehensive benchmark of several classical and modern imputation techniques was performed. They analysed the data and results under realistic circumstances, with different missingness mechanisms and percentages of missing values. A wide range of real-life datasets were used in order to get a good understanding of the performances under different conditions, with a maximum of 25 variables and 100.000 observations. The authors compared several imputation techniques, including Mean/Mode, k-NN, Random Forest, Discriminative Deep Learning, Variational Autoencoder Imputation and Generative Adversarial Network Imputation. For the evaluation of imputation accuracy of continuous variables and of regression tasks, Jäger et al. (2021) used the RMSE metric. For the evaluation of categorical variables and classification tasks, they used the macro F1-score metric. Jäger et al. (2021) found that simpler supervised learning methods often obtain similar results, and sometimes

even outperform, modern generative methods. The authors do point out that the deep learning approaches can be significantly slower than other methods, due to the time it takes to optimize and train the model. They concluded that Random Forest Imputation yielded the most accurate results, also accounting for post-imputation data mining performance.

Hameed and Ali (2022) analyzed seventeen different MVI techniques. These included, but were not limited to, mean/mode, k-NN, regression, Multiple Imputation by Chained Equations (MICE), Expectation Maximization (EM) Imputation and Support Vector Machine (SVM) Imputation. They found that the performance of these methods depends on various factors, such as the field of study, the performance matrix used and the characteristics of the dataset (Hameed & Ali, 2022). They found that MICE worked well in the medical field, whilst also highlighting that popular methods like mean imputation, k-NN and MICE are not necessarily the most efficient, meaning that they either become computationally intensive for large datasets (k-NN, MICE) or their reliability for imputations is questionable (mean imputation).

The study by Faisal and Tutz (2021) focuses on the modern biomedical field where missing data is a common issue. The field of biomedical sciences often contains a large number of mixed-type variables, which not all imputation methods can handle well. The authors proposed a novel imputation method that uses a weighted version of nearest neighbours to accommodate mixed data. They compared it to existing methods, namely k-NN and Random Forest imputation. Faisal and Tutz (2021) tested their method on a number of different, real or simulated, datasets and found that their proposed method yielded smaller imputation errors than the two aforementioned methods. The imputation error did however heavily rely on the size of the correlation in the data. Additionally, they note that their proposed method was relatively time-consuming due to the hyperparameter optimization being essential for good results. The authors used the proportion of falsely imputed categories (PFC) and mean squared imputation error (MSIE) for the evaluation of categorical and continuous imputations, respectively.

Jadhav et al. (2019) aimed to gain insight into data missingness mechanisms and imputation performance. They did this by analying seven different imputation algorithms, including mean imputation, median imputation, k-NN imputation, predictive mean matching and Bayesian Linear Regression. Five different datasets from the University of California, Irvine repository were used to test these methods. It should be noted that these datasets were quite small, as none of them exceeded more than 1030 observations and 13 attributes, and that they were all numerical datasets. They analyzed the methods using the Normalized Root Mean Squared Error (NRMSE) metric. They found that the k-NN algorithm outperformed all others and that its relative performance was independent of the dataset and the amount of missing values used (Jadhav et al., 2019).

In a paper by Randahl (2022), the effect of missing values and imputation techniques on the forecasted number of fatalities as an effect of political violence in various countries was investigated. He investigated all three main missingness mechanisms using 7200 simulations. The

imputation algorithms that were tested were mean imputation, Random Forest, k-NN, Predictive Mean Matching, Expectation Maximization and Bayesian linear regression. He found that the choice of imputation algorithm significantly affected the predicted casualties, and perhaps surprisingly found that single imputation methods worked best overall, especially the k-NN and Random Forest algorithms (Randahl, 2022).

To conclude, the existing body of literature relies mostly on small datasets with few variables and often do not investigate all missingness mechanisms. Exceptions do exist, but the combination of sizable datasets, all mechanisms and utilizing both direct and indirect evaluation remains a gap in the literature. This thesis will aim to bridge this gap, by using both small and bigger datasets, exploring all missingness mechanisms, and using both direct and indirect evaluation of the methods. Some studies suggest that the type of data is paramount to choosing the right imputation technique (Faisal & Tutz, 2021). This will also be looked at in further detail. Additionally, this thesis will aim to gain insight into which imputation technique works best for business applications, incorporating computation time, feasibility, and robustness to the different types of missingness as well. Finally, very few papers exist on the Type 2 MNAR mechanism and its effect on the optimal imputation strategy. This thesis will aim to find answers to the questions that arise from this gap in the literature.

3. Data

This section will explain the data that is used in this research. It will be divided into two parts: public data sets and simulated data sets. All transformations and other steps in data preparation will be explained in order to make this research reproducible. An overview of all datasets and their characteristics can be found in Table 3.1.

3.1 Public data sets

All public data sets are retrieved from either the University of California, Irvine (UCI) Machine Learning repository, or Kaggle.

3.1.1 Iris

The Iris dataset by Fisher (1988) is a very well-known dataset, used in many studies for classification or other machine learning tasks (Eirola, Doquire, Verleysen & Lendasse, 2013; Kiasari, Jang & Lee, 2017; Silva-Ramírez, Pino-Mejías & López-Coello, 2015; Hameed & Ali, 2022; Gautam & Ravi, 2015). It contains 150 observations of 5 variables, one of which is the target variable. The four predictor variables are height and width measurements of the petals and sepals of the Iris flower, which are used to predict which of the three species of Iris flower it is. All predictor variables are continuous, ranging from 0.1 to 7.9 cm. The four predictor variables do not have a multivariate normal distribution, as can be seen from the Mardia's tests in Table A.1 in Appendix A (Mardia, 1970). When categorizing by the outcome variable however, they do approximate multivariate normality. This dataset only needed one transformation, where the target variable was transformed from a string to a categorical variable by coding the strings to integer values.

3.1.2 Car Evaluation

The car evaluation dataset by Bohanec (1997) is a dataset consisting of exclusively categorical variables. It has 1,728 observations containing 6 predictor variables and 1 multinomial target variable with 4 levels (unacceptable, acceptable, good, very good). The 6 predictor variables can be divided into two groups: price (buying price, maintenance price) and technology (number of doors, capacity, size of luggage boot, estimated safety). The goal is to predict the acceptability of the car. This dataset can be used in order to evaluate the performance of imputation methods for categorical variables. As the variables are all categorical, they follow a multinomial distribution. Most predictor variables are balanced, meaning that the categories of each attribute are

approximately equally present in the dataset. The multinomial target variable 'Acceptability' is imbalanced however, with 'unacc' (unacceptable) being the most frequent (1210 out of 1728 observations). This dataset was transformed in the same way as the Iris dataset, where all string (categorical) values were coded to be represented as integer values.

3.1.3 Banknote Authentication

The banknote dataset by Lohweg (2013) is a dataset often used for training of binary classification tasks. It consists of 1,372 observations with four continuous predictor variables and one binary target variable (0 for authentic, 1 for inauthentic). The predictor variables are the Variance, Skewness and Kurtosis of a wavelet-transformed image, as well as the overall entropy of the image. All values come from real-life images of banknotes. The variables are not normally distributed, but rather skewed (See Table A.2 in Appendix A). This dataset did not need any transformations or adjustments before use.

3.1.4 Adult/Census Income

The adult dataset (also known as Census Income) by Becker and Kohavi (1996) is a dataset that is commonly used for classification tasks. The goal is to predict whether an individual's earnings exceed \$50.000 per year based on various factors. The dataset contains 32,561 observations of 15 variables, of which one is the target variable where 0 denotes an income of less than \$50.000 and 1 denotes an income of \$50.000 or more. The data contains both categorical/binary and continuous variables. The categorical/binary variables are *workclass*, *education*, *marital-status*, *occupation*, *relationship*, *race*, *gender* and *native-country*. The continuous variables are *age*, *fnlwgt*, *education-num*, *capital-gain*, *capital-loss* and *hours-per-week*. The categorical and binary variables were again mapped to be represented by integers.

Table 3.1: Summary of datasets and their characteristics

Dataset Name	**Observations**	**Variables**	**Variable Type**	**Domain**
Iris	150	5	Continuous	Biology
Banknote Authentication	1,372	5	Continuous	Finance
Car Evaluation	1,727	7	Categorical	Automotive
Adult/Census Income	32,561	15	Mixed	Demographics
Simulated Data	1,000	18	Mixed	Simulation

3.2 Simulated Data

In the interest of obtaining data that adhere to the assumptions of most models, namely the multivariate normal assumption, simulated datasets were created. This way, a multivariate normal dataset could be created in combination with categorical or binary variables with explanatory power. This was done by first simulating a continuous multivariate normal classification dataset with a binary target variable using the *sim_classification* function from R package **modeldata** (Kuhn, 2023). This function takes a few parameters, including the number of observations,

number of linear predictors and an intercept value. The intercept value (α) was set at -5, with 1000 observations and 5 linear predictors. The predictors are simulated in two sets: first, two multivariate normal predictors (β_1 and β_2) are simulated with a correlation of approximately 0.65 and a zero mean. These change the log-odds using main effects and an interaction term as follows:

$$\alpha - 4\beta_1 + 4\beta_2 + 2\beta_1\beta_2$$

The intercept can be changed to introduce class imbalance. An intercept of -5 gives a slight imbalance towards the positive class, meaning that approximately 550 observations will have a 1 as their outcome variable. The second set of predictors ($\gamma_k, k = 1, ..., 5$) are linear, and contribute to the log-odds using alternating signs and coefficients having a constant sequence of values between 2.5 and 0.25. Their contribution to the log-odds is as follows:

$$-2.5\gamma_1 + 1.9375\gamma_2 - 1.375\gamma_3 + 0.8125\gamma_4 - 0.25\gamma_5$$

All linear predictors follow a standard univariate normal distribution with mean 0 and a standard deviation of 1.

After that, four random categorical variables were added that had no explanatory value for the target. These variables attain values ranging from 0 to 4, with equal probability. To add explanatory categorical variables, conditional probabilities were used where the probability of a low number was higher when the target value was 0, and conversely, the probability of a high number was higher when the target value was 1. The categorical variables range in values from 1 to 4. When the target variable is 0, the probabilities of getting a 1, 2, 3 or 4 were $[0.5, 0.4, 0.1, 0.0]$, respectively. When the target variable is 1, these probabilities change to $[0.1, 0.2, 0.3, 0.4]$ respectively.

In the interest of seeing what happens to the imputation performance when data is not normally distributed, we create another dataset where we add skewness to our originally simulated data. For this, we use the exact same simulated dataset as explained before, but add a transformation to the first two normally distributed variables. These are also the variables where missing values will be introduced. The transformation consists of two parts. First, we apply a Box-Cox transformation on the variable after we make all values positive, see Equation 3.1. Then, we shift the data back, see Equation 3.2.

$$X_{skewed} = \frac{(X - \min(X) + 1)^{\lambda} - 1}{\lambda} \tag{3.1}$$

$$X_{skewedback} = X_{skewed} + \min(X) - 1 \tag{3.2}$$

After this transformation, the range of the variables was different than the original variables. In order to still be able to compare the two in our direct evaluation, a linear transformation was applied. This transformation consists of two steps: first, we make sure that the difference between the minimum value and the maximum value is the same. This can be done by

multiplying the skewed data by a scaling factor, which is calculated as follows:

$$\frac{\max(X_{original}) - \min(X_{original})}{\max(X_{skewed}) - \min(X_{skewed})} \tag{3.3}$$

After that, the only thing left to do is to shift the data so that they have the same range. This can be done by adding the following to the scaled data:

$$\frac{\max(X_{original}) + \min(X_{original})}{2} - \frac{\max(X_{scaled}) + \min(X_{scaled})}{2} \tag{3.4}$$

In Equation 3.1, the λ parameter determines how much skewness is introduced. When $0 < \lambda < 1$, the data will become right-skewed. When $\lambda > 1$, the data will become left-skewed. Smaller values of λ will result in more right-skewness. In this thesis, λ will be set to 0.01. See Figure 3.1 for the histograms of the two transformed of our simulated data, before and after performing the transformation.

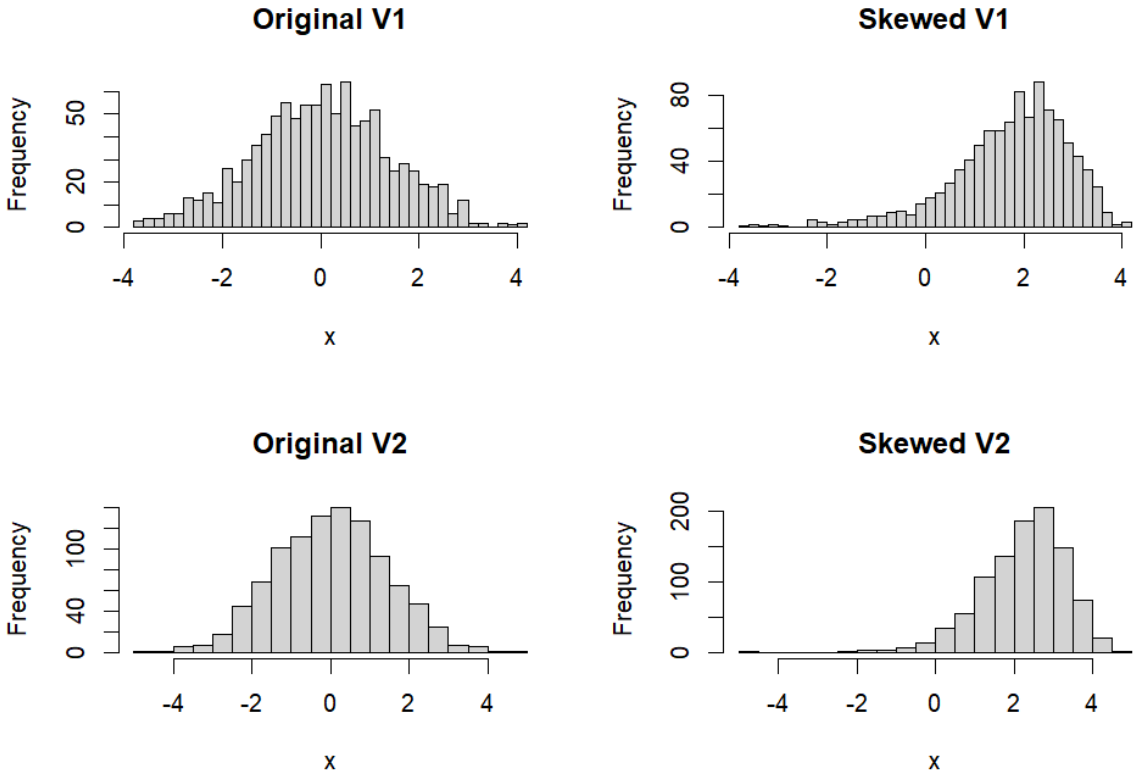

Figure 3.1: Histograms of the first simulated variable, before and after Box-Cox transformation

4. Methods

This section is structured as follows: Section 4.1 and 4.2 will explain all imputation methods that are used in this paper. It will cover how they work, what kind of assumptions they make and the potential advantages and disadvantages that they bring. After that, Section 4.4 will explain the three main missingness mechanisms, and Section 4.5 will explain the extension on this paper with an extra, less explored, missingness mechanism in the Type 2 MNAR missingness.

4.1 Single Imputation

This section will outline each of the single missing value imputation methods. It will explain how they are used, how they work and their potential advantages or disadvantages.

4.1.1 Listwise deletion

Listwise deletion, or complete-case analysis, refers to the analysis where only the complete observations are considered. All observations that have one or more missing attributes get removed from the dataset, after which the required data mining task is performed on the remaining data. This method is simple, but may introduce bias, affects variability and causes a big loss of data and precision (Hameed & Ali, 2022).

4.1.2 Mean/Median - Mode Imputation

Mean/mode or median/mode imputation is used as a baseline imputation method. This imputation technique is most often used in practice because it is the easiest to implement (Ambler, Omar & Royston, 2007). These methods work by simply taking the mean or median value of all observed values for each continuous variable, and imputing the missing values with this number. For categorical or binary variables, we take the mode of each variable and impute the missing values accordingly. When using the mean for the imputation of continuous variables, the algorithm can be sensitive to outliers. Using the mode circumvents this problem. However, it should be noted that both methods can significantly change the mean and standard deviation of the data post-imputation (Hameed & Ali, 2022).

4.1.3 k-NN Imputation

K-nearest neighbours (k-NN) imputation is a non-parametric method that uses a predefined distance metric to quantify the degree of proximity among observations. In this research, the *kNN()* function from R package **VIM** was used to implement this imputation method. The

algorithm consists of two steps. First, the distances between the observation with a missing value to all other observations are calculated. Then, the k closest neighbours are aggregated to calculate the imputed value.
The measure of distance between two observations is computed as a weighted mean of the individual contributions from each variable. The selection of weights reflects the significance of each variable in the analysis. Consequently, the distance between the i-th and j-th observations is defined as follows:

$$d_{i,j} = \frac{\sum_{k=1}^{p} w_k \delta_{i,j,k}}{\sum_{k=1}^{p} w_k} \tag{4.1}$$

where $d_{i,j}$ is the distance between observation i and j, w_k is the weight of variable k and $\delta_{i,j,k}$ is the contribution of the k-th variable. The weights w_k are calculated using a Random Forest regression, where the variable weights are based on the variable importance according to a Random Forest regression. These feature importances range from 0 to 1, and thus can be directly used as weights for the k-NN imputation. The main advantage of this method is that the algorithm can assign higher weights to variables that are important for predicting the outcome variable, which can lead to more accurate imputations. For continuous variables, the method involves calculating the absolute distance between elements and then dividing it by the total range of the variables:

$$\delta_{i,j,k} = \frac{|x_{i,k} - x_{j,k}|}{r_k} \tag{4.2}$$

where $x_{i,k}$ is the value of the k-th variable of the i-th observation and r_k is the range of the k-th variable, i.e. its maximum value minus its minimum value (Kowarik & Templ, 2016). For ordinal variables, the categories are converted to integers, after which it follows the same contribution equation (Equation 4.2). As such, if an ordinal variable has values 1, 2, 3 and 4, the different values are seen as equidistant. One can change these values to adjust the distance between the unique values.
For nominal and binary variables, a regular 0/1 distance is used:

$$\delta_{i,j,k} = \begin{cases} 0 & \text{if } x_{i,k} = x_{j,k}, \\ 1 & \text{if } x_{i,k} \neq x_{j,k}. \end{cases} \tag{4.3}$$

After all the distances are calculated, the k nearest neighbours to the observation with the missing value (based on $d_{i,j}$) are aggregated to calculate the imputed value. For continuous variables, the imputed value will simply be the median value of these k nearest neighbours. For categorical variables, the mode of the k nearest neighbours is used to impute the missing value.

An advantage of this technique is that it is suitable for both continuous and categorical variables. Furthermore, it eliminates the necessity of generating a predictive model for every individual missing data attribute. This is particularly advantageous in scenarios involving multiple instances of missing values (Hameed & Ali, 2022). However, as this method requires the algorithm to go through each observation to calculate its distance to the incomplete observation,

it gets computationally intensive as the data set grows large.

4.2 Multiple Imputation

The idea of Multiple Imputation (MI) was first proposed by Rubin (1978) and refers to the process of imputing a certain missing values multiple times, each time generating a slightly different value. The main idea behind MI is that it tries to encapsulate the inherent uncertainty that comes with missing value imputation. With single imputation, subsequent analyses are performed based on the assumption that the imputed value is the true value, which is a very strong assumption and often not true. MI aims to circumvent this problem by imputing the missing values M times, and performing the subsequent analyses on all M datasets, treating each imputed dataset as if it were the one real dataset (Rubin, 1978). After that, the resulting parameter estimates get pooled together in order to get a more accurate estimate with the uncertainty of imputation included.

In this thesis, all MI methods except for Expectation Maximization were performed using R package **mice** (Van Buuren & Groothuis-Oudshoorn, 2011). MICE is an acronym for Multiple Imputation by Chained Equations. It is a widely used approach in the broad framework of MI. MICE works as follows: It starts with the initialization step, where the algorithm performs a simple imputation, namely mean or mode, depending on the data type. Then, the iterative procedure starts. Each variable that had missing values is marked as missing. One by one, each variable with missing values is used as a target variable, with the other variables being used as predictors, including the simply imputed variables. Each time, a model is trained with those predictors and target variables. This is where the different model selections play into effect.

In this thesis, we use Linear Bayesian Regression, Predictive Mean Matching and Random Forest. The missing values are imputed based on the predictions of the selected model. The variables get imputed one by one, each time also using the most recent imputed values in the models. After all variables have been updated, the entire process repeats itself multiple times in a chained fashion, meaning that each time the newly imputed values are used to predict the missing values. The amount of iterations can be adjusted. In this thesis, the *maxiter* variable was set to 10 each time, as the creators of the algorithm state that good results should be obtained after as few as 5 or 10 iterations (Van Buuren & Oudshoorn, 2000). I chose to keep it at 10, and not more, due to time and computational restraints. After these 10 iterations, we have imputed one dataset. This entire process is repeated M times, each time getting new estimates for the missing values. Normally, one would perform analyses on all M datasets, but as we are focusing on the imputation performance we will take a different approach. Instead, we will average all M imputations to get the final estimates of our missing values. For categorical variables, the average values will be rounded to obtain the final imputed value. We use the rounded mean, as the categorical variables that we impute generally have an order to them, making the rounded mean represent a central tendency of the imputations. The only dataset where this does not fully apply is the Adult dataset, where we impute the *maritalstatus* variable.

The rest of this section will outline each of the multiple imputation techniques that are used in this research. It will explain how they are used, how they work and their potential advantages or disadvantages (Van Buuren & Oudshoorn, 2000).

4.2.1 Expectation Maximization Imputation

The Expectation Maximization (EM) Imputation algorithm makes use of an iterative procedure in order to estimate the missing values. It achieves this by determining the maximum likelihood estimates for the parameters of the distribution that the entire dataset is assumed to follow. The algorithm assumes multivariate normality, thus it estimates the means and (co)variances of the multivariate normal distribution. In the context of missing value imputation, this method seeks to estimate the missing values in a dataset by maximizing the expected log-likelihood of that value. The iterative procedure contains two steps. The first step is the E-step, or the Expectation step. The algorithm evaluates the conditional expectation of the log-likelihood of the data (Krishnan & McLachlan, 2012).

$$Q(\theta|\theta^{(t)}) = \mathbb{E}_{Z|X,\theta^{(t)}}[\log L(\theta; X, Z)] \tag{4.4}$$

Here, the expectation is taken with respect to the missing data (Z), conditional on the complete or observed data (X) and the parameters of the previous iteration ($\theta^{(t)}$). Additionally, Z are the latent variables. In the context of data imputation, the missing values are treated as latent variables. In the E step, these missing values are estimated given the current estimated parameters and the observed data.

The second step is the M-step, or the Maximization step. Here, the algorithm maximizes the expected log-likelihood that was evaluated in the E-step in order to obtain new estimates for the parameters (Ghomrawi, Mandl, Rutledge, Alexiades & Mazumdar, 2011).

$$\theta^{(t+1)} = \underset{\theta}{\operatorname{argmax}}\, Q(\theta|\theta^{(t)}) \tag{4.5}$$

In other words, the algorithm finds the parameters that maximize the new expected log-likelihood, using both the observed and missing data in the process. These two steps are repeated in an iterative manner until the algorithm reaches convergence, when the change in log-likelihood from one iteration to the next gets smaller than a certain threshold, or the difference in parameter estimates crosses a certain threshold (Lin & Tsai, 2020; Krishnan & McLachlan, 2012).

The EM imputation algorithm has some drawbacks. First, it is possible that, if multiple local maxima exist, the EM algorithm can converge to a local maximum of the log-likelihood instead of a global maximum (Redner & Walker, 1984). Furthermore, if the estimation of the complete-data maximum likelihood is complicated, the algorithm can become computationally intensive (McLachlan & Krishnan, 2007). This is especially the case for large datasets. Lastly, even though the EM algorithm does not have many assumptions, it does assume multivariate normality and a MAR missingness mechanism, which can be particularly challenging to satisfy for categorical or binary data (McLachlan & Krishnan, 2007). The implementation of the EM algorithm in this thesis is done using R package **Amelia** (Honaker, King & Blackwell, 2011).

4.2.2 Linear Bayes Imputation

The Linear Bayes (LB) imputation algorithm uses a Bayesian framework to impute the missing values. It specifies a linear model, which implies that it assumes a linear relationship between variables. This linear model takes the following form:

$$Y = \beta_0 + \sum_{i=1}^{p} \beta_i x_i + \epsilon \tag{4.6}$$

where Y is the dependent variable, β_0 is the constant term, the β_i's are the coefficients corresponding to variables x_i and ϵ is the error term, which is assumed to be normally distributed (Raftery, Madigan & Hoeting, 1997).

To estimate the imputed values, it first estimates the parameters of the linear model. Using the Bayesian method, it treats each parameter β as a random variable, and provides a distribution for these parameters, capturing the uncertainty around them. These distributions are called priors. These priors can be a standard normal distribution, but they can also take other forms when information is already available from previous studies for example. Using the observed data, these priors are updated to form posterior distributions using Bayes' Theorem (Joyce, 2003). For each missing value, the model uses the estimated parameters to generate a value that fits the data and the model. Because it is a Bayesian technique, it samples these values from the posterior distribution which means it encapsulates the uncertainty of the imputation. Because of that uncertainty, the LB algorithm is used in a multiple imputation setting, where it generates multiple plausible values for one missing cell.

4.2.3 Random Forest Imputation

Random Forest Imputation, also known as Missing Forest in some literature, is a non-parametric imputation method that harnesses the strengths of the machine learning algorithm Random Forest (RF). The conceptual framework of RF, which serves as the cornerstone of this imputation method, is founded on the idea of constructing multiple decision trees on the training set and outputting a value that is the mode of the classes from individual trees for classification problems (missing values in categorical variables), or a mean prediction of the individual trees for regression problems (missing values in continuous variables) (Breiman, 2001).

The algorithm works as follows: first, the dataset is split into a training set and testing set. For all model building purposes, only the training set is used. Then, a decision tree is fitted to the data. A decision tree is built by recursively splitting the data based on (random) feature variables, where each time the best split is chosen among these variables (see Figure 4.1 for a simplified example). For each Random Forest, a large amount of trees are made.

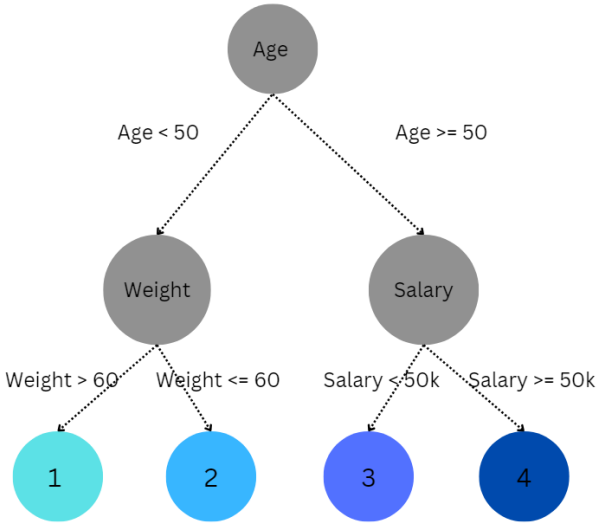

Figure 4.1: Simplified Decision Tree example.

All trees are then combined in order to make a final prediction. For categorical values that are missing, the formula is as follows:

$$\hat{Y}(X) = \underset{j \in \{1,2,\ldots,k\}}{\arg\max} \sum_{b=1}^{B} I(f_b(X) = C_j) \tag{4.7}$$

Here, $\hat{Y}(X)$ is the final predicted value, k is the number of categories for this categorical variable, B is the total amount of trees, I is an indicator function which is 1 if the condition is true and 0 otherwise, and $f_b(X)$ corresponds to the prediction of tree b. In other words, the function counts which class was predicted most often by all trees, and selects that class as the final prediction. For continuous variables, the equation is as follows:

$$\hat{Y}(X) = \frac{1}{B} \sum_{b=1}^{B} f_b(X) \tag{4.8}$$

Again, $\hat{Y}(X)$ denotes the final prediction, B is the number of trees and $f_b(X)$ denotes the predicted value of tree b. In other words, this function calculates the mean value of all predictions and selects that as the final predicted value.

In the application of missing value imputation, the strategy is relatively straightforward: a Random Forest model is fitted to the observed data while treating the missing data as the target variable, on a variable-by-variable basis for each column that contains missing values. Following this, the model predicts each missing value and fills it in accordingly. Notably, this process does not require the specification of a model beforehand and makes no assumptions concerning the distribution of the complete data. Moreover, it effectively captures non-linear relationships and can handle mixed-type data, presenting a considerable advantage in data analysis (Shah, Bartlett, Carpenter, Nicholas & Hemingway, 2014; Stekhoven & Bühlmann, 2012).

4.2.4 Predictive Mean Matching Imputation

Predictive Mean Matching (PMM) imputation, as proposed by R. J. A. Little (1988), is a semi-parametric method that tries to retain the distributional properties of the data (R. J. Little, 1988; Robins & Wang, 2000). Where parametric methods make distributional assumptions such as multivariate normality, PMM makes use of the observed data to impute the missing values. In order to efficiently explain how this method works, let us introduce some new notations. Let $\mathbf{v}$ be the variable that contains missing values, and let X be the other variables. The method fits a multivariate linear regression of $\mathbf{v}$ on X, based solely on the observed data. Using this regression, it predicts a value for $\mathbf{v}$ for each observation, which we will call $\hat{\mu}_i$. It then checks the data for which variable $\mathbf{v}$ is observed for potential candidate donors, and selects 5 (this varies across implementations and packages, R package **mice** uses 5) candidates that are closest in predicted mean $\hat{\mu}_i$ based on absolute differences. From these 5 so-called 'donors' that are closest to the missing observation in terms of predicted mean $\hat{\mu}$, one is randomly chosen. It is not the value of $\hat{\mu}_j$, where j is the selected donor, that the missing value is imputed with, but rather the true value of that donor in that variable, namely $\mathbf{v}_j$. This way, it maintains the variability and distributional characteristics of the dataset (Morris, White & Royston, 2014). As explained in Section 4.2, the MICE algorithm then iteratively imputes all variables.

Aside from not assuming a certain distribution, another one of the advantages of PMM is that it works particularly well when the data is not normally distributed or when there is a non-linear relationship between variables. Literature has shown that it also works well for categorical variables, and has computational advantages when the number of categories is large (Van Buuren & Groothuis-Oudshoorn, 2011). Additionally, it can handle both continuous and categorical variables. A final advantage is that because of how the algorithm works, imputed values are always restricted to the observed values (R. J. Little, 1988). The downside of PMM is that it can be more computationally intensive than other methods, but not as intensive as methods like Random Forest imputation or k-NN imputation (R. J. Little, 1988).

4.3 Evaluation

As mentioned in Section 1, this research will make use of several evaluation metrics. These can be split up as direct evaluation, indirect evaluation and feasibility.

4.3.1 Direct Evaluation

Direct evaluation refers to the direct imputation accuracy of each method. This will be quantified by using the Root Mean Squared Error (RMSE) of the imputed values compared to the original true values. The RMSE is calculated as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2} \tag{4.9}$$

Here, n denotes the total number of missing values. x_i and $\hat{x}_i$ denote the true value and imputed

value, respectively. Lower values for the RMSE correspond with a better imputation accuracy. The RMSE is a metric that can be used correctly for continuous variables, but not for categorical or binary values. Therefore, for the evaluation of discrete variables the Percentage of Correct Predictions (PCP) is used (Nishanth & Ravi, 2016; Valdiviezo & Van Aelst, 2015).

$$PCP = 100 * \frac{\#\ of\ correctly\ imputed\ values}{\#\ of\ total\ predictions} \tag{4.10}$$

Higher PCP values correspond with a better imputation accuracy for the discrete variables.

A problem that arises when comparing some methods, is that not all methods are suited for multiple imputation. This is because MI needs a random component to the imputation procedure in order to obtain different results for each imputation. This can be in the form of a random noise that is added or in other ways that introduce randomness to an algorithm, like the way in which Random Forests are built. Methods like k-NN imputation do not have a random aspect to them, but are deterministic. This means that each time that you run the algorithm, you will get exactly the same imputed values. To circumvent this problem and be able to compare all methods to each other directly, this research uses multiple amputation. Multiple amputation refers to the process of introducing missing values to the same dataset multiple times, so that in each amputed dataset the values that are missing differ from each other. One then imputes these datasets separately, after which statistics like the mean and standard deviation of the RMSE and PCP can be calculated in order to compare all methods. For the methods that can be used for multiple imputation, each amputed dataset gets multiple (5) imputations as explained in Section 4.2, after which the mean values of all imputations are taken as the final imputed value. This way, the power of multiple imputation is upheld, while subsequently being able to compare all methods to each other.

4.3.2 Indirect Evaluation

Indirect evaluation refers to the process of evaluating the methods by their post-imputation data mining task performance. This research will focus on classification performance. Each imputed dataset will be run through a simple XGBoost algorithm, after which the classification accuracy will be calculated.

The XGBoost algorithm, like the Random Forest, is a nonparametric decison tree-based ensemble method. It was first introduced by Chen and Guestrin (2016). The efficiency, effectiveness and scalability is widely lauded since its inception, as it was dominating machine learning challenges (Chen & Guestrin, 2016). XGBoost is an acronym for eXtreme Gradient Boost. The boosting refers to the practice where weak learners, usually decision trees, are sequentially built. In this process, each new tree learns and corrects the errors that are made by previous trees. This works via an additive model, where the models are added together to correct residuals from previous models. This process can be defined in the following way:

$$F_t(x) = F_{t-1}(x) + \eta * h_t(x) \tag{4.11}$$

Here, $F_t(x)$ denotes the model at iteration t, $h_t(x)$ is the new 'weak learner' that is added at iteration t, and η is the learning rate which has a default value of 0.3. The algorithm optimizes an objective function based on the predictions of the previous trees, the contribution of the current tree and regularization terms:

$$L(\phi) = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \tag{4.12}$$

Here, ϕ is a set of parameters for the model, y_i is the true outcome value of observation i, $\hat{y}_i^{(t-1)}$ denotes the model's prediction based on the previous $t-1$ trees, and $f_t(x_i)$ is the prediction of the new tree for observation i. The regularization term $\Omega(f_t)$ is there to prevent overfitting. This works by adding a penalizing term to the objective function when the model gets too complex. $l(\cdot)$ is the loss function, which changes depending on what type of outcome variable your dataset has. For binary classification tasks, the loss function is typically a binary logistic loss function, also known as log-loss. The log-loss function for a single observation is as follows:

$$l(y, p) = -(y \log(p) + (1 - y) \log(1 - p)) \tag{4.13}$$

Here, y is the true class label, and p is the predicted probability of having an outcome value of 1. This loss function penalizes predictions that are confidently wrong. For multinomial classification tasks, the loss function takes on a slightly different form. This loss function is called the softmax loss function. This function is also for a single observation.

$$l(\mathbf{y}, \mathbf{p}) = -\sum_{c=1}^{M} y_c \log(p_c) \tag{4.14}$$

Here, $\mathbf{y}$ is a one-hot encoded vector with length M that denotes the true class label of the observation, with $y_c = 1$ for the class that it is in and 0 otherwise. M is therefore the total number of classes. The vector $\mathbf{p}$ is also a vector of length M, and contains the predicted probabilities of being in each class. y_c and p_c denote the true label and predicted probability of being in class c. Note that in the objective function (Equation 4.12), we have y_i, $\hat{y}_i^{(t-1)}$ and $f_t(x_i)$ as 'parameters' in the equation, which is different to the parameters y, $\mathbf{y}$, p and $\mathbf{p}$ in Equations 4.13 and 4.14. This is because the objective function aims to explain how each new tree ($f_t(x_i)$) is added to the model's existing predictions ($\hat{y}_i^{(t-1)}$). The loss functions however, aim to explain a single observation's prediction, namely p or p_c. These probabilities are essentially the same as $\hat{y}_i^{(t-1)} + f_t(x_i)$), but just captured into one term.

Because of the use of multiple amputation, mean and standard deviation values can be calculated for the classification accuracy as well. The XGBoost algorithm is used because it has the addition of being able to handle missing values. Because it can handle missing values, we can also perform the classification task on the incomplete dataset, and get a baseline classification accuracy. If a company is only interested in a classification task, it might not be worth imputing the values if the classification accuracy of the XGBoost model is comparable or even higher than the post-imputation classification accuracy. For each classification, 5-fold cross-validation was

used in order to obtain more accurate results for classification accuracy.

4.3.3 Computation Time And Feasibility

The evaluation on the basis of feasibility relies on two things. One is the computation time, or the time it takes each algorithm to finish the imputation procedure. This is internally measured in the amount of seconds from the start of the procedure until the end. For all imputation tasks, a HP EliteBook x360 1040 G6 was used, with an Intel Core i7-8665U processor. The power mode of the laptop was set to maximum at all times. As we are using multiple amputation, the total time it takes for all 20 iterations will be averaged. It should be noted that the methods that use multiple imputation will have to impute the data multiple times, which will clearly impact the computation time. However, this will not be corrected as in practice, one will also want to use multiple amputation for these methods because that is where their strength lies. Additionally, due to the nature of how this research is set up, the computing time of the different methods will also include saving each imputed dataset and calculating RMSE values. This added time is negligable when compared to the total computation time of the algorithms. Additionally, these extra computations are the same for each method, which means it will not affect the overall conclusions. The second part consists not of a metric, but more of a nuanced perspective on feasibility. The amount of hyperparameters that need tuning can become large for some methods, which is not always desirable in a business setting. Therefore, this will also be taken into consideration in the final evaluation of this thesis.

4.4 Missingness Mechanisms

In the realm of missing data, three main so-called missingness mechanisms can be identified. Those three mechanisms are: Missing Completely At Random (MCAR), Missing Not At Random (MNAR) and Missing At Random (MAR) (Rubin, 1976). The differences between these mechanisms are quite subtle. To help clarify these nuances, this section will offer fictive examples to illustrate each one more clearly, as was done by Jäger et al. (2021).

In this study, each of the three mechanisms will be simulated in order to see the different effects they might have on the efficiency and results of the imputation techniques. This is done using the *ampute* function in R package **mice** (Van Buuren & Groothuis-Oudshoorn, 2011). The percentage of missing values was set to 50%. This means that in total, 50% of observations will have a missing value in one of its columns. Furthermore, it should be noted that for each dataset, two variables were amputed. This means that for each variable that has been amputed, approximately 25% of the observations will be missing. There are no observations with a missing value in both variables. These variables that are amputed were selected based on feature importances from a base XGBoost model on the full dataset. If the dataset was continuous or categorical only, like the Iris and Car datasets, the two most 'important' variables were selected for amputation. If the dataset has mixed variables, the most 'important' continuous and categorical variables were selected. See Appendix A.1 for all feature importance graphs.

In order to provide adequate mathematical formulas for the different missingness probabilities,

we need to establish some notation. Let the variables (excluding the Y variable) of our dataset be represented as the $n \times p$ matrix X, with n observations and p variables. So x_{ij} is therefore the i-th observation of the j-th column. Let x_{ik}, where $k = 1, ..., p; k \neq j$ denote the vector of values for observation i, except for column j. So

$$x_{ik} = [x_{i1}, x_{i2}, ..., x_{i,j-1}, x_{i,j+1}, ..., x_{i,p-1}, x_{ip}]$$

Furthermore, let R_{ij} be a binary indicator variable, indicating 1 when the value of x_{ij} is missing, and 0 otherwise. Lastly, let u_i denote all unobserved variables that are not present in the data. It should be noted that in this thesis it is assumed that all missingness probabilities of all observations are independent from each other. In a survey setting, this would mean that the answers that one person fills in (or does not fill in), do not affect the probability that someone else does or does not fill in their answer.

4.4.1 MCAR

MCAR, or Missing Completely At Random, is the most commonly investigated missingness mechanism in the current literature on data imputation techniques. MCAR means that values are missing independent of any other variables, see Equation 4.15 and Table 4.1. The missing values are perfectly randomised.

$$\text{MCAR: } P(R_{ij} = 1) = Q \tag{4.15}$$

Here, R_{ij} denotes whether the value of observation i and variable j is missing or not, with $R_{ij} = 1$ denoting a missing value and $R = 0$ otherwise. Q denotes the percentage of missing values, which is a constant under MCAR.

Height	Gender	$\text{Height}_{\text{MCAR}}$
179.0	F	?
192.0	M	?
189.0	M	189.0
156.0	F	156.0
175.0	M	?
170.0	F	170.0
181.0	M	181.0
197.0	M	?
156.0	F	?
160.0	F	160.0

Table 4.1: Applying the MCAR condition to column height with 50% missing values removes five out of ten height values, independent of height or gender.

4.4.2 MAR

The MAR mechanism, or Missing At Random, denotes the missingness mechanism where the missing values are correlated with another variable's values, but independent of its own value. Equation 4.16 and Table 4.2 illustrate this.

$$\text{MAR: } P(R_{ij} = 1|x_{ij}, x_{ik}) = P(R_{ij} = 1|x_{ik}) \tag{4.16}$$

Again, $R_{ij} = 1$ denotes a missing value being present in the j-th column of the i-th observation. x_{ij} denotes the i-th observation of column j, and x_{ik} denotes all columns of observation i except for the j-th column. We can see that the probability of having a missing value does not depend on the true value itself, but solely on other observed variables (excluding the outcome variable).

Height	Gender	$\text{Height}_{\text{MAR}}$
179.0	F	179.0
192.0	M	?
189.0	M	?
156.0	F	156.0
175.0	M	?
170.0	F	170.0
181.0	M	?
197.0	M	?
156.0	F	156.0
160.0	F	160.0

Table 4.2: Applying the MAR condition to column height with 50% missing values removes five out of ten height values, independent of its own value but dependent on the gender value. Observations with gender 'Male' correspond to missing values.

4.4.3 MNAR

The MNAR mechanism, or Missing Not At Random, has some ambiguous definitions. One says that MNAR is simply missing data that is neither MCAR nor MAR (Polit & Beck, 2008). Other definitions say that values are at least missing dependent on its own (missing) value (Equation 4.17), but may also depend on observed information (Equation 4.18) (Santos et al., 2019). In the first ever paper on missingness mechanisms by Rubin (1976), the case where the missing probability depends on the value itself is called *censored data*. For an example, see Table 4.3. Unfortunately, this mechanism is most often found in real-life scenarios (Laaksonen, 2018). Because the missing values may also depend on observed data, MAR and MNAR can become indistinguishable from each other based on your data, as can be seen when comparing Equations 4.16 and 4.18. Because in real life we do not know the value of x_{ij} if it is missing, MAR and MNAR can not be distinguished. MNAR is also called non-ignorable missingness, because performing analysis on data that has MNAR missingness can significantly bias results of parameter estimations (Fielding, Fayers & Ramsay, 2009).

$$\text{MNAR (Type 1): } P(R_{ij} = 1|x_{ij}, x_{ik}) = P(R_{ij} = 1|x_{ij}) \tag{4.17}$$

$$\text{MNAR (Type 2): } P(R_{ij} = 1|x_{ij}, x_{ik}) = P(R_{ij} = 1|x_{ij}, x_{ik}, u_i) \tag{4.18}$$

Again, $R_{ij} = 1$ denotes a missing value being present, and x_{ik} and x_{ij} denote the other observed variables and the variable with a potential missing value, respectively. Furthermore, we see that in Equation 4.18, the probability of having a missing value can also depend on factors that are not included in the dataset at all, denoted by u_i.

Height	Gender	$\text{Height}_{\text{MNAR}}$
179.0	F	179.0
192.0	M	192.0
189.0	M	189.0
156.0	F	?
175.0	M	?
170.0	F	?
181.0	M	181.0
197.0	M	197.0
156.0	F	?
160.0	F	?

Table 4.3: Applying the MNAR condition to column height with 50% missing values removes five out of ten height values, independent of gender but dependent on the height value. Lower values correspond to missing values.

In this thesis, the Type 1 MNAR missingness is introduced using the *ampute* function from R package **mice**, as explained before. The Type 2 MNAR missingness is not available in any package, so we implement a novel approach to introducing missing values corresponding to MNAR Type 2 missingness. This is an extension of this paper and will be a separate part of the research because how it is implemented differs substantially from the other missingness mechanisms, making it unsuitable to compare directly. The full method will be explained in the following section, Section 4.5.

4.5 Extension: Type 2 MNAR

In order to introduce missing values dependent on factors outside of the dataset, we will base the probability of having a missing value on a certain correlated variable that is also in the dataset, after which we will delete this correlated variable before starting the imputation procedure. This way, there is a definitive process behind the missingness probability, whilst simultaneously simulating the MNAR 2 mechanism where the missingness depends on factors that are outside the dataset.

4.5.1 Variable Association Metrics

First, we need to determine which variables are suitable to base the missingness probability on, i.e. which variables are correlated with the variables where we want to introduce missing values. For continuous variables, the correlation coefficient is straightforward as we can use the standard Pearson correlation for these variables. However, as we have mixed data as well, we can not always use the Pearson correlation (which is only suitable for correlation between two continuous variables). Table 4.4 shows an overview of suitable correlation measures for each combination of variable types that we use.

Variable Type	**Continuous**	**Binary**	**Nominal**	**Ordinal**
Continuous	Pearson	Point-Biserial	ANOVA	n/a
Binary	Point-Biserial	n/a	Cramer's V	Cramer's V
Nominal	ANOVA (η^2)	Cramer's V	Cramer's V	n/a
Ordinal	n/a	Cramer's V	n/a	Spearman's ρ

Table 4.4: Appropriate correlation methods for different types of variable combinations that occur in our datasets. Combinations with value n/a do not occur in this thesis.

Pearson Correlation

As stated in Table 4.4, we use Pearson correlation for two continuous variables (Pearson, 1895). The Pearson correlation coefficient, or Pearson's r, is calculated between two continuous variables (x and y) by the following formula:

$$r = \frac{\sum(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum(x_i - \overline{x})^2 \sum(y_i - \overline{y})^2}} \tag{4.19}$$

Here, x_i and y_i are individual observations for variable x and y, and $\overline{x}$ and $\overline{y}$ are their respective means. This coefficient takes on a value between -1 and +1, where -1 indicates a perfect negative linear correlation between the variables, and +1 indicates a perfect positive linear correlation.

Point-Biserial Correlation

Pearson's r can be applied to the case with one continuous and one binary variable as well. This is called the Point-Biserial correlation coefficient. In this case, the formula simplifies to:

$$r_{pb} = \frac{\mu_1 - \mu_0}{s} \cdot \sqrt{\frac{n_1 n_0}{n^2}} \tag{4.20}$$

Here, μ_1 and μ_0 are the sample means of the continuous variable for each category of the binary variable. The standard deviation of the continuous variable is denoted by s. Furthermore, n_1 and n_0 denote the number of observations where the binary variable is 1 and 0, respectively, and n is the total number of observations. The point-biserial correlation is essentially a measure of how the means of the continuous variable differ across the categories, adjusted for how big each category is.

ANOVA and η^2

When we want to test the correlation between a nominal categorical variable and a continuous variable, we use the ANOVA (Analysis of Variance), which essentially evaluates whether the mean of the continuous variable differs between groups of the nominal variable. This is evaluated by calculating the total sum of squares (SST) and the between-group sum of squares (SSB), whose formulas are described below.

$$SST = \sum_{i=1}^{N} (x_i - \overline{x})^2 \tag{4.21}$$

$$SSB = \sum_{j=1}^{k} n_j (\overline{x}_j - \overline{x})^2 \tag{4.22}$$

In these equations, x_i denotes a single observation, and $\overline{x}$ is the mean of x over all N observations. The number of categories of the nominal variable is denoted by k, n_j is the number of observations in group j and $\overline{x}_j$ is the mean of each group j. The between-group sum of squares is divided by the total sum of squares to obtain a measure of association, η^2. The value of η^2 gives the proportion of the variance of the continuous variable that can be attributed to the variance of the nominal variable.

$$\eta^2 = \frac{SSB}{SST} \tag{4.23}$$

Cramer's V

In order to get a measure of association for two nominal variables (where at least one has more than 2 categories), we use Cramer's V (Cramér, 1999). Cramer's V gives a measure of association between two nominal variables but does not give a direction (positive or negative association). For this research, the direction of the association is not important, so Cramer's V is suitable. The calculation starts by creating a contingency table of the two nominal variables and calculating the expected frequency of observations for each combination of the variables (E_{ij}). This is then compared to the observed frequency for each combination (O_{ij}) to obtain a χ^2 statistic (Equation 4.24).

$$\chi^2 = \sum_{i,j} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \tag{4.24}$$

This statistic is then used to calculate Cramer's V in the following way:

$$V = \sqrt{\frac{\chi^2}{n \times \min(k-1, r-1)}} \tag{4.25}$$

Here, n is the total number of observations, and k and r are the number of categories in the two nominal variables, i.e. the number of rows and columns of the aforementioned contingency table.

Spearman's ρ

Lastly, when we want to obtain a measure of association between two ordinal variables, we use Spearman's ρ (Spearman, 1961). Spearman's ρ, also called Spearman's rank correlation coefficient, is a non-parametric measure of dependence between two variables. Where Pearson's correlation metric assumes a linear relationship between two variables, Spearman's ρ does not assume this. However, it does assume a monotonic relationship between the variables, i.e. relationships that consistently increase or decrease. As the full name suggests, it uses ranks instead of true values to obtain the correlation coefficient. All values in each of the variables are assigned ranks, based on their value respective to the other values of that variable. Then, for each observation, the difference in ranks (d_i) is calculated by subtracting the rank of the first variable from the rank of the second variable for that observation. These values of d_i are then used as stated in Equation 4.26.

$$\rho_{spearman} = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \tag{4.26}$$

Here, d_i are the differences in ranks as previously explained, and n is the number of observations. Spearman's ρ takes on a value between -1 and 1, with -1 indicating a perfect negative monotonic relationship, 1 indicating a perfect positive monotonic relationship, and 0 indicating no monotonic relationship.

4.5.2 MNAR 2 Algorithm

After determining a correlated variable for each of our variables that we later want to impute, we introduce missing values. The way in which the missing values are introduced depends on which variable type the correlated variable is. This subsection will explain how the algorithm works. For the full algorithm, see Appendix B.

As stated before, a distinction needs to be made for the different types of correlated variables. For a continuous correlated variable, the process is as follows: the algorithm first determines a threshold value based on the desired missing percentage. We will denote the desired percentage as Q. The missing percentage, Q, is multiplied by 1.5. Then, based on the values of the correlated variable, the value of the $(1 - (1.5Q))$-th percentile is calculated and set as the threshold value. All values that are above this threshold will be considered candidate observations. From these candidate observations, $Q * n$ observations are sampled without replacement to obtain the observations where we will introduce missing values. The reason behind multiplying Q by 1.5 and subsequently sampling from that pool of candidates is to introduce randomness. We need this random component in order to be able to ampute the dataset multiple times without selecting the same observations to be missing each time.

When we are dealing with a binary correlated variable, the process is different. For binary correlated variables, we only need to make a distinction between values 0 or 1. If the value is 0, we assign a low probability of 0.2 times the desired missing percentage, and if the value is 1 we assign a high probability of 1.9 times the desired missing percentage. For example, if

the missing percentage is set to 50%, observations with a 0 in the correlated variable will be assigned a probability of 0.1 and observations with a 1 will be assigned a probability of 0.95. For each observation, a random value between 0 and 1 is generated and compared to the respective probability of that observation. If the probability is higher than the randomly generated value, it will be set as missing.

When dealing with a categorical correlated variable, the process is similar to the binary case. The only difference is the way in which the probability is calculated. If the categorical variable x has k levels, each observation will get a probability corresponding to $\frac{x_i}{k} * Q$. For example, with a missing percentage of 50% and a categorical correlated variable with 4 levels, the probabilities for levels 1, 2, 3 and 4 will be 12.5%, 25%, 37.5% and 50%, respectively. These probabilities are again compared to a randomly generated value between 0 and 1, after which the observation will be set as missing if the generated value is smaller than the probability.

After the missing indices are generated, a check is performed. If the number of missing indices is greater than the number of missing values that we want, new indices are sampled from the list of missing indices to obtain the desired amount. On the other hand, if the number of missing indices is smaller than the desired number of missing values, additional random missing values are assigned to the remaining observations to meet the specified missing percentage. Finally, the correlated value should be removed from the dataset. For compatibility with the subsequent imputations and other procedures, the correlated value is not removed but instead transformed to a series of random values between -0.0001 and 0.0001. This way, the column structure of the data is retained whilst making sure that the correlated variable itself is uninformative to the subsequent imputation process, adhering to the Type 2 MNAR mechanism.

5. Main Results

This section is organized to give a review of the main results obtained in this thesis. Section 5.1 will provide the direct evaluation of all methods, where the RMSE and PCP values can be compared across all methods and datasets. After that, Section 5.2 will compare the post-imputation classification performance of all methods, providing insight into how each method prepares data for classification tasks. Finally, Section 5.3 will discuss the computational aspect and feasibility of each method.

5.1 Direct Evaluation

This section will show and explain the results of our imputation procedures based on the RMSE and PCP metrics. Table 5.1 shows all RMSE values. These values are the mean RMSE that each method achieved over 20 imputation runs. The full distribution by way of IQR graphs of the RMSE and PCP values of all 20 imputations can be found in Appendix A.2 and A.3.

The first thing we notice is that, as expected, Mean/Mode and Median/Mode imputation perform the worst across all datasets. Furthermore, Random Forest imputations show exceptional performance in all datasets, but specifically in the Adult dataset. It should be noted that the RMSE values in the Adult dataset are relatively large in comparison to the others, due to the fact that one of the imputed variables in the Adult dataset was *capital_gain*, which has a lot of zero values but also a lot of very high values. This causes the RMSE to be larger by default. In all other datasets, RF either has the lowest RMSE value or at least is very competitive with the other imputation methods. This can be explained by the fact that RF is very good at encapsulating complex and/or nonlinear relationships in the data. We also see that the Linear Bayesian Regression is often underperforming compared to the other methods. This is not unexpected, as the Linear Bayes algorithm assumes a linear relationship between variables, which is a strong assumption as this is not always the case. This reasoning is further strengthened by the fact that the LB algorithm does perform well for the Iris dataset, as well as the simulated and skewed datasets. This is because the Iris dataset has strong linear relationships between the predictor variables, as can be seen in Appendix A.6, Figure A.29. Both the simulated data and its skewed version also have a strong linear relationship between the variables that have been imputed, see Figure A.30 and A.31 in Appendix A.6. Furthermore, we can see that the k-NN algorithm performs very well, especially in the Banknotes and Skewed datasets. Due to the nonparametric nature of the k-NN algorithm, it obtains very good results for datasets that have non-normal (or not close to normal) distributions. Conversely, we see that the EM algorithm performs well for the Iris dataset, as well as the simulated dataset. This makes sense as the EM algorithm assumes

multivariate normality, which is true for the simulated dataset, and roughly the case for the Iris dataset. Lastly, the predictive mean matching imputation method also performs well for the Iris, simulated and skewed datasets. It is surprising that it slightly outperformed methods such as EM and Linear Bayes, as these assume multivariate normality (LB assumes this through the MICE algorithm), and PMM is a nonparametric method. Therefore, it was expected that PMM would outperform these methods in those datasets that did not have multivariate normality.

Table 5.1: Mean RMSE results over 20 imputations. Bold values denote lowest (best) value for that dataset.

Dataset	**Method** / **Mechanism**	Mean	Median	k-NN	EM	LinBayes	pmm	RF
Adult	**MCAR**	7546	7626	8049	5633	5649	5574	**5438**
	MAR	8768	8872	9365	6388	6387	6492	**6287**
	MNAR	10658	10786	10822	7496	7472	7381	**7336**
Banknotes	**MCAR**	4.62	4.63	**0.98**	1.69	1.69	1.54	1.09
	MAR	4.88	5.01	**1.04**	1.66	1.67	1.53	1.16
	MNAR	4.67	4.60	**1.02**	1.76	1.76	1.57	1.13
Iris	**MCAR**	1.38	1.41	0.28	0.28	0.28	**0.26**	0.27
	MAR	1.41	1.27	0.34	0.29	0.30	0.29	**0.28**
	MNAR	1.42	1.23	0.33	0.29	0.29	**0.28**	0.29
Simulated	**MCAR**	1.38	1.38	1.01	0.96	0.96	**0.94**	1.01
	MAR	1.32	1.32	1.03	**0.95**	0.97	0.98	1.03
	MNAR	1.41	1.41	1.09	**1.01**	**1.01**	1.03	1.10
Skewed	**MCAR**	1.22	1.24	0.84	0.84	0.85	**0.83**	0.84
	MAR	1.09	1.07	**0.78**	0.80	0.80	0.79	0.79
	MNAR	1.06	0.97	0.76	0.78	0.78	**0.75**	0.78

When we look at the imputation performance for the categorical variables in Table 5.2, we see some interesting results. When we look at the 'baseline' performance of the Mean/Mode and Median/Mode methods (which obviously obtain the same PCP for categorical variables), we see that its performance is heavily reliant on which dataset is used. This makes sense, as datasets with categorical variables where one value dominates other values will yield good imputation results when using the mode. This performance still depends on which missingness mechanism is present. This can be easily explained by an example: take a categorical variable with three levels: 1, 2 and 3. Let us say that 85% of the observations contain a 1, 10% contains a 2 and 5% contains a 3. Under MCAR, approximately 85% of the missing values would be a 1. As this is the mode, each imputation would have approximately 85% chance of being correct, resulting in a very high PCP. Under MNAR however, it can occur that high values correspond with a higher probability of being missing. Therefore, the PCP would decrease drastically if all these values are imputed with the mode (1).

We see that in the Adult dataset, imputing the missing values with the mode obtains very competitive results compared to the other methods, outperforming all of them except for k-NN

and RF. This is because for the variable where missing values were introduced, 46% had a value of 2 (see Figure A.32 in Appendix A.6). k-NN and RF are both nonparametric methods, which clearly excelled at identifying the underlying structures in the Adult dataset.

Table 5.2: Mean PCP values over all 20 imputations. Values are in percentages.

Dataset	**Method** / **Mechanism**	Mode	k-NN	EM	LB	PMM	RF
Adult	**MCAR**	45.98	**81.79**	41.11	41.13	44.81	72.75
	MAR	48.56	**80.70**	40.18	39.87	43.99	71.52
	MNAR	45.98	**75.85**	40.09	39.72	42.51	67.06
Car	**MCAR**	31.25	37.99	40.65	42.05	**45.17**	41.16
	MAR	29.75	42.29	41.95	42.78	**46.07**	41.40
	MNAR	15.17	32.37	38.56	40.36	**40.92**	38.48
Simulated	**MCAR**	30.20	36.18	35.49	**36.33**	35.21	35.06
	MAR	27.83	34.96	33.81	**35.26**	33.36	34.25
	MNAR	21.47	32.74	32.73	**32.90**	32.04	32.13
Skewed	**MCAR**	30.25	**37.27**	35.94	35.64	36.56	36.51
	MAR	27.42	34.90	34.97	**35.41**	34.44	34.95
	MNAR	20.28	**33.50**	32.52	32.68	32.82	33.09

For the Car dataset we see that PMM obtained the highest PCP values, although EM, Linear Bayes and RF are not much worse. The variables that were imputed in the Car dataset were very balanced, with each value accounting for approximately one-third of all observations see Figures A.33 and A.34 in Appendix A.6). We also clearly see that Mode imputation is by far the worst when facing the MNAR missingness mechanism. This is due to the reasons stated before. For both the simulated and skewed datasets, we see that all methods are very competitive, except for the baseline Mode imputation. The differences in performance of all methods between the simulated and skewed datasets are very small. Linear Bayes performed slightly better than the rest, but not significantly. This is partly due to the fact that the variables that are imputed are identical in both datasets, as the categorical variables were unchanged. The slight difference in categorical imputation performance is likely a result of random noise, as the observations that were made missing were randomly selected each time.

5.2 Indirect Evaluation

This section will show and explain the post-imputation classification accuracy results. The XGBoost classifier was not tuned for any dataset or method to keep the comparisons consistent and straightforward. First, the XGBoost classifier was run separately on each complete dataset, in order to get a first benchmark for classification accuracy. Second, the reason for using the XGBoost classifier was that it can handle missing data by itself. Therefore, all datasets with missing values were also directly used to obtain a classification performance when no values were imputed. Table 5.3 shows all classification accuracies in one overview.

Table 5.3: Average XGBoost accuracy for all datasets and methods over 20 imputations.

		Dataset					
Method	**Mechanism**	**Adult**	**Banknotes**	**Car**	**Iris**	**Simulated**	**Skewed**
Full Data		**86.96**	**99.20**	**99.02**	**94.67**	**94.30**	**94.30**
No Imp.	**MCAR**	86.58	96.65	84.26	94.33	93.49	93.76
	MAR	86.68	96.70	83.14	83.14	93.67	94.07
	MNAR	86.38	96.88	87.81	87.81	93.73	93.61
Listw. Del.	**MCAR**	86.53	98.85	97.25	94.45	94.45	94.79
	MAR	87.71	98.97	96.61	95.77	94.57	94.20
	MNAR	86.91	98.65	97.15	95.57	94.43	94.73
MeanM	**MCAR**	86.55	96.25	78.21	95.13	93.04	93.19
	MAR	86.60	96.17	70.28	70.28	93.36	93.19
	MNAR	86.42	96.59	70.11	70.11	93.38	93.21
MedianM	**MCAR**	86.56	96.25	78.21	95.13	92.94	93.26
	MAR	86.49	96.10	70.28	70.28	93.36	93.17
	MNAR	86.38	96.59	70.11	70.11	93.39	93.21
k-NN	**MCAR**	86.80	99.38	97.45	95.90	95.63	95.62
	MAR	86.77	99.35	97.51	97.51	95.59	95.54
	MNAR	86.59	99.35	97.81	97.81	95.53	95.39
LB	**MCAR**	87.52	99.16	88.53	95.53	95.19	95.49
	MAR	87.66	99.17	89.14	89.14	95.38	95.42
	MNAR	87.41	99.13	92.42	92.42	95.26	95.18
RF	**MCAR**	87.46	99.28	95.65	95.70	94.90	95.40
	MAR	87.53	99.34	95.37	95.37	95.11	95.24
	MNAR	87.21	99.23	96.92	96.92	95.15	95.19
PMM	**MCAR**	88.14	99.13	95.48	95.53	95.39	95.68
	MAR	88.30	99.27	95.24	95.24	95.43	95.43
	MNAR	87.89	99.19	96.79	96.79	95.39	95.35
EM	**MCAR**	87.59	99.21	89.04	95.53	95.30	95.54
	MAR	87.67	99.20	89.28	89.28	95.38	95.40
	MNAR	87.44	99.15	92.29	92.29	95.24	95.15

The first thing that becomes obvious when looking at the classification results is that the difference between the missingness mechanisms almost fully disappears. For the Iris dataset, we see that there is still a relatively large difference in the performance of the benchmark methods (i.e. No imputation, Mean/Mode and Median/Mode). This means that not imputing the values in this dataset, or imputing them with the mean or median (as the Iris dataset does not contain any categorical variables) can be a risk to your classification performance. Using the Linear

Bayes or EM methods can also be detrimental to classification performance, but to a lesser extent than the aforementioned methods. For all these methods, the classification performance was still reliable under the MCAR mechanism.

When we look at the results for the Car dataset, we see some interesting results. For all non-benchmark methods, we see that the post-imputation classification performance was actually slightly higher for the MNAR mechanism compared to the other mechanisms. This can be related to the characteristics of the dataset itself. It is possible that under MNAR, certain values are more often missing. If those values are then imputed by values that have even stronger correlation with the outcome variable, it can lead to improved classification performance. We also see that none of the imputation methods were able to attain the same classification accuracy post-imputation as the complete dataset. This is interesting, as the imputations actually improved classification accuracy across all other datasets compared to their complete counterparts. This could be due to the fact that the Car dataset consists solely of categorical variables, and not all methods are designed to be able to impute those. We see that k-NN imputation, which is suitable for categorical imputation, performs the best for this dataset.

When we look at the results for the Simulated and Skewed datasets, we see that they are very similar across all methods. This is likely due to the fact that the transformation did not change any other relationships within the datasets. For both datasets, using (non-benchmark) imputation methods actually improved classification performance. This is likely because the imputation algorithms find the relationships between variables, and preserve them very well (Finney & DiStefano, 2006). When outliers go missing, and are subsequently imputed, these imputed values can sometimes be more useful for post-imputation classification tasks. This improved classification performance was also noticeable in the Adult and Banknotes datasets, as well as in the Iris dataset when using k-NN or RF imputations. Furthermore, we clearly see that it is often not desirable to impute values with the mean/median and mode when preparing data for a classification task: for all datasets, not imputing the values attained at least the same classification accuracy compared to imputing them with the mean/median and mode.

5.3 Computation Time And Feasibility

This section will discuss the computational requirements for each method. As explained in Section 4, the computational time that was measured was the time in seconds that it took each algorithm to impute one dataset. For MI methods, the entire process of imputing the dataset 5 times and then taking the mean counts as one imputed dataset. Figure 5.1 shows the computation time across different dataset sizes. As expected, the most simple algorithms (Mean/Mode and Median/Mode) were also the fastest, across all dataset sizes. We see that after that, the EM algorithm was fastest overall.

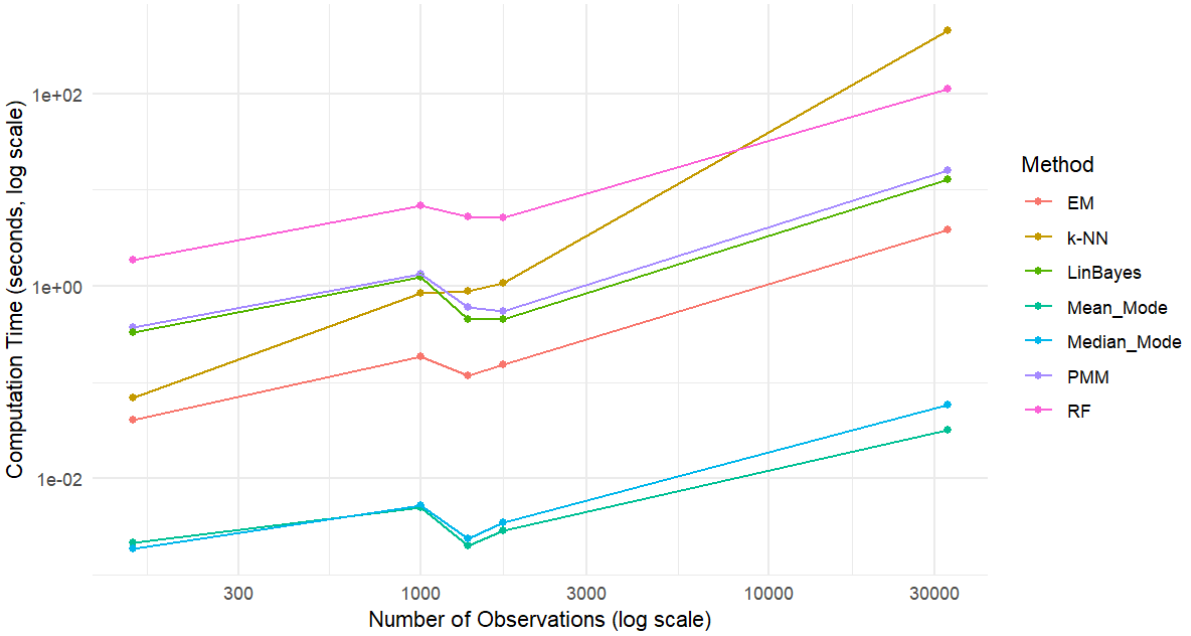

Figure 5.1: Computation time for each method to complete one dataset. Both axes are on a logarithmic scale.

Linear Bayesian regression and PMM had very similar computation times, being the second and third slowest for small datasets. For the bigger datasets, they became faster than k-NN. k-NN itself was computationally very competitive for the smaller (< 1000 observations) datasets. However, as the number of observations grows, it quite quickly becomes very computationally intensive. This is because for each missing value (which also grows with dataset size), the algorithm has to check every single other observation to see which are the nearest neighbours. For the Adult dataset, with more than 30.000 observations, the k-NN algorithm took approximately 7 minutes to complete one dataset. For all datasets except for Adult, the Random Forest algorithm took the longest time to complete. The full averaged values of all computation times can be found in Appendix A.5.

All computation times logically increase when the number of observations increase. However, we see that for the Car and Banknotes datasets (at 1728 and 1372 observations respectively), there are breaks in the lines. The computation times were faster than for the simulated datasets, which have less observations. This can be explained by the fact that both the simulated and skewed datasets contained 18 variables, compared to 7 and 5 for Car and Banknotes, respectively. This explains the decrease in computation times, even though the number of observations was slightly larger.

6. Extension: Type 2 MNAR Results

This section will provide an overview and interpretation of the results obtained for the Type 2 MNAR mechanism. Due to the nature of how this research was conducted, this section should be seen as separate from the main results in Section 5. The structure will be the same: first, the used association measures are discussed briefly, after which the direct evaluation metrics will be reviewed in Section 6.1. After that, Section 6.2 will analyze the indirect evaluation in the form of post-imputation classification accuracy.

As explained in Section 4.5, we use different correlation or association metrics for different types of variables. The variables where we introduce Type 2 MNAR missingness are the same as in the main results, i.e. the two variables with the highest XGBoost-based feature importance for each dataset are imputed. For each of those variables, we test which other variable has the highest correlation and use that correlated variable to introduce missingness. Table 6.1 shows an overview of which variables had the highest correlation, as well as the value of the correlation/association. The last column shows the XGBoost-based feature importance for the correlated variable.

Dataset	Var to Impute	Correl. Var	Metric	Value	Importance
Adult	Cap_gain	Income	P-B	0.22	n/a
	Maritalstatus	Age	ANOVA (η^2)	0.18	0.11
Banknotes	Variance	Target	P-B	-0.71	n/a
	Skewness	Kurtosis	Pearson	-0.79	0.14
Car	Persons	Acceptability	Spearman's ρ	0.39	n/a
	Safety	Acceptability	Spearman's ρ	0.47	n/a
Iris	Petal Length	Sepal Length	Pearson	0.87	0.01
	Petal Width	Sepal Length	Pearson	0.82	0.01
Simulated	Two Factor 1	Two Factor 2	Pearson	0.65	0.13
	Cat1	Target	Cramer's V	0.64	n/a
Skewed	Two Factor 1	Two Factor 2	Pearson	0.65	0.13
	Cat1	Target	Cramer's V	0.64	n/a

Table 6.1: Correlation Metrics and Values for Different Datasets

As can be seen in the above table, not all variables had another variable that was highly correlated. Especially the variables in the Adult dataset did not have a highly correlated variable. When looking at the outcomes of the direct and indirect evaluation, we should look at

the strength of the association as well, as this can influence how heavily the Type 2 MNAR mechanism affects the missing values.

6.1 Direct Evaluation

For the direct evaluation, the analysis is twofold: for the continuous variables, we examine the RMSE of the imputed values compared to the original values. For the categorical variables, we analyze the PCP for each variable. Table 6.2 shows the RMSE values under the Type 2 MNAR missingness mechanism. Full IQR graphs can be found in Appendix A.4.1.

Table 6.2: Mean RMSE results for Type 2 MNAR mechanism. Bold values denote lowest (best) value for that method. Values in parentheses denote standard deviations.

Dataset	**Adult**	**Banknotes**		**Iris**		**Simulated**	**Skewed**
Method	Cap Gain	Variance	Skewness	PetalLength	PetalWidth	TwoFactor1	TwoFactor1
Mean	10,294 *(80.7)*	4.13 *(0.04)*	6.51 *(0.11)*	2.08 *(0.11)*	0.89 *(0.05)*	**1.46** *(0.04)*	1.19 *(0.04)*
Median	10,332 *(77.9)*	4.41 *(0.04)*	6.55 *(0.09)*	3.11 *(0.29)*	1.24 *(0.08)*	1.49 *(0.04)*	**1.08** *(0.04)*
k-NN	10,339 *(71.5)*	3.46 *(0.17)*	5.17 *(0.19)*	**0.47** *(0.03)*	0.24 *(0.03)*	1.82 *(0.05)*	1.51 *(0.06)*
LB	10,288 *(74.3)*	4.27 *(0.10)*	**4.92** *(0.09)*	0.54 *(0.02)*	**0.21** *(0.01)*	1.64 *(0.04)*	1.40 *(0.05)*
EM	10,292 *(71.2)*	4.21 *(0.11)*	4.97 *(0.11)*	0.55 *(0.03)*	**0.21** *(0.01)*	1.62 *(0.04)*	1.39 *(0.05)*
RF	10,306 *(74.4)*	**3.14** *(0.12)*	5.57 *(0.20)*	0.64 *(0.13)*	0.27 *(0.05)*	1.61 *(0.05)*	1.35 *(0.06)*
pmm	**10,281** *(74.6)*	4.18 *(0.09)*	4.94 *(0.11)*	**0.47** *(0.04)*	0.23 *(0.02)*	1.64 *(0.05)*	1.42 *(0.06)*

The first thing that stands out is the performance of the mean and median imputations. Unsurprisingly, for the Adult, Banknotes and Iris dataset they underperform compared to the other methods. However, for the Simulated and Skewed datasets they achieve the lowest RMSE out of all methods. Mean imputation performed best for the Simulated data, which had a normal distribution. For the Skewed data, median imputation worked slightly better. This is in line with what we would expect, as the median is more robust to outliers which are more present in the Skewed dataset.

For the Adult dataset, LB, EM and PMM were the best performing methods. Surprisingly, k-NN was the worst for this dataset, even achieving a higher RMSE than mean and median imputation. This could be due to the nature of the data, as the Cap Gain variable contains a lot of zero values. It should also be noted that the RMSE values could be close due to the weak association in the underlying Type 2 MNAR mechansim, with a point-biserial correlation of only 0.22.

When we look at the Banknotes dataset, we see some surprising results. Methods that impute the Variance variable well (k-NN, RF), do not achieve the same results for the Skewness variable. Conversely, methods that impute Skewness well (LB, EM, PMM) perform worse for Variance. This could be due to the underlying mechanism in which the Type 2 MNAR missingness is introduced. For Variance, the Target variable was used to introduce missingness.

For Skewness, the correlated variable was Kurtosis. This suggests that in this case, the Target variable is more important in the imputation process for LB, EM and PMM than the Curtosis variable is. Conversely, LB, EM and PMM seem to handle the imputation process better in the absence of the Curtosis variable.

For the Iris dataset, all methods except mean and median imputation achieved more similar results than in the other datasets. However, k-NN and PMM did seem to work best across both variables. Random Forest imputation shows that it needs more of data to be able to compete with the other methods, as the Iris dataset only contains 150 observations.

When we evaluate the PCP for all categorical variables, the results are more clear. Table 6.3 shows the mean Percentage of Correct Predictions for each method and dataset. Full IQR graphs can be found in Appendix A.4.2.

Table 6.3: Mean PCP results for Type 2 MNAR mechanism. Bold values denote highest (best) value for that method. Values in parentheses denote standard deviations.

Dataset	**Adult**	**Car**		**Simulated**	**Skewed**
Method	Maritalstatus	Persons	Safety	Cat1	Cat1
Mode	16.35 *(0.17)*	31.17 *(0.69)*	30.75 *(0.85)*	12.71 *(4.90)*	13.07 *(5.49)*
k-NN	**75.47** *(0.23)*	29.16 *(0.88)*	28.99 *(1.71)*	27.89 *(2.12)*	27.52 *(2.39)*
LB	41.68 *(0.33)*	**33.54** *(1.22)*	**33.35** *(1.16)*	**28.05** *(1.68)*	**28.14** *(2.13)*
EM	41.84 *(0.36)*	33.51 *(1.51)*	32.79 *(1.27)*	27.65 *(1.70)*	27.29 *(1.83)*
RF	68.46 *(0.23)*	31.11 *(1.13)*	30.92 *(1.22)*	27.02 *(2.08)*	26.69 *(1.97)*
pmm	40.84 *(0.39)*	33.45 *(1.22)*	32.88 *(1.35)*	27.43 *(1.84)*	27.49 *(1.79)*

We immediately see that with the exception of the Adult dataset, the Linear Bayesian algorithm outperforms all other algorithms, albeit marginally. Only one method significantly outperformed all other methods in a dataset, which was the k-NN algorithm for the Adult dataset. We see that for the other datasets, the best performing methods were LB, EM and PMM. Interestingly, Mode imputation often performs worse than expected due to the Type 2 MNAR mechanism. For the Adult dataset, the mode value for *Maritalstatus* accounts for 46% of all observations. However, mode imputation only achieved a PCP of 16.35%. Similarly, it achieved a PCP of approximately 13% for both the Simulated and Skewed dataset, where the mode accounts for 33% of all observations. In the Car dataset the difference is smaller: a PCP of approximately 31% where the mode is present in 33% of observations. With the exception of k-NN and RF in the Adult dataset, no method achieved a PCP value that was higher than the percentage of observations that correspond to the mode. This suggests that k-NN and RF achieve better results when more data is available. It also highlights how the Type 2 MNAR mechanism can heavily influence imputation performance, as the underlying missingness is much harder to predict for our imputation algorithms.

6.2 Indirect Evaluation

For the indirect evaluation, we again look at the average XGBoost classification accuracy over 20 imputation runs. Table 6.4 shows the summarized results for each method and variable.

Table 6.4: Average XGBoost accuracy for all datasets and methods over 20 imputations under Type 2 MNAR missingness. Values are percentages, bold values denote highest accuracy. 'Full' column denotes XGBoost accuracy for full dataset.

Dataset	**Variable**		**Method**							
		Full	**No Imp.**	**Mean**	**Median**	**k-NN**	**LB**	**RF**	**PMM**	**EM**
Adult	CapGain	*86.96*	**89.89**	89.88	84.85	84.74	89.69	85.08	85.35	89.68
	Maritalstatus.	*86.96*	86.54	86.58	86.58	86.55	**87.17**	86.60	87.12	87.13
Banknotes	Variance	*99.20*	97.27	**97.43**	97.35	94.82	95.20	95.03	94.91	95.05
	Skewness.	*99.20*	95.65	95.18	95.07	**97.91**	94.89	95.71	95.42	94.84
Car	Persons	*99.02*	**85.81**	74.05	74.05	78.65	82.99	81.52	83.47	82.97
	Safety	*99.02*	**80.38**	71.73	71.73	73.08	76.51	75.54	76.46	76.51
Iris	PetalLength	*94.67*	94.47	94.47	94.43	**97.43**	95.10	95.73	95.30	95.67
	PetalWidth	*94.67*	93.00	93.03	93.03	**97.33**	97.10	95.03	97.07	97.27
Simulated	TF1.	*94.30*	90.65	90.59	90.60	**92.82**	91.96	91.39	92.29	92.24
	Cat1	*94.30*	**96.56**	94.39	94.39	94.14	94.20	94.29	94.11	94.38
Skewed	TF1.	*94.30*	90.74	90.66	90.59	93.08	92.53	92.37	**93.18**	93.14
	Cat1	*94.30*	**96.66**	94.62	94.62	93.85	94.31	94.00	94.04	94.47

The first thing that we notice is that k-NN seems to be the best overall method when it comes to preserving post-imputation classification accuracy. However, perhaps more importantly, the results show that in some cases it is better to not impute the variables at all if one is looking to optimize classification accuracy. Under Type 2 MNAR missingness, the imputation algorithms seem to struggle to find the underlying missingness pattern and data structure, resulting in more noise due to bad imputations and lower classification accuracy. Surprisingly, Random Forest imputation does not outperform the other methods in any dataset, although it is competitive everywhere.

When we look at the types of data, we see that the imputation algorithms mostly struggle with categorical data. The cases where a categorical variable had to be imputed almost all had higher classification accuracy when the missing values were not imputed, with *Maritalstatus* being the exception.

Because we essentially remove the correlated variable from the dataset, we should take into account how important that variable was for a classification task. As was shown in Table 6.1, the variables *Maritalstatus*, *Skewness* and *Two Factor 1* (for both the Simulated and Skewed datasets) were amputed using a correlated variable that had a significant XGBoost feature importance. We see that for these variables, the classification accuracy was on average lower when

compared to the other variable from that dataset. This makes sense, as the variable itself had to be imputed, which affects classification performance, but the correlated variable was 'lost' as well. Interestingly, the k-NN algorithm seems to perform well in these cases, often even exceeding the classification accuracy for when the other variable was missing.

Another thing that stands out is that in some cases, the classification accuracy is higher post-imputation compared to the full data. This is the case for the Iris dataset when imputed with any of the non-benchmark methods, but also for Adult (CapGain), Simulated (Cat1) and Skewed (Cat1) when the variables were not imputed at all. The latter suggests that for these variables, the imputations were bad enough to have a detrimental effect on the classification accuracy by adding more noise than relevant information.

Although most methods were competitive for most datasets, for continuous data the most robust results were obtained by using the k-NN algorithm. For categorical data, the most robust method that generalizes well over different datasets was actually not imputing the values at all but rather letting the XGBoost algorithm deal with the missing values.

7. Conclusion and Limitations

In this thesis, we set out to explore a range of imputation methods to find out which method is most suitable for business applications. We analyzed how these methods performed under different missingness mechanisms (MCAR, MAR, MNAR 1, MNAR 2), and tested them all on six different datasets. The methods that were tested in this research consisted of both single and multiple imputation methods. We used Mean/Mode and Median/Mode imputation as benchmark methods, and compared them to the performance of the k-NN, Expectation Maximization, Linear Bayesian Regression, Random Forest and Predictive Mean Matching imputation methods.

To effectively be able to compare both single imputation and multiple imputation methods, we made use of multiple amputation, where a dataset is amputed and imputed multiple times in order to obtain more reliable results. All methods were evaluated by way of RMSE (Root Mean Squared Error) and PCP (Percentage of Correct Predictions), post-imputation XGBoost classification performance and computational requirements. We have found that unsurprisingly, there is no one-size-fits-all solution. Each method had its strengths and weaknesses under different conditions.

In our main findings, for our dataset with only categorical variables, PMM outperformed all other methods in the PCP metric, with LB being the second best. However, when we look at the subsequent classification performance, k-NN showed that it was best at maintaining the underlying structures in the dataset, therefore obtaining the highest post-imputation classification accuracy. For continouous datasets, k-NN performed best when the data was not approximately normally distributed. When the data did follow an approximately normal distribution, PMM and RF performed best based on the RMSE metric. When looking at the post-imputation classification performance, these three methods were also the best at maintaining accuracy, regardless of the type of missingness mechanisms. For our mixed-type datasets, RF appeared to be the most consistent across all datasets and mechanisms based on the RMSE metric. However, for the categorical imputations, k-NN was again the best or very competitive for all mixed datasets. Post-imputation classification performance showed that PMM and k-NN were the best performing methods.

In the extension regarding the Type 2 MNAR mechanism, we obtained different results. Although there was no clear winner, the k-NN algorithm appears to be most robust to different datasets when dealing with continuous variables, obtaining competitive RMSE values most of the time. When dealing with categorical imputations under Type 2 MNAR missingness, the

Linear Bayes algorithm outperformed the other methods in five out of six cases, albeith marginally.

When looking at the results of the classification accuracy under Type 2 MNAR missingness, we see surprising results. When looking at post-imputation classification accuracy for categorical variables, it often was better to not impute the variables at all. The imputation methods only introduced more noise in the dataset, resulting in lower XGBoost classification accuracy post-imputation compared to letting the XGBoost classifier deal with the missing values itself. This shows that algorithms can have real difficulty dealing with missing data that stems from the Type 2 MNAR mechanism.

Whilst k-NN and RF may seem the clear winners, when we incorporate computation times they do not appear as favourable. k-NN and RF showed to be the slowest algorithms by far when dealing with larger (> 1000 observations) datasets. For our biggest dataset with approximately 32 thousand observations, k-NN took more than 7 minutes to impute one dataset. This is obviously not desirable, as real-life datasets often grow even larger than that. However, it is also worth noting that in a business setting, a method that generalizes well across different missingness mechanisms is desirable, as one can not test for these mechanisms in practice. In this thesis, k-NN seemed most stable across these mechanisms.

Concluding, we can say that the imputation algorithm that is most suitable heavily depends on the data type that one has, as well as the intended goal of the imputations and underlying missing mechanism. When the goal is to obtain highest post-imputation classification accuracy, k-NN, PMM and RF are best suitable, as long as the missingness is in some way related to your data, i.e. it is not Type 2 MNAR missingness. If there are time constraints, one should choose PMM out of these three. If the goal is to get the most accurate imputations, RF or k-NN could be best suitable if computation time is not an issue.

This paper has some limitations. First, for all methods that were implemented using the **mice** package, the *maxiter* variable was set to 10. Even though the literature states that convergence should be reached after as few as 5-10 iterations, further research can be done to see if more iterations provide better results. Second, due to time and computational restraints, we could not optimize the XGBoost classifier and k-NN algorithm. The base XGBoost model was used for each classification, and the k parameter in the k-NN algorithm was set to 3 for each run. Optimization could lead to more accurate results. Future research can see if this optimization changes the results in any way. Lastly, the percentage of missing values was constant throughout this research, having been set at 50%. Due to time constraints it was not possible to investigate the effects of different percentages on the performance of the imputation techniques.

Further research could improve on this paper by considering smaller and/or larger missing value proportions. Another addition could be to try to improve computation time in the k-NN and RF algorithms, whilst trying to maintain imputation performance. Furthermore, the Type

2 MNAR mechanism is not yet widely studied, so more analyses and more thorough studies should be done to further investigate how one should deal with this type of missingness.

References

Adnan, F. A., Jamaludin, K. R., Wan Muhamad, W. Z. A. & Miskon, S. (2022). A review of the current publication trends on missing data imputation over three decades: direction and future research. *Neural Computing and Applications*, *34*(21), 18325–18340.

Ambler, G., Omar, R. Z. & Royston, P. (2007). A comparison of imputation techniques for handling missing predictor values in a risk model with a binary outcome. *Statistical Methods in Medical Research*, *16*. doi: 10.1177/0962280206074466

Becker, B. & Kohavi, R. (1996). *Adult.* UCI Machine Learning Repository. (DOI: https://doi.org/10.24432/C5XW20)

Bohanec, M. (1997). *Car Evaluation.* UCI Machine Learning Repository. (DOI: https://doi.org/10.24432/C5JP48)

Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5–32.

Chen, T. & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785–794).

Cramér, H. (1999). *Mathematical methods of statistics* (Vol. 26). Princeton university press.

Eirola, E., Doquire, G., Verleysen, M. & Lendasse, A. (2013). Distance estimation in numerical data sets with missing values. *Information Sciences*, *240*, 115–128.

Faisal, S. & Tutz, G. (2021). Imputation methods for high-dimensional mixed-type datasets by nearest neighbors. *Computers in Biology and Medicine*, *135*, 12. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0010482521003711` doi: https://doi.org/10.1016/j.compbiomed.2021.104577

Farhangfar, A., Kurgan, L. A. & Pedrycz, W. (2007). A novel framework for imputation of missing values in databases. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *37*(5), 692–709.

Fielding, S., Fayers, P. M. & Ramsay, C. R. (2009). Investigating the missing data mechanism in quality of life outcomes: a comparison of approaches. *Health and Quality of Life Outcomes*, *7*(1), 1–10.

Finney, S. J. & DiStefano, C. (2006). Non-normal and categorical data in structural equation modeling. *Structural equation modeling: A second course*, *10*(6), 269–314.

Fisher, R. A. (1988). *Iris.* UCI Machine Learning Repository. (DOI: https://doi.org/10.24432/C56C76)

Folino, G. & Pisani, F. S. (2016). Evolving meta-ensemble of classifiers for handling incomplete and unbalanced datasets in the cyber security domain. *Applied Soft Computing*, *47*, 179–190.

Fuechsel, G. (1960s). *Origin of the phrase 'garbage in, garbage out'.* (Unpublished article, but widely attributed to G. Fuechsel)

Gautam, C. & Ravi, V. (2015). Data imputation via evolutionary computation, clustering and a neural network. *Neurocomputing*, *156*, 134–142.

Ghomrawi, H. M., Mandl, L. A., Rutledge, J., Alexiades, M. M. & Mazumdar, M. (2011, 5). Is there a role for expectation maximization imputation in addressing missing data in research using womac questionnaire? comparison to the standard mean approach and a tutorial. *BMC Musculoskeletal Disorders*, *12*, 1-7. Retrieved from https://bmcmusculoskeletdisord.biomedcentral.com/articles/10.1186/1471-2474-12-109 doi: 10.1186/1471-2474-12-109/FIGURES/2

Hameed, W. M. & Ali, N. A. (2022). Comparison of seventeen missing value imputation techniques. *Hunan Daxue Xuebao/Journal of Hunan University Natural Sciences*, *49*. doi: 10.55463/issn.1674-2974.49.7.4

Honaker, J., King, G. & Blackwell, M. (2011). Amelia ii: A program for missing data. *Journal of statistical software*, *45*, 1–47.

Jadhav, A., Pramod, D. & Ramanathan, K. (2019). Comparison of performance of data imputation methods for numeric dataset. *Applied Artificial Intelligence*, *33*. doi: 10.1080/08839514.2019.1637138

Jäger, S., Allhorn, A. & Bießmann, F. (2021). A benchmark for data imputation methods. *Frontiers in big Data*, *4*, 693674.

Joyce, J. (2003). Bayes' theorem.

Kiasari, M. A., Jang, G.-J. & Lee, M. (2017). Novel iterative approach using generative and discriminative models for classification with missing features. *Neurocomputing*, *225*, 23–30.

Kowarik, A. & Templ, M. (2016). Imputation with the r package vim. *Journal of statistical software*, *74*, 1–16.

Krishnan, T. & McLachlan, G. J. (2012). The em algorithm. *Handbook of computational statistics: concepts and methods*, 139–172.

Kuhn, M. (2023). Package 'modeldata'.

Laaksonen, S. (2018). Sampling principles, missingness mechanisms, and design weighting. *Survey methodology and missing data: Tools and techniques for practitioners*, 49–76.

Lin, W. C. & Tsai, C. F. (2020). Missing value imputation: a review and analysis of the literature (2006–2017). *Artificial Intelligence Review*, *53*. doi: 10.1007/s10462-019-09709-4

Little, R. J. (1988). Missing-data adjustments in large surveys. *Journal of Business & Economic Statistics*, *6*(3), 287–296.

Little, R. J. & Rubin, D. B. (2019). Statistical analysis with missing data. *Statistical Analysis with Missing Data*. doi: 10.1002/9781119482260

Little, R. J. A. (1988). Missing-data adjustments in large surveys. *Journal of Business Economic Statistics*, *6*(3), 287–296. Retrieved 2024-01-10, from http://www.jstor.org/stable/1391878

Lohweg, V. (2013). *banknote authentication.* UCI Machine Learning Repository. (DOI: https://doi.org/10.24432/C55P57)

Mardia, K. V. (1970). Measures of multivariate skewness and kurtosis with applications. *Biometrika*, *57*(3), 519–530.

McLachlan, G. J. & Krishnan, T. (2007). *The em algorithm and extensions*. John Wiley & Sons.

Morris, T. P., White, I. R. & Royston, P. (2014). Tuning multiple imputation by predictive mean matching and local residual draws. *BMC medical research methodology*, *14*, 1–13.

Nishanth, K. J. & Ravi, V. (2016). Probabilistic neural network based categorical data imputation. *Neurocomputing*, *218*, 17–25.

Pearson, K. (1895). Vii. note on regression and inheritance in the case of two parents. *proceedings of the royal society of London*, *58*(347-352), 240–242.

Polit, D. F. & Beck, C. T. (2008). *Nursing research: Generating and assessing evidence for nursing practice*. Lippincott Williams & Wilkins.

Raftery, A. E., Madigan, D. & Hoeting, J. A. (1997). Bayesian model averaging for linear regression models. *Journal of the American Statistical Association*, *92*(437), 179–191.

Randahl, D. (2022). What's missing?: The effect of missing data and imputation techniques on predictive performance in forecasting civil war violence.

Redner, R. A. & Walker, H. F. (1984). Mixture densities, maximum likelihood and the em algorithm. *SIAM Review*, *26*(2), 195-239. Retrieved from `https://doi.org/10.1137/1026034` doi: 10.1137/1026034

Robins, J. M. & Wang, N. (2000). Inference for imputation estimators. *Biometrika*, *87*(1), 113–124.

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

Rubin, D. B. (1978). Multiple imputations in sample surveys-a phenomenological bayesian approach to nonresponse. In *Proceedings of the survey research methods section of the american statistical association* (Vol. 1, pp. 20–34).

Santos, M. S., Pereira, R. C., Costa, A. F., Soares, J. P., Santos, J. & Abreu, P. H. (2019). Generating synthetic missing data: A review by missing mechanism. *IEEE Access*, *7*, 11651–11667.

Schafer, J. L. & Olsen, M. K. (1998). *Multiple imputation for multivariate missing-data problems: A data analyst's perspective* (Vol. 33). doi: 10.1207/s15327906mbr3304$_5$

Shah, A. D., Bartlett, J. W., Carpenter, J., Nicholas, O. & Hemingway, H. (2014, 3). Comparison of random forest and parametric imputation models for imputing missing data using mice: A caliber study. *American Journal of Epidemiology*, *179*, 764-774. Retrieved from `https://dx.doi.org/10.1093/aje/kwt312` doi: 10.1093/AJE/KWT312

Silva-Ramírez, E.-L., Pino-Mejías, R. & López-Coello, M. (2015). Single imputation with multilayer perceptron and multiple imputation combining multilayer perceptron and k-nearest neighbours for monotone patterns. *Applied Soft Computing*, *29*, 65–74.

Spearman, C. (1961). The proof and measurement of association between two things.

Statista. (2022). *Volume of data/information created worldwide from 2010 to 2025.* Retrieved from `https://www.statista.com/statistics/871513/worldwide-data-created/` (Statista)

Stekhoven, D. J. & Bühlmann, P. (2012). Missforest-non-parametric missing value imputation

for mixed-type data. *Bioinformatics*, *28*. doi: 10.1093/bioinformatics/btr597

Valdiviezo, H. C. & Van Aelst, S. (2015). Tree-based prediction on incomplete data using imputation or surrogate decisions. *Information Sciences*, *311*, 163–181.

Van Buuren, S. & Groothuis-Oudshoorn, K. (2011). mice: Multivariate imputation by chained equations in r. *Journal of statistical software*, *45*, 1–67.

Van Buuren, S. & Oudshoorn, C. G. (2000). *Multivariate imputation by chained equations.* Leiden: TNO.

A. Tables and Figures

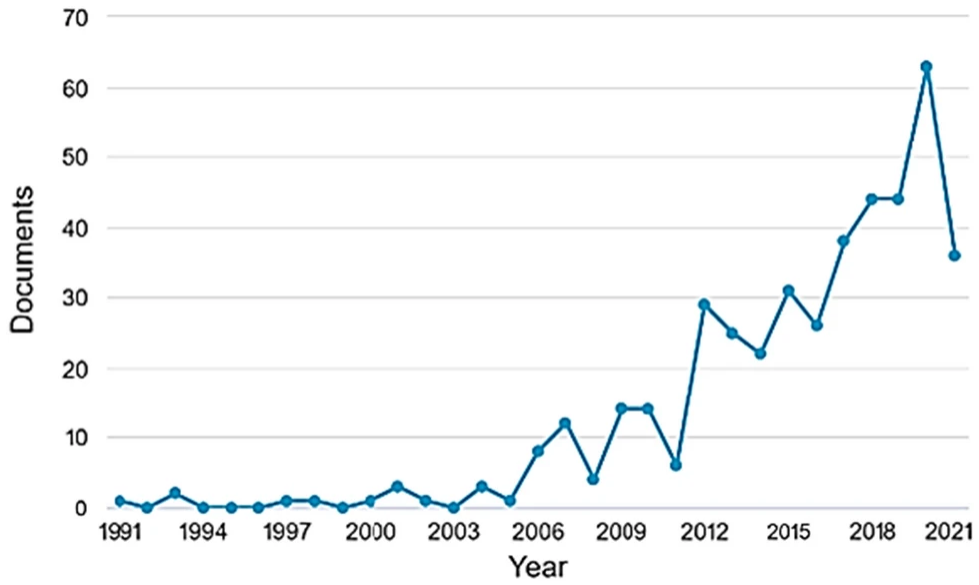

Figure A.1: Number of imputation publications by year (Adnan et al., 2022)

Table A.1: Results of Mardia's Tests for Multivariate Normality in the Iris Dataset

Test	Statistic	p-value	Result
Mardia Skewness	67.431	0.000	NO
Mardia Kurtosis	−0.230	0.818	YES
MVN	NA	NA	NO

Table A.2: Results of Mardia's Tests for Multivariate Normality in the Banknotes dataset

Test	Statistic	p-value	Result
Mardia Skewness	1025.282	0.000	NO
Mardia Kurtosis	0.620	0.535	YES
MVN	NA	NA	NO

A.1 Feature Importances

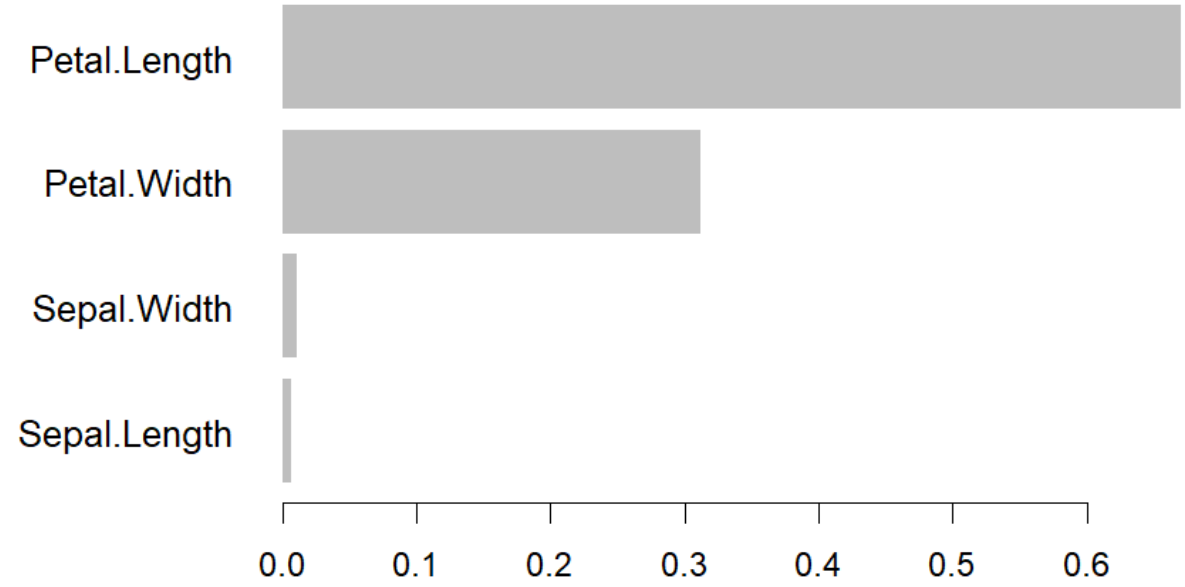

Figure A.2: Feature importances from base XGBoost model for Iris dataset

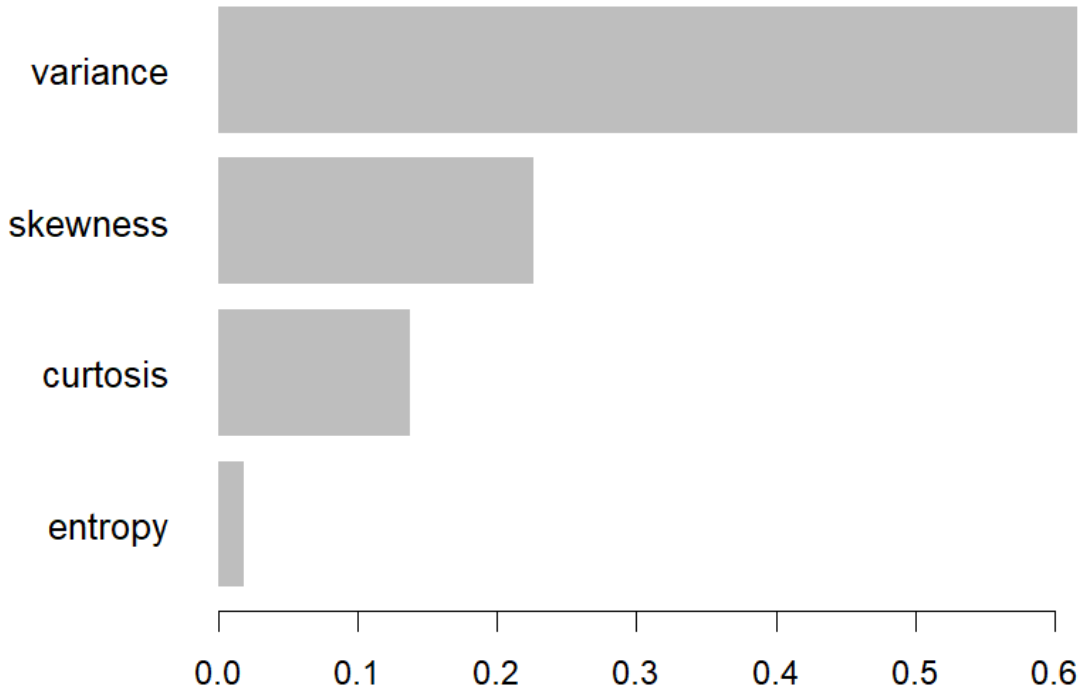

Figure A.3: Feature importances from base XGBoost model for Banknotes dataset

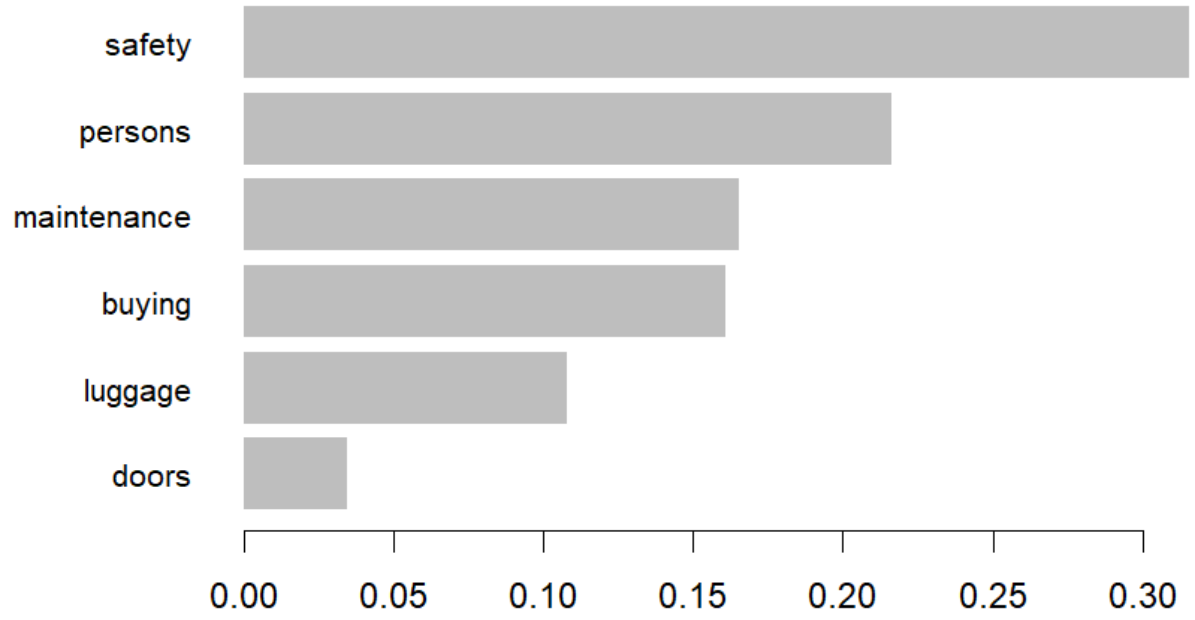

Figure A.4: Feature importances from base XGBoost model for Car dataset

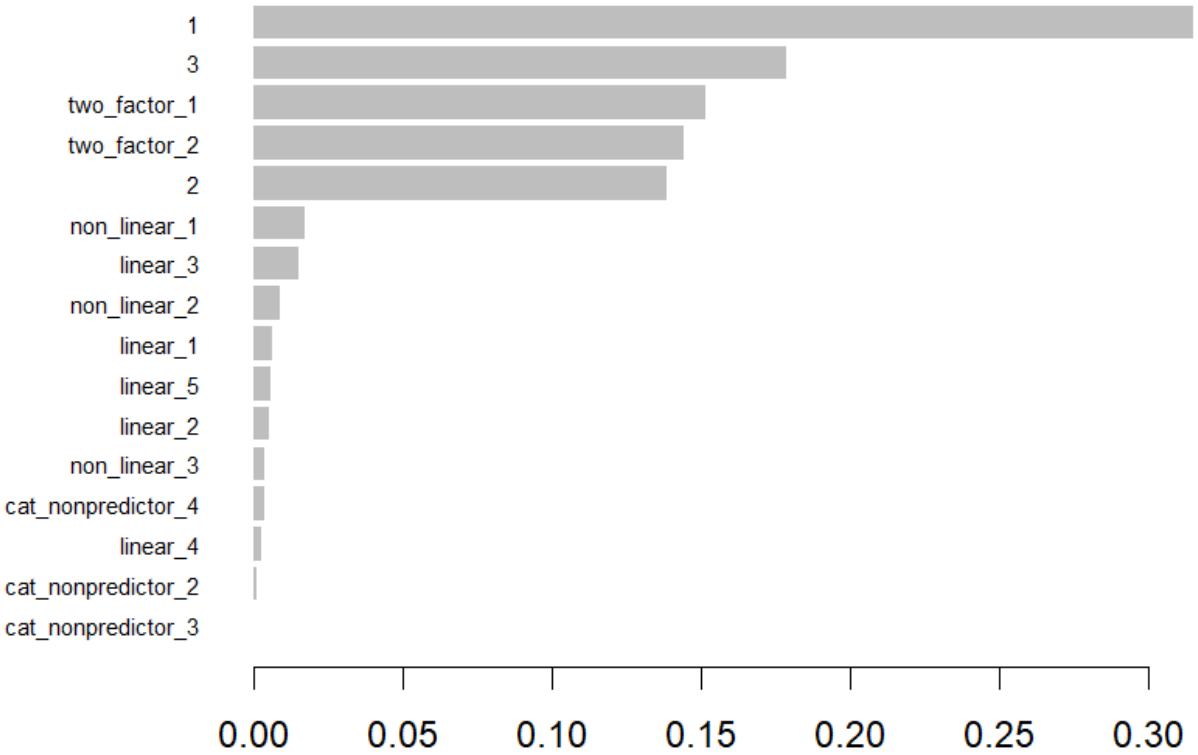

Figure A.5: Feature importances from base XGBoost model for Simulated dataset. 1, 2 and 3 are the categorical predictor variables. two_factor_1 and two_factor_2 are continuous predictor variables.

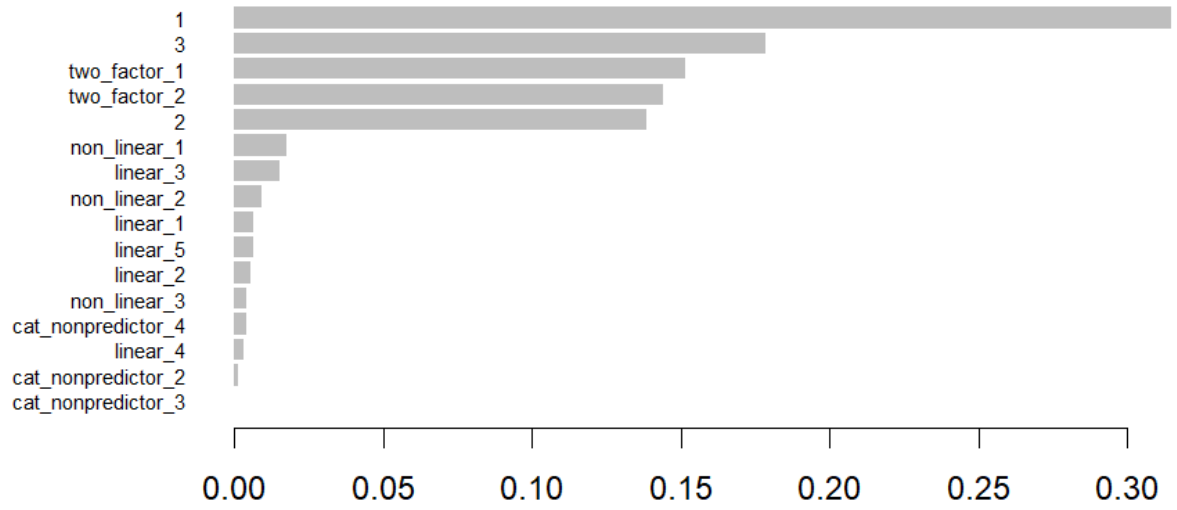

Figure A.6: Feature importances from base XGBoost model for Skewed dataset. 1, 2 and 3 are the categorical predictor variables. two_factor_1 and two_factor_2 are continuous predictor variables.

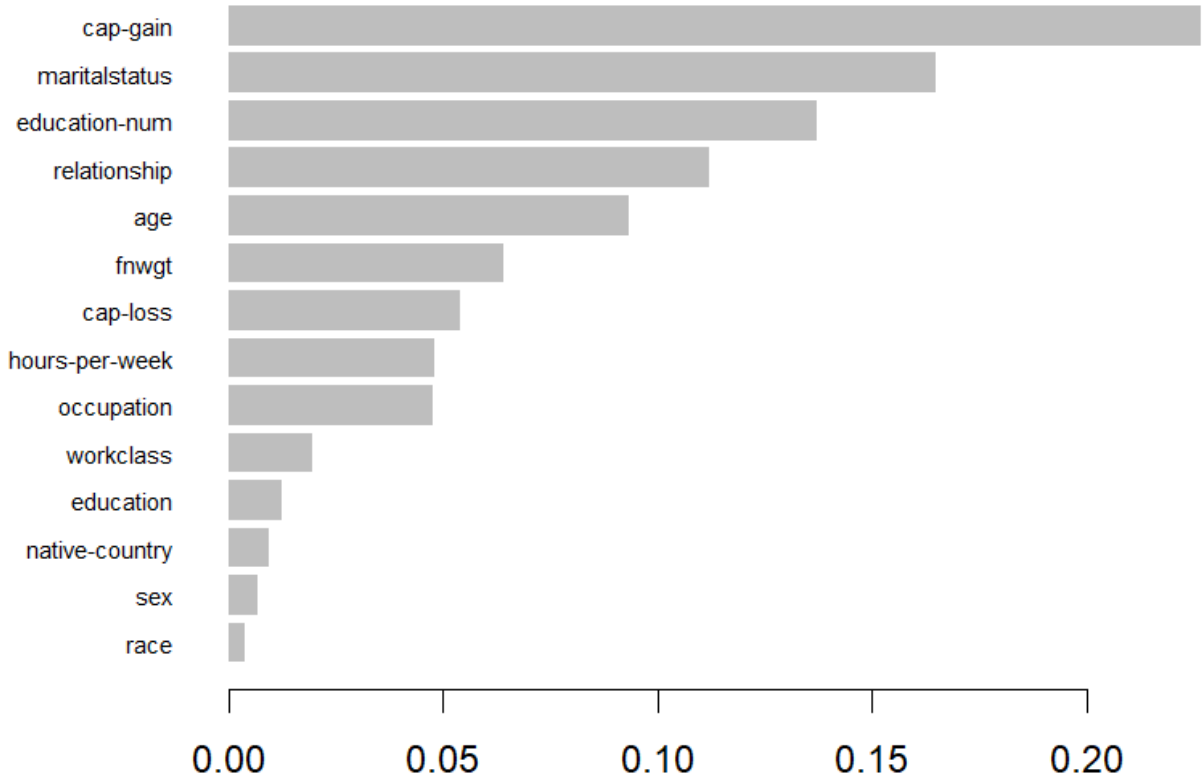

Figure A.7: Feature importances from base XGBoost model for Adult dataset

A.2 RMSE Graphs

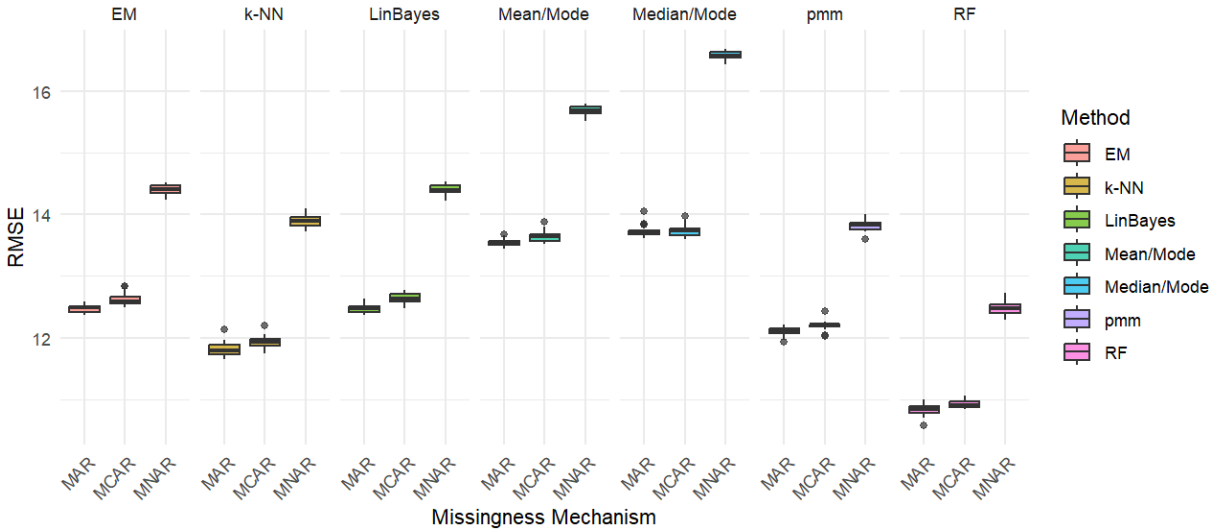

Figure A.8: IQR Graph of RMSE values for the Adult dataset

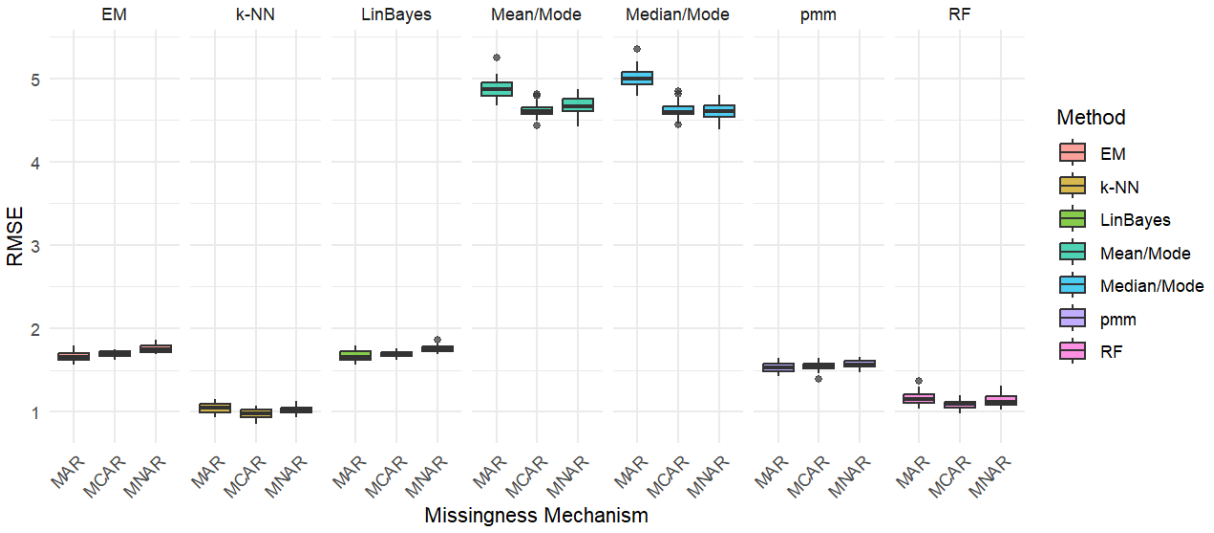

Figure A.9: IQR Graph of RMSE values for the Banknotes dataset

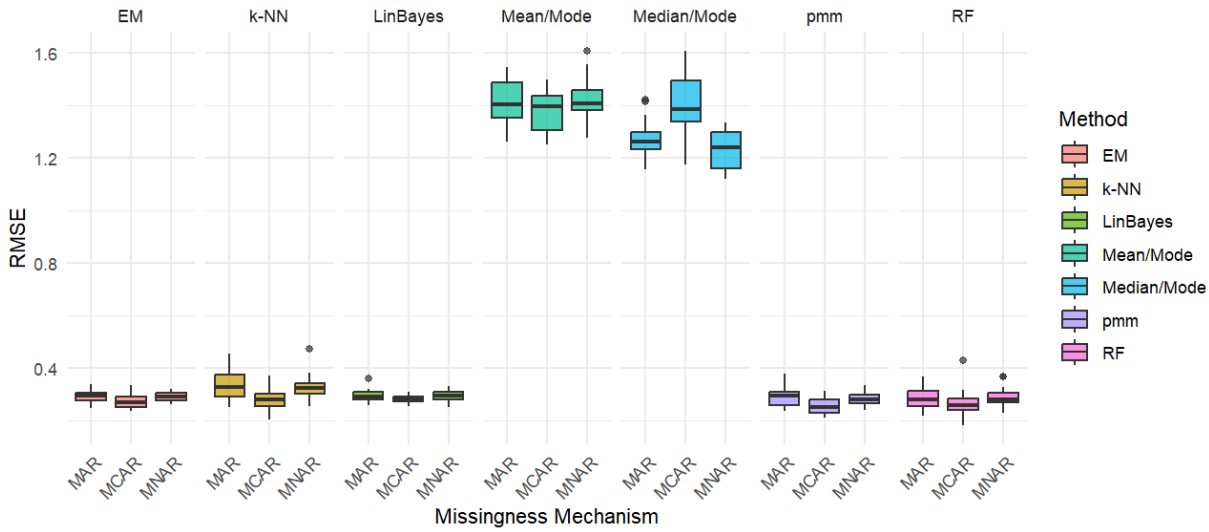

Figure A.10: IQR Graph of RMSE values for the Iris dataset

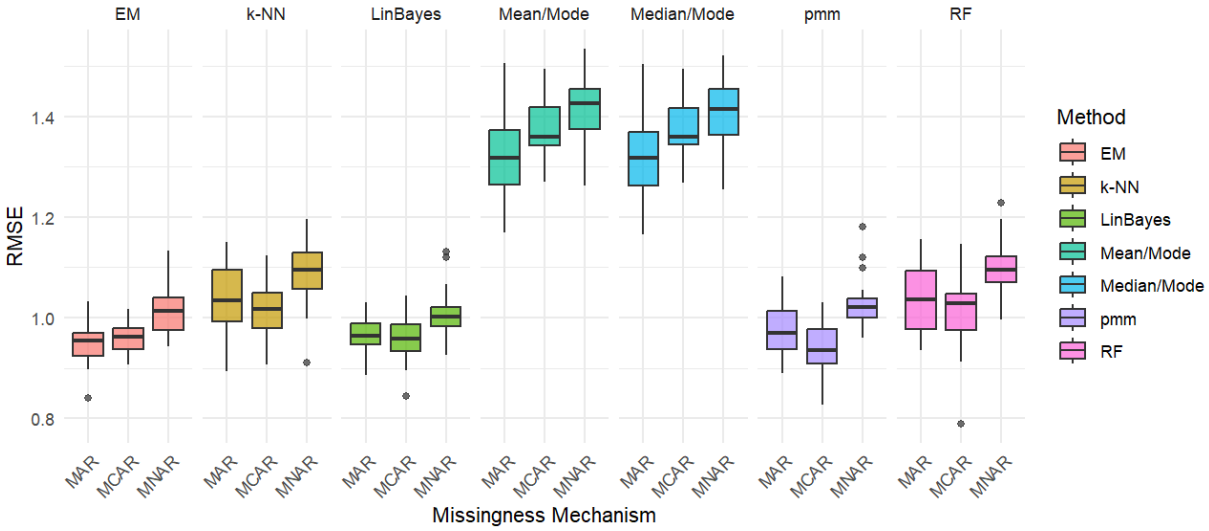

Figure A.11: IQR Graph of RMSE values for the Simulated dataset

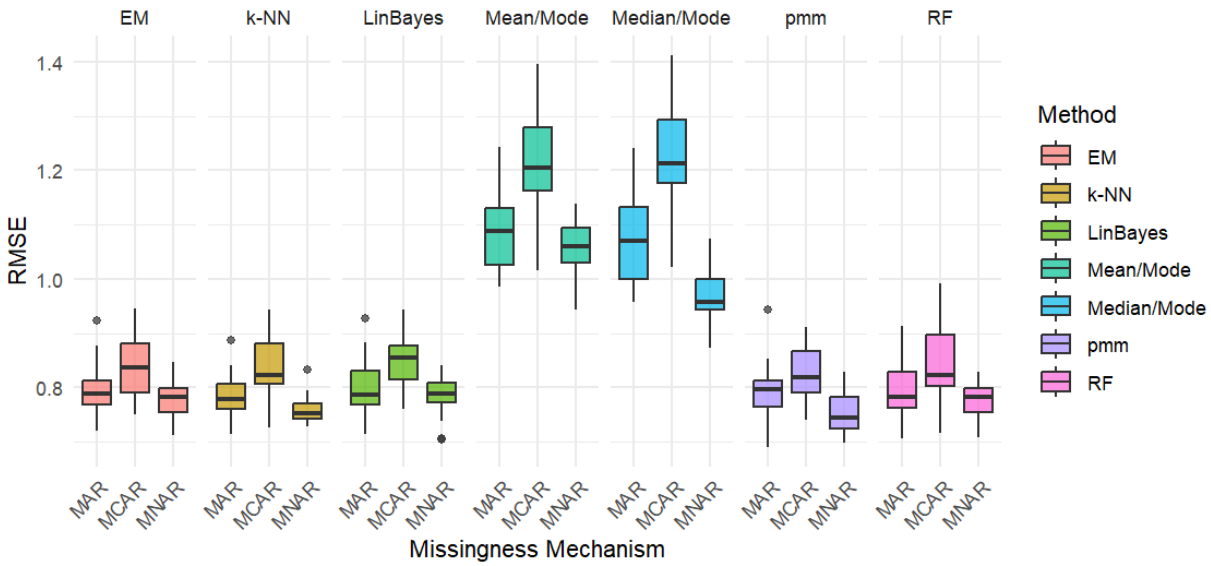

Figure A.12: IQR Graph of RMSE values for the Simulated skewed dataset

A.3 PCP Graphs

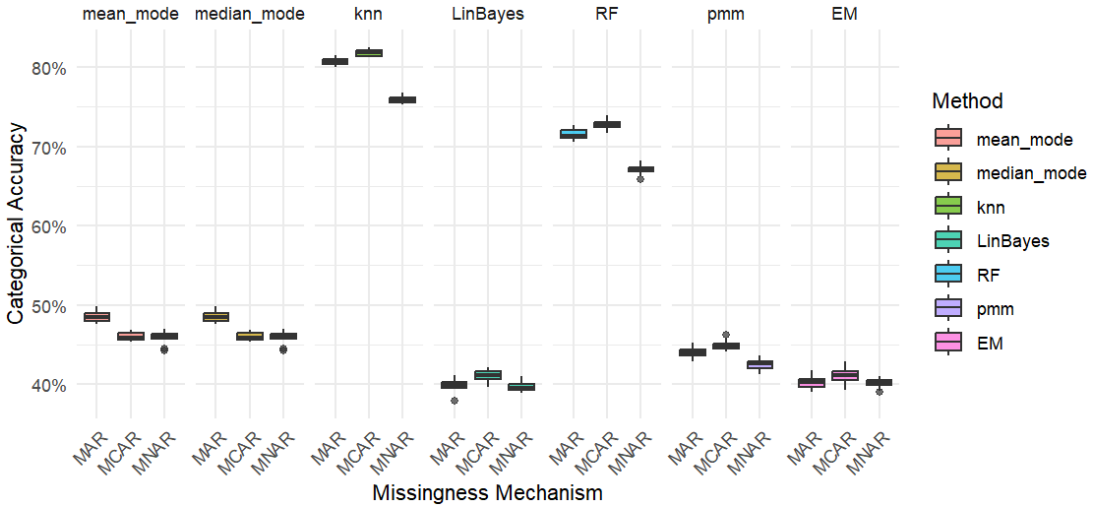

Figure A.13: IQR graph of PCP values for the Adult dataset

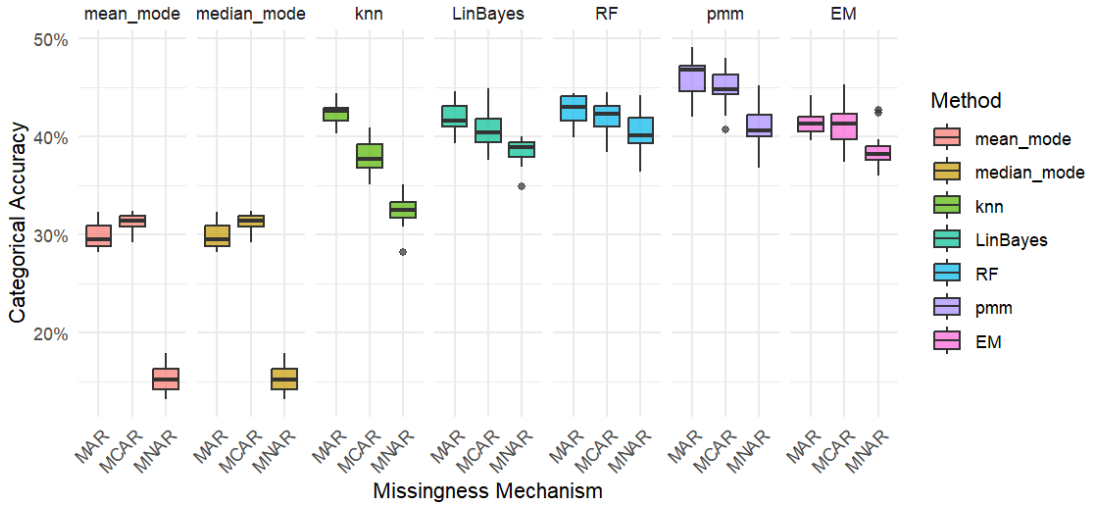

Figure A.14: IQR graph of PCP values for the Car dataset

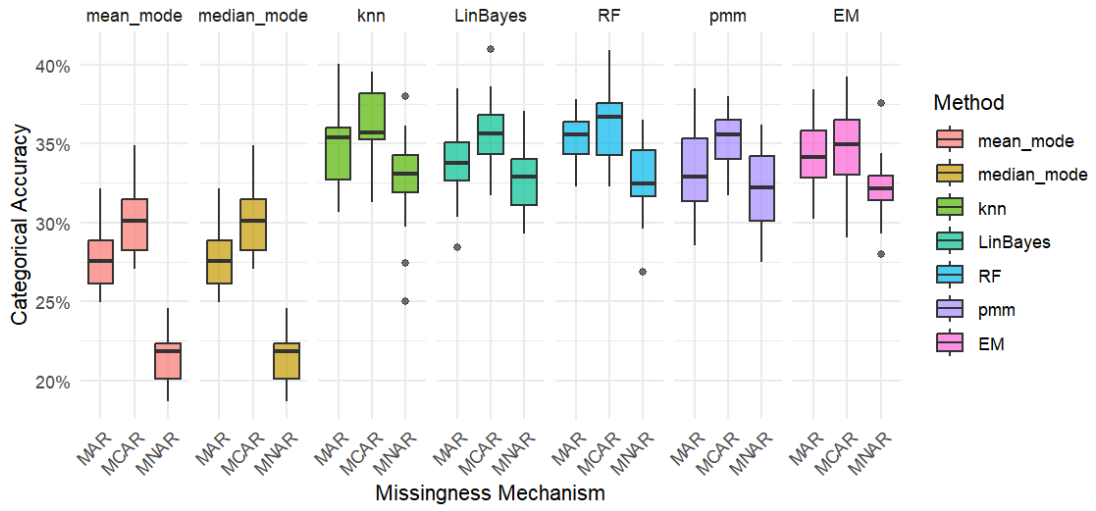

Figure A.15: IQR graph of PCP values for the Simulated dataset

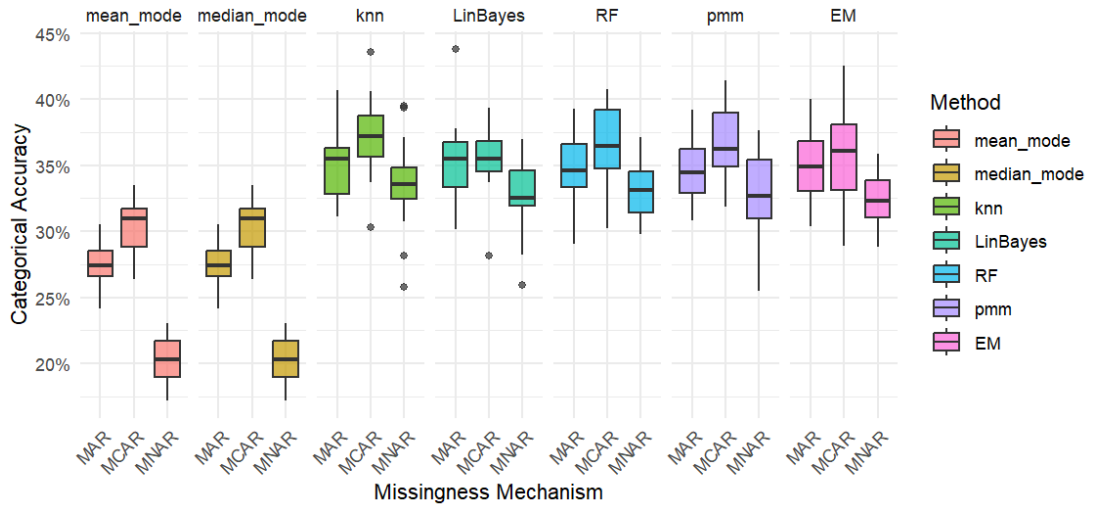

Figure A.16: IQR graph of PCP values for the Skewed dataset

A.4 Type 2 MNAR results

A.4.1 RMSE

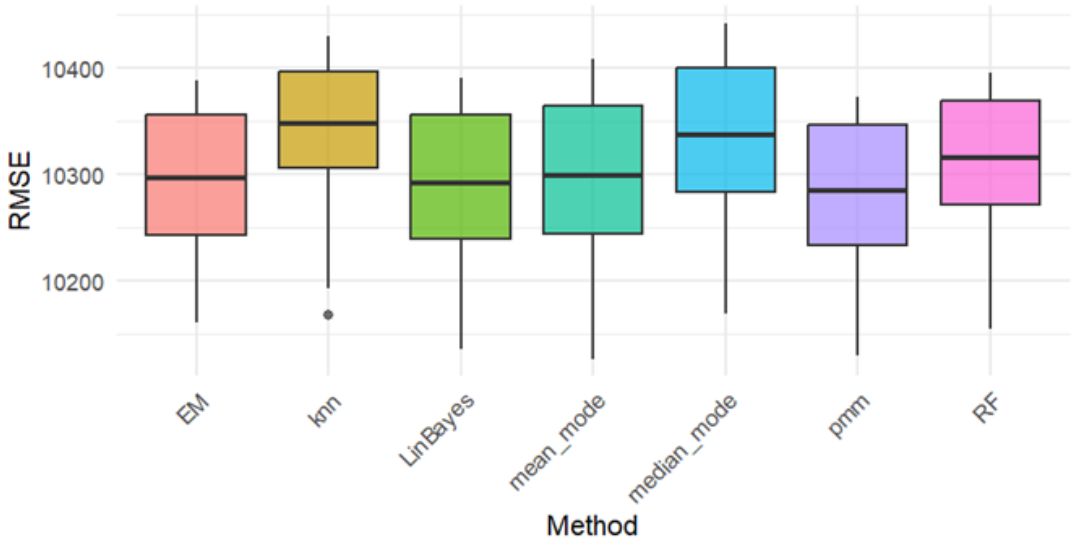

Figure A.17: IQR graph of RMSE values for the Cap-Gain variable in the Adult dataset under Type 2 MNAR

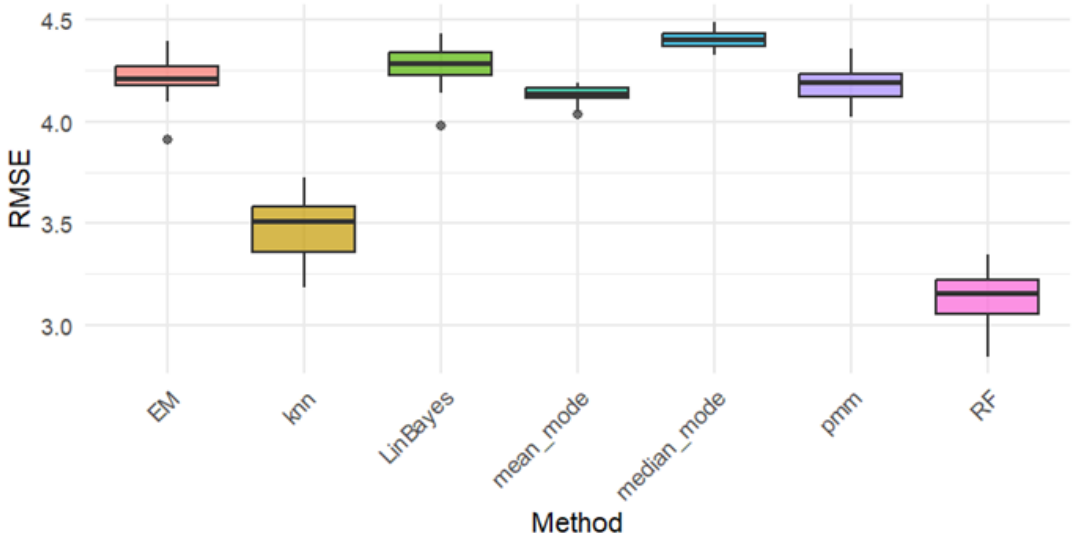

Figure A.18: IQR graph of RMSE values for the Variance variable in the Banknotes dataset under Type 2 MNAR

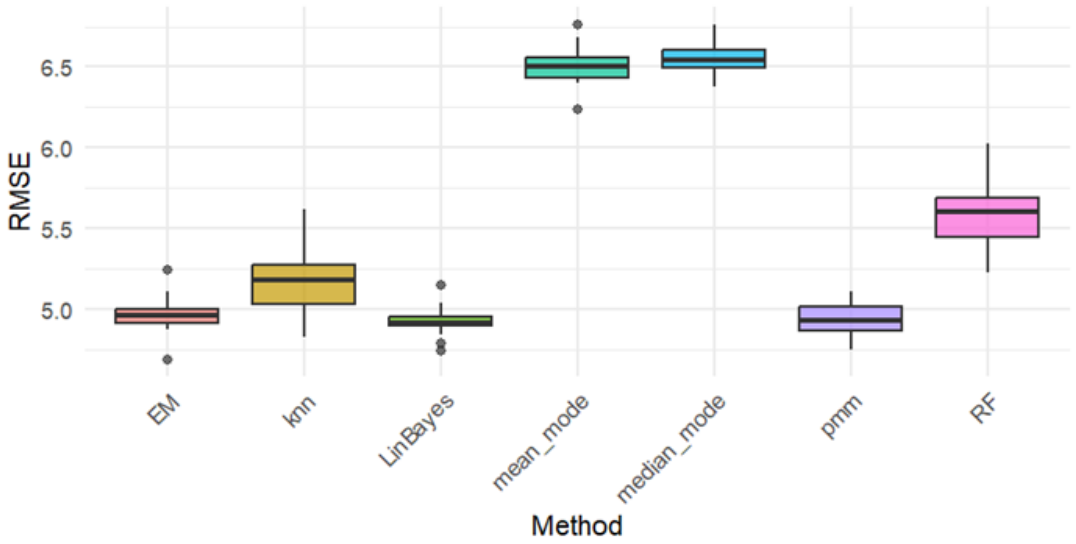

Figure A.19: IQR graph of RMSE values for the Skewness variable in the Banknotes dataset under Type 2 MNAR

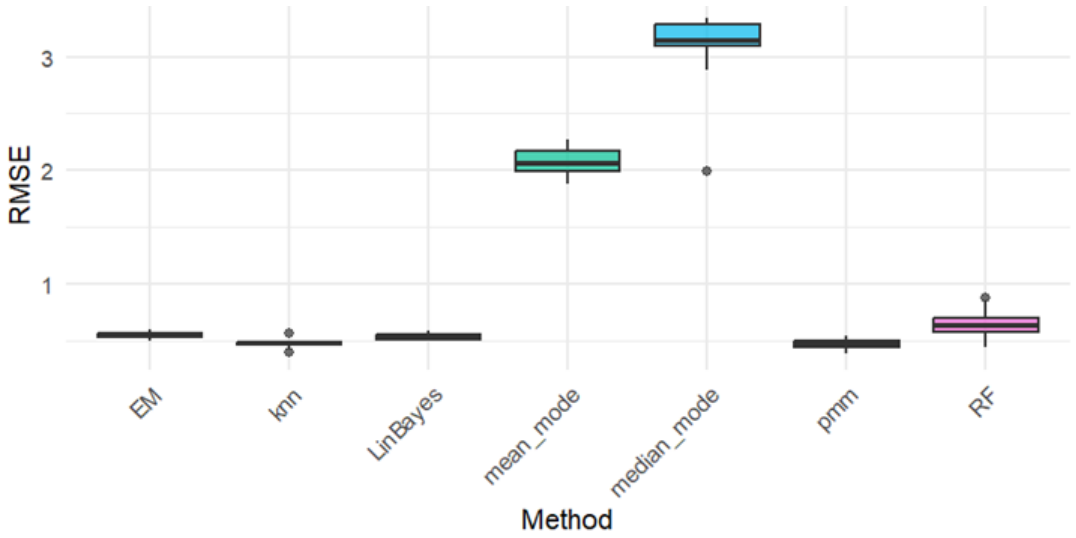

Figure A.20: IQR graph of RMSE values for the Petal Length variable in the Iris dataset under Type 2 MNAR

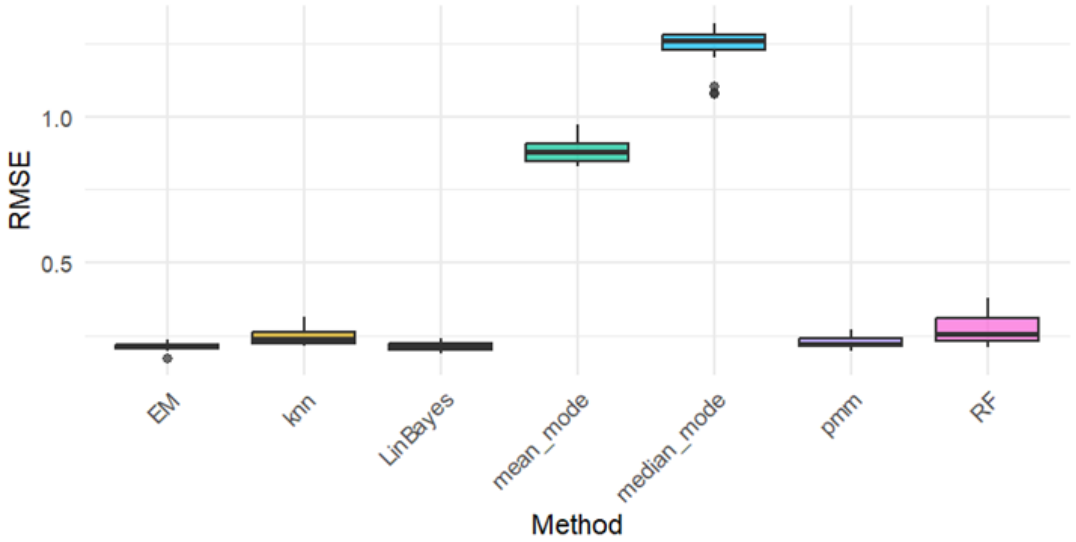

Figure A.21: IQR graph of RMSE values for the Petal Width variable in the Iris dataset under Type 2 MNAR

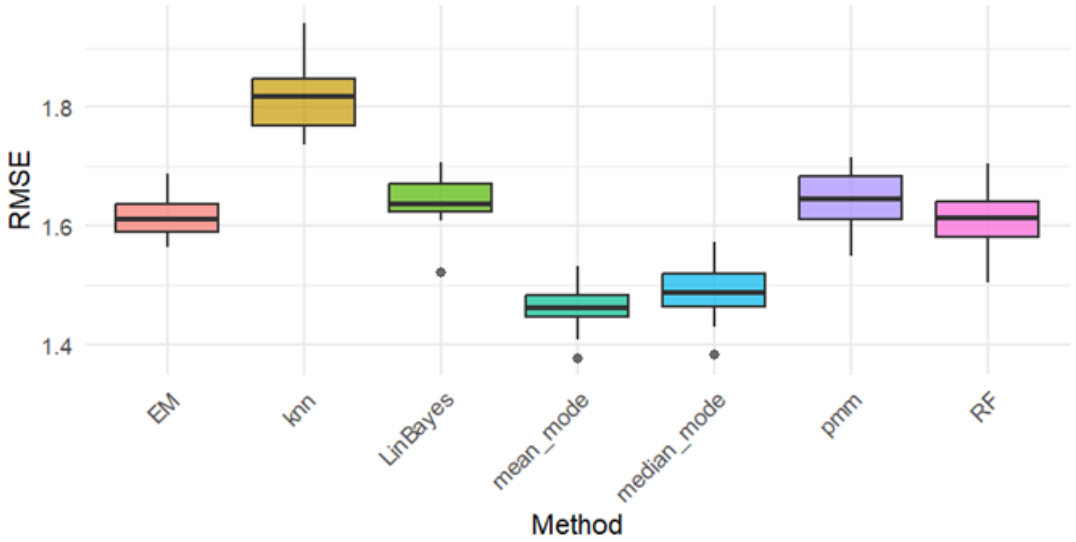

Figure A.22: IQR graph of RMSE values for the Two Factor 1 variable in the Simulated dataset under Type 2 MNAR

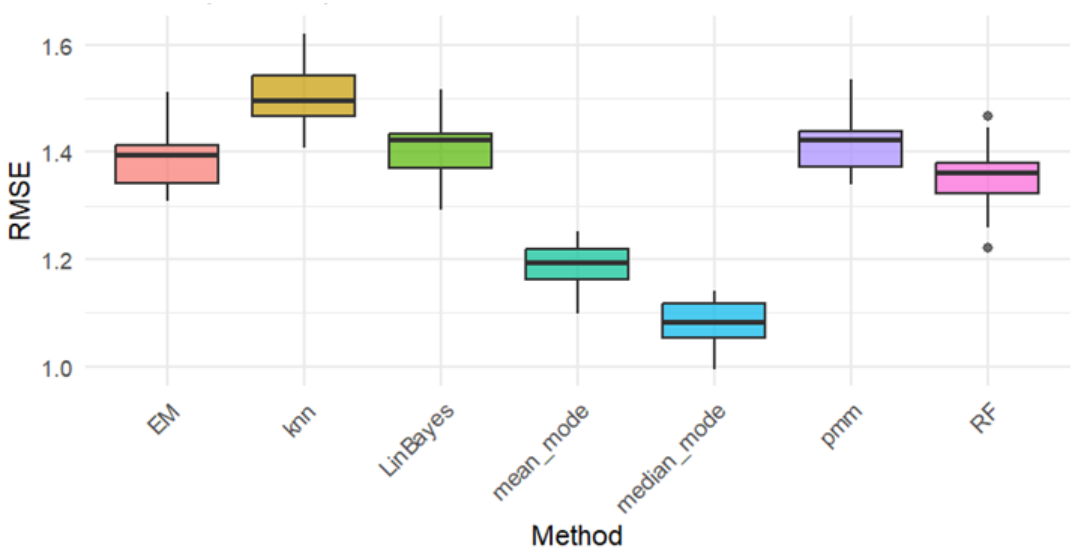

Figure A.23: IQR graph of RMSE values for the Two Factor 1 variable in the Skewed dataset under Type 2 MNAR

A.4.2 PCP

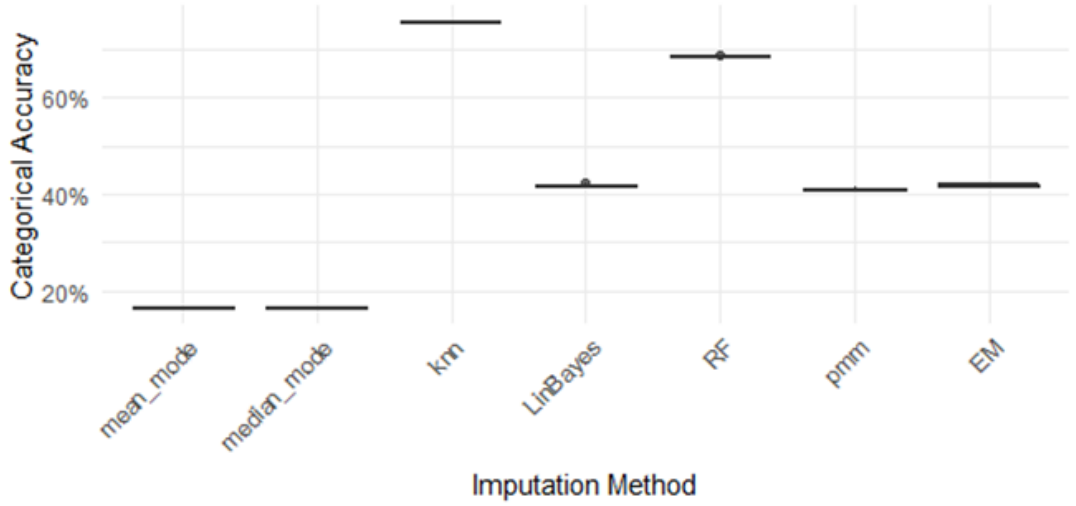

Figure A.24: IQR graph of PCP values for the Maritalstatus variable in the Adult dataset under Type 2 MNAR

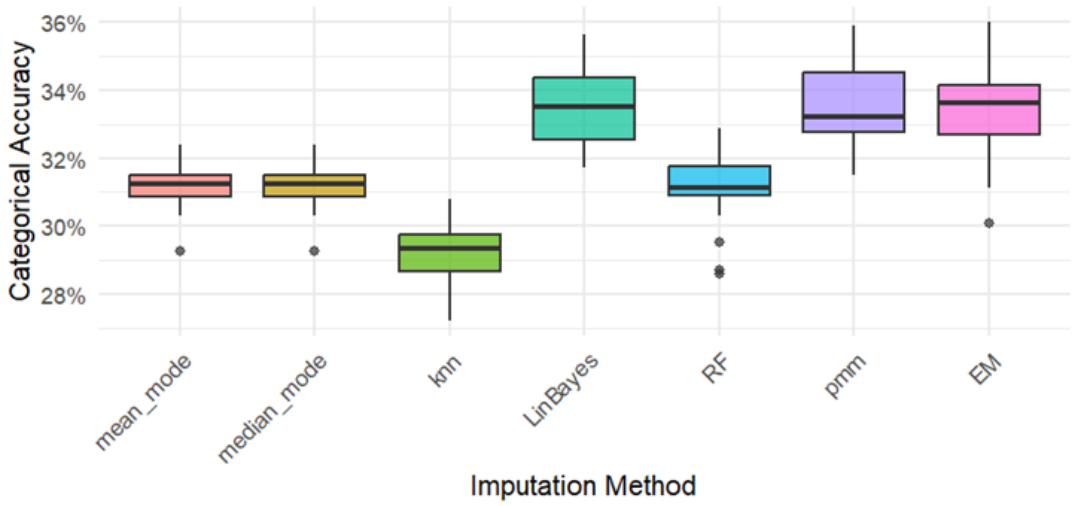

Figure A.25: IQR graph of PCP values for the Persons variable in the Car dataset under Type 2 MNAR

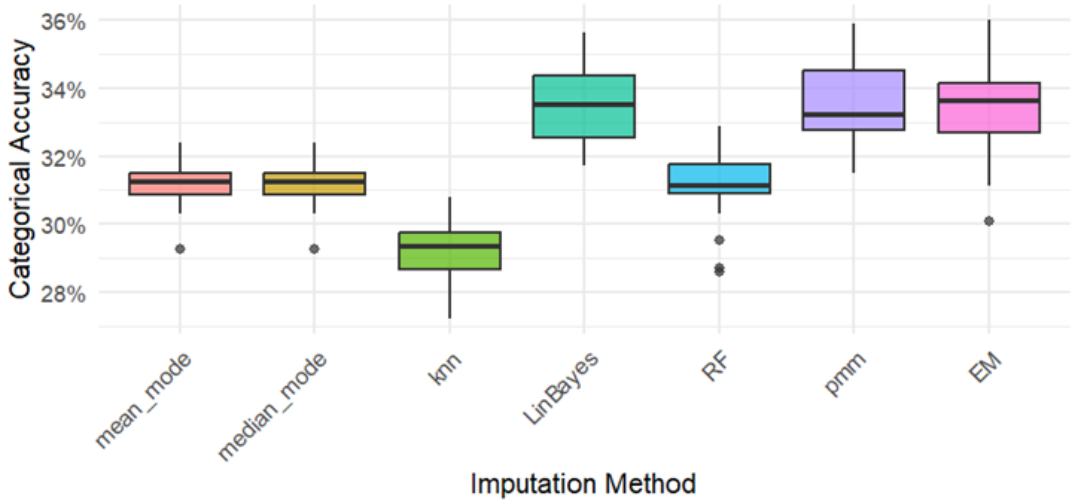

Figure A.26: IQR graph of PCP values for the Safety variable in the Car dataset under Type 2 MNAR

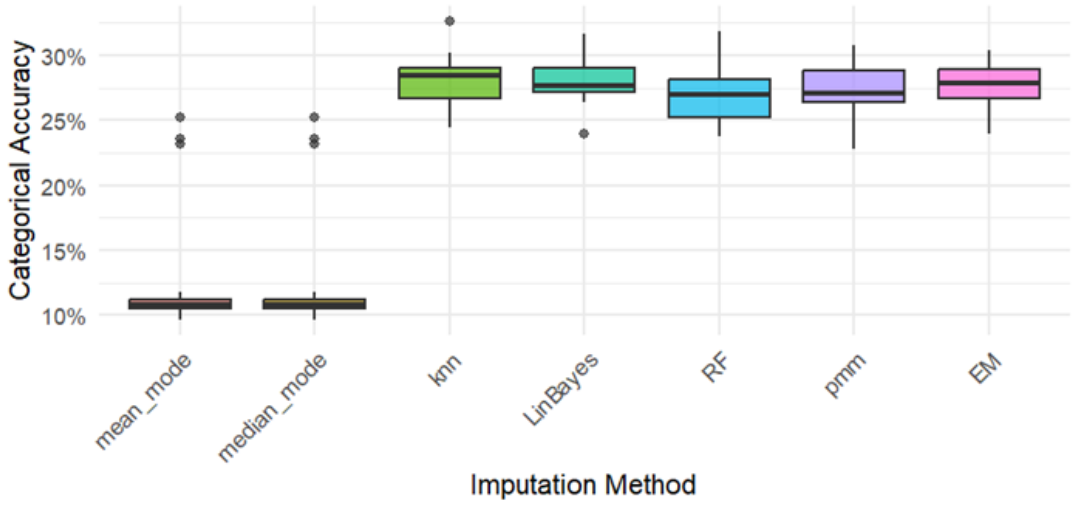

Figure A.27: IQR graph of PCP values for the Cat1 variable in the Simulated dataset under Type 2 MNAR

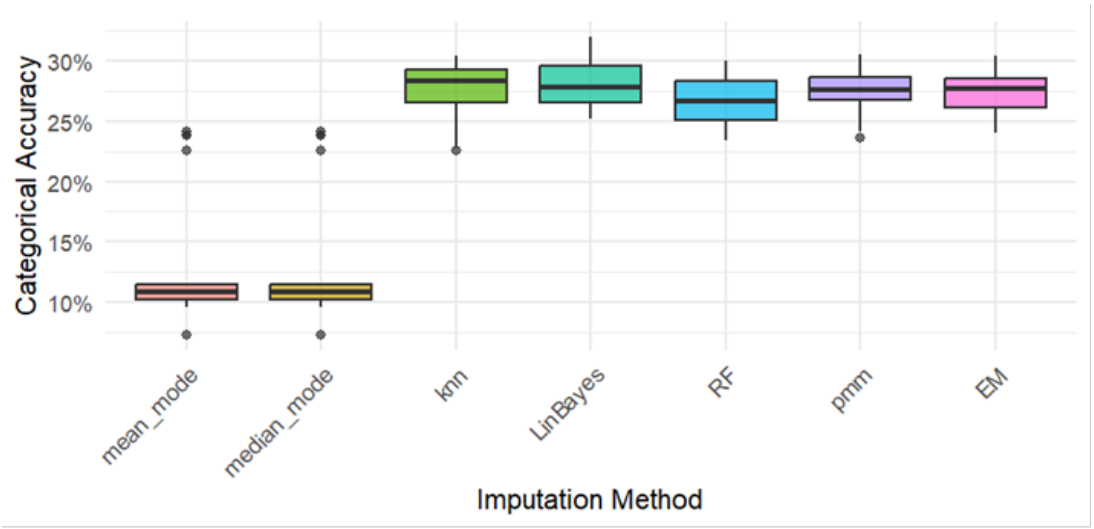

Figure A.28: IQR graph of PCP values for the Cat1 variable in the Skewed dataset under Type 2 MNAR

A.5 Computation times

Table A.3: Computation time (in seconds) for different imputation methods across various dataset sizes. Times are for a single imputation.

Method	**Iris**	**Simulated**	**Skewed**	**Banknotes**	**Car**	**Adult**
#Obs.	**150**	**1.000**	**1.000**	**1.372**	**1.728**	**32.561**
Mean/Mode	0.003	0.007	0.010	0.003	0.003	0.033
Median/Mode	0.003	0.007	0.011	0.003	0.003	0.057
k-NN	0.07	0.85	1.39	0.89	1.08	452.75
LinBayes	0.33	1.25	2.00	0.45	0.45	12.92
RF	1.88	6.82	10.5	5.26	5.20	112.45
PMM	0.37	1.32	2.09	0.60	0.54	15.83
EM	0.04	0.18	0.265	0.12	0.15	3.87

A.6 Correlation and frequency graphs

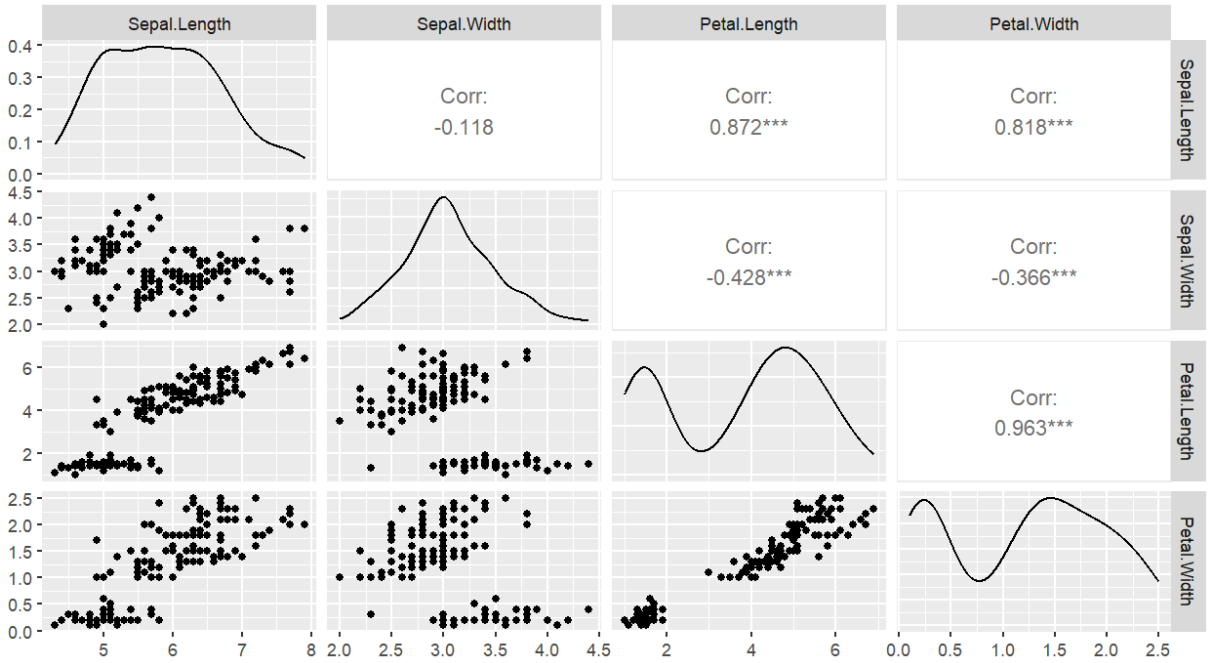

Figure A.29: Linear relationship between variables in Iris dataset. Upper right triangle shows correlation coefficients, with *** denoting significance level. Diagonal shows density plots, and bottom left shows correlation plots.

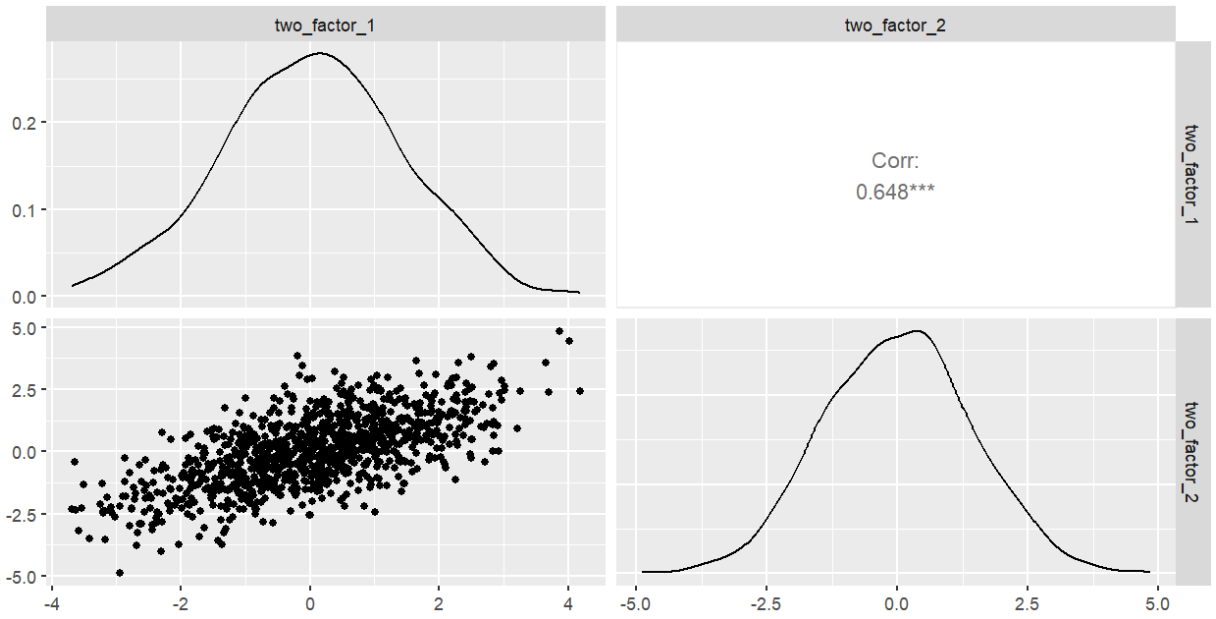

Figure A.30: Correlation between first two variables of Simulated dataset.

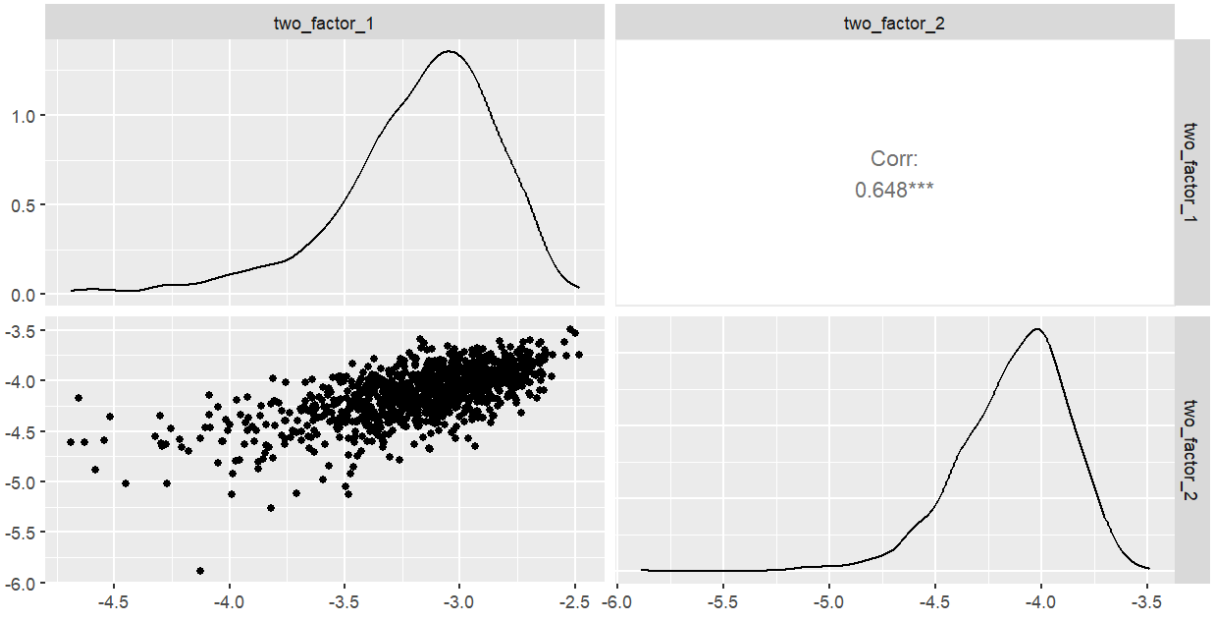

Figure A.31: Correlation between first two variables of Skewed dataset.

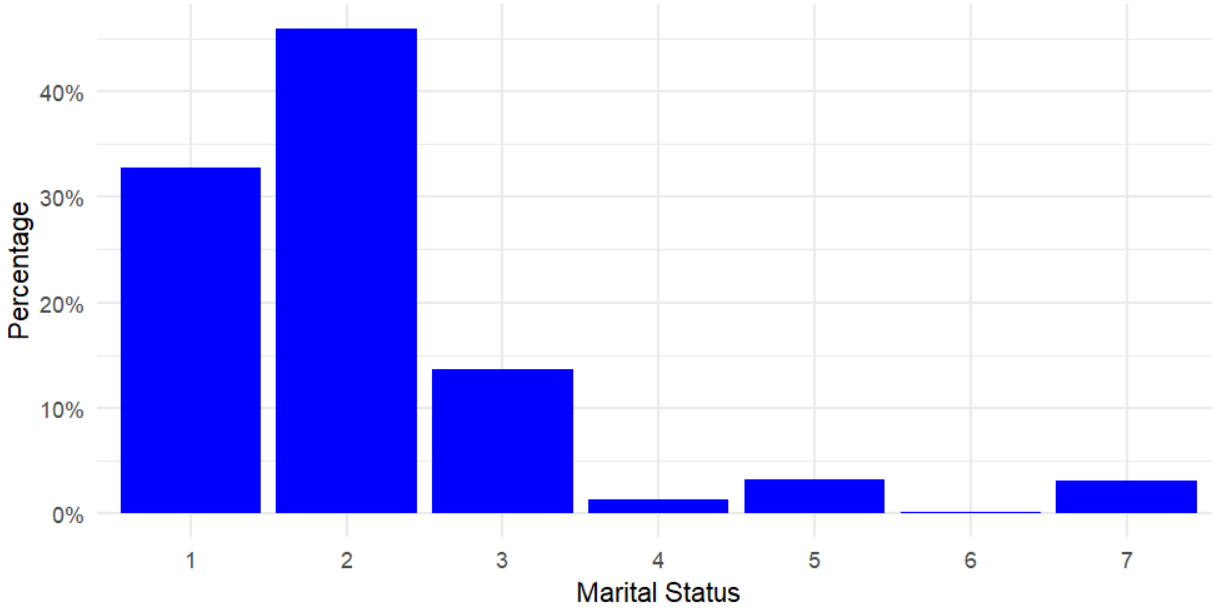

Figure A.32: Distribution of values for Marital Status in the Adult dataset

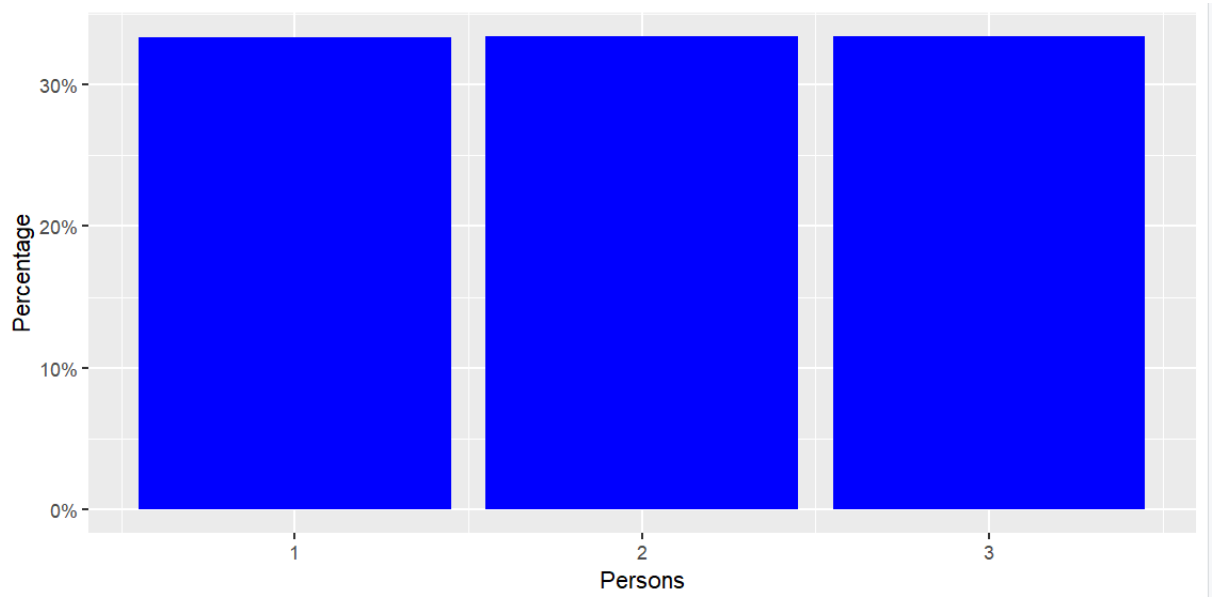

Figure A.33: Distribution of values for Persons in the Car dataset

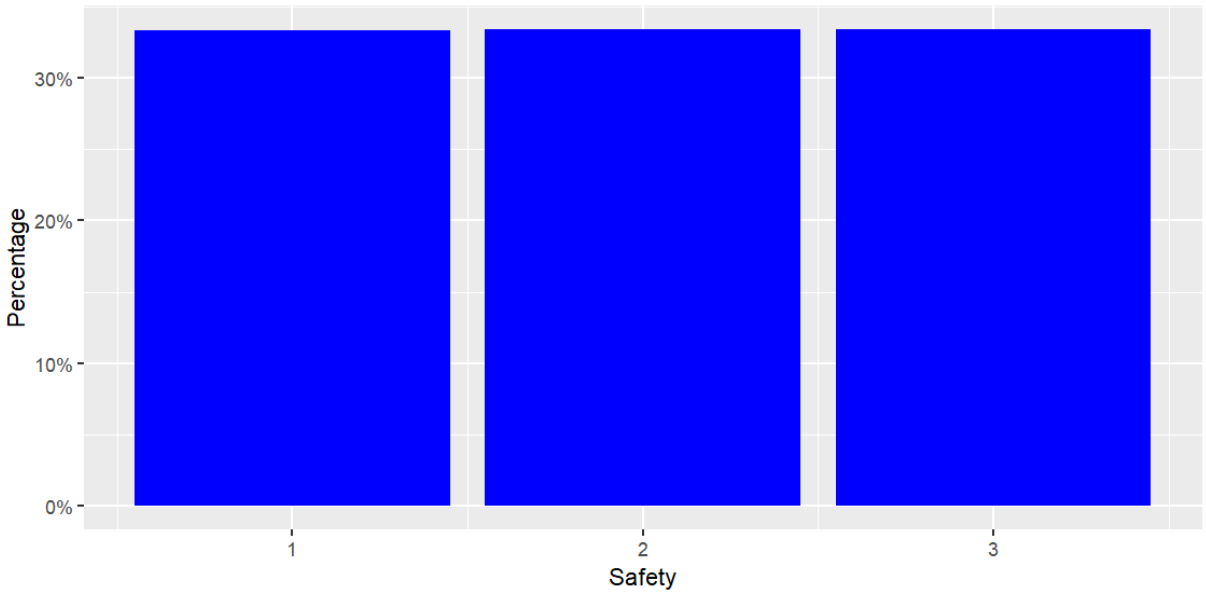

Figure A.34: Distribution of values for Safety in the Car dataset

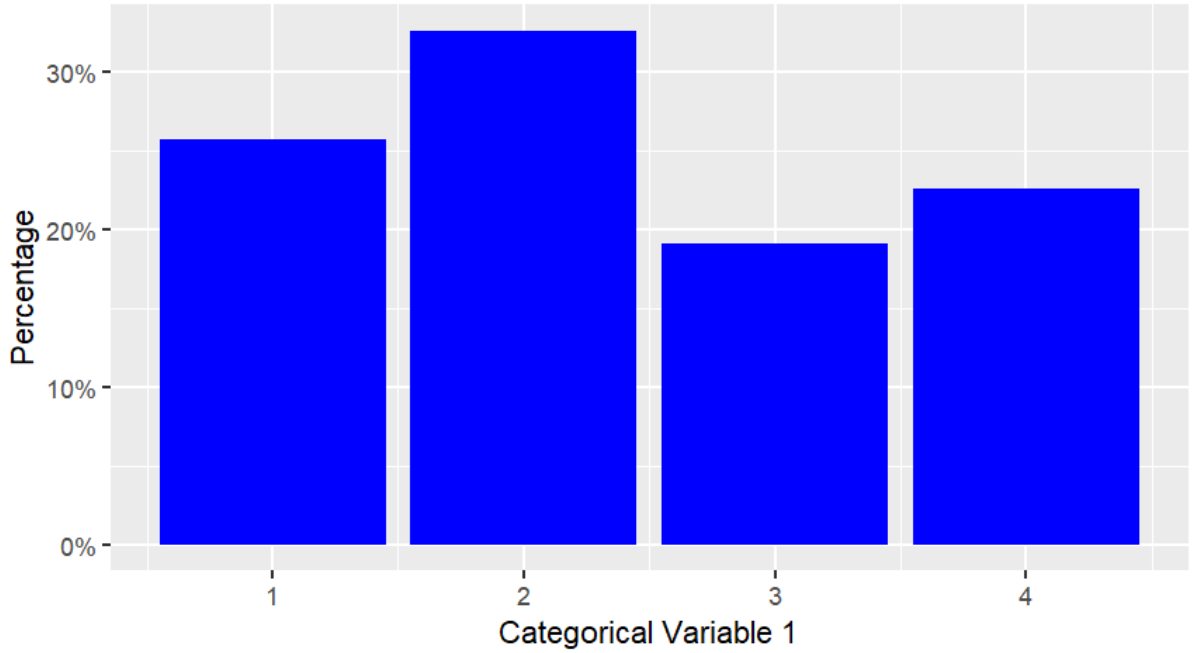

Figure A.35: Distribution for Cat1 in the Simulated and Skewed datasets

B. Programming code

Algorithm 1 MNAR 2 Mechanism

Input: data, dataframe_name, var_to_impute, correlated_var, proportion
Output: data with MNAR 2 missingness
set.seed(NULL) {For randomness}
$n \leftarrow$ nrow(data)
$num\_missing \leftarrow$ round($n *$ proportion)
$cat\_var\_indices \leftarrow$ get_cat_var_and_target(dataframe_name) {Returns indices of all cat. var}
$is\_categorical \leftarrow$ which(names(data) == correlated_var) $\in cat\_var\_indices$
if !is_categorical **then**
    $expanded\_proportion \leftarrow \min(1, 1.5 *$ proportion)
    $threshold \leftarrow$ quantile(data[[correlated_var]], probs = 1 - expanded_proportion)
    $candidate\_indices \leftarrow$ which(data[[correlated_var]] $\geq$ threshold)
    $missing\_indices \leftarrow$ sample(candidate_indices, size = num_missing, replace = FALSE)
else
    if is binary **then**
        $high\_prob \leftarrow$ proportion $* 1.9$
        $low\_prob \leftarrow$ proportion $* 0.2$
        $prob\_missing \leftarrow$ ifelse($data[[correlated\_var]] == 1, high\_prob, low\_prob$)
        $is\_missing \leftarrow$ runif(n) $< prob\_missing$
    else
        $prob\_missing \leftarrow data[[correlated\_var]]/\max(data[[correlated\_var]])$
        $prob\_missing \leftarrow$ prob_missing $*$ proportion
        $is\_missing \leftarrow$ runif(n) $< prob\_missing$
    end if
    $missing\_indices \leftarrow$ which($is\_missing$)
    if length(missing_indices) $\leq$ num_missing **then**
        $missing\_indices \leftarrow$ sample($missing\_indices$, size = $num\_missing$)
    end if
    $actual\_num\_missing \leftarrow$ length($missing\_indices$)
    if actual_num_missing $\leq$ num_missing **then**
        Sample from not yet selected observations until enough missing
    end if
end if
$data[missing\_indices, var\_to\_impute] \leftarrow NA$
$data[[correlated\_var]] \leftarrow$ runif($nrow(data), \min = -0.0001, \max = 0.0001$)
return data

Note: Algorithm works well for proportion $\leq$ 50%. For higher proportions missing, algorithm will work as MCAR when correlated variable is continuous.