The Single Imputation Technique in the Gaussian Mixture Model Framework

THE SINGLE IMPUTATION TECHNIQUE IN THE GAUSSIAN MIXTURE MODEL FRAMEWORK

B.M.J AISYAH

PhD

2018

The Single Imputation Technique in the Gaussian Mixture Model Framework

Binti Mat Jasin AISYAH

Submitted for the Degree of
Doctor of Philosophy

Faculty of Engineering and Informatics
University of Bradford

2018

Abstract

Binti Mat Jasin Aisyah
The Single Imputation Technique in the Gaussian Mixture Model Framework
Keywords: Missing data imputation, Gaussian mixture model, Wild bootstrap resampling, Predictive mean matching

Missing data is a common issue in data analysis. Numerous techniques have been proposed to deal with the missing data problem. Imputation is the most popular strategy for handling the missing data. Imputation for data analysis is the process to replace the missing values with any plausible values. Two most frequent imputation techniques cited in literature are the single imputation and the multiple imputation.

The multiple imputation, also known as the golden imputation technique, has been proposed by Rubin in 1987 to address the missing data. However, the inconsistency is the major problem in the multiple imputation technique. The single imputation is less popular in missing data research due to bias and less variability issues. One of the solutions to improve the single imputation technique in the basic regression model: the main motivation is that, the residual is added to improve the bias and variability. The residual is drawn by normal distribution assumption with a mean of 0, and the variance is equal to the residual variance. Although new methods in the single imputation technique, such as stochastic regression model, and hot deck imputation, might be able to improve the variability and bias issues, the single imputation techniques suffer with the uncertainty that may underestimate the R-square or standard error in the analysis results.

The research reported in this thesis provides two imputation solutions for the single imputation technique. In the first imputation procedure, the wild bootstrap is proposed to improve the uncertainty for the residual variance in the regression model. In the second solution, the predictive mean matching (PMM) is enhanced, where the regression model is taking the main role to

generate the recipient values while the observations in the donors are taken from the observed values. Then the missing values are imputed by randomly drawing one of the observations in the donor pool. The size of the donor pool is significant to determine the quality of the imputed values. The fixed size of donor is used to be employed in many existing research works with PMM imputation technique, but might not be appropriate in certain circumstance such as when the data distribution has high density region. Instead of using the fixed size of donor pool, the proposed method applies the radius-based solution to determine the size of donor pool. Both proposed imputation procedures will be combined with the Gaussian mixture model framework to preserve the original data distribution.

The results reported in the thesis from the experiments on benchmark and artificial data sets confirm improvement for further data analysis. The proposed approaches are therefore worthwhile to be considered for further investigation and experiments.

Declaration

The candidate confirms that the work submitted is his own and that appropriate credit has been given where reference has been made to the work of others.

Binti Mat Jasin AISYAH

Acknowledgements

Thanks to my supervisors, families and friends.

Table of Contents

List of Figures

List of Tables

Glossary

MI	Multiple Imputation
EM	Expectation Maximization
SEM	Structure Equation Model
GMM	Gaussian mixture model
OLS	Ordinary Least Square
PMM	Predictive Mean Matching
MCAR	Missing completely at random
MAR	Missing at random
MNAR	Missing not at random
CC	Complete case
ML	Maximum Likelihood
MLE	Maximum Likelihood Estimation
E-step	Expectation-step
M-step	Maximization-step
PDF	Probability Density Function
EMB	Expectation Maximisation with Bootstrapping
CRVE	Cluster Robust Variance Estimator
2D	2 Dimensions
MAPE	Mean Absolute Percentage Error
MAE	Mean Absolute Error
RMSE	Root Mean Square Error
MDP	Missing Data Percentage
MICE	Multivariate Imputation by Chained Equations
Hmisc	Harrell Miscellaneous
BaBooN	Bayesian Bootstrap Predictive Mean Matching
MIDAS	Multiple Imputation Using Distance-Aided Selection of Donors
MIDAStouch	Touch-up version of MIDAS
NN	Nearest Neighbour
ABB	Approximate Bayesian Bootstrap

SPSS	Statistical Package for the Social Sciences
SAS	Statistical Analysis System
KDTree	K Dimensional Tree
i.i.d	Independent and Identically Distributed
PMMr	Predictive Mean Matching by Radius

Publication

Mat Jasin A., Neagu D., Csenki A. (2018) The Wild Bootstrap Resampling in Regression Imputation Algorithm with a Gaussian Mixture Model. In: Perner P. (eds) Machine Learning and Data Mining in Pattern Recognition. MLDM 2018. Lecture Notes in Computer Science, vol 10935. Springer, Cham

List of notations

D	Notation for a donor pool
d	Size of a donor pool D
δ_{max}	Maximum distance
$\lvert\delta_{ij}\rvert$	Distance between observed data and central data points
ε_i	A noise/error term generated from random error or residual
$\hat{y}_i = \beta_0 + \sum_{j=1}^{P} \beta_j x_{ij} + \varepsilon_i$	Simple linear regression model for an estimated a dependent variable
$\hat{\varepsilon}_i = y_i - \hat{y}_i$	A residual value from an observed value and estimated value
k	A counter value for number of K Gaussian component
K	A total number of Gaussian components
π_1, π_2	The mixture weight / mixture coefficient as a prior probability
$\boldsymbol{\mu}$	A vector of mean values
$\boldsymbol{\Sigma}$	A vector of covariance matrices
$\mathbf{x}_l$	A vector of dependent variable
$\mathbf{X}$	A data with ($N \times P$) matrix
$\mathbf{X}^1$	A data $N \times (P + 1)$ matrix where 1 represents a row vector of ones
$\mathbf{X}_P$	A dependent variables with ($N \times P$) matrix
N	A total number of observations or samples
P	A total number of variables / dimensions / column vectors
$x_{ij} = (x_{i1}, \dots, x_{iP})$	A row in data matrix with P variables
x_{ij}	An element in a data matrix $\mathbf{X}$
i, j	Counter for N and P where $i = 1, \dots, N$ and $j = 1, \dots, P$
M	A missing data indicator. The value is 1 or 0
$\mathbf{X}^C = (\mathbf{X}^O, \mathbf{X}^M)$	A complete data contains 2 components $\mathbf{X}_{obs}, \mathbf{X}_{miss}$.
$\mathbf{X}^O$	An observed data
$\mathbf{X}^M$	The variable(s) contain missing elements
T	Superscript T denotes the transpose of a matrix or vector
$\boldsymbol{\theta}$	Unknown parameter where $\boldsymbol{\theta}$ consists of mean vector $\boldsymbol{\mu}$ and covariance matrices $\boldsymbol{\Sigma}$
$P(S \mid \mathbf{X}, \boldsymbol{\theta})$	The probability of missingness S conditioned on data matrix $\mathbf{X}$ and unknown parameter θ
n_1	A size of observed data
n_0	A size of missing data

$\boldsymbol{\beta}$	A P-dimensional vector of regression coefficient
β_0	A intercept value
β_1	A slope
μ_{x_l}	A mean value for dependent variable x_l
μ_x	A mean value for independent variable x_l
$(\mathbf{X}^{1T}\mathbf{X}^1)^{-1}$	An inverse matrix
m	A counter in iteration process in MI technique
$\mathcal{L}(.)$	The likelihood function
$p(.)$	A PDF
t	An iteration counter in EM algorithm
$\boldsymbol{Z}$	A vector of hidden variable
$\hat{\boldsymbol{\tau}}$	A posterior probability of each data points belonging to a k^{th} Gaussian component
π_k	Prior information of the mixing coefficient / weight
$\widehat{\boldsymbol{\beta}} = (\mathbf{X}^{1T}\mathbf{X}^1)^{-1}\mathbf{X}^{1T}\mathbf{x}_l$	The estimated regression coefficient applied in matrix notation
$\mathbf{H} = \mathbf{X}^1(\mathbf{X}^{1T}\mathbf{X}^1)^{-1}\mathbf{X}^{1T}$	The hat matrix
$\mathbf{x}_i^1 = [1 \quad x_{11} \quad \ldots \quad x_{1P}]$	A vector of regressor variables
$w_i = \mathbf{x}_i^{1T}(\mathbf{X}^{1T}\mathbf{X}^1)^{-1}\mathbf{x}_i^1$	The diagonal elements in the Hat matrix
$\hat{v} = \hat{\sigma}^2(\mathbf{X}^{1T}\mathbf{X}^1)^{-1}$	A variance estimator
$t_i^* = a_i$	
$\mathbf{x}_i^{1T}\widehat{\boldsymbol{\beta}} = \beta_0 + \sum_{p=1}^{P}\beta_p x_{ip}$	The fitted regression model

Chapter 1

Introduction

1.1 Research Context

Data quality is one of the most current significant issue which have serious consequences on cost, effectiveness, efficiency and reliability to organizations and business. From a research perspective, data quality is of the utmost importance and has been addressed in the areas including statistics, management, and computer science. The quality of data is measured to achieve the specific standard based on several dimensions: (1) Accuracy and precision, (2) Reliability, (3) Timeliness and currency, (4) Completeness and (5) Consistency. Missing data has been the most common data quality issue since 1970 and has a significant impact on the conclusion of data analysis inference [1].

1.2 Research Problem

Missing data problems are some of the most pervasive and common problems in the medical, finance, social and many more research areas. Some data are missing due to data are not properly collected, being lost, not available or problems occurring in processing steps. Due to that, dealing with missing data is significant as it affects the reliability of analysis results.

There are several techniques for handling missing data problems. The most common approach is eliminating units with missing values. This method is known as complete case analysis or listwise deletion. The main problems of this approach in existing research solutions are directed towards the resulting tendency of substantial bias and loss of precision particularly when the missing data are ignored in data analysis, such as in the cases of unbalanced data classifications [2][3].

The imputation technique is an alternative approach of the complete case analysis method. This technique aims to replace the missing values with plausible values. Little and Rubin [4][5] gave an overview and analysis of the missing data techniques. The significant elements needed to be considered in the imputation procedure [5] are : (1) maximizing the use of observed data as to improve accuracy, performance and minimize error between the actual data and the imputed data, (2) preserving the underlying data distribution, (3) taking into account the variance estimation caused by data imputation approach.

Basically, there are two main types of imputation techniques. The single imputation technique substitutes missing values with mean, median or mode values. This technique is simple and easy to apply but this approach will lead to the biased analysis result [4]. The single imputation techniques such as cold deck, hot deck and regression imputation fail to account the uncertainty and underestimate the variance [2] .The more advanced techniques are multiple imputation (MI) and expectation maximization (EM) [4]. The MI technique [6] is to improve the drawbacks in the single imputation technique. Rubin [7] in 1970's formulated the idea of MI technique in the Bayesian framework that requires random draws from the posterior distribution. This technique will create M different copy of datasets and replace the missing values with different plausible values into the M copied datasets. The analysis in the MI technique is implemented in three different phases. On the other hand, the EM algorithm is introduced by Dempster [8], primarily to find maximum likelihood for parameter estimation of the mean vector and covariance matrices in the presence of missing data. This method is an iterative process where each iteration contains two-step procedures: the e-step and, the m-step. At present, the EM algorithm is extensively employed in the more complex model such as mixture model [9], structure equation model(SEM) [10] and many more in statistics literature.

Several techniques to handle missing values in which appropriately learning the distribution of the data have been discussed in the literature. These include the imputation method based on the multivariate normal distribution introduced by Schafer [3]. The Gaussian mixture model (GMM) [2] is a semi-parametric probabilistic model that has been studied extensively to learn the distribution of dataset.

The purpose of GMM model in this study is to estimate the density of the data. The conditional mean imputation (also known as regression imputation or ordinary least square, OLS) is employed to impute the missing values incorporate with the GMM model. There are two types of regression: (i) deterministic, where the imputed data generated from regression model and fall along the regression line [11]. The outcome of this technique does not portray uncertainty and might cause severe bias [12], (ii) stochastic, where the regression imputation require noise or a normal distributed residual term to be added [13]. This technique may improve variability and produce unbiased parameter estimate. However, it tends to attenuate standard errors [13]. As mentioned by Di Zio [5], the regression technique has the advantage to preserve the data distribution. (iii) random draw is the simple random imputation from probability distribution given the observed values of current sub population. Consequently, this technique causes too much variability on the imputed data [14].

Consistent with data sparsity issue in the stochastic regression imputation, the result will be distorted if the imputed data yielded outside from the observed data range. The Predictive mean matching (PMM) model can explicitly address this concern. Little and Rubin [4] revealed that the PMM is less vulnerable to model misspecification. The PMM method is the combination of parametric and non-parametric imputation methods. Each missing value is replaced by randomly drawn values from a set of candidate donors in a donor pool. The donor pool is created by taking the observed values that closest to the regression-predicted scores. There are several ways to determine the size of a donor pool:

a) Fixed size of the donor pool D
b) fixed δ_{max} [15]
c) probability inversely proportional function $|\delta_{ij}|$ [16] [17] [18].

But it is quite often that the number of candidate donors D are typically fixed and containing either $D = 1,3, 5$ or 10 members [12].

1.3 Research Focus

This research primarily employs the GMM to learn the underlying data distribution. The GMM is unsupervised learning approach in machine learning method.

The random error ε_i in $\hat{y}_i = \beta_0 + \sum_{j=1}^{J} \beta_j x_{ij} + \varepsilon_i$ can be obtained in two ways [19]:

1) by drawing a random error with an underlying assumption that it is independent and identically distributed (i.i.d); and by following a Gaussian distribution with zero mean and finite variance;

2) by drawing a random error with a replacement from the empirical distribution of the estimated residuals $\hat{\varepsilon}_i = y_i - \hat{y}_i$ [20].

Problems can occur in the random error and residual $\hat{\varepsilon}_i$ in method (1) that will create the sparsity problem whereas the random $\hat{\varepsilon}_i$ generation will be either too large or too small although the normality distribution assumption is met. The sparsity of data in a method (1) will be inconsistent if the data distribution has a different cluster and each cluster consists of a different density. The sparsity of data creates some problems such as increases the variance between the imputed and original data. My first aim in this thesis is to preserve of the error distribution $\hat{\varepsilon}_i$ in the conditional meal imputation and to minimize the deviation between predicted missing data (imputed data) and original data.

According to Little and Rubin [4], the PMM method does not require an explicit model to learn the data distribution of the missing values. The PMM itself can impute the missing values within the specific data range. However, there are situations where the data distribution is supposed to be the mixture of k

distribution where $k = 1,2,\dots,K$. In such situations, the density in each distribution have different variances. Therefore, the problem might occur where the selection of the large number of fixed possible donors D (for instance, $d = 10$ where d refers to the size of a donor D as mentioned by Little and Rubin [4]) will not improve bias and increase variances or variability of the imputed data. As the second aim, our research is motivated to improve this bias and variance trade-off issue in regression imputation in the mixture of Gaussian data distribution that have different densities.

1.3.1 Research Questions

There are four main research questions in this thesis:

1. How to minimize the error of the random error distribution on the imputed data?
2. How can the bootstrap resampling procedure in GMM framework be applied?
3. Which number of selection donor in predictive mean matching that appropriate for improving error deviation between imputed and original data and simultaneously preserve data distribution in GMM?
4. How it can be guaranteed that the selection number of a donor will occur within the current distribution?

The research questions (a) and (b) are answered and described in Chapter 3. The research questions (c) and (d) are answered and described in Chapter 4.

1.3.2 Research Objectives

The aim of this research is to study the accuracy of missing data methods applied in the GMM framework. Therefore, to achieve the target, there are five objectives identified for this research:

1) To explore the current methods and issues to handle the missing data problem specifically in the GMM framework.
2) To implement a combination of the bootstrap resampling technique in the GMM model in the context of missing data where both the bootstrap and GMM apply iterative procedures.
3) To compare the implementation of PMM in the data which is has different densities and variances.
4) To compare the implementation of PMM with or without GMM to learn data distribution. This is because, according to Little and Rubin [4] and Buuren [12], the PMM method is the explicit model that less vulnerable to model misspecification. Therefore, this study may investigate the appropriateness of PMM application in GMM framework.
5) To investigate the impact of combination of PMM's implementation in GMM.

1.4 Research Contributions

There are four major contributions of this research:

1) To introduce and demonstrate the combination of the wild bootstrap resampling technique in GMM in handling missing data imputation process.
2) To minimize the error between the original data and predicted missing data by applying the wild bootstrap resampling technique.
3) To introduce the flexible number of candidate members in donor pool applied in the PMM method to impute the missing values.
4) To propose the PMM imputation method with a flexible number of candidates in the donor pool applied in GMM as to maintain the predicted missing values lies in the original data distribution

1.5 Dataset

The experimental work is conducted in the two separate experiments in the Chapter 3 and Chapter 4. The Old Faithful geyser dataset applied in both experimental works in Chapter 3 and Chapter 4. This dataset is recorded by Azzalini and Bowman [21] to measure the waiting time to the next eruption of the geyser found in the Yellowstone National Park . This dataset contains 272 records for the time interval between the next eruptions. It is composed of 2 variables:

i) waiting time between eruptions presented in Y axis and,

ii) the duration of eruptions presented in X axis.

It has two clusters where the eruptions take less than 2.5 minutes in first cluster and take more than 3.5 minutes in the second cluster. The measurement of time and duration values are taken in minutes.

For the artificial case study, the values are randomly sampled with 1000 observations of two Gaussian classes with different position mean values and positive-negative correlation.

The Figure 1.1 below illustrates the random data generated from the parameter values shown in the Table 1.1:

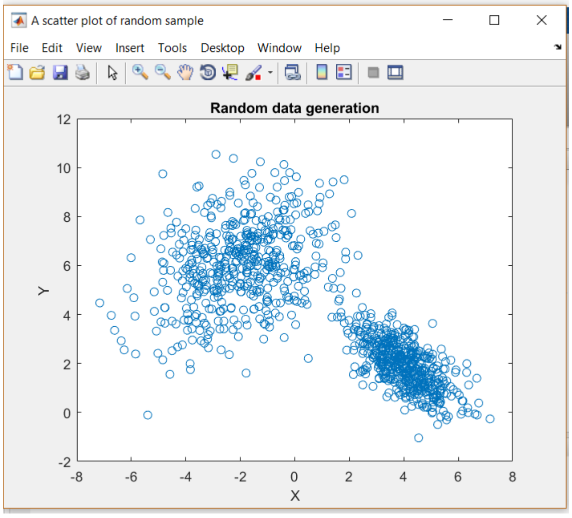


Figure 1.1 : Random data generated for case 1

The second random data is generated with 2-dimensional data and the of size
N = 1000 of two gaussian classes. In this second random normal data, the variance of X and Y are reduced in the second covariance matric to 1.5 values. Therefore, the second simulation study in Chapter 4, two types of random data were employed:

Table 0.1 : The parameters values for two random generation data

Random data generation for Case 1	Random data generation for Case 2
$\pi_1 = 0.5, \ \pi_2 = 0.5$	$\pi_1 = 0.5, \ \pi_2 = 0.5$
$\mu_1 = (4.2)^T, \mu_2 = (-2.6)^T$	$\mu_1 = (4.2)^T, \mu_2 = (-2.6)^T$
$\Sigma_1 = \begin{pmatrix} 1 & -0.7 \\ -0.7 & 1 \end{pmatrix}, \Sigma_2 = \begin{pmatrix} 3 & 0.9 \\ 0.9 & 3 \end{pmatrix}$	$\Sigma_1 = \begin{pmatrix} 1 & -0.7 \\ -0.7 & 1 \end{pmatrix}, \Sigma_2 = \begin{pmatrix} 1.5 & 0.9 \\ 0.9 & 1.5 \end{pmatrix}$

The Figure 1.2 below illustrates the random data generated from the second parameter values shown in the Table 1.1:

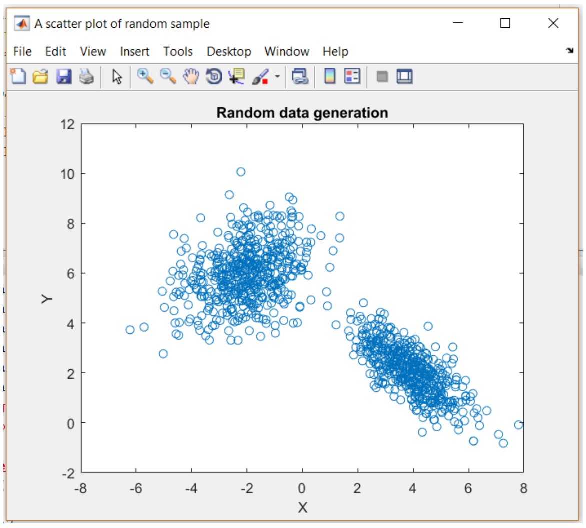


Figure 1.2 : Random data generated for case 2

1.6 Thesis Structure

This thesis contains five chapters.

The Chapter 1 introduces the research context, research problem, research focus, research questions, research objectives and research contributions. This chapter also describes about the data sets being used in the experimental works in the following chapter 3 and chapter 4.

In Chapter 2, the author introduces the background information for understanding the fundamental concept of the missing data mechanisms, missing data pattern and a data set containing missing data structure. We also present a few classical and advance theoretical methods in handling the missing data issue. Due to the fact that the GMM framework is utilised in this research work to learn the underlying gaussian data distribution, the GMM model framework theory is presented as well.

Chapter 3 demonstrates the imputation technique that can be used to minimize the error between original and predicted missing values occur in the single dependent variable $\mathbf{x}_l$. Through this experimental work, the proposed single imputation technique in GMM framework is compared with the well-known multiple imputation technique implemented in the R-package AMELIA II.

In Chapter 4, we consider that our proposed size of donor pool is based on flexible values whereby most of the existing research works applied $D = 1,3,5,10$. One of these candidates D will be matched by a random sample to replace each missing value applied in the PMM imputation technique. We demonstrate and compare the two fixed-radius values $r =$ 0.5 and $r =$ 0.7 used in the experimental work. We also compare the proposed single imputation technique with a few multiple imputation techniques in R packages software. The purpose of this experimental study is to compare the accuracy and performance between the original data and the imputed data.

During the implementation of this research works, we identified advantage and disadvantages of the proposed methods. We also outline several recommendations as a guideline in future research works. All these constraints and recommendation are elaborated in Chapter 5.

Chapter 2

Background

2.1 Introduction

This chapter presents the fundamental concepts of missing data including missing data pattern, missing data mechanisms, missing data methods especially single imputation technique and few other advance missing data imputation techniques. The Gaussian mixture model framework also introduced as to learn the underlying data distribution and parameter optimization through the maximum likelihood estimation and expectation maximization algorithm.

2.2 Missing data patterns.

Missing data pattern is a primary concern in missing data research. The pattern of missingness describes the position of missing values and observed values in a dataset. There are several types of missing data patterns such as (1) univariate pattern, (2) unit non-response pattern, (3) monotone pattern, (4) general pattern, (5) planned missing pattern and (6) latent variable pattern [13]. There are three general missing data patterns called univariate, monotone and arbitrary are commonly discussed in the missing data literature. In this research context, the univariate missing data pattern is applied.

Let $\mathbf{X}$ denote is the $(N \times P)$ data matrix with $i = 1, \ldots, N$ units of the samples on $p = 1, \ldots, P$ variables where x_{ip} is a single element of a variable $\mathbf{X}_p$. A dichotomous variable $\mathbf{S}$ is defined as a missing data indicator. For instance, $s_{ip} = 0$ if x_{ip} is missing and $s_{ip} = 1$ if x_{ip} is observe. Only two missing data patterns are considered in this research work: the univariate missing data pattern and the general data pattern.
Figures 2,1, 2.2 and 2.3 demonstrate the most common missing data patterns where the blocks representing variables contain a sub-block shaded with blue colour to indicate that the variable has missing data. Otherwise, the variables

shaded with white colour are called observed data and will be used as predictors.

The first missing data pattern is known as a univariate missing data pattern. Generally, the pattern is defined as univariate if only one variable contains missing data. The example shows a data matrix $\mathbf{X}$ containing variables $\mathbf{X}_1$, $\mathbf{X}_2$ and $\mathbf{X}_3$. The variables $\mathbf{X}_1$ and $\mathbf{X}_2$ are fully observed while the single variable $\mathbf{X}_3$ contains missing values [4]. The example of univariate missing data pattern is shown in Figure 2.1. Although this type of missing data pattern is relatively rare [22], but it is still researched, for example the multiple imputation by chains equation (MICE) specified the univariate imputation model in [23].

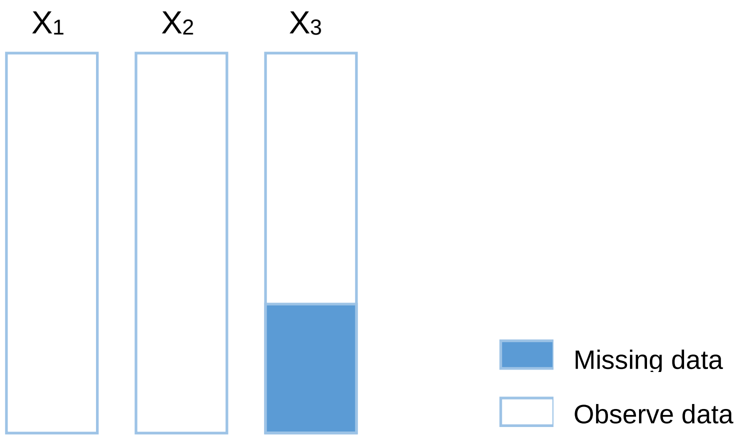


Figure 2.1 : The univariate missing data pattern

The monotone missing data pattern is a part of univariate pattern and typically occurs in longitudinal studies for clinical trials. It is called a monotone pattern when a single element is missing in $\mathbf{X}_p$, then the missing elements increased in each of the subsequent variables $\mathbf{X}_{p+1}, \mathbf{X}_{p+2}, \ldots, \mathbf{X}_P$. The example is shown in Figure 2.2.

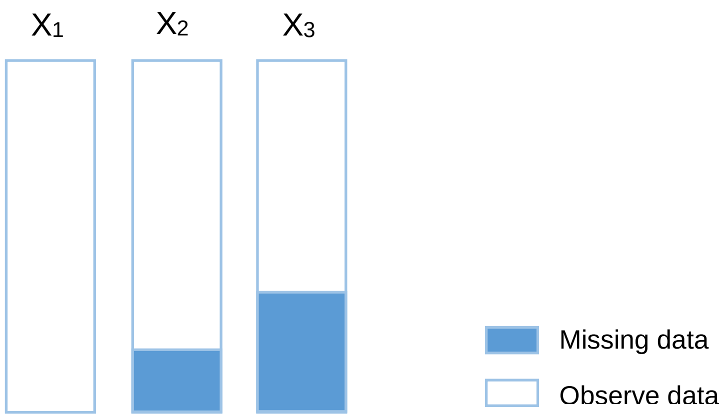


Figure 2.2: The monotone missing data pattern

The general pattern is also known as arbitrary pattern [1]. As seen in Figure 2.3, the missing elements occur in all variables $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_P$ in a random style and disperse in the data structure. Although this type of missing data pattern is seemingly deceptive [22], there are research works that focus on the arbitrary data pattern [24].

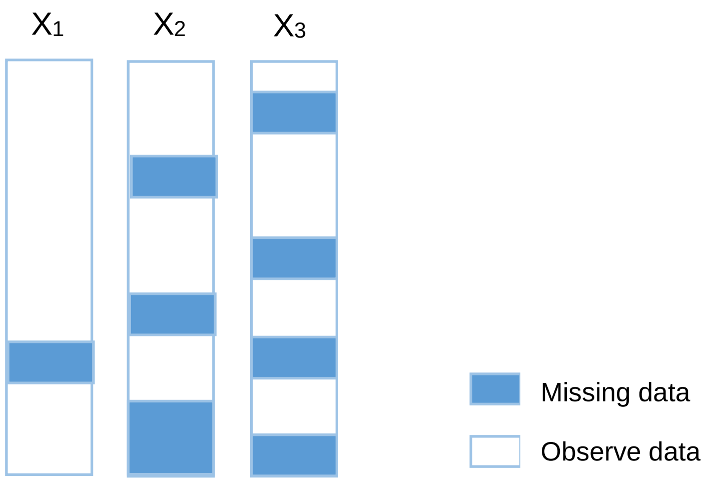


Figure 2.3 : The general missing data pattern

The missing data pattern is important to identify the location of missing data in a data set while the reason for missingness will be discussed in the missing data mechanism in which it is significant to decide on how to handle the missing data.

The next aspect in missing data research is the missing data mechanism will be discussed in the next sub-section.

2.3 Missing data mechanism.

The missing data mechanisms are important to describe the underlying concept and reason of missing data occurrence. These mechanisms follow the probabilistic-based theory and were initiated by Rubin in 1976 [7]. The aims of the mechanisms are to define precisely the key reason of missing problems and the relationship on the observed variables using a set of random variables with joint probability distribution [4]. The missing data mechanisms basically has three categories: missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR).

Let $\mathbf{X}$ denote a data matrix that contains missing values. The generic notation of a complete data matrix is denoted as $\mathbf{X}^C = (\mathbf{X}^O, \mathbf{X}^M)$ where $\mathbf{X}^C$ is a complete data matrix decomposed into $\mathbf{X}^O$ and $\mathbf{X}^M$,denoted as observed variables or complete entry and missing variables respectively . $\mathbf{S}$ denotes a missing data indicator to indicate that the current data on the specific variable is missing or complete. For instance, let $x_i = (x_1, \dots, x_N)^T$ represents the values of a random variable of N observations. Let $s_i = (s_{1p}, \dots, s_{NP})$ be a $(N \text{ x } P)$ data matrix as $\mathbf{X}$. It represents the indicator variable of N observations to indicate that the $s_{ip} = 0$ if the current x_{ip} is observed or $s_{ip} = 1$ if the current x_{ip} is missing. These mechanisms will apply the conditional distribution such as $P(S|\mathbf{X}, \theta)$ where θ is unknown parameters. There are three main missing data mechanisms describe by Little and Rubin[4].

2.3.1 MCAR

The missing completely at random (MCAR) is defined if the probability of missingness neither depends on any observed variable $\mathbf{X}^O$ nor missing variable $\mathbf{X}^M$. The MCAR assumption is also defined as a random sample of $\mathbf{X}^C$. Therefore, removing the small percentage of missing data will not lead to a biased result but the standard error (SE) will be increased since the size of data decreased[25]. The example of missing data occurring due to MCAR is that of the participant ignoring a few questions in a survey form due to unrelated reasons[26].
The notation of MCAR is as follows [1]:

$$p(S|\mathbf{X}^C, \boldsymbol{\theta}) = p(S|\boldsymbol{\theta}) \quad (2.1)$$

2.3.2 MAR

The missing at random (MAR) is in the case of missing data occur in $\mathbf{X}$ do not depend on the missing variable $\mathbf{X}^{M}$ or itself but the probability of missingness depends to the observed variable $\mathbf{X}^{O}$. Schafer and Graham in [1] provide a good example of the missing data with MAR mechanism. Systolic blood pressure data recorded in January (X) and February (Y). The data is MAR due to the date recorded if the measurement data recorded in January exceeded 140 (X > 140). This implies that the missing data occur based on observed data recorded in January (X). The notation of MAR is as follow [1]:

$$p(S|\mathbf{X}^{C}) = p(S|\mathbf{X}^{O}, \theta) \tag{2.2}$$

2.3.3 MNAR

The missingness is regarded to be missing not at random (MNAR), occurs if the probability of missing data depends on observed data $\mathbf{X}^{O}$ and the value of missing variable $\boldsymbol{X}^{M}$ itself where the information may not have existed [26]. Based on the example described in [22], the missing values in the job performance rating variable are because a number of individual's records of the job performance rate itself is low and unemployed. The notation of MNAR is as follow [1], [22] :

$$p(S|\mathbf{X}^{C}) = p(S|\mathbf{X}^{O}, \boldsymbol{X}^{M}, \theta) \tag{2.3}$$

The missing data mechanism concept of MCAR, MAR and MNAR is key aspect to be considered in missing data research because the reason for missingness will give an effect to the mean and standard deviation on the predicted values. The missing data pattern, missing data mechanism and missing data proportion are the important aspect need to consider in missing data research prior to deal with imputation procedure in missing data prediction[27].

2.4 Missing data Structure.

Suppose the data set $\mathbf{X}$ has N units of independent and identically distributed (i.i.d) data points with P variables can be written as follows:

Observation (index)	$\mathbf{X}_1$	$\mathbf{X}_2$	..	$\mathbf{X}_l$	..	$\mathbf{X}_P$
1	x_{11}^O	x_{12}^O	..	x_{1l}^O	..	x_{1P}^O
..	:	:	..	..	..	:
n_1	$x_{n_1 1}^O$	$x_{n_1 2}^O$	..	$x_{n_1 l}^O$	..	$x_{n_1 P}^O$
$n_1 + 1$	$x_{(n_1+1)1}^M$	$x_{(n_1+1)2}^M$	..	NA	..	$x_{(n_1+1)P}^M$
:	:	:	..	..	..	:
N	x_{N1}^M	x_{N2}^M	..	NA	..	x_{NP}^M

Figure 2.2 : A table structure with missing values

Figure 2.3 illustrates a data set that contains missing values (highlighted with NA in the relevant cells). Let $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_l, .., \mathbf{X}_P\}$ be the random variable of the $N \times P$ data matrix where $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_P$ are the dependent (or predictor) variables and completely observed while to clearly differentiate between the independent variables and dependent variable, let $\mathbf{x}_l$ be the dependent (or response) variable. The dependent variable is also known as a response variable $\mathbf{x}_l$, be the size of $n_1 \times 1$ vector contain observed data and n_0 be the size of missing values in $\mathbf{x}_l$. In the imputation process, Rao and Shao [28] suggested creating a set of respondents $\mathbf{X}^O$ and a set of non-respondents $\mathbf{X}^M$ separately. The variable $\mathbf{X}^O$ denotes the $n_1 \times P$ data matrix where n_1 is the size of observed data while $\mathbf{X}^M$ denotes the $n_0 \times P$ matrix where $n_0 = N - n_1$ is the number of missing values. Let $\mathbf{x}_l$ be the size of $n_1 \times 1$ vector contain observed data and n_0 be the size of missing values in $\mathbf{x}_l$.

2.5 Missing data imputation methods

Missing data are ubiquitous and often discovered in the pre-processing of data analysis. Dealing with missing data is a significant and difficult task because it will cause a significant effect to any statistical analysis on prediction. There are several techniques in handling the missing data problem in both traditional and advance methods. There are two main categories of the missing data imputation namely single imputation and multiple imputation. The single imputation predicts each missing element by a single value such as complete case analysis (CC), mean imputation, regression imputation and stochastic regression imputation. The multiple imputation (MI) is a method that creates a several copies of imputed dataset and yield several parameter estimates. These parameter estimates are combined to generate a single inference.

2.5.1 Complete case analysis

Complete case (CC) analysis refers to the method that removes the incomplete data points or records from a data sets and the analysis only runs on a complete data set. This method is effective if the proportion of missing data is small. This method is easy to implement but it distorts the parameters estimated in the statistical results such mean and standard deviation [13]. If the missing data proportion is small and missing data is under MCAR case, the complete case causes unbiased result. However, according to Schafer and Graham [1], the statistical result will performs badly if the data is under MAR case. Due to the fact that discard observations are not recommended and have a serious impact on the statistical result, the mean imputation method is proposed to improve some drawbacks in complete case analysis method.

2.5.2 Mean imputation

The mean imputation or unconditional mean imputation performs better than discarding missing observations in complete case analysis method. Prior to analysis, the missing observations will be filled in with either mean, median or mode values estimated from the non-missing observations or observed variables. Although the unconditional mean imputation is a simple a method, but it might contribute to a biased estimate of the mean, standard deviation, correlation between variables. It is also may not preserve the original data distribution. Regardless of a MCAR or MAR case, it severely distorts the parameter estimation in an analysis result.

2.5.3 Regression imputation

Regression imputation is an imputation method to preserve the correlation between observe and missing variables. The regression imputation is also known as the conditional mean imputation. The imputed values are obtained by fitting a regression model. Identify the outcome variable(s) as the dependent variable(s) while the rest as the independent variables. The independent variable(s) is also known as a predictor because the independent variable(s) will influence the output of dependent variable(s). The complete case analysis is applied to obtain the observed component and estimate the regression coefficient to fit a regression model. In the next step, the missing values in the dependent variable(s) are predicted from a fitted linear regression model. The main constraint of this imputation procedure is that the imputed data might suffer with a bias in estimated values and lack residual variation due to the imputed values lying on the regression line. Consequently, this model might yields high correlation in the r-square result regardless of whether the data is MCAR or MAR [13]. If this model is combined with another iterative model, it might affect to account for the uncertainty of the parameters estimation in the next iterations. The rest of the notations for the regression model are presented by following the table structure illustrated in Figure 2.3 [29].

Based on notations described in the section 2.4, x_l is defined as the response variable while x_p is the predictor variable. Generally, the basic linear regression model is [30]:

$$x_{li} = \beta_0 + \sum_{p=1}^{P} \beta_p x_{ip} + \varepsilon_i \qquad (2.3)$$

where β_0 is the intercept value, β_p is the slope value of $p = 1, \ldots, \mathrm{P}$ number of independent variables and $\varepsilon_i, i = 1, \ldots, \mathrm{N}$ is the random error that follow the normal distribution with mean zero and unknown variance and N is the number of observations.

From the probabilistic point of view underlying linear regression, assume that the response variable $\mathbf{x}_l$ and predictor variable $\mathbf{x}$ are random variables in bivariate case. Then, the probability distribution of any x_l is explained by x .The expected value of x_l given by x is a straight line in which it is defined by:

$$E(x_l|x\ ) = \beta_0 + \beta_1 x \qquad (2.4)$$

It is shows that the mean of response variable $\mathbf{x}_l$is a linear on each possible value of x .

The variance estimation in simple linear regression model is:

$$Var(x_l|x\ ) = Var(\beta_0 + \beta_1 x\ +\ \varepsilon\ ) = \sigma^2 \qquad (2.5)$$

In bivariate case, the purpose of regression is to predict the dependent variable $\mathbf{x}_l$ based on independent variables $\mathbf{x}$. The joint distribution of $\mathbf{x}$ and $\mathbf{x}_l$ yields the parameter estimate of mean $\boldsymbol{\mu}_l = (\mu_x , \mu_l)$ and covariance matrices:

$$\sum = \begin{pmatrix} \sum_{xx} & \sum_{xx_l} \\ \sum_{x_l x} & \sigma^2{}_{x_l x_l} \end{pmatrix}$$

The 2-dimensional vector of coefficients $\beta = (\beta_0, \beta_1)$ is generated by:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{N} x_{li} x_i - \frac{(\sum_{i=1}^{N} x_{li})(\sum_{i=1}^{N} x_i)}{N}}{\sum_{i=1}^{n} x_i^2 - \frac{(\sum_{i=1}^{N} x_i\ )^2}{N}}$$

$$= \frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{\sum_{i=1}^{N} x_{li}(x_i - \bar{x})}$$

$$= \frac{S_{xx_l}}{S_{xx}} \quad (2.6)$$

$$= \frac{\Sigma_{xx_l}}{\Sigma_{xx}}$$

$$= \Sigma_{xx_l}\Sigma_{xx^{-1}}$$

and

$$\hat{\beta}_0 = \mu_{x_l} - \hat{\beta}_1\mu_x \quad (2.7)$$

where

$$\mu_{x_l} = \frac{1}{N}\sum_{i=1}^{N} x_{li} \quad \text{and} \quad \mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i \quad \begin{matrix}(2.8a)\\(2.8b)\end{matrix}$$

5. Filling in Least square estimate with Yates's method [31].

The Yates's method is one of the classic methods for imputing missing data since 1933 [4]. According to the yates' method, each row in missing variable will be replace with least square estimate as shown in Equation (2.3). However, the values of parameter $\boldsymbol{\beta}$ are generated by the matrix representation to generate the least square estimate. The matrix representation is also possible for prediction using multiple regression setting.

Let $\mathbf{X}$ be a $N \times P$ data matrix as shown in the Figure 2.3, where $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_l, \ldots, \mathbf{X}_P\}$ are the independent variables and $\mathbf{x}_l$ is a dependent variable. The missing data $n_0 = N - n_1$ in the dependent vector $\mathbf{x}_l = (x_{ln_1+1}, \ldots, x_{lN})^T$ can be predicted by the following least square regression model [29]:

$$\mathbf{x}_l = \mathbf{X}^1\boldsymbol{\beta} + \boldsymbol{\varepsilon} \quad (2.9)$$

The matrix development of Equation (2.9) is presented as follows:

$$\mathbf{x}_l = \begin{bmatrix} x_{1l} \\ \vdots \\ x_{Nl} \end{bmatrix}, \mathbf{X}^1 = \begin{bmatrix} 1 & x_{11} & \dots x_{1P} \\ \vdots & \vdots & \dots \ \vdots \\ 1 & x_{N1} & \dots x_{NP} \end{bmatrix}, \boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_P \end{bmatrix}, \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_N \end{bmatrix}$$

In general, $\mathbf{x}_l$ is an $N \times 1$ vector of the observations contains missing values, $\mathbf{X}^1$ is a $N \times (P + 1)$ data matrix where 1 represents a column vector of ones, $\boldsymbol{\beta}$ is a $P \times 1$ vector of the regression coefficients and $\boldsymbol{\varepsilon}$ is a $N \times 1$ vector of random errors. Where the least square of $\widehat{\boldsymbol{\beta}}$ is estimated by:

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}^{1T}\mathbf{X}^1)^{-1}\mathbf{X}^{1T}\mathbf{x}_l \tag{2.10}$$

where the least squares estimate of $\widehat{\boldsymbol{\beta}} = (\beta_0, \beta_1, \dots, \beta_{P-1})$ is based on the observed n_1 rows. This assumes that inverse matrix $(\mathbf{X}^{1T}\mathbf{X}^1)^{-1}$ is non-singular and full rank. According to Yates's method [4], one way to yield the correct least square estimates $\widehat{\boldsymbol{\beta}} = (\beta_0, \beta_1, \dots, \beta_{P-1})$ is by using the equation shown in Equation (2.10).The equation follows the design matrix based on the complete case applied to n_1 rows.

2.5.4 Stochastic Regression imputation

Stochastic regression is extended of the linear regression model. This is because the conditional mean imputation or linear regression imputation might contribute to the substantial biased result especially when the data mechanism is MAR.

Consider the basic regression in Equation (2.4), then add the unobservable random error ε_i as shown in Equation (2.3) with the assumption that the random error ε_i is normally distributed with zero mean and variance one. The step to substitute the missing values in stochastic linear regression resembles the basic linear regression imputation as discussed in Section 2.5.3. An important feature of the stochastic regression is that the lost variability and; attenuated variance and covariances of the imputed data in the basic linear regression imputation will be improved. In addition, the stochastic regression is also able to improve the correlation between variables.

There are two ways to implement the stochastic regression. The random error component ε_i can be generated either with $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ or local residual draw. According to Srivastava [20], to impute the missing data in n_0 using stochastic regression, the residual ε_i can be generated by:

$$\varepsilon_i = \left(\frac{n_1}{n_1 - p}\right)^{\frac{1}{2}} (x_{li} - \hat{x}_{li}) = (\varepsilon_i, \dots, \varepsilon_{n_1})^T \tag{2.11}$$

Take the random sample of size n_0 with a replacement from $(\varepsilon_1, \dots, \varepsilon_{n_1})^T$ to impute the missing values in n_0 . The residual placed in the new linear stochastic regression is denoted by $(\tilde{\varepsilon}_1, \dots, \tilde{\varepsilon}_{n_0})^T$ and added to the Equation (2.4):

$$\mathbf{x}_{ln_0} = \mathbf{X}^1_{n_0}\boldsymbol{\beta} + \tilde{\varepsilon}_{n_0} \tag{2.12}$$

where

$$\tilde{\boldsymbol{\varepsilon}}_{n_0} = (\tilde{\varepsilon}_1, \dots, \tilde{\varepsilon}_{n_0})^T \tag{2.13}$$

2.5.5 Multiple imputation

Multiple imputation(MI) was introduced by Donald B. Rubin [7][12] in 1970s. This method employs a Bayesian framework in handling the missing data imputation. It is designed to quantify the uncertainty of the missing values by creating several copies of a dataset, then the several copies of completed data set are analysed and combined to produce a single set of inference.

The main idea of multiple imputation is to improve few drawbacks in the single imputation such as mean or mode or median, linear regression imputation and random draw imputation. This substitution process entails three phases:

Imputation phase. The missing values are identified initially. Impute these missing values with an appropriate model to generate a complete dataset. The stochastic regression model imputation is applicable to replace the missing values in the MI technique [22] while according to Rubin [31], the implicit and explicit imputation techniques can be considered in the MI technique.

Practically, the imputation procedure in MI will create M = 3 or 5 completed datasets where $m = 1, \ldots, M$ be the number of imputation process. Traditionally, M = 3 or M = 5 number of imputation is recommended in literatures but later, the M = 20 to 100 completed dataset is the best approach [12].

Analysis phase. At this stage, it is required to perform separately the statistical analysis on M completed data. The desired multiple set of mean parameters estimate, p-values, and standard errors is generated.

Pooling phase. The result of parameter estimates and test statistics (standard errors, confidence intervals and t-tests) obtained from M completed dataset analysis are combined into a single set of results of accurate parameter estimates and test statistics.

The multiple imputation point estimate is defined by Rubin [4] to create an average of the parameters estimate [12]:

$$\bar{Q} = \frac{1}{M} \sum_{m=1}^{M} \hat{Q}_m \tag{2.14}$$

where $\hat{Q}_m$ is the parameter estimates of the m^{th} completed dataset, $\bar{Q}$ is the average parameter estimate over imputed datasets and M refers to the number of copied dataset.

The standard error generated in the multiple imputation consist of two variance components called within-variance and between-variance. Let denote T_M be the posterior variance of parameter estimate Q [12]:

$$T_M = \overline{U}_M + BV_M \tag{2.15}$$

where $\overline{U}_M$ and BV_M are the within-variance and between-variance respectively of the M number of imputations. The posterior variance is applied in the Rubin's formula and to generate the posterior variance is simply squared the standard error. Then, as in equation (2.15), the average of within -variance $\overline{U}$ is calculated by [12]:

$$\overline{U} = \frac{1}{M}\sum_{m=1}^{M}\overline{U}_m \tag{2.16}$$

where $\overline{U}_m$ denotes the variance-covariance matrix of $\hat{Q}_m$ of the m^{th} imputed dataset and M is the number of imputations.

While BV is between-imputation variance of m imputed dataset calculated by [12]:

$$BV = \frac{1}{M-1} + \sum_{m=1}^{M}(\hat{Q}_m - \bar{Q})(\hat{Q}_m - \bar{Q})^T \tag{2.17}$$

where $\hat{Q}_m$ is the parameter estimate m^{th} imputed dataset and $\bar{Q}$ is the average of parameter estimate calculated in (2.14). Thus, the total variance combines the within-variance $\overline{U}$ and between-imputation variance BV. There is an additional term BV/M requires an average of BV due to the parameters estimate using finite M . The total variance is calculated by [12]:

$$\begin{aligned} T &= \overline{U} + BV + \frac{BV}{M} \\ &= \overline{U} + \left(1 + \frac{1}{M}\right)BV \end{aligned} \tag{2.18}$$

Therefore, the standard error SE of the total variance T is depicted by taking a squared root of T as follow [22]:

$$SE = \sqrt{T} \tag{2.19}$$

The t test can be calculated as [12] :

$$T^{-1/2}(Q - \bar{Q}) \sim t_v \tag{2.20}$$

where total variance T derived in the equation (2.18), $\bar{Q}$ is the average of the parameters in the equation (2.14) and t_v is the student-t distribution with the degree of freedom v is as follow [31] [32]:

$$v = (M - 1)\left[1 + \frac{\bar{U}}{(1 + M^{-1})BV}\right]^2 \tag{2.21}$$

The two-tailed p-value to test the probability of null hypothesis $Q = Q_0$ in which Q_0 is a specified value [3]:

$$2P\big(t_v > T^{-1/2}(\bar{Q} - Q')\big) \tag{2.22}$$

where t_v is the student-t distribution with the degree of freedom v, $\bar{Q}$ is the average of the parameter estimates and Q' is the transpose of Q.

The detail process of MI is illustrated in Figure 2.4 and (2.5). In the imputation phase, M copy of datasets are created where M = 3. The missing values in each of copied data set is fill-ed in with different plausible values and different parameter estimates such as regression coefficient B_1, B_2 and B_3 and standard errors (SE). These parameter estimates of each copy of dataset are analysed in the analysis phase as shown in the step (3) in Figure 2.4. Then, in the pooling phase, each parameter estimates of each dataset will be combined (or pooled) and yielded a single set parameter estimates. This is shown in the step (4) in the Figure 2.4.

For a better understanding of the MI process, the detailed process is shown in Figure 2.4 below [32]:

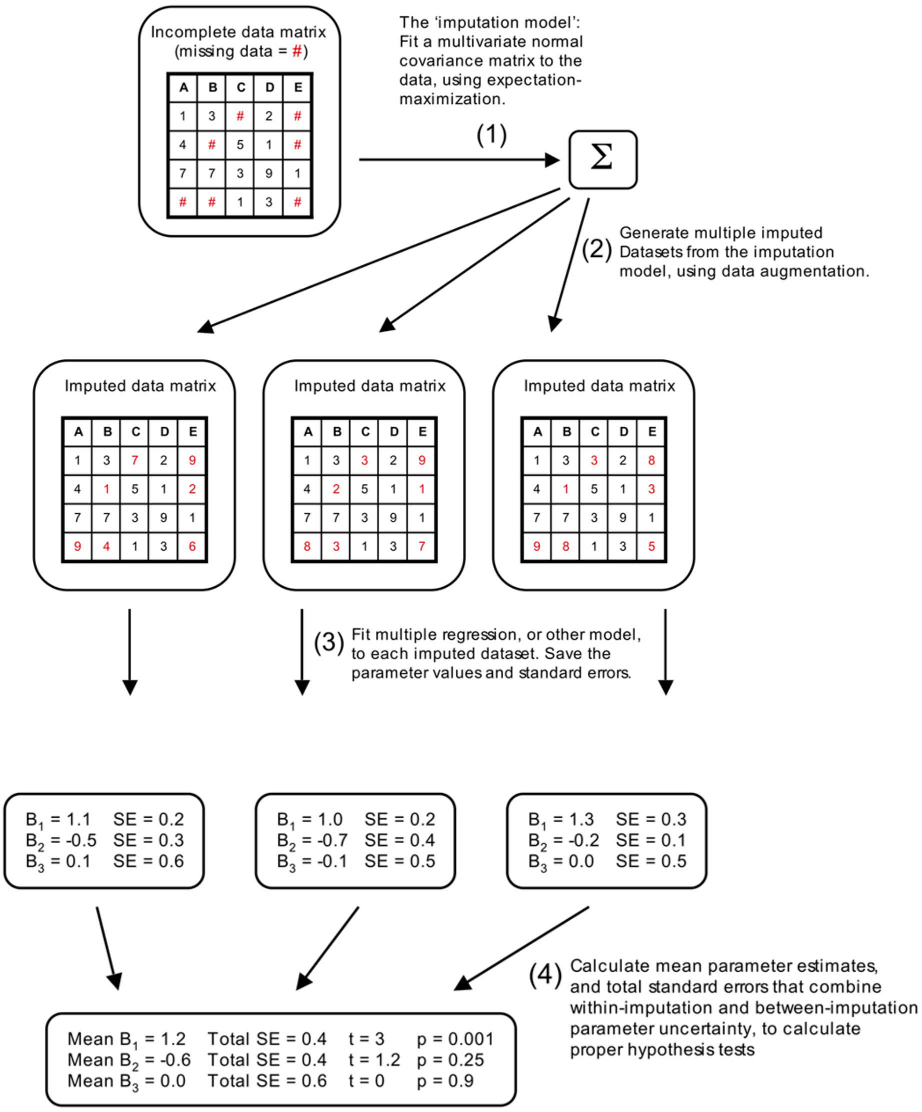


Figure 2.3 : Flowchart of the multiple imputation process [32].

Figure 2.5 illustrates the implementation steps of the multiple imputation [12].

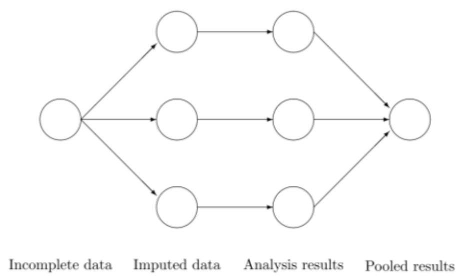


Figure 2.4 : Scheme of the main steps in multiple imputation [12]

2.5.6 Maximum likelihood imputation

Maximum likelihood (ML) [33] is a statistical procedure for handling a missing data problem. The goal of maximum likelihood is to estimate the unknown parameter from the population or sample data. The estimation process for the maximum likelihood begins by specifying the data distribution because this method relies on the probability density function. Basically, the ML computes the likelihood for the individual case in the sample data. Due to the likelihood value is too small and prone to a rounding error, the log-likelihood calculation using a natural logarithm is a more tractable calculation. Then the log-likelihood value is computed on a sample data and the parameter estimate are repeatedly replaced in the incremental search until the log-likelihood maximised for the optimal parameter estimate.

In this research context, the maximum likelihood is implemented in the multivariate normal data distribution. Let N denote the number of observations and P is the number of variables. Basically, without considering the missing values, the likelihood function based on the notations in Figure 2.3 is as follows:

$$\mathcal{L}(\boldsymbol{\theta}) = \prod_{i=1}^{N} p\,(x_{i1,}x_{i2}, \dots, x_{iP}; \boldsymbol{\theta}) \tag{2.15}$$

where $\boldsymbol{\theta}$ is the parameter estimates. In the context of missing data, due to the vector $\mathbf{x}_l$ containing missing values in $N - n_1$, then the product of likelihood of complete N observations is defined as follow [34]:

$$\mathcal{L}(\boldsymbol{\theta}) = \prod_{i=1}^{n_1} p\,(x_{i1,}x_{i2}, \dots., x_{il}, .., x_{iP}; \boldsymbol{\theta}) \prod_{n_1+1}^{N} p\,(x_{i1,}x_{i2}, \dots, x_{iP}; \boldsymbol{\theta}) \tag{2.16}$$

where x_i is a score value, $p(.)$ is a joint probability density function for multivariate normal distribution. The $\boldsymbol{\theta}$ is a set of parameters mean $\boldsymbol{\mu}$ and covariance matric $\boldsymbol{\Sigma}$. The score of the likelihood function $\mathcal{L}(\boldsymbol{\theta})$ of parameter $\boldsymbol{\theta}$

given by an independent observation $x_{i1}, x_{i2}, \ldots, x_{iP}$ describe the normal curve at score values. The log-likelihood applies the natural logarithm in the likelihood function and it is preferable than the likelihood values which are prone to rounding error. With the multivariate normality assumption, the individual log- likelihood is calculated by [13]:

$$log\mathcal{L}_i = -\frac{v_i}{2}log(2\pi) - \frac{1}{2}\log|\mathbf{\Sigma}| - \frac{1}{2}(x_i - \boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(x_i - \boldsymbol{\mu}) \tag{2.17}$$

where v_i refers to each observed data in a dataset. Later, the log-likelihood with regard to parameter estimation is maximised:

$$\hat{\theta}(\mathbf{X}) = \arg\overset{max}{\theta} \ \log\mathcal{L}\,(\boldsymbol{\theta}|\mathbf{X}) \tag{2.18}$$

Originally, the likelihood is the product of N individual likelihood for the i.i.d of the sample data. Since the log-likelihood function monotonically increases and simplifies the difficulty of the likelihood computation, the application of log-likelihood transforms the product of likelihood into a sum of individual log-likelihood values:

$$\widehat{\boldsymbol{\theta}}(x_i) = \arg\overset{max}{\theta}\prod_{i=1}^{N} p\,(x_i|\boldsymbol{\theta}) = \arg\overset{max}{\theta}\sum_{i=1}^{N} log\ p(x_i|\boldsymbol{\theta}) \tag{2.19}$$

Estimating the parameter with the maximum likelihood estimation(MLE) is relatively straightforward but the derivative of the log-likelihood function is intractable when data are missing. Therefore, iterative optimisation may reduce the complex solution in terms of the difficulty of the derivative process of MLE. The expectation maximisation (EM) algorithm is a well-known method to optimise the parameter estimation in MLE especially when the data is incomplete.

2.5.7 Expectation Maximization (EM) algorithm

The theoretical study of the EM algorithm is introduced by Dempster, Laird and Rubin [35]. The EM algorithm is an iterative process to compute the maximum likelihood estimation of the set parameter $\boldsymbol{\theta}$ such as mean, covariances and regression coefficients of a model using a statistical approach when the data has missing values. This method finds the MLE of parameter $\boldsymbol{\theta}$ in two steps. The expectation step (E-step) uses any appropriate model to fill in the missing values. Then, the expectation of the log-likelihood on the observed data based on the current parameter estimated values is computed. The maximisation (m-step) maximises the expected of the log-likelihood with respect to parameter values obtained from the new parameter set $\boldsymbol{\theta}$. The recently updated new $\boldsymbol{\theta}$ is used in the E-step in the next iteration process. The M-step guarantees that the log-likelihood values increase at each iteration. This iterative process continues until it must be converged where the deviation of the log-likelihood values in each iteration satisfy a certain threshold criterion. The basic e-step and m-step are as follow [4] [26]:

The e-step: Based on the current parameter estimate $\boldsymbol{\theta}^t$ and the iteration counter t, the expected complete data log-likelihood is obtained by [36]:

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^t) = \int log(\boldsymbol{\theta}|\ \mathbf{x}_l) f(\ \mathbf{x}_l|\mathbf{X}^O, \boldsymbol{\theta} = \boldsymbol{\theta}^t) d\mathbf{X}_l \tag{2.20}$$

where $\boldsymbol{\theta}$ consists of the mean $\boldsymbol{\mu}$ and covariance matrices $\boldsymbol{\Sigma}$ since the context of this research is Gaussian distribution. $\mathbf{X}^O$ refers to independent observed variables and $\mathbf{x}_l$ is the dependent variable with size n_1.

The m-step: the expected value complete data log-likelihood based on the current estimated $\boldsymbol{\theta}^{t+1}$ is maximised:

$$Q(\boldsymbol{\theta}^{t+1}|\boldsymbol{\theta}^t) \geq Q(\boldsymbol{\theta}|\boldsymbol{\theta}^t) \tag{2.21}$$

where $\boldsymbol{\theta}^{t+1}$ is obtained after the missing values in $\mathbf{x}_l$ are imputed. Then the new $\boldsymbol{\theta}^{t+1}$ will be re-estimated in the m-step.

The EM generally is not necessarily used to impute the missing data. The most important role of the EM algorithm is to maximise the MLE. However, it is difficult to implement the direct derivative on the density function in Equation (2.20) if the data are incomplete. The hidden variable $\boldsymbol{Z}$ is introduced.

In the m-step, the $\boldsymbol{\theta}^{t+1}$ will be maximized in the t^{th} iteration based on the convergence criteria 10^{-1}:

$$\boldsymbol{\theta}^{t+1} = \arg \frac{max}{\theta} Q(\boldsymbol{\theta}^{t}|\boldsymbol{\theta}^{t+1}) \tag{2.22}$$

In the m-step, the new parameters $\boldsymbol{\theta}^{t+1}$ are replaced by the old $\boldsymbol{\theta}^{t}$ and construct the difference between the current log-likelihood and the old log-likelihood values:

$$\mathcal{L}(\boldsymbol{\theta}^{t+1}) - \mathcal{L}(\boldsymbol{\theta}^{t}) \tag{2.23}$$

Monotonically, the log-likelihood values in each iteration decrease where the updated likelihood is deducted with the old likelihood value.

$$\mathcal{L}(\boldsymbol{\theta}^{t+1}) \geq \mathcal{L}(\boldsymbol{\theta}^{t}) \tag{2.24}$$

In m-step, to find the re-estimated of $\boldsymbol{\theta}^{t+1} > \boldsymbol{\theta}^{t}$

$$\mathcal{L}(\mathbf{X}|\boldsymbol{\theta}) = \log p\,(\mathbf{X}|\boldsymbol{\theta}) - \log p\,(\mathbf{X}||\boldsymbol{\theta}^{t}) \tag{2.25}$$

$$= \log \left\{\sum\nolimits_{z} p(\mathbf{X}|\mathbf{Z}=z,\boldsymbol{\theta})p(\mathbf{Z}=z|\boldsymbol{\theta}) \log p(\mathbf{X}|\boldsymbol{\theta}^{t})\right\}$$

$$= \log \left\{\sum\nolimits_{z} p(\mathbf{X}|\mathbf{Z}=z,\boldsymbol{\theta})p(\mathbf{Z}=z|\boldsymbol{\theta}) \frac{p(\mathbf{Z}=z|\mathbf{X},\boldsymbol{\theta}^{t})}{p(\mathbf{Z}=z|\mathbf{X},\boldsymbol{\theta}^{t})} \log p(\mathbf{X}|\boldsymbol{\theta}^{t})\right\}$$

$$= \log\left\{\sum\nolimits_{z} p\,(\mathbf{Z}=z|\mathbf{X},\boldsymbol{\theta}^{t}) \frac{p(\mathbf{X}|\mathbf{Z}=z,\boldsymbol{\theta})p(\mathbf{Z}=z|\boldsymbol{\theta})}{p(\mathbf{Z}=z|\mathbf{X},\boldsymbol{\theta}^{t})}\right\} logp(\mathbf{X}|\boldsymbol{\theta}^{t})$$

$$\geq \sum\nolimits_{z} p\,(\mathbf{Z}=z|\mathbf{X},\boldsymbol{\theta}^{t}) log \left\{\frac{p(\mathbf{X}|\mathbf{Z}=z,\boldsymbol{\theta})p(\mathbf{Z}=z|\boldsymbol{\theta})}{p(\mathbf{Z}=z|\mathbf{X},\boldsymbol{\theta}^{t}}\right\}$$

$$= \sum\nolimits_{z} p(\mathbf{Z}=z|\mathbf{X},\boldsymbol{\theta}^{t}) log \left\{\frac{p(\mathbf{X}|\mathbf{Z}=z,\boldsymbol{\theta})p(\mathbf{Z}=z|\boldsymbol{\theta})}{p(\mathbf{Z}=z|\mathbf{X},\boldsymbol{\theta}^{t})p(\mathbf{X}|\boldsymbol{\theta}^{t})}\right\}$$

$$= \Delta(\boldsymbol{\theta}|\mathbf{X},\boldsymbol{\theta}^{t})$$

Therefore:

$$\log \quad p(\mathbf{X}|\boldsymbol{\theta}) \geq p(\mathbf{X}|\boldsymbol{\theta}^{t}) + \Delta(\boldsymbol{\theta}|\mathbf{X},\boldsymbol{\theta}^{t}) \equiv Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{t}) \tag{2.26}$$

Maximizing $\boldsymbol{\theta}^{t+1}$ is shown by:

$$\theta^{t+1} = \arg \overset{max}{\boldsymbol{\theta}} Q(\boldsymbol{\theta}|\boldsymbol{\theta}) \tag{2.27}$$

$$= arg \overset{max}{\boldsymbol{\theta}} \left\{log\ p(\mathbf{X}|\boldsymbol{\theta}^{t}) \sum\nolimits_{z} p(\mathbf{Z}=z|\mathbf{X},\boldsymbol{\theta}^{t}) \frac{p(\mathbf{X}|\mathbf{Z}=z,\boldsymbol{\theta})p(\mathbf{Z}=z|\boldsymbol{\theta})}{p(\mathbf{Z}=z|\mathbf{X},\boldsymbol{\theta}^{t})p(\mathbf{X}|\boldsymbol{\theta}^{t})}\right\}$$

$$= arg \overset{max}{\boldsymbol{\theta}} \left\{\sum\nolimits_{z} p(\mathbf{Z}=z|\mathbf{X},\boldsymbol{\theta}^{t}\right\} logp(\mathbf{X}|\mathbf{Z}=z,\boldsymbol{\theta})p(\mathbf{Z}=z|\boldsymbol{\theta})$$

$$= arg \overset{max}{\boldsymbol{\theta}} \left\{\sum\nolimits_{z} p(\mathbf{Z}=z|\mathbf{X},\boldsymbol{\theta}^{t}) \frac{p(\mathbf{X}|\mathbf{Z}=z,\boldsymbol{\theta})p(\mathbf{Z}=z|\boldsymbol{\theta})}{p(\mathbf{Z}=z|\mathbf{X},\boldsymbol{\theta}^{t})}\right\}$$

$$= arg \overset{max}{\boldsymbol{\theta}} \left\{\sum\nolimits_{z} p(\mathbf{Z}=z|\mathbf{X},\boldsymbol{\theta}^{t}) logp(\mathbf{X},\mathbf{Z}=z|\boldsymbol{\theta})\right\}$$

$$= \arg \overset{max}{\boldsymbol{\theta}} \{E_{\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{t}} \log p(\mathbf{X},\mathbf{Z}=z|\boldsymbol{\theta})\}$$

The E-step and M-step are alternately repeated until the small difference is obtained and $\mathbf{\Phi}^{t+1}$ meets the convergence criteria as shown in Equation (2.22).

2.6 Gaussian mixture model

In the exploratory data analysis, data may be present with several subpopulations. The mixture model is a semi-parametric probabilistic modelling algorithm used in data clustering based on the density estimation [5]. This model learns the data distribution by probability density function (PDF) to approximate the probability of observing each data clustered in the component. The EM algorithm is the most popular strategy used to find the maximum likelihood for the parameter estimations. In this research context, the GMM is applied to work with Gaussian distribution specifically. The GMM applied in missing data problems has been studied extensively [37][5][38].

The GMM is an unsupervised learning method. Therefore, it does not require prior information on class label to determine the cluster group. However, the GMM requires prior parameter values to estimate the PDF. Let $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_P\}$ is a continuous random variable of size N with P -dimensional random vector on $\mathbb{R}^P$,the probability density function of $\mathbf{X}$ is defined by

$$p(\mathbf{X}) = \sum_{k=1}^{K} \pi_k p_k(\mathbf{X}) \tag{2.28}$$

where $p_k(\mathbf{X})$ is a mixture of PDF of $\mathbf{X}$ and each of $p_k(\mathbf{X})$ is a component of mixture density. The parameter π_k is so-called as mixing coefficient or weights in the mixture model framework. The mixing coefficients π_k for the k^{th} component must satisfy the conditions $0 < \pi_k < 1$ and $\sum_{k=1}^{K} \pi_k = 1$. The GMM is a dynamic model which does not necessarily determine any column vector to be an input or output variable particularly.

The main goal of the EM algorithm in the GMM framework is to maximise the parameter values. However, it is difficult to maximise the likelihood with respect to current parameter values when the data is incomplete. The hidden variable $\mathbf{Z}$ is included so that the difficulty of derivation in maximum likelihood estimation for incomplete data would be resolved by an iterative process as discussed in the previous sub-section 2.57.

Performing the GMM requires the unobserved hidden variable to place the component label of the feature vector $\mathbf{X}$. The multinomial hidden variable, $\mathbf{Z} = z_{ik,}, \dots, z_{NK}$, $(i = 1, \dots, N; k = 1, \dots, K)$ be a categorical variable placed with values zero-one for the Gaussian component $k = 1, \dots, K$ such that $z_{ik} = 1$ if x_i belongs to a component k or otherwise. Given the parameter mixing coefficient $\pi_1, \dots, \pi_k$ consisting of the probability of drawing x_i , the z_{ik} follows a multinomial distribution denoted by $\boldsymbol{z}_k \sim Multinomial\ (\pi_1, \dots, \pi_k)$. Let $\mathbf{z}_k$ be a multinomial vector, then it is defined by:

$$z_{ik} = \begin{cases} 1 & if\ x_i\ belongs\ to\ component\ k \\ 0 & otherwise \end{cases} \tag{2.29}$$

In order to implement the hidden variable in the mixture model, the marginal distribution $p_k(\mathbf{X})$ in Equation (2.28) is the mixture of the $k = 1, \dots, K$ Gaussian distributions. Letting the hidden variable $\mathbf{Z}$, the joint distribution $p(\mathbf{X}, \mathbf{Z})$ is formed with the continues vector $\mathbf{X}$. In the further notations, the joint distribution is represented by the marginal of hidden variable $p(\mathbf{Z})$ and the conditional distribution $p(\mathbf{X}|\mathbf{Z})$. Specifying the membership of each observation x_i, with respect to the set of parameter $\mathbf{\Phi}$, the likelihood of the complete data is obtained by:

$$p(\mathbf{X}|\mathbf{\Phi}) = p(\mathbf{X}, \mathbf{Z}|\mathbf{\Phi}) = p(\mathbf{Z}|\mathbf{\Phi})p(\mathbf{X}|\mathbf{Z}, \mathbf{\Phi}) \tag{2.30}$$

Then, the log-likelihood is

$$logp(\mathbf{X}|\mathbf{\Phi}) = \log\{p(\mathbf{X}, \mathbf{Z}|\mathbf{\Phi})\} = \log\{p(\mathbf{Z}|\mathbf{\Phi})p(\mathbf{X}|\mathbf{Z}, \mathbf{\Phi})\} \tag{2.31}$$

where $\mathbf{\Phi} = (\pi_1, \dots, \pi_K, \mathbf{\theta}_1, \dots, \mathbf{\theta}_K)$ and the $\mathbf{\theta}_k$ is the $\mathrm{P} \times 1$ vector of unknown parameters of mean $\boldsymbol{\mu}_k$ and $\mathrm{P} \times \mathrm{P}$ covariance matrices $\mathbf{\Sigma}_k$ for multivariate case.

In the log-likelihood calculation with respect to the set of parameters $\mathbf{\Phi}$, the complete log-likelihood is as follows [39]:

$$\begin{aligned} logL_C(\mathbf{\theta}|\mathbf{X}, \mathbf{Z}) &= logp(\mathbf{X}, \mathbf{Z}|\mathbf{\theta}) \\ &= log \prod_{i=1}^{N} p(x_i, z_i|\mathbf{\theta}) \\ &= log \prod_{i=1}^{N} p(z_i)p(x_i|z_i, \boldsymbol{\theta}) \\ &= log \prod_{i=1}^{N} \left(\prod_{k=1}^{K} \pi_k^{Z_{ik}} \right) \left(\prod_{k=1}^{K} N(x_i|\boldsymbol{\theta}_k)^{Z_{ik}} \right) \\ &= \sum_{i=1}^{N} \sum_{k=1}^{K} z_{ik} \log(\pi_k N(x_i|\boldsymbol{\theta}_k)) \end{aligned} \tag{2.32}$$

Using the Bayes theorem, with respect to the parameters of mean vector $\boldsymbol{\mu}_k$ and covariance matrices $\mathbf{\Sigma}_k$, to calculate posterior probability of each data point belonging to the k^{th} Gaussian component with the prior probability π_k :

$$\begin{aligned} \tau_{ik} &= p(\boldsymbol{Z}_k = 1|\mathbf{X}) \\ &= \frac{p(\mathbf{Z}_k = 1)p(\mathbf{X}|\mathbf{Z}_k = 1)}{\sum_{j=1}^{K} p(\mathbf{Z}_j = 1)p(\mathbf{X}|\mathbf{Z}_j = 1)} \\ &= \frac{\pi_k \mathcal{N}(x_i|\boldsymbol{\mu}_k, \mathbf{\Sigma}_k)}{\sum_{j=1}^{K} \pi_i \mathcal{N}(x_i|\boldsymbol{\mu}_j, \mathbf{\Sigma}_j)} \end{aligned} \tag{2.33}$$

The next sub-section provides the discussion on the application of EM in the Gaussian mixture model framework.

2.6.1 EM algorithm in GMM framework

In the GMM framework, the EM algorithm is applied to get maximum likelihood estimation (MLE) of the parameter estimation. A further discussion of the EM algorithm can be found in [40][41]. The set of parameters estimations $\mathbf{\Phi}$ in the Gaussian mixture model are estimated using the EM algorithm by maximizing the log-likelihood.

As mentioned in the previous sub-section, GMM needs prior estimation of parameter values $\mathbf{\Phi}$ for calculating the posterior distribution $\hat{\boldsymbol{\tau}}$. The purpose of $\hat{\boldsymbol{\tau}}$ is to quantify the probability of responsibility of each observation belonging to the k^{th} Gaussian component. The $\hat{\boldsymbol{\tau}}$ is computed in the e-step in the EM algorithm.

The e-step:

Let t denote an iteration counter in the EM iterative process, then the posterior distribution using Bayes's rule is as follow:

$$\hat{\tau}_{ik}^{(t)} = \frac{\hat{\pi}_k \, f\left(x_i \middle| \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k\right)}{\sum_{j=1}^{K} \hat{\pi}_k \, f\left(x_i \middle| \hat{\boldsymbol{\mu}}_j, \hat{\boldsymbol{\Sigma}}_j\right)} \tag{2.34}$$

The conditional expectation of the log-likelihood is defined as:

$$Q(\mathbf{\Phi}|\mathbf{\Phi}^t) = E_{\mathbf{\Phi}^t}\{logL_C(\mathbf{\Phi})|\mathbf{X}, \mathbf{\Phi}^t\} \tag{2.35}$$

The m-step:

Based on the current update of the posterior distribution $\hat{\boldsymbol{\tau}}$ in the e-step, the updated set of parameters $\mathbf{\Phi}$ are obtained in the m-step with respect to the k Gaussian component.

The new set of parameters $\mathbf{\Phi}^{(t+1)}$ are updated as follows [39]:

$$N_k^{(t+1)} = \sum_{i=1}^{N} \hat{\tau}_{ik} \tag{2.36}$$

$$\hat{\pi}_k^{(t+1)} = \frac{N_k}{N}, k = 1, \dots, K \tag{2.37}$$

$$\hat{\boldsymbol{\mu}}_k^{(t+1)} = \frac{1}{N_k} \sum_{i=1}^{N} \hat{\tau}_{ik} x_{ik} \tag{2.38}$$

$$\hat{\mathbf{\Sigma}}_k^{(t+1)} = \frac{1}{N_k} \sum_{i=1}^{N} \hat{\tau}_{ik} \left[(x_{ik} - \hat{\boldsymbol{\mu}}_k)(x_{ik} - \hat{\boldsymbol{\mu}}_k)^T \right] \tag{2.39}$$

The conditional expectation of the log-likelihood is maximised as:

$$\mathbf{\Phi}^{(t+1)} = \arg \underset{\mathbf{\Phi}}{max} \, Q(\mathbf{\Phi}|\mathbf{\Phi}^{(t)}) \tag{2.40}$$

The process of e-step and m-step will be iteratively continuing until the different between $\mathcal{L}\left(\mathbf{\Phi}^{(t+1)}\right) - \mathcal{L}(\mathbf{\Phi}^{(t)})$ give small amount and meets the convergence criteria.

2.6.2 EM algorithm in GMM framework with missing values

In this section, the EM algorithm applied in data set with missing values will be discussed. A further detail description can be found by Rubin and Dempster in [8]. Let denote the complete data $\mathbf{X}^C = (\mathbf{X}^O, \mathbf{X}^M)$ is divided into the feature vectors containing observed data and missing data. Both represent with the O and M subscripts for the observe and missing data respectively [39].

$$log\mathcal{L}(\mathbf{\Phi}|\mathbf{X}^O) = log\ p(\mathbf{X}^O|\boldsymbol{\Phi}) = \sum_{i=1}^{N} \log\left(\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i^o|\boldsymbol{\mu}_k \mathbf{\Sigma}_k) \right) \tag{2.41}$$

The complete log-likelihood that account for missing data is computed by:

$$logL_i = \left\{-n_1\left(\frac{v_i}{2}log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}_i|\right) - \frac{1}{2}\sum\nolimits_{i=1}^{n_1}(x_i - \boldsymbol{\mu})\right\} \tag{2.42}$$

$$-n_0\left(\frac{v_i}{2}\log(2\pi) - \frac{1}{2}\log|\sigma^2|\right)$$

$$-\frac{1}{2\sigma^2}\sum\nolimits_{i=1}^{n_0}(x_i - \boldsymbol{\mu})^2$$

$$= \{logL_{obs}\} + logL_{miss}$$

where n_1 denotes the size of observed component, n_0 denotes the size of missing component and v_i refers to the complete data element(s) in each of the current row vector. Equation (2.42) shows the computation of the log-likelihood on the entire population data and separated by a partition of observed and missing components. The first part represents the log-likelihood of the observed data while the next partition is for the missing components.

Since the EM algorithm applied in a complete data set $\mathbf{X}^C = (\mathbf{X}_p^O, \mathbf{X}_l^M)$ where $\mathbf{X}_p^O$ is observed data contains p variables and vector $\mathbf{X}_l^M$ contains missing values, $1, \dots, n_0$, $\mathbf{X}_l^M$ is included in the log-likelihood calculation:

$$Q\left(\boldsymbol{\Phi}\middle|\boldsymbol{\Phi}^{(t)}\right) = E[logL_C(\boldsymbol{\Phi}|\mathbf{X}_p^O, \mathbf{X}_l^M, \mathbf{Z})|\mathbf{X}_p^O, \boldsymbol{\Phi}^{(t)}] \tag{2.43}$$

The basic idea of the EM in GMM framework discussed in Section 2.6.1. In the e-step, the posterior distribution $\hat{\tau}_{ik}^t$ for all the data in the feature vector $\mathbf{X}^O$ is conditioned on the current parameter $\boldsymbol{\Phi}_k$. Let assume that t is the iteration counter:

$$\hat{\tau}_{ik}^{(t)} = \frac{\hat{\pi}_k\, f\left(x_i^O\middle|\hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k\right)}{\sum_{j=1}^{K} \hat{\pi}_k\, f\left(x_i^O\middle|\hat{\boldsymbol{\mu}}_j, \hat{\boldsymbol{\Sigma}}_j\right)} \tag{2.44}$$

Generally, each missing value in the feature vector $\mathbf{X}_l^M$ is replaced by conditional mean imputation. It is necessary to compute the conditional variance in the feature vector $\mathbf{X}^M$ required to compute in the E-step [14]:

$$\widehat{\boldsymbol{\mu}_{ik}^{M}} = E\left[x_i^M \middle| x_i^O, z_{ik} = 1\right] = \boldsymbol{\mu}_k^M + \boldsymbol{\Sigma}_k^{MO}(\boldsymbol{\Sigma}_k^{OO})^{-1}(x_i^O - \boldsymbol{\mu}_k^O) \tag{2.45}$$

$$\widehat{\boldsymbol{\Sigma}_{ik}^{MM}} = Cov\left[x_i^M \middle| x_i^O, z_{ik} = 1\right] = \boldsymbol{\Sigma}_k^{MM} - \boldsymbol{\Sigma}_k^{MO}(\boldsymbol{\Sigma}_k^{OO})^{-1}\boldsymbol{\Sigma}_k^{OM} \tag{2.46}$$

In the M step,the new parameters $\boldsymbol{\Phi}^{(t+1)}$ are re-estimated. The $\widehat{\boldsymbol{\Sigma}}_{ik}$ added in (2.50), the covariance matrices in account the missing component $\mathbf{X}^M$:

$$N_k = \sum_{i=1}^{N} \hat{\tau}_{ik} \tag{2.47}$$

$$\hat{\pi}_k^{(t+1)} = \frac{1}{N_k}, k = 1, \dots, K \tag{2.48}$$

$$\widehat{\boldsymbol{\mu}}_k^{(t+1)} = \frac{1}{N_k}\sum_{i=1}^{N} \hat{\tau}_{ik} x_{ik} \tag{2.49}$$

$$\widehat{\boldsymbol{\Sigma}}_k^{(t+1)} = \frac{1}{N_k}\sum_{i=1}^{N} \hat{\tau}_{ik} \left[(x_{ik} - \widehat{\boldsymbol{\mu}}_k)(x_{ik} - \widehat{\boldsymbol{\mu}}_k)^T\right] + \widehat{\boldsymbol{\Sigma}_{ik}^{MM}} \tag{2.50}$$

The procedure of e-step and m-step iteratively executed until the different of log-likelihood $\mathcal{L}\left(\boldsymbol{\Phi}^{(t+1)}\right) - \mathcal{L}(\boldsymbol{\Phi}^{(t)})$ meets the convergence criteria.

2.7 Chapter Summary

Although the EM algorithm is commonly applied in the Gaussian mixture model framework as the parameter optimization by maximizing the log-likelihood value, this algorithm is also known with the slow convergence process.

Chapter 3

Resampling technique

3.1 Introduction

In this chapter, we propose a combination of the bootstrap resampling technique with the regression imputation technique in the Gaussian mixture model framework.

We address the issue of data sparsity and uncertainty issues in the ordinary multiple imputation technique in the Amelia II package in R. The multiple imputation (MI) technique is discussed in the previous chapter. This method is known as a golden method to address the missing data issue. However, if the data are distributed on different cluster densities, the outcome from imputed data predicted by MI changes the original distributional features.

In this chapter, we present a scenario which takes advantage of the wild bootstrap and the Gaussian mixture model as to preserve the original distributions of data and to produce a better result for the predicted values.

3.1.1 Bootstrap resampling

The bootstrap is a re-sampling technique using a replacement method from original sample data proposed by Efron [42]. This re-sampling method primarily provides a procedure for estimating a standard error, confidence interval in various types of distributions. Later developments were enhanced by Bickel and Freedman [43]. The best feature of this non-parametric method is that it does not require a pre-assumption of the data distribution. The process of resampling is repeated up to the finite iteration number. This method was extended in [44] and [45] to create the random error ε_i in the regression model.

Let begin with the basic notation of the bootstrap. Supposing that the original i.i.d sample $\mathbf{x} = (x_1, x_2, \dots, x_N)^T$ of size N drawn from unknown distribution F is written as [42]:

$$F \to (x_1, x_2 \dots, x_N) \tag{3.1}$$

Since the sample data $\mathbf{x} = (x_1, x_2, \dots, x_N)^T$ has unknown distribution, the parameter of $\widehat{\boldsymbol{\theta}}$ is also unknown. Performing the bootstrap resampling from the original sample $\mathbf{x} = (x_1, x_2, \dots, x_N)^T$ with B number of iteration process will replicate the B unordered bootstrap sample indicated by (b) notation $\mathbf{x} = (x_1^1, x_2^2, \dots, x_N^B)$. The empirical distribution function $\hat{F}$ will assign the weight $1/N$ in each data points x_i [46].

$$P\left(x_i^* = x_j \middle| \mathbf{x}\right) = N^{-1} \quad 1 \le i, j \le N \tag{3.2}$$

According to Efron [47], the number of B is typically range from 50 to 200. The detail procedure to choose the right number of B is discussed in [48].

If the original sample dataset contains missing values, the missing values should be filled in with any appropriate imputation techniques such as mean, regression, and stochastic regression or any other imputation technique from implicit and explicit approaches, before generate the bootstrap estimate $\widehat{\boldsymbol{\theta}}^{(1)}, \dots, \widehat{\boldsymbol{\theta}}^{(B)}$. According to Rubin [31], the way to estimate parameter $\boldsymbol{\theta}$ in the bootstrap application is written as follow:

Algorithm 3.1: Basic resampling technique

For $b = 1,2,, \dots, B$:

1. Create a bootstrap replicated $\mathbf{X}^{(b)}$ from the original sample $\mathbf{X}$ and each observation is assigned with weight $1/N$.
2. Fill in the bootstrap replicate with an appropriate imputation technique and the imputed bootstrap replicates is become $\widehat{\mathbf{X}}^{(b)}$.
3. Estimate the bootstrap parameter estimates $\widehat{\boldsymbol{\theta}}^{(1)}, \dots, \widehat{\boldsymbol{\theta}}^{(B)}$ from $\widehat{\mathbf{X}}^{(b)}$.

3.1.2 Heteroscedasticity vs Homoscedasticity

Heteroscedasticity refers to the nonconstant variance in data dispersion across the predictor variable. The prevalence of heteroscedasticity will create variability of residual in the ordinary least square (OLS) regression which widens or narrows in a megaphone pattern when the independent variable increases in the residual plot. Ignoring the heteroscedasticity in the fitted values will cause a biased result in the estimate of regression coefficient. The example of a heteroscedastic presence in the household food expenditure dataset is shown in Figure 3.1:

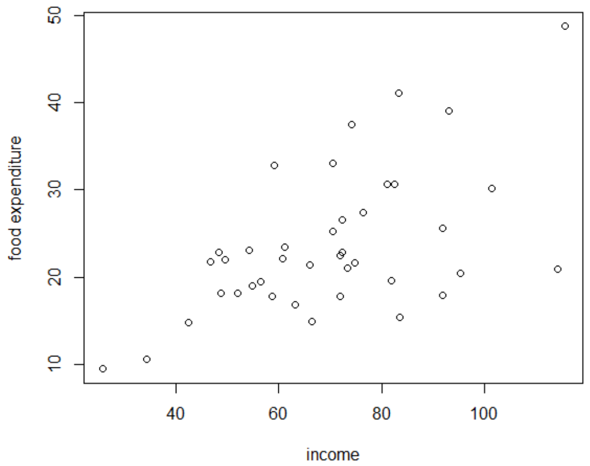


Figure 3.1 : The household food expenditure dataset

In contrast, homoscedasticity refers to residuals (or errors) in the OLS regression)which have constant variance across the predicted variable. There are several remedies to stabilise the unequal variance across the dependent variable in the literature.

There are several ways to check that the regression model works well on the data. The residual vs fitted plot and the scale location plot test are among the common built-in diagnostic plots in the regression model analysis. The first plot test is the residual vs fitted plot shows the linearity between the two variables and equal variance along the regression line as shown in the Figure 3.2 (a) and (b). The heteroscedasticity presence in the Figure 3.2(a) where the residuals clearly shows a megaphone pattern along the regression line while the residuals show improvement in the Figure 3.2 (b) where the errors are homoscedastic. However, the best outcome if the errors are shapeless and symmetrically distributed along the regression line. The second plot is the scale-location plot use a red line to test homoscedastic errors. The red line goes up and shows a steep angle if errors spread wider across the x-axis. This is shown in the Figure 3.2 (c). The errors spread equally along the regression line if the red line appear horizontal on the equal variance errors. This is shown in the Figure 3.2 (d).

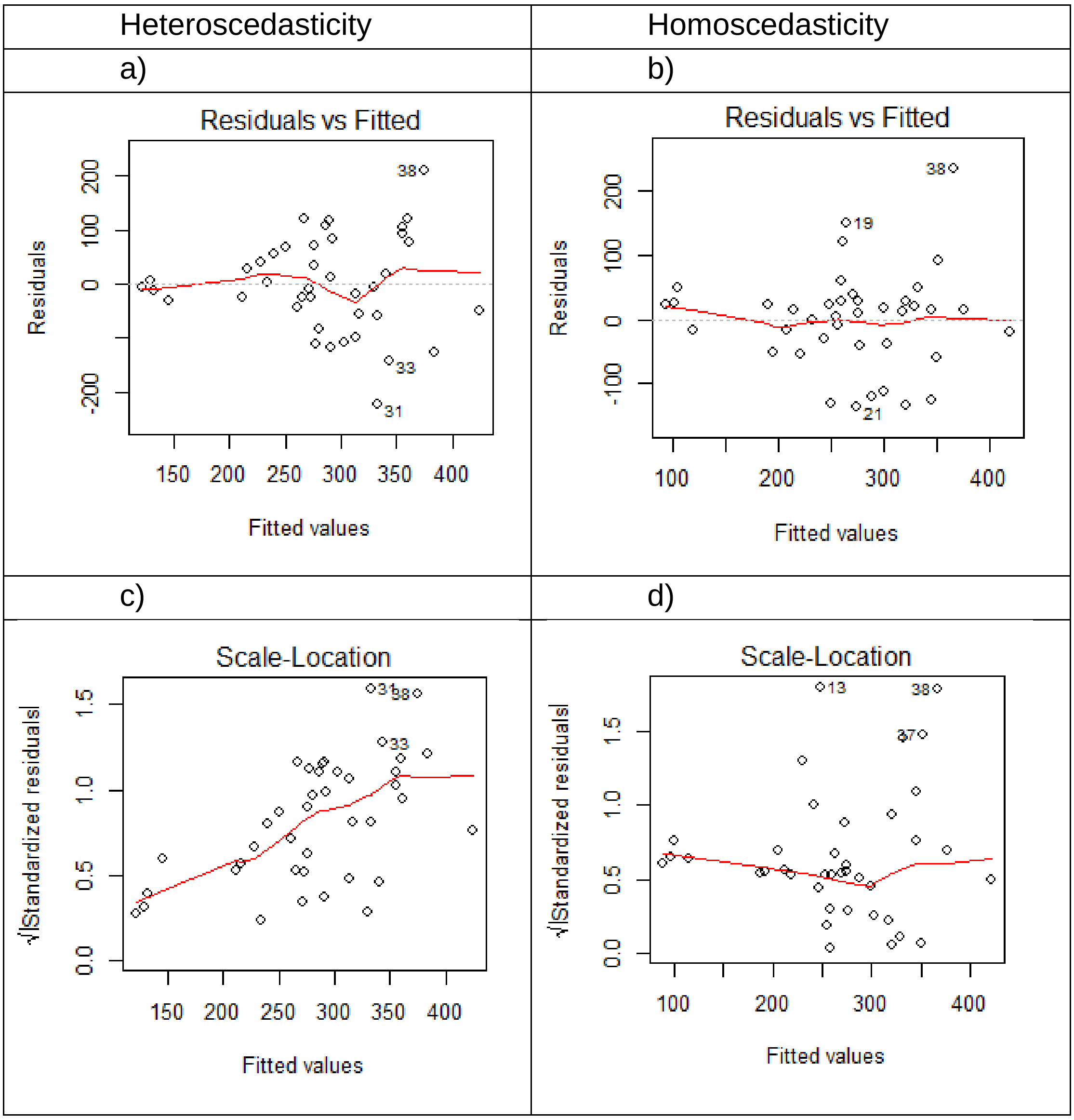


Figure 3.2 : Heteroscedasticity vs Homoscedasticity

3.1.3 Wild bootstrap

The wild bootstrap is applicable to resolve the heteroscedasticity in error variance. The inconsistency of the error variance will cause inconsistency in the regression coefficient estimation, the variance estimation from a linear regression model will be biased and the end result of the estimation test might be inaccurate [49] . Let consider again the general regression model as shown in Equation (2.9):

$$\mathbf{x}_l = \mathbf{X}^1\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{3.3}$$

where $\boldsymbol{\varepsilon}$ may be heteroscedastic. The consistency of error variance in the homoscedastic case is $1 - w_i$ where $w_i = \mathbf{x}_i^T(\mathbf{X}^{1T}\mathbf{X}^1)^{-1}\mathbf{x}_i$ is the i^{th} diagonal element of the Hat matrix $\mathbf{H} = \mathbf{X}^1(\mathbf{X}^{1T}\mathbf{X}^1)^{-1}\mathbf{X}^{1T}$. The procedure for wild bootstrap is to resample the residual $\boldsymbol{\varepsilon}$ rather than on the actual data. According to Wu [50], the variance estimation on the basic bootstrap resembles the common variance estimator $\hat{v} = \hat{\sigma}^2(\mathbf{X}^{1T}\mathbf{X}^1)^{-1}$ where $\hat{\sigma} = {}^1\!/_{N\,-\,P}\sum_1^n r_i^2$ in the regression model. Therefore, Wu modified the basic bootstrap resampling on $\boldsymbol{\varepsilon}$ by proposing the following equation:

$$x_{li}^{(b)} = \mathbf{x}_i^{1T}\hat{\beta} + t_i^* \frac{\hat{\varepsilon}_i}{\sqrt{1 - w_i}} \tag{3.4}$$

where $w_i = \mathbf{x}_i^T(\mathbf{X}^{1T}\mathbf{X}^1)^{-1}\mathbf{x}_i$ is the diagonal of the Hat matrix $\mathbf{H}$, ε_i is the i^{th} residual, the fitted regression model $\mathbf{x}_i^{1T}\widehat{\boldsymbol{\beta}} = \beta_0 + \sum_{p=1}^{P}\beta_p x_{ip}$ in which $\mathbf{x}_i^1 = [1 \quad x_{11} \quad ... \quad x_{1P}]$, t_i^* is drawn by mean is equal to zero and variance is equal to one.

Due to inconsistency issue of $t_i^*\hat{\varepsilon}_i$, then Cribari-Neto and Zarkos in [49] proposed a new mechanism to replace the $t_i^*\hat{\varepsilon}_i$.

The sample a_i is drawn with a replacement as follows:

$$t_i^* = a_i = \frac{\hat{\varepsilon}_i - \bar{\hat{\varepsilon}}_i}{\sqrt{n^{-1}\sum_{i=1}^{n}(\hat{\varepsilon}_i - \bar{\hat{\varepsilon}})^2}} \tag{3.5}$$

where $\hat{\varepsilon}_i = x_{li} - \mathbf{x}_i^{1T}\widehat{\boldsymbol{\beta}}$ in which $\mathbf{x}_i^{1T}\widehat{\boldsymbol{\beta}} = \beta_0 + \sum_{p=1}^{P}\beta_p x_{ip}$ and $\bar{\hat{\varepsilon}} = n^{-1}\sum_{i=1}^{n}\hat{\varepsilon}_i$.

There are a few types of implementation of the bootstrap in the literature. The basic implementation of bootstrap is discussed in Efron [42]. However, Wu [51] applied the following steps to improve heteroscedastic error in the regression model [49]:

Algorithm 3.2: Wu's bootstrap

---

1. Each random t_i^* is generated from a distribution with mean 0 and unit variance as shown in Equation (3.5).
2. Create a wild bootstrap sample follows Equation (3.4):
   $x_{il}^{(b)} = \mathrm{x}_i^{1T}\hat{\beta} + t_i^* \frac{\hat{\varepsilon}_i}{\sqrt{1-w_i}}$
3. Draw regression coefficients $\hat{\beta}^* = (\mathrm{X}^{1T}\mathrm{X}^1)^{-1}\mathrm{X}^{1T}\mathrm{x}_l^*$ from a wild bootstrap sample
4. Repeat steps 1,2, 3 until B times, where B is the total of iteration in number in bootstrap process.
5. Compute the variance of the B vectors obtained from the steps 1 through 4.

The second wild bootstrap technique employed in this study is the Liu's bootstrap [43]. Liu [43] proposed t_i^* in Wu [51] by resampling a set of central residual with zero mean and unit variance that has third central moments equal to one. Liu proposed two procedures to draw random numbers t_i^*. However, we consider the second procedure as it is appropriate for normal distribution.

Liu's bootstrap is conducted by drawing random numbers:

$$t_i = D_{1i}D_{2i} - E(D_{1i})E(D_{2i}) \tag{3.6}$$

Liu constructed two points of D_{1i} and D_{2i} where $D_{1i},..,D_{1N}$ are i.i.d normally distributed with mean $1/2\left(\sqrt{17/6}+\sqrt{1/6}\right)$ and variance $1/2$ while $D_{2i},..,D_{2N}$ are i.i.d normally distributed with mean $1/2\left(\sqrt{17/6}-\sqrt{1/6}\right)$ and variance $1/2$. Performing Liu's bootstrap in the Algorithm (3.2) is undertaken by changing the equation in step (2) with Equation (3.6).

Figure in 3.3 and 3.4 are the example of applying Wu's and Liu's wild bootstrap algorithm on the household food expenditure.

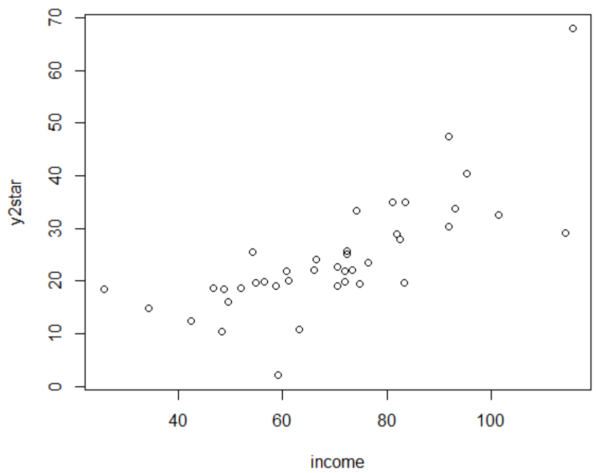


Figure 3.3 : The use of Wu's wild bootstrap algorithm to improve the variance

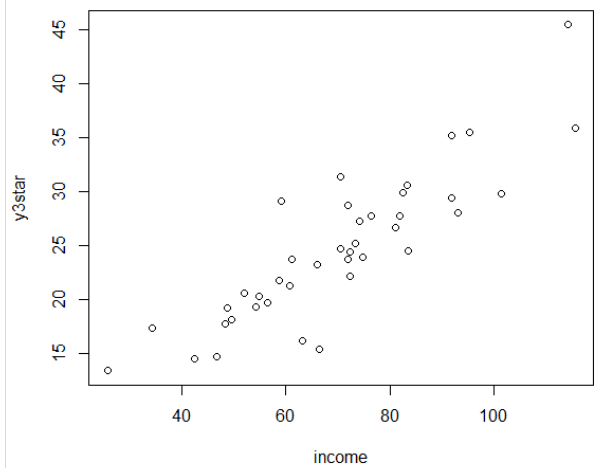


Figure 3.4 : The use of Liu's wild bootstrap algorithm to improve the variance

3.1.4 A literature review of the bootstrap resampling technique in the GMM

The bootstrap resampling technique emerged to improve the parameter estimation and validate the statistical model result such as standard error calculation used for an approximate confidence interval for a parameter estimate and improve the variability and the sparsity of data in the regression model. The problem often arises in a small dataset and only involves the single distribution. In the missing data issue, Takahashi [52] proposed the ratio imputation in the expectation maximisation with bootstrapping (EMB) algorithm. The use of bootstrap is more efficient compared to the ordinary multiple imputation because this method does not require estimation of the χ^2 distribution to overcome the Cholesky decomposition. Likewise, the use of the bootstrap resampling technique for imputing the missing values in [20] generates a good result as does multiple imputation.

The later concern when the data distributed in different subgroups, the concept of mixture of distribution introduced to resolve the complex issue in data clustering. Several purposes are highlighted to employ bootstrap resampling in the mixture distribution such as identifying the total number of components, estimating the standard error or standard deviation for different clusters and validating the identifying issue[53].

There are several simulation studies in diverse areas applied the bootstrap in the mixture model. Grün and Leisch [53] [54] applied the parametric bootstrap to diagnose the model identifiability, model fit and model restriction of the mixture model. Ciarlini and Regoliosi [55] measured uncertainty by the use of the bootstrap method to approximate the distribution function in a mixture distribution. In [56], the authors studied the empirical comparison of the use of jackknife and bootstrap sampling methods for calculating the standard error in validating the variance in parameter estimation on the mixture of data distributions. Foster, Hill and Lynos [57] incorporated the non-parametric bootstrap which is called the sampling artefact with the mixture model to improve the ecological pattern and statistical inference in their ecological study. The proposed bootstrap in the mixture model is also implemented in the FLEXMIX package in R Software [54]. Schlattmann [58] also proposed the resampling method incorporate with the mixture model as implemented by Grün and Leisch [53] [54] in Algorithm 3.3 as an alternative tool of the log-likelihood-ratio statistic (LRS) to test the number of component in infinite mixture model.

The application of resampling bootstrap recommended by Grün and Leisch [53] [54]; and Schlattmann [58]:

Algorithm 3.3: Bootstrap resampling in GMM framework

---

1. Estimate each data points belongs to k^{th} component, $\hat{a}(\mathbf{X}_N) \in A_K$ and estimate parameters $\widehat{\Theta} \subset \Omega$.
2. Draw B bootstrap sample with replacement $\mathbf{X}_N^b(b = 1, \ldots, B)$ independently with the parametric bootstrap $\mathbf{X}_N^b \sim \hat{a}(\mathbf{X}_N)$
3. Fit the mixture model incorporate with EM algorithm on the bootstrap samples in either
    a. Random initialization: $\hat{a}^b(\mathbf{X}_N^b)\epsilon A_{K_0}$ with possibly $K_0 \neq K$ or
    b. Initialization in $\widehat{\Theta}$: $\hat{a}^b(\mathbf{X}_N^b, \widehat{\Theta}) \in A_{K_0}$with $K_0 \neq K$.

The bootstrap resampling procedure implemented by Grün and Leisch [53] [54] is appropriate to deal with the complete dataset and the model is straightforward to generate. Since our research work deals with the wild bootstrap, we need to make a transformation of the i^{th} residual ε_i as shown in Equation (3.4). Previously, Cameron, Gelbach and Miller [59] and, MacKinnon and Webb [60] extended the use of wild bootstrap to produce statistical inference for different cluster of heteroscedasticity data. However, both papers employed the cluster robust variance estimator (CRVE) procedure for the wild bootstrap sampling to deal with data sets contain several clusters.

MacKinnon and Webb [61] mentioned that the wild bootstrap might not performs well if the CRVE deals with these constraints:

- The unlimited number of cluster group.
- Each cluster groups might have the same error correlation.
- The total number of observations in each cluster group are equal.

In fact, the mixture model which typically use the EM algorithm to estimate parameters with missing data, is known as a flexible model in various data distribution in different research domains. Mixture of multivariate normal distribution is the most popular in mixture modelling. It is proven by [62], to show that the GMM is flexible to fit the 2D image data in variety of angles. Due to that, we take the advantage of the great flexibility of mixture modelling in our research and combine the wild bootstrap to improve the variability in the imputed data.

3.1.5 Contribution to the missing data imputation using the wild bootstrap

The algorithm 3.3a starts with initiating the set of parameters that belong to the the k^{th} component as for a proper setting for the prior information of the set of parameters. According to Schlattmann [58], a vetor $\mathbf{x}_p = x_1, \dots x_{n1}$ is still unobserved heterogeneity each x_i is still does not assigned to any k component. From step 2 (a) in Algorithm 3.3a, a vector of the latent variable $\mathbf{Z}$ determines the component of each x_i after the estimation of posterior distribution in the step 2(a). Then in the step 2(b), the vector of residual $\hat{\boldsymbol{\varepsilon}}_k$ is computed based on the prior parameters' information of the component $k = 1, \dots, K$.

The bootstrap procedure based on the resample approach in the GMM is described in the following Algorithm 3.3a and 3.3b:

Algorithm 3.3a: Generating random error from a GMM framework

1. Use the K-means algorithm to estimate the initial values of the set of parameters $\mathbf{\Phi}$
2. Generate the residual $\hat{\boldsymbol{\varepsilon}}_k$ values for each Gaussian component:
   a. Fit Gaussian mixture model using the parameter values from the step 1 by estimating the posterior probability of each x_i using the observed values n_{1k}:
   $\hat{\tau}_{ik}^{(\mathrm{t})} = \frac{\hat{\pi}_k f(x_i|\hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k)}{\sum_{j=1}^{K} \hat{\pi}_j f(x_i|\hat{\boldsymbol{\mu}}_j, \hat{\boldsymbol{\Sigma}}_j)}$
   b. Compute the residual: $\hat{\boldsymbol{\varepsilon}}_k = \mathbf{x}_l - \mathbf{x}_i^{1T}\hat{\boldsymbol{\beta}}$ where k is the Gaussian component $k = 1, \dots, K$.

The purpose of the Algorithm 3.3(b) is to draw the vector $\hat{\boldsymbol{\varepsilon}}_k$ independently according to the number of B bootstrap replications. The new set of independent bootstrap replications $\hat{\boldsymbol{\varepsilon}}_k$ generated in each of the b^{th} bootstrap resampling process. There are two options with respect to the Wu's [51] or Liu's [43] wild bootstrap procedure. In the step 3.b.i, the Gaussian mixture model is fitted by computing the posterior distribution in the e-step again. In the following m-step, the missing values are predicted by stochastic regression where the random error $\hat{\varepsilon}_{ik}$ is added from the wild bootstrap generated in 3.b.ii . At the final step, the parameters $\mathbf{\Phi}$ are updated.

Algorithm 3.3b: Applying the wild bootstrap in GMM framework

3. For b = 1,.., B
   a. Draw a vector $\hat{\boldsymbol{\varepsilon}}_k$ of n_{1k} i.i.d sample with a simple random sampling with replacement. The vector $\hat{\boldsymbol{\varepsilon}}_k$ is generated from

step 2b with respect to the option of the Wu's [51] or Liu's[43] bootstrap procedure as discussed in Sub-section 3.13.

b. Fit Gaussian mixture model using the parameter values from the step 1.

   In the E-step,

   i. Compute the posterior probabilities vector $\hat{\tau}_{ik}$ in Equation (2.34) on the observed data:

$$\hat{\tau}_{ik}^{(t)} = \frac{\hat{\pi}_k f(x_i|\hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k)}{\sum_{j=1}^{K} \hat{\pi}_j f(x_i|\hat{\boldsymbol{\mu}}_j, \hat{\boldsymbol{\Sigma}}_j)}$$

   In the M-step,

   ii. Impute the missing values of size n_{0k} using a linear regression model in Equation (2.9) based on OLS estimator $\hat{\boldsymbol{\beta}}_k^O$ in Equation (2.10):

$$x_{lik} = \hat{\beta}_{0k}^O + \hat{\beta}_{1k}^O + \hat{\varepsilon}_{ik}$$

c. Update the new parameter $\boldsymbol{\Phi}$ for each component in GMM as shown in (2.37), (2.38) and (2.39)

3.2 Experimental Results

The purpose of this experiment is to evaluate the use of non-parametric bootstrapping incorporate with the regression model to impute the missing values in the finite gaussian mixture model framework for both latent class enumeration and parameter estimation.

In this study, we compare the imputation accuracy using MAPE and MAE whilst measuring the performance using RMSE and R-Squared of three methods: single regression imputation combined with Wu's and Liu's wild bootstrap and MI. The better results are highlighted in bold font in Table 3.1 and Table 3.2.

A plot of the results shown in Figure 3.5 compare the outcome between multiple imputation techniques in the Amelia II package in R and the proposed methods.

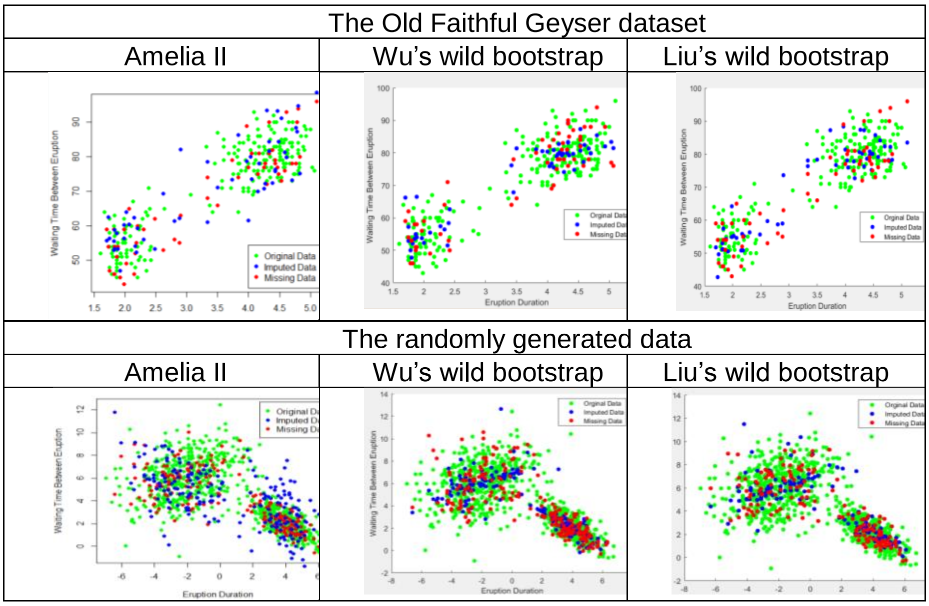


Figure 3.5 : The scatter plot of two datasets using the Amelia II and the proposed methods.

Table 3.1 summarizes the performance and prediction accuracy of the three methods on the Old Faithful Geyser dataset while Table 3.2 shows the result estimation on the random data generation. The result of the proposed methods in RMSE shows better performance and significantly different between the MI with the proposed Wu’s and Liu’s method in all MDP proportions. This is shown in the 5% MDP, Wu and Liu method yielded 7.8225 and 7.8879 respectively while MI gained 9.8719. It is also found in 10%, 15% and 20% MDP where the Wu's and Liu's method have outperformed the MI where the result of Wu's shows 7.0955, 6.6819 and 6.7349 while Liu shows 7.8746, 7.0150 and 7.2354 in RMSE. In contrast, the MI obtained 8.4187, 8.7004 and 8.9103 higher than Wu's and Liu's method in 10%, 15% and 20% MDP respectively.

The R-squared values are used to quantify the overall model performance of variance in response variable explained by the independent variables. The larger the R-squared means the more variability is explained by the linear regression model. The result of R-squared presented in Table 3.1 showed that the proposed method gives the best performance with 0.6338% for 5% of MDP proportion followed by 0.6836, 0.7683 and 0.7127 for Wu, while Liu's obtained 0.6894, 0.6869 and 0.7050 for the 10%, 15% and 20% of MDP respectively on the Faithful data set. The R-squared obtained by the proposed method in the random generation data in Table 3.2 showed less than 0.6% for all MDP percentages. In contrast, the MI in Amelia gives a lower variance than the proposed method in all MDP proportions with R-squared ranging from 0.03 to 0.2.

The imputation accuracy is measured based on the average relative error between predicted missing data and the original data using mean absolute percentage error (MAPE) and mean absolute error (MAE).

The result of MAE in the Table 3.1 showed that Wu's and Liu's methods are consistently outperformed the MI method on the Old Faithful Geyser dataset. In contrast, in the Table 3.2, Liu's method offered consistent and better accuracy than MI method. Meanwhile Wu's method showed inconsistent improvement in the measure of average error magnitude to MI method on the random data generation.

Table 3.1 : The bias, MSE and RMSE estimate on the Old Faithful Geyser dataset

		Faithful data			
		MAPE	MAE	R square	RMSE
5%	amelia	0.6220	6.5928	0.4960	9.8719
	wu	**0.2758**	**2.6453**	**0.6338**	**7.8225**
	liu	**0.1713**	**1.7281**	**0.4212**	**7.8879**
10%	amelia	0.0827	1.5636	0.6450	8.4187
	wu	**0.0379**	**0.6959**	**0.6836**	**7.0955**
	liu	**0.0190**	**0.3594**	**0.6894**	**7.8746**
15%	amelia	0.0346	1.0379	0.5184	8.7004
	wu	**0.0012**	**0.0342**	**0.7683**	**6.6819**
	liu	**0.0167**	**0.5018**	**0.6869**	**7.0150**

	amelia	0.0432	1.6841	0.4959	8.9103
20%	wu	**0.0054**	**0.2144**	**0.7127**	**6.7349**
	liu	**0.0104**	**0.3994**	**0.7050**	**7.2354**

The imputation accuracy is measured based on the average relative error between predicted missing data and the original data using mean absolute percentage error (MAPE) and mean absolute error (MAE).

The result of MAE in the Table 3.1 showed that Wu's and Liu's methods are consistently outperformed the MI method on the Old Faithful Geyser dataset. In contrast, in Table 3.1, the Liu's method offered consistent and better accuracy than MI method. Meanwhile the Wu's method showed inconsistent improvement in the measure of average error magnitude to MI method on the random data generation.

Table 3.2 : The bias, MSE and RMSE estimate on Randomly Generated data

		Random data			
		MAPE	MAE	R square	RMSE
	amelia	1.5798	0.7748	0.2439	2.0230
5%	wu	**0.0469**	**0.1055**	**0.2593**	**1.8642**
	liu	**0.0721**	**0.1604**	**0.3754**	**1.8066**
	amelia	0.1259	0.1477	0.2206	2.2926
10%	wu	0.1344	0.5811	0.3977	2.1557
	liu	**0.0089**	**0.0377**	**0.5803**	**1.6889**
	amelia	0.1674	0.3105	0.0272	2.2817
15%	wu	**0.0272**	**0.1812**	**0.1316**	**2.3463**
	liu	**0.0203**	**0.1282**	**0.5197**	**1.7214**
	amelia	*0.0501*	*0.1159*	*0.1206*	*2.7461*

20%	wu	**0.0435**	**0.3528**	**0.1954**	**2.1858**
	liu	**0.0127**	**0.1070**	**0.4586**	**1.8340**

As can be observed from the MAPE values obtained in Table 3.1, Wu's and Liu's proposed methods performed better imputation on the Old Faithful Geyser dataset.

Meanwhile, by observing the MAPE values gained in Table 3.2, Liu's method showed consistent to defeat the MI method compared with Wu's method.

As expected, the results show that the use of wild bootstrap for Wu's and Liu's options in the Gaussian mixture model framework may be able to improve the imputed data performance and accuracy, and perform better than the multiple imputation technique proposed in the Amelia II R package. These research experiments are published in the Publication 1.

3.3 Chapter Summary

In this chapter, we developed the single imputation to replace missing values. The predicted missing values were derived from the combination of regression imputation and the wild bootstrap resampling technique.

Based on the results, we summarise that the single missing data imputation combined with the wild bootstrap is preferable to the MI technique for data containing several Gaussian distributions. Furthermore, the imputation process for the Gaussian mixture model could be relevant to preserve the originality of data distribution.

Chapter 4

Predictive mean matching (PMM)

4.1 Introduction

This chapter presents the implementation procedure of the predictive mean matching method in handling the missing data problem.

The PMM method is becoming increasingly popular and is applied in some research domains such bio-statistics, the medical field and so forth. This widely accepted method is an extension of the ordinary least square method incorporating several other techniques such as nearest neighbour and hot-deck. The detail of each technique and detailed technical procedure of PMM are described in this chapter.

4.2 Predictive Mean Matching imputation technique

PMM [63] is a semi-parametric method which was introduced by Rubin [6] as a single imputation that is able to deal with misspecification issue compared to other existing imputation methods. It was then extended by Little [64] in the multiple imputation framework to work with multivariate continuous data. This PMM method has attracted more attention in the recent literature on methods for handling missing data due to its ability to preserve the data distribution without pre-assumption of the data distribution [12] and the fact that it is less sensitive on the model misspecification including non-linear associations, heteroscedastic residuals, and deviations from normality [65]. Therefore, the PMM imputation was proven to guarantee that the predicted missing values will lie only within the observed data range [4].

At present, the algorithm of PMM already exists in several statistical software such as [65]:

- Software packages in R - mice, Hmisc, BaBooN, mi

- SAS - Proc mi, MIDAS, MIDAStouch
- SPSS
- Stata – mi, impute, pmm, ice, match

4.2.1 K-Nearest neighbour technique

The nearest neighbour (NN) technique is a tool to examine the distance between key points and the closest data points of key points. It plays an important role in PMM implementation. It was introduced by Fix and Hodges in 1951 [66]. It is widely used in pattern recognition and classification where the class is initially known. Later, T. Cover and P. Hart [67] extended the use of k number of attributes or variables in the NN technique to specify the number of possible nearest data points to be considered in the class 's candidates. However, for a standardization in this thesis, we will use P for the total number of variables in this thesis. In PMM method, the nearest neighbour technique is employed to find the distance between the observed values that closest to the predictive mean values.

The imputation of PMM process initially started by estimating the predictive mean values of size n_0 using a linear regression model $\mathbf{x}_l = \mathbf{X^{1^{\mathit{O}}}}\boldsymbol{\beta}$. The next prediction is performed on the observed values of size n_1. Let $\hat{x}_{li}$ be the predicted values of size n_0 where $i = n_1 + 1, \dots, N$ while $\hat{x}_{lj}$ is the observed values of size n_1 where $j = 1, \dots, n_1$. The basic predictive mean matching can be written as follows:

$$\delta(i,j) = |\hat{x}_{li} - \hat{x}_{lj}| \tag{4.1}$$

where $\delta(.)$ is a distance function. There are multiple choices of distance calculation in the nearest neighbour technique. The employed distance measure in this research is the Euclidean distance:

$$\delta(i,j) = \left(|\hat{x}_{li} - \hat{x}_{lj}|^2\right)^{1/2} \tag{4.2}$$

On the subsequent step, in the common practice [12] [16] [17] [18] the missing values with size n_0 imputed by using the hot deck imputation technique by randomly selecting a value from the donor pool.

4.2.2 Hot deck technique

Hot deck imputation generally is a kind of a single imputation introduced by Burns [68] , Rao and Shao [28]. This technique imputes the missing values with observed values and it is necessary to include the variance estimation. It was then implemented in the multiple imputation framework that proposed by Rubin [6]. Although the hot deck can improve uncertainty in the multiple imputation method, but an improvement is required for the variance underestimation issue when it is being combined with the multiple imputation method. Di Zio [5] employed the hot deck random sample incorporates with the Gaussian mixture model framework as to improve the variability in the conditional mean imputation technique. The hot deck random sample imputation was then incorporated with the PMM method for three main reasons highlighted by Siddique and Berlin in [16], [17]:

a) The sampled data in hot deck is based on the observed value.
b) The imputed data are generated within the specified range.
c) It is not necessary to learn the data distribution since the imputed data come from the observed values.

Although it sought to be relevant in the multiple imputation process, it is still does fully account for the uncertainty issue.

The basic hot-deck imputation technique:

$$\hat{x}_{li} = x_d^* \tag{4.3}$$

where $\hat{x}_{li}$ for $i = 1,2, \dots, n_0$ and d is an index of the candidate donors in a donor pool DP_D where $d = 1, \dots, D$ from the observed data. Hot deck imputation technique replaces a missing value by randomly selecting an observation unit taken from a donor pool with respect to appropriate distance measure.

4.2.3 The basic implementation of PMM

Let $\mathbf{x}_l$ denote the dependent variable and $\mathbf{X}_P$ denote the independent variables. For the complete data, we define $\mathbf{X}^C = (\mathbf{X}^O, \mathbf{X}^M)$ where $\mathbf{X}^O$ and $\mathbf{X}^M$ are the observe data and missing data respectively. The detail notation of the missing data structure shown in the Sub-section 2.4. Algorithm 4.1 describes the basic implementation of PMM.

The basic steps of the PMM implementation incorporate with MI are shown as follow [69]:

Algorithm 4.1: Basic algorithm of the PMM

---

1) Estimate the regression coefficient $\widehat{\boldsymbol{\beta}}$, $\widehat{\sigma}$ and $\hat{\boldsymbol{\varepsilon}}$ based on n_1 complete cases.
2) Use A as a χ^2 variate with $n_1 - \mathrm{P}$ degree of freedom to draw $\sigma^{2*} = \hat{\boldsymbol{\varepsilon}}^T \hat{\boldsymbol{\varepsilon}} / A$
3) Using estimate $\widehat{\boldsymbol{\beta}}$ in the posterior distribution to generate $\boldsymbol{\beta}^*$ with covariance $\sigma^{2*} \left(\mathbf{X}^{O^{\boldsymbol{T}}} \mathbf{X}^O \right)^{-1}$
4) Compute predictive mean with the regression model of $\hat{\mathbf{x}}_l^O = \mathbf{X}^O \widehat{\boldsymbol{\beta}}$ and $\hat{\mathbf{x}}_l^M = \mathbf{X}^M \widehat{\boldsymbol{\beta}}$ for the size of n_1 and n_0 respectively.
5) For each element of missing values in x_{li} where $i = 1, \dots, n_0$, calculate the distance between $\mathbf{x}_l^O$ and $\mathbf{x}_l^M$, $\delta(i,j) = \left| \hat{\mathbf{x}}_l^O - \hat{\mathbf{x}}_l^M \right|$ where $j = 1, \dots, n_1$ and $i = 1, \dots, n_0$
6) Create donor pool sets called DP_D for each n_0 where each set of DP_D contains D number of closest candidates from observed data $\hat{\mathbf{x}}_l^O = \mathbf{X}^O \widehat{\boldsymbol{\beta}}$.

---

7) Impute each missing values $j = 1, \ldots, n_0$ by random draw $\hat{\mathbf{x}}_{li}^{M} = \hat{\mathbf{x}}_{ld}^{*}$ where $d = 1, \ldots, D$ is an index of a candidate donor in each donor pool DP_D

4.2.4 A literature review of selecting number of donors in PMM technique.

The next consideration is to describe the selection technique for candidates donor in a donor pool DP_D. The number of closest data points are defined as a donor pool DP_D and $D > 0$. The closest term means that the observed data has the smallest distance to the predictive mean value. Prior to the hot deck imputation in the PMM method, it is important to specify the number of data points in a pool of donor DP_D and measure the distance between the observed values and predictive mean values or central points in each set of donor DP_D. The chosen number of nearest neighbour in each donor DP_D is still has variants such as $D = 1,3,5,10$ [12] where D refers to the size of nearest observed values containing in a donor pool DP.

In the most research works [12], the size of a donor pool DP_D is fixed such as $D = 1,3,5,10$. However, the fix and small size of DP_D will produce the biased result of the imputed data. Although the PMM method was introduced by Rubin two decades ago, there are relatively few research works investigating particularly the set of the candidate donor D. Andridge and Little [70] distinguish four methods to select the candidates in a donor DP and also highlighted by Bureen in his book [12] :

Table 4.1 : Four methods to select the donor DP [70][12]

Number	Method of selecting donor
1	Replace a non-respondent i by randomly sampling a donor from $\delta(i,j) < \eta$ where η has a pre-specified threshold.
2	Replace a non-respondent i by a donor from $\delta(i,j) < \eta$ either using nearest neighbour hot deck, or deterministic hot deck in-stead.

3	Replace a non-respondent i by randomly sample a donor by a pre-specified size of $D = 1,3,5,10$ which taken from $\delta(i,j) < \eta$
4	Replace a non-respondent $i$by randomly sample a donor estimated from a probability that depends on | $\hat{x}_{li} - \hat{x}_{lj}$ | [16] .

There are several types to define the matching distance introduced by Little in Equation (4) in [64]. However, Little in [64] estimates $\hat{\beta}$ without taking into account the sampling variability. The type of distance measures later expanded by Heitjan and Little [71] take into account of the sampling variability. Let $\hat{x}_{li}$ be the predicted values of size n_0 where $i = n_1 + 1, \ldots, N$ while $\hat{x}_{lj}$ is the observed values of size n_1 where $j = 1, \ldots, n_1$. The three matching types are as follows [12]:

Type 0: matching $\delta(i,j) = \left|\hat{x}_{lj} = X_{obs}\,\hat{\beta} - \hat{x}_{li} = X_{miss}\,\hat{\beta}\right|$ (4.4)

Type 1: matching $\delta(i,j) = \left|\hat{x}_{lj} = X_{obs}\,\hat{\beta} - \dot{x}_{li} = X_{miss}\,\dot{\beta}\right|$ (4.5)

Type 2: matching $\delta(i,j) = \left|\dot{x}_{lj} = X_{obs}\,\dot{\beta} - \dot{x}_{li} = X_{miss}\,\dot{\beta}\right|$ (4.6)

From the equations (4.4), (4.5) and (4.6), $\hat{x}$ and $\hat{\beta}$ are the estimates of x and β respectively. The symbol of $\dot{\beta}$ in which the literal contains a dot above the β shows that is generated by randomly drawn from the posterior distribution. Type 0 matching does not involve sampling variability by using the bootstrap resampling method while while type 1 matching draws imputation parameter $\dot{\beta}$ only in $X_{miss}\,\dot{\beta}$ and type 2 matching applies sampling variability in the donors and recipients. In this research work, the type 1 matching that is being used due to sampling variability in the bootstrap resampling method will cause an iterative process. The EM algorithm is the GMM framework that applies iterative processes. Too many iterative processes though will cause increased computational complexity in the algorithm.

The main concern for the research contribution in this chapter is about to identify the number of donors in a donor pool. Table 4.2 shows the review on the development of various options to specify the size of donor D in the literature.

Table 4.2 : The development of existing research works that focusing on the size of donor DP_D

Reference	Author	Number of proposed D	Description on methods
[72]	Stef van Buuren, Karin Groothuis-Oudshoorn	3	Applied in MICE package in R software. Applied the method number (1) without specifying the type of distance calculation.
[73]	Frank E Harrell Jr and others	3	Applied in Hmisc package in R software. Apply the weighted multinomial probability sampling using the tricube function as the weights of donors. Provide two options either to use the closest donor to generate the imputed value or apply D closest data points and take one of the D closest points to generate the imputed value
[74]	Florian Meinfelder, Thorsten Schnapp	1	Applied in BaBooN package in R software. Apply Bayesian Bootstrap The higher the manual weight the higher in the influence of the corresponding variable's predictor on the overall distance.
[16], [17]	Juned Siddique,	Proposed **$c = 3$ as	Apply the distance-based donor selection

	Thomas R. Belin	a good deal.	incorporates with the Approximate Bayesian Bootstrap (ABB) approaches. The function is known as MIDAS and applied in SAS statistical software. A function $\lvert\delta_{ij}\rvert$is used to calculate the distance of each candidate's donor from the central data points based on the probability inversely proportional. This probability outcome will be used to select donors with $^{**}c$ closeness parameters.
[18], [17]	Juned Siddique, Thomas R. Belin	Proposed the value of the closeness parameter $^{**}c = \{-1,0,1,2,3\}$.	Apply the distance-based donor selection incorporates the different types of Approximate Bayesian Bootstrap (ABB) approaches. The function is known as MIDAS and applied in SAS statistical software. Expand the probability inversely proportional function $\lvert\delta_{ij}\rvert$ in [16] and introduce additional parameters α and β to select any candidate of donor with closeness parameter $^{**}c$.
[75]	Daniel McNeish	small	Small observations but does not specifically mentioned the method and size of donor.
[76]	Gerko Vink,	5	Partition the data equally into P size in each cluster. Find the 5 closest observed data from the P size of partitioned data.

	Goran Lazendic, Stef van Buuren		Impute each missing value by randomly draw an observed value from a donor pool of 5 candidate donors.
[77]	Tim P Morris, Ian R White, Patrick Royston	1,3,5 and 10	Apply the type 1 and type 2 matching distance but does not specify the type distance calculation.
[78]	Gabriele B. Durrant and Chris Skinner	10	An imputed value is randomly sampled from D size of possible values. Apply the acceptance–rejection procedure that incorporate with weighted bootstrap to generate the potential of imputed data.
[79]	Katherine J. Lee and John B. Carlin	10	Apply the type 1 matching distance in the linear regression model and type 2 matching distance in the multiple imputation framework but both implementations do not specify the type distance calculation.
[65]	Philipp Gaffer, Florian Meinfelder, Volker Bosch		MIDAS Touch is an extended version of MIDAS [17]. Applied the distance-based donor selection based on probability functionintroduced by Siddique & Belin [16] [17] [18] but modify the fixed $**c$ in that function form but added with coefficient of determi-

			nation as the function's parameter to improve few drawbacks in using c as the closeness parameter.

***c is the closeness parameter used to adjust the probability selection function for each possible closest donors*

The problem of the fixed size of donor will potentially yield a bias issue on the imputed values especially if the missing values lie in a distribution with the high-density region. The imputed data might possibly appear too close to the regression line with a nonzero slope. As far as Enders is concerned in [13], the outcome of such an imputation procedure will consequently overestimate the correlation and R^2 Statistic and lack variability regardless of whether the missing data are MCAR or MAR.

There are several existing research methods proposing a flexible value to determine the size of the donor pool DP_D. For instance, Siddique and Berlin [16] proposed probability inversely proportional to estimate the closest data point to the central point 'donee'. The exponent parameter c (in paper [16], they put it as k closest value) is used as an adjustable closest parameter in the probability inversely proportional function to determine the selection of candidate donors. The reasonable value proposed by Siddique and Berlin is $c = 1{:}\,10$ but the best deal is $c = 3$.

Siddique and Berlin then extended the probability inversely proportional function in [18] with a new transformation in which the new parameters α and β added in the probability function to selection any possible donors. They also proposed to use $c = \{-1,0,1,2,3\}$. This proposed method is also taking the consideration of the possibility value of dependent variable Y . If Y values are assumed to have a large value, then $c > 0$ is chosen while if Y has small values or negative, then $c = -1$. All these to ensures the selection probability must be positive.

Gaffert, Meinfelder, and Bosch [65] expanded the probability function applied in *MIDAS* [16] [17] [18] by adding the coefficient of determination R^2_{obs} in calculating the probability function to determine the candidate of donors. Then, this function will be incorporate with the ABB approach. The size of donors will be determined by the values R^2_{obs} in the probability function. If $R^2_{obs} = 0$, then the size of donor will be equal to 0, while $R^2_{obs} = 0.9$, the size could be 10 and if $R^2_{obs} = 0.29$, the size of donor will be equal to 3. Due to the variance issues if the ABB is applied in *MIDAS* [16] [17] [18], the *MIDAStouch* employed bias-correction factor PLF with slightly changes on the n .

Although the proposed size of donors in *MIDAS*, the efficiency of c depends on the value of the dependent variable Y . The function also needs some adjustment of c when Y has negative values. Furthermore, if c is small, then this will imply lower variance and higher bias on the imputed values. However, if $c = 0$, then the sample value equivalent to the simple random hot deck imputation. The *MIDAS* touch is also flexible; however, the proposed function depends on the value of R^2_{obs} . The function performs well if $R^2_{obs} = 0.9$ and $R^2_{obs} = 0.29$ while it will have few impractical consequences if $R^2_{obs} = 0$.

In this research work, we propose a modification to improve the size of DP_D with a flexible value. The size of DP is depends on the number of nearest data points within the radius r .For further consideration of this proposed technique, we take the idea of Di Zio [80], whereby the missing data imputation is implemented by applying the PMM imputation techniques through the Gaussian mixture model framework. The parameters estimation to impute the missing data will be applied based on the respective k^{th} Gaussian distributions. However, in [80], the number of candidates donor have not specified. Therefore, in this research work, the primary concern is the number of candidates in a donor pool DP_D will be flexible and specified by the nearest data points within a specific radius value r . Although Buuren [76] proved that PMM guarantees the preservation of the underlying distributions of the data without using the model specification, the implementation of sampling for the nearest data points within the specific radius might take the observed from the different distribution especially if the chosen radius value r is quite large. Hence, the use of the Gaussian mixture model is useful to circumvent any possibility of sampling the imputed data with the nearest data observed within the specific radius arbitrarily taken from a different distribution of data. Thus, the imputed data will lie on the current distribution and retain result accuracy.

Assume that the collection DP_D of n_1 size of observed data. The Euclidean distance between two points $\hat{x}_{li}$ and $\hat{x}_{lj}$ is shown as $|\hat{x}_{li} - \hat{x}_{lj}|$ in Equation (4.1). Given the radius value r, all pair of data points $\hat{x}_{li}$ and $\hat{x}_{lj}$ must be within the radius r as follow:

$$|\hat{x}_{li} - \hat{x}_{lj}| < r \tag{4.7}$$

Finding the nearest neighbour within the radius value was firstly introduced by Bently [81]. A detailed discussion on the fixed radius nearest neighbour will be undertaken in the next subsection.

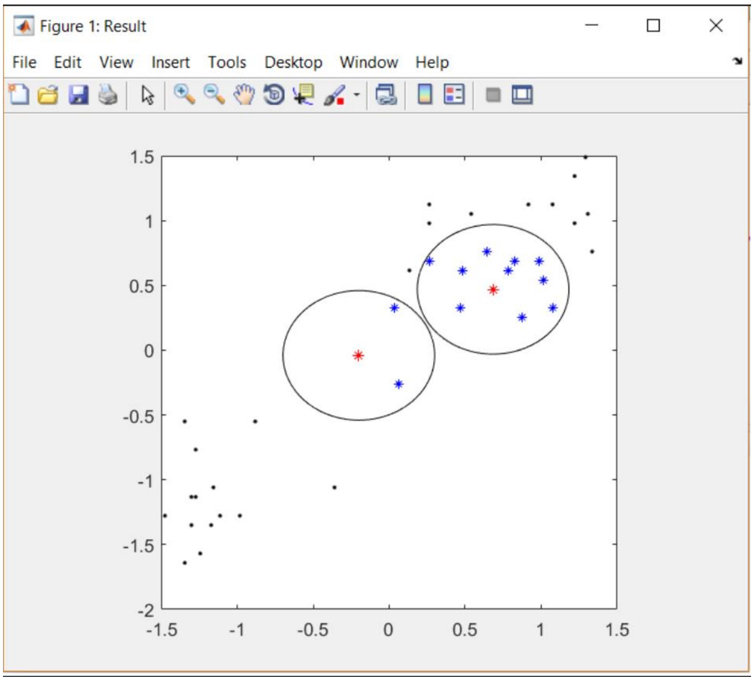


Figure 4.1 : The fixed-radius nearest neighbour

4.2.5 Contribution to the PMM imputation with the fixed radius NN search

In this research work, the *rangesearch* function in MATLAB R2017a is applied as a black-box solution to find the nearest neighbour data points within the specific radius where the radius r is a scalar value. The *rangesearch* function uses KD-tree algorithm since the number of variables in this research work is two and the Euclidean distance calculation is chosen.

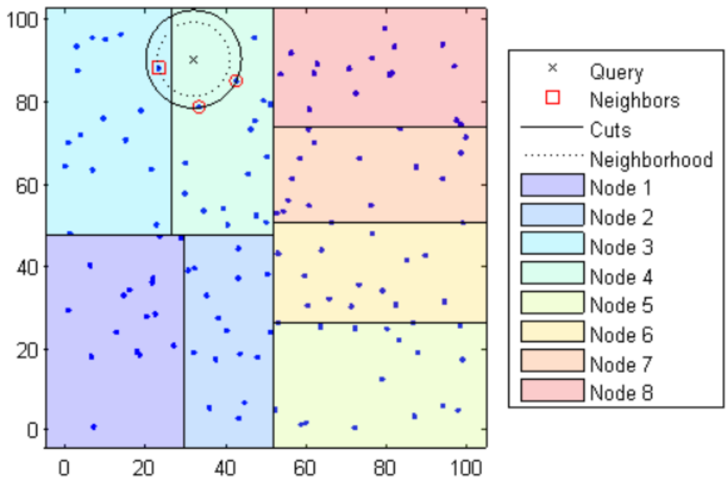


Figure 4.2 : The concept using patch objects to colour code the different "buckets." [82]

KD-tree is a data-structure built in a recursive way to find the nearest neighbour data points. Data is encapsulated with links that points to right and left node. The splitting strategy is either by median of the highest variance or middle of max-spread. In the first strategy, chose which dimension has higher variance of the data and split the dimension with the median value. The second strategy is by choosing which dimension has larger spread range by choice the closest pivot to the median value.

There are several concepts relevant in the KD-tree concept introduced by Bentley [83]. Basically, the notation is described Bentley for the "*kdtree*" algorithm where k refers to the number of dimensions and d refers to the short term of the word dimension. For the standardization of the whole notations in this thesis, we will use P to represents the number of dimensions. The root of the tree represents the parent node that connected to all child nodes. The non-terminal node contains two child nodes according to the partitioning of the node. These two nodes are named as the left nodes, and the right node. All the nodes in the left subspace represented by the left sub nodes, and the right subspace represented by the right sub nodes. In the p-dimensional tree, a record contains P keys. For instance, the 2-dimensional point has two values as x-coordinate and y-coordinate. Each node is either partitioned to the left of right side, it is determined by the key values. A record is assigned as the left child node if the key values are less than the partition value or assigned as the right child node if the key values are more than the partition value. The terminal node is a node that does not has child node contains small number of data records and it is called as bucket.

Figure 4.3(a) and Figure 4.3(b) visualizes the implementation of KD-tree. Consider the five data points (4,6), (3,8), (7,10), (8,9), (6,11). The node at coordiates (4,6) is the first node and considered as the root node. The node at coordiates (3,8) whose the value$\{(x,y)|x<4\}$, the node is assigned to the left side. The node at coordiates (7,10) whose the value $\{(x,y)|y>4\}$, then it is assigned to the right side. The subsequent node (8,9) whose the value $\{(x,y)\,|\,x>4\,\&y<9\}$, it is assigned to the left side after sub-root node (7,10).

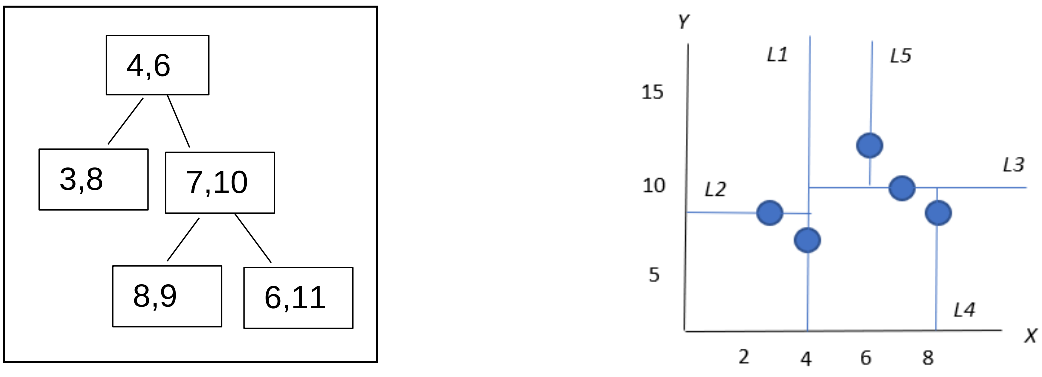


Figure 4.3 : (a) On the left side is the Binary Tree. (b) On the right side is the division method in KD-tree

The key values in the 2-dimensional record serve as a discriminator for partitioning the subspace which is the partitioning procedure initially starting at the x-coordinate. The partition value will be a root, and the two-subspace stored recursively in the two sub-trees. In Figure 4.3(b), since the node at coordinate (4,6) is the root, the partitioning starts at x-coordinate and split the points with a vertical line at $L1$. The second node at coordinate (3,8), split the y-coordinate with a horizontal line $L2$. Since the third node at coordinate (7,10) has the same level with the second node, then it splits the y-coordinate horizontally at L3. This build a tree procedure is performed in Algorithm 4.2 [83].

Algorithm 4.2 : Build a Tree

---

6. Root← Build Tree ($file$)
7. Node procedure Build Tree ($file$)
8. Begin
9. Local $file, j, d$ $maxspread, p$
10. If size($subfile$)<= b, then return (Make Terminal($file$))
11. $maxspread$ ← 0
12. For j ← 1 step 1 until k do
13. If spreadest ($j, file$) > $maxspread$ then
14. $maxspread$ ← spreadest($j, file$)
15. d ← j
16. p ← Median($d, file$)
17. Return Make Non Terminal (d, p, Build Tree(Left Sub File($d, p, file$)), Build Tree(Right Sub File($d, p, file$)))

The application of the algorithm (4.2) to (4.5) are relies of the equations. Let consider $\mathbf{X}$ be the set of N data points in a coordinate space with P dimensions drawn from $p(\mathbf{X})$. Let $S_D(\mathbf{X}_q)$ be the small circle and $\mathbf{X}_q$ be the central point of the circle that containing number of D nearest data points. The $S_D(\mathbf{X}_q)$ is defined as:

$$S_D(\mathbf{X}_q) = \{\mathbf{X}: \delta\left(\mathbf{X}, \mathbf{X}_q\right) < \delta\left(\mathbf{X}_q, \mathbf{X}_D\right)\} \tag{4.8}$$

where D is number of donors that nearest to the central point $\mathbf{X}_q$ and $\delta(.)$ is a distance function to estimate the number of D data points in a circle.

The volume of circle region $v_D(\mathbf{X}_q)$ is estimated by:

$$v_D(\mathbf{X}_q) = \int_{S_D(\mathbf{X}_q)} \mathrm{d}\mathbf{X} \tag{4.9}$$

while, to estimate the probability of each data point belonging to the circle is defined by

$$u_D(\mathbf{X}_q) = \int_{S_D(\mathbf{X}_q)} \mathrm{p}(\mathbf{X})\mathrm{d}\mathbf{X} \tag{4.10}$$

where $0 \ll u_D(\mathbf{X}_q) \ll 1$.

This probability distribution of $u_D(\mathbf{X}_q)$ describes the possible of each data point fall inside a circle region by following the beta distribution $B(D, N)$ where the parameter D is the number of samples lie in the circle region and N is the number of total samples:

$$p(u_D) = \frac{N!}{(D-1)!(N-D)!}(u_D)^{D-1}(1-u_D)^{N-D} = B(D, N) \tag{4.11}$$

Therefore, the moment of the beta distribution is

$$p(u_D) = \frac{D}{(N+1)} \tag{4.12}$$

According to Bentley [83], the searching procedure in data structure is performed by partitioning with median values and probably the average number of data points in each partition would be small. Then, $\mathrm{p}(\mathbf{X})$ is approximately lie within the circle $S_D(\mathbf{X}_q)$ by averaging $\bar{\mathrm{p}}(\mathbf{X})$. Then, the probability content of the region with the average of $\bar{\mathrm{p}}(\mathbf{X})$ approximated by:

$$u_D(\mathbf{X}_q) \cong \bar{\mathrm{p}}(\mathbf{X}_q)v_D(\mathbf{X}_q) \tag{4.13}$$

and the expected of area of region $v_D(\mathbf{X}_q)$

$$E[v_D(\mathbf{X}_q)] = \frac{D}{(N+1)\bar{\mathrm{p}}(\mathbf{X}_q)} \tag{4.14}$$

where the $\bar{\mathrm{p}}(\mathbf{X}_q)$ represents the average of small region $S_D(\mathbf{X}_q)$.

Since bk denotes the size of the bucket in the leaf node in which the number of data points in each bucket is represented, then the partitioning procedure by the median values will ensure the size of each partition is nearly to bk. Taking the equation (4.15) corresponding to the approximation of hypercube subregion, the expected area of hypercube subregion is

$$E[v_{bk}(\mathbf{X}_{bk})] = \frac{D}{(N+1)\bar{\mathrm{p}}(\mathbf{X}_{bk})} \tag{4.15}$$

where $\mathbf{X}_{bk}$ the data points fall within the hypercube subregion.

The volume of hypercube $V_D(\mathbf{X}_q)$ is proportional to $v_D(\mathbf{X}_q)$ containing the region $S_D(\mathbf{X}_q)$ which locate closest data points D to central point $\mathbf{X}_q$, is estimated by:

$$V_D(\mathbf{X}_q) = G(d)v_D(\mathbf{X}_q) \tag{4.16}$$

where the constant $G(d)$ depends on the dimension P and $v_D(\mathbf{X}_q)$ is the hypercube as in equation (4.9).

The expected volume of $V_D(\mathbf{X}_q)$ is

$$E[V_D(\mathbf{X}_q)] = \frac{DG(d)}{(N+1)\bar{\mathrm{p}}(\mathbf{X}_{bk})} \tag{4.17}$$

By assumption, the hypercube containing D data points is almost similar with the hyper-cubical bucket of the leaf containing bk data points $\mathbf{X}_D \approx \mathbf{X}_{bk}$. After simplifying the detail equation (10) presented in [83], the volume of a hypercube with 2 side r for the average number of bucket overlapped with the small ball $S_D(\mathbf{X}_q)$is estimated by:

$$\left([DG(d)]^{1/P} + 1\right)^P \tag{4.18}$$

where D is the closest data points to central point $\mathbf{X}_q$, P is the number of current dimensions and the constant $G(d)$ is defined by:

$$G(d) = \frac{(2r)^P \Gamma(P/2+1)}{\pi^{P/2} r^P} = \frac{2^P \Gamma(P/2+1)}{\pi^{P/2}} \tag{4.19}$$

From equation (4.18) and (4.19), upper bound will be:

$$\left(d^{1/P} \frac{2\Gamma(P/2+1)^{1/P}}{\sqrt{\pi}} + 1\right)^P \tag{4.20}$$

In the search algorithm, if the current node identified as a terminal node, all the records in the bucket are examined. Then the search continues with the *ball within bound* function. This function can determine the ball is within the domain of the node.

The subsequent search algorithm is to determine the location of the query point. If the current query point is less than the current partition value, the current partition values is assigned as the current coordinate upper bound. This is means that the search of the current node will be occurred in the left side of the partition value. The similar query point procedure applied for node at the right side of the tree. This query process is implemented in Algorithm (4.3)

Algorithm 4.3: **Search Node in KD Tree**

1. Procedure ($node$)
2. local $p, d, temp$
3. If node is terminal
4. Call Ball Within Bound function
5. d $\leftarrow$ discriminator [$node$]
6. p $\leftarrow$ partition[$node$]
7. If $X_q[d] \leq p$ then
8. $temp \leftarrow B_+[d]$
9. $B_+[d] \leftarrow p$
10. Search(leftson($node$))
11. $B_+[d] \leftarrow temp$
12. End
13. Else
14. $temp \leftarrow B_-[d]$
15. $B_+[d] \leftarrow p$
16. Search(rightson($node$))
17. $B_-[d] \leftarrow temp$
18. End

Algorithm 4.4: Search node when the ball within the bound

1. If BALL WITHIN BOUNDS Then
2. Begin
3. Local d
4. For d $\leftarrow$step 1 until k do
5. If COORDINATE DISTANCE (d , $X_q[d], B_-[d]) \leq PQD[1]$
6. If COORDINATE DISTANCE (d , $X_q[d], B_+[d]) \leq PQD[1]$
7. Then return (false)
8. return (true)
9. End

The ball within bound function in Algorithm 4.4 is to determine circle region is exactly lie within the boundary of the current node. Then the closest data point within the circle will be updated.

Algorithm 4.5: Search node when the circle region overlaps with other boundary

---

1. If BOUND OVERLAP BALL Then
2. local d
3. For d ←step 1 until k do
4. If $X_q[d] \leq B_-[d]$
5. Then begin “lower than low boundary”
6. *Sum* ← *sum* + COORDINATE DISTANCE (d , $X_q[d], B_-[d]$)
7. If DISSIM(sum)> $PQD[1]$ then return (true)
8. End
9. Else If $X_q[d] \leq B_+[d]$
10. Then begin “higher that high boundary”
11. Sum ← sum + COORDINATE DISTANCE (d , $X_q[d], B_+[d]$)
12. If DISSIM(sum)> $PQD[1]$ then return (true)
13. End

The bound overlap ball function in Algorithm 4.5 is applied to determine any data point lie beyond the boundary or partition line then is considered overlapped with other region. If the data points located in the lower boundary, or if the data points located at the upper boundary, then find the dissimilarity by applying the distance function to calculate the closest points within the radius.

4.2.6 Normalization

The PMM is a distance-based method uses distance calculation between non-respondent item i and respondent item j. Normalization technique is required to adjust new range for the data on different scales and transform the data within the range 0 to 1. There are several techniques for the normalization process;(1) min-max normalization, (2) z-score normalization and (3) max normalization. In this research work, the z-score is selected because the scaling calculation is based on the mean and standard deviation appropriate with the normal distribution assumption of the current data set where the mean and standard deviation values are given. The z-score is calculated as follow:

$$z\_score_i = \frac{x_i - \mu}{\sigma} \tag{4.21}$$

where μ is the mean and i is the standard deviation value.

4.2.7 Experiment procedure

We performed this simulation study to gain insight the appropriateness of PMM approach in the GMM model.

Software: the proposed method in these experiments were conducted using MATLAB version 2017a in this simulation studies. The proposed methods, PMMr 0.5 and PMMr0.7 show that the radius values 0.5 and 0.7 are applied in this experimental result. Both methods are compared with few R-packages that apply PMM with different approaches namely:

a) package Amelia II, version 1.7.4
b) package mice , version 2.46.0
c) package Hmisc, version 4.1-1
d) package BaBoon, version 0.2-0

Imputation implementation: the missing data are imputed based on PMM which combined the parametric model such as regression and nearest neighbour technique and non-parametric model such as hot deck technique. In common practice, the existing research has applied the fixed size of the nearest neighbour donors such as D is equal to 1,3,5,10 [84]. However, in our context, we proposed the flexible approach to determine the size of donors d by applying the use of radius value to estimate the size donors.

Before imposing the missing data imputation procedure, the missing values in variable Y are created randomly based on 5%, 10%, 15% and 20% MDP.

Evaluation criteria: These experiments are designed to measure the performance and prediction accuracy between predicted and actual values. RMSE computes the deviation between predicted and actual values that employed by most missing data imputation studies. The greater the deviation means the greater variance between them. Therefore, the lower value shows better performance:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{N}} \tag{4.14}$$

MAE was used to measure the average error of each different in imputation:

$$MAE = \frac{1}{N} \times \sum_{i=1}^{N}|\hat{y}_i - y_i| \tag{4.15}$$

R-squared values were used to describe the variance in goodness-of-fit for the regression models between observed data and the expected values of the dependent variable. The range of R-squared is between 0 and 1:

$$R^2 = \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2} \tag{4.16}$$

4.2.8 The proposed algorithm

The following algorithm provides the detail steps to implement the imputation process using PMMr 0.5 or 0.7.

Algorithm 4.6 : Implementation of PMM in GMM framework

---

1. Initiate the set of parameters $\mathbf{\Phi}$ with K-means algorithm.
2. Define the radius values.
3. Fit the Gaussian mixture model using the parameter values estimated in step1.
4. In the E-step,
   i. Compute the posterior probabilities vector $\hat{\tau}_{ik}$ in Equation (2.33) on the observed data
5. In the M-step,
   i. Generate the missing values of size n_{0k} and observed values of size n_{1k} using linear regression model in Equation(2.4) and (2.3) respectively based on OLS estimator $\widehat{\boldsymbol{\beta}}_k^O$:

$$\hat{x}_{lik}^M = \hat{\beta}_{0k}^O + \hat{\beta}_{1k}^O x_{ik} \text{ and } x_{ljk}^O = \hat{\beta}_{0k}^O + \hat{\beta}_{1k}^O x_{ik} + \varepsilon_i$$

This imputed data will be the central points of the nearest data point.

   ii. Apply feature scaling $z\_score_i = \frac{x_i - \mu}{\sigma}$ to normalize data for all variables.
   iii. For each missing data in $\hat{x}_{li}^M$, find $\Delta = \left|\hat{x}_{lik}^M - x_{ljk}^O\right| \leq r$ where x_{ljk}^O is observed in the current k Gaussian component.
   iv. Construct predicted missing data point sets with size n_0 named DP_D. Each set will contain the number of D donors which taken from $\Delta = \left|\hat{x}_{lik}^M - x_{ljk}^O\right| \leq r$. Apply the algorithm (4.2), (4,3)

---

and (4.4) to search the nearest data points within radius r.

v. Draw one donor i_j which has the maximum distance from DP_D where $i = 1, \ldots, n_{0k}$ and $j = 1, \ldots, D$

vi. Impute each missing value in n_{0k} with donor i_j in the step vi.

6. Update the new parameter Φ for each component in GMM as shown in (3), (4) and (5).

4.2.9 Experimental result

Table 4.1 ,4.2 and 4.3 provide the experimental result of the PMM imputation implemented on both the Old Faithful Geyser, random data generation dataset for case 1 and random data generation dataset for case 2 respectively where the number of donors D is based on the data points within the radius values 0.3, 0.5, 0.7 and 1.0. The results in Table 4.2 and 4.2 are implemented on the data set containing 20% missing data percentage. The aim of this experiment is to measure and compare the performance of PMMr in different radius values. For simplicity, we will use PMMr 0.3,PMMr0.5, PMMr0.7 and PMMr1.0 to show that the PMM employs radius 0.3, 0.5,0.7 and 1.0 respectively.

Figures 4.1, 4.2 and 4.3 present the scatterplot of the implementation of the PMMr in the different radius values for different datasets. The results show that a small radius value infuses a perfect correlation on the predicted values but may attenuate the variability of the predicted values. The highest radius value may create variability on the predicted values. However, too much variability will increase the variance of the predicted values and distort the performance of the MAPE, MAE and RMSE results. This is shown in Table 4.1 and Table 4.2 for the radius =1 where the MAPE, MAE and RMSE values are 0.1197, 4.5185, 0.2514 and 11.4778 for the Old Faithful Geyser dataset while the values are 0.0425, 0.3095, 0.0389 and 2.1735 for the random data generation. The results obtained in both Tables 4.1 and 4.2 show that the PMMr 0.5

performed better compared to other radius values in both datasets. However, in Table 4.3, the result for PMMr 0.3 is outperformed the other radii 0.5, 0.7 and 1.0 values. This is because, the variance of X and Y in the sample data generation for case 2 is reduced from 3 to 1.5. Subsequently, the chance to reduce the error between predicted values and the actual values may be feasible.

According to the results in the Table 4.1 and Table 4.2, the performance of *PMMr* 0.5 was superior compared to other radius values. Therefore, we will continue the next experiment using *PMMr* 0.5 to compare the performance with other techniques in PMM to impute the missing data based on 0.05, 0.1, 0.15 and 0.2 missing data percentage.

Table 4.3 : PMM imputation based on the radius values R =0.3 ,0.5, 0.7 and 1.0 on the Old Faithful Geyser data

Radius	MAPE	MAE	R square	RMSE
0.3	0.0440	1.6481	0.7692	6.6207
0.5	**0.0029**	**0.1111**	**0.7411**	**6.7495**
0.7	0.0083	0.3148	0.3762	8.9474
1.0	0.1197	4.5185	0.2514	11.4778

Table 4.4 : PMM imputation based on the radius values R =0.3, 0.5, 0.7 and 1.0 on the random data generation case 1

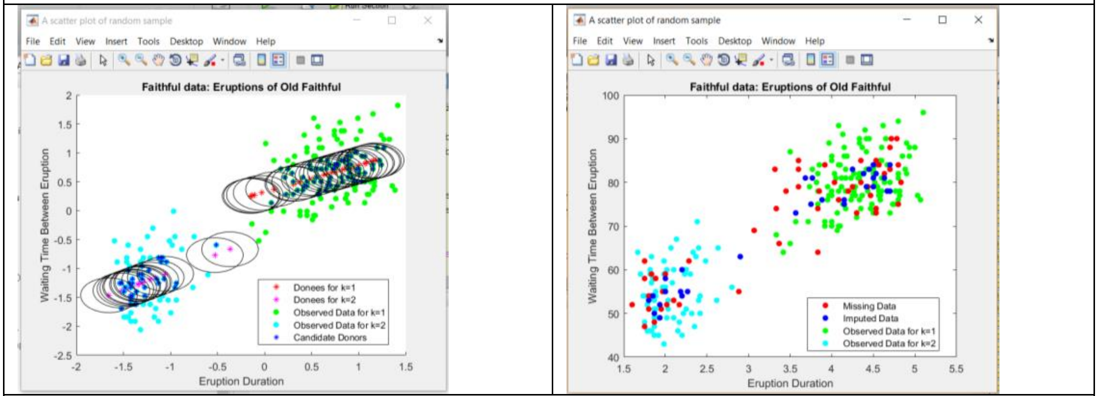
Radius	MAPE	MAE	R square	RMSE
0.3	0.0165	0.1419	0.6430	1.5148
0.5	**0.0061**	**0.0487**	**0.6646**	**1.5072**
0.7	0.0086	0.0684	**0.5502**	**1.7454**
1.0	0.0425	0.3095	0.0389	2.1735

Table 4.5 : PMM imputation based on the radius values R =0.3, 0.5, 0.7 and 1.0 on the random data generation case 2

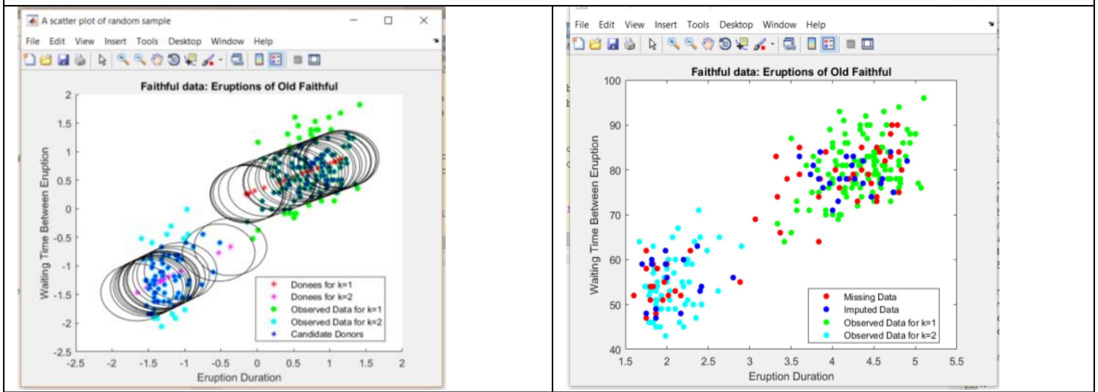
Radius	MAPE	MAE	R square	RMSE
0.3	**0.0055**	**0.0483**	**0.7360**	**1.1415**

0.5	0.0152	0.1341	0.6864	1.2439
0.7	0.0124	0.1098	0.5607	1.4724
1.0	0.0217	0.1916	0.2582	1.9133

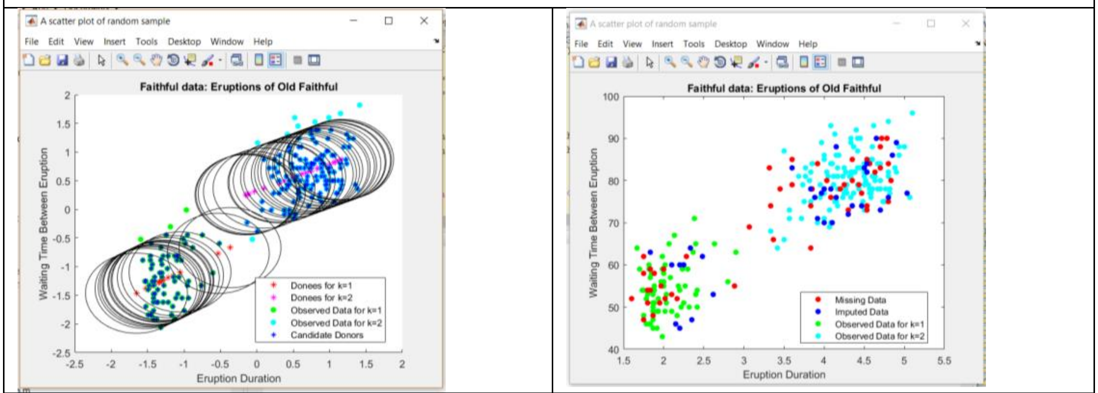
R = 0.3



R = 0.5



R =0.7



R =1.0

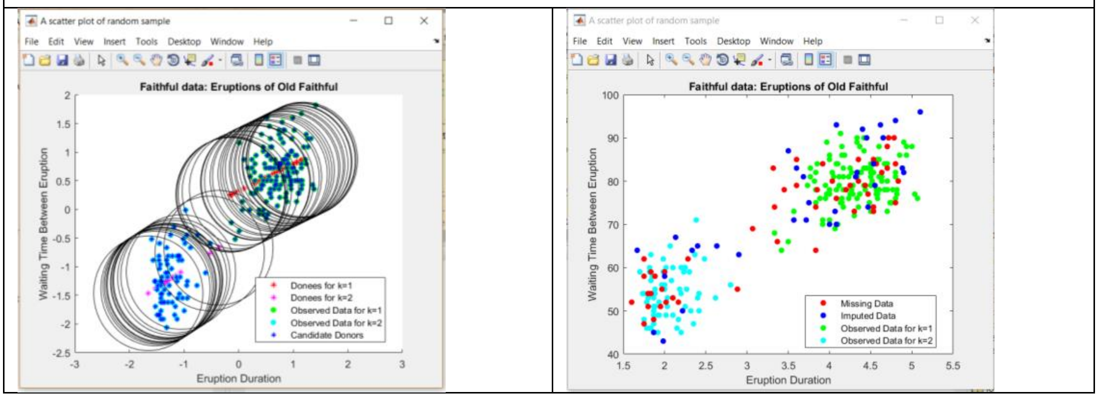
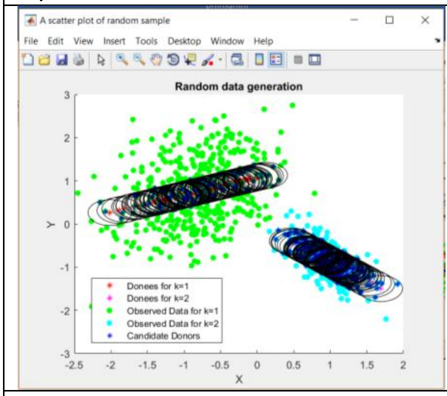
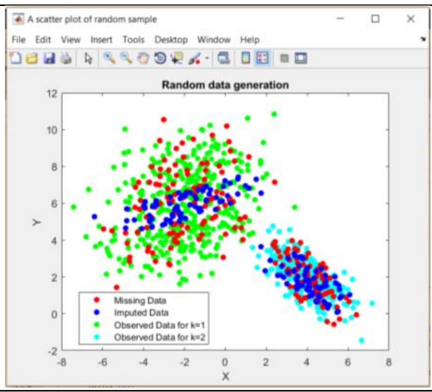


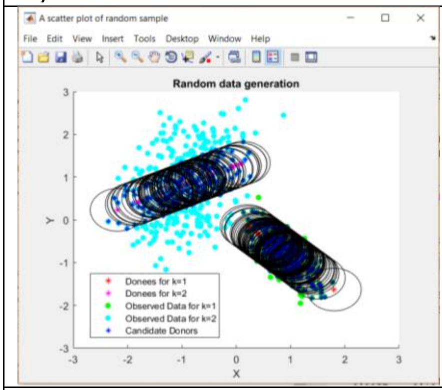
Figure 4.4 : PMM imputation based on the radius values R =0.3, 0.5, 0.7 and 1.0 on the Old Faithful Geyser data.

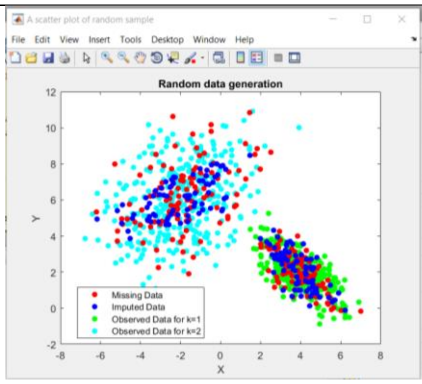
a) R = 0.3



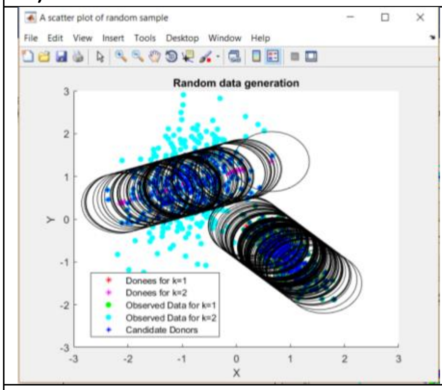


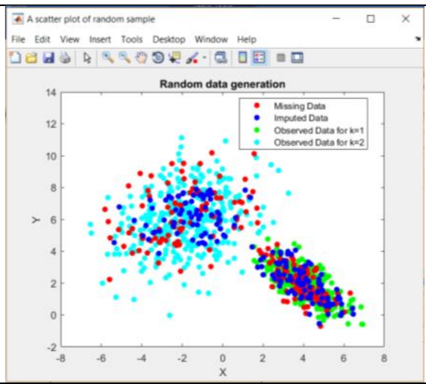
b) R = 0.5



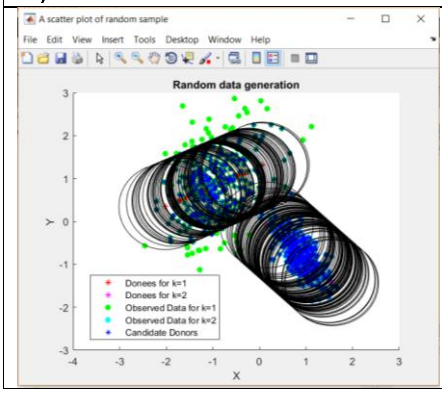


c) R = 0.7





d) R = 1



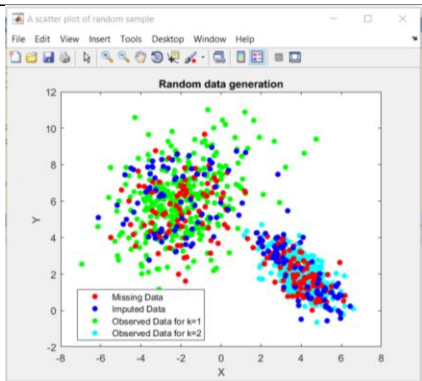


Figure 4.5 : PMM imputation based on the radius values R =0.3, 0.5, 0.7 and 1.0 on the random data generation

Table 4.6 : The comparison of PMMr 0.5 and PMMr 0.7 with other techniques in PMM on the Old Faithful Geyser

		MAPE	MAE	R square	RMSE
5%	PMMr0.5	0.2395	2.4285	**0.8526**	**5.3050**
	PMMr0.7	0.4015	4.0714	0.6518	8.1548
	Amelia	0.4667	4.7327	0.7287	7.197
	Mice	0.162	1.6428	0.6704	7.9327
	Hmisc	0.2817	2.8571	0.5623	9.1417
	bbpmm	**0.0915**	**0.9285**	0.6727	7.9056
10%	PMMr0.5	**0.0170**	**0.3333**	**0.6017**	**8.4174**
	PMMr0.7	**0.0019**	**0.0370**	**0.6242**	**8.1763**
	Amelia	0.0661	1.3008	0.648	7.9123
	Mice	0.064	1.2592	0.6235	8.183
	Hmisc	0.0207	0.4074	0.677	7.5792
	bbpmm	0.0339	0.6666	0.6497	7.8928
15%	PMMr0.5	**0.02**	**0.5853**	0.7075	7.2415
	PMMr0.7	0.0935	2.731	0.6352	8.0879
	Amelia	0.0456	1.333	**0.7451**	**6.7604**
	Mice	0.0467	1.3658	0.6079	8.3840
	Hmisc	0.0233	0.6829	0.6025	8.442
	bbpmm	0.0684	2	0.6525	7.8925
20%	PMMr0.5	**0.0091**	**0.3518**	**0.7815**	**6.3377**
	PMMr0.7	**0.0083**	**0.3148**	0.5451	8.9474
	Amelia	0.0561	2.1065	0.7693	6.5143
	Mice	0.0162	0.6111	0.6688	7.8066
	Hmisc	0.0256	0.9629	0.6594	7.9162
	bbpmm	0.03403	1.2777	0.6734	7.7519

Table 4.7 : The comparison of PMMr0.5 and PMMR 0.7 with other techniques in PMM on the random data generation case 1

		MAPE	MAE	R square	RMSE
5%	PMMr0.5	0.1403	0.2982	**0.6101**	**1.6642**
	PMMr0.7	0.0941	0.2	**0.5448**	**1.7981**
	Amelia	**0.0336**	**0.2778**	0.1441	2.3844
	Mice	0.1997	0.3592	0.4391	1.6754
	Hmisc	0.2401	0.4320	0.2457	1.943
	bbpmm	0.1283	0.2308	0.5027	1.5777
10%	PMMr0.5	**0.0552**	**0.2368**	**0.5420**	**1.7416**
	PMMr0.7	**0.0591**	**0.2534**	**0.5388**	**1.7477**
	Amelia	0.07075	0.3036	0.1902	2.3156
	Mice	0.101	0.4335	0.5059	1.8087
	Hmisc	0.102	0.4379	0.5363	1.7521
	bbpmm	0.1064	0.4567	0.47302	1.8681
15%	PMMr0.5	**0.0060**	**0.0365**	**0.6151**	**1.6241**
	PMMr0.7	0.0261	0.1588	0.5470	1.7621
	Amelia	0.0351	0.1953	0.2493	2.1199
	Mice	0.02	0.1142	0.529	1.6541
	Hmisc	0.0339	0.1931	0.4831	1.7328
	bbpmm	0.0427	0.2435	0.4173	1.8398
20%	PMMr0.5	**0.0061**	**0.0487**	**0.6646**	**1.5072**
	PMMr0.7	0.0086	0.0684	**0.5502**	**1.7454**
	Amelia	0.0162	0.1342	0.1209	2.4166
	Mice	0.007	0.0581	0.4686	1.8788
	Hmisc	0.0260	0.2151	0.3869	2.0181
	bbpmm	0.0109	0.09	0.4669	1.8818

Table 4.8 : The comparison of PMMr 0.5 and PMMr 0.7 with other techniques in PMM on the random data generation case 2

MDP	Methods	MAPE	MAE	R square	RMSE
5%	PMMr0.5	0.0437	0.0776	**0.7324**	**1.0148**
	PMMr0.7	**0.0007**	**0.0011**	0.5912	1.2543
	Amelia	**0.0014**	**0.0026**	0.3062	1.6339
	Mice	0.2138	0.3802	0.5518	1.3131
	Hmisc	0.0471	0.0838	0.7274	1.024
	bbpmm	0.0572	0.1018	0.6625	1.1395
10%	PMMr0.5	**0.0019**	**0.0081**	0.7146	1.1828
	PMMr0.7	**0.0079**	**0.0335**	0.4880	1.5843
	Amelia	0.0737	0.3117	0.3419	1.796
	Mice	0.0196	0.0832	0.6367	1.3343
	Hmisc	0.0094	0.0401	0.6524	1.3052
	bbpmm	0.0254	0.1075	**0.7291**	**1.1522**
15%	PMMr0.5	**0.0024**	**0.0154**	**0.6938**	**1.2594**
	PMMr0.7	0.0135	0.0851	0.5882	1.4607
	Amelia	0.0128	0.0806	0.3936	1.7723
	Mice	0.0071	0.0452	0.6106	1.4201
	Hmisc	0.0034	0.022	0.6607	1.3257
	bbpmm	0.0172	0.1083	0.605	1.4304
20%	PMMr0.5	0.0152	0.1341	0.6864	**1.2439**
	PMMr0.7	0.0124	0.1098	0.5607	1.4724
	Amelia	0.0056	0.0501	0.3426	1.8009
	Mice	0.021	0.1863	0.6264	1.3576
	Hmisc	**0.0017**	**0.0157**	**0.5881**	1.4255
	bbpmm	0.007	0.0626	0.6269	1.3567

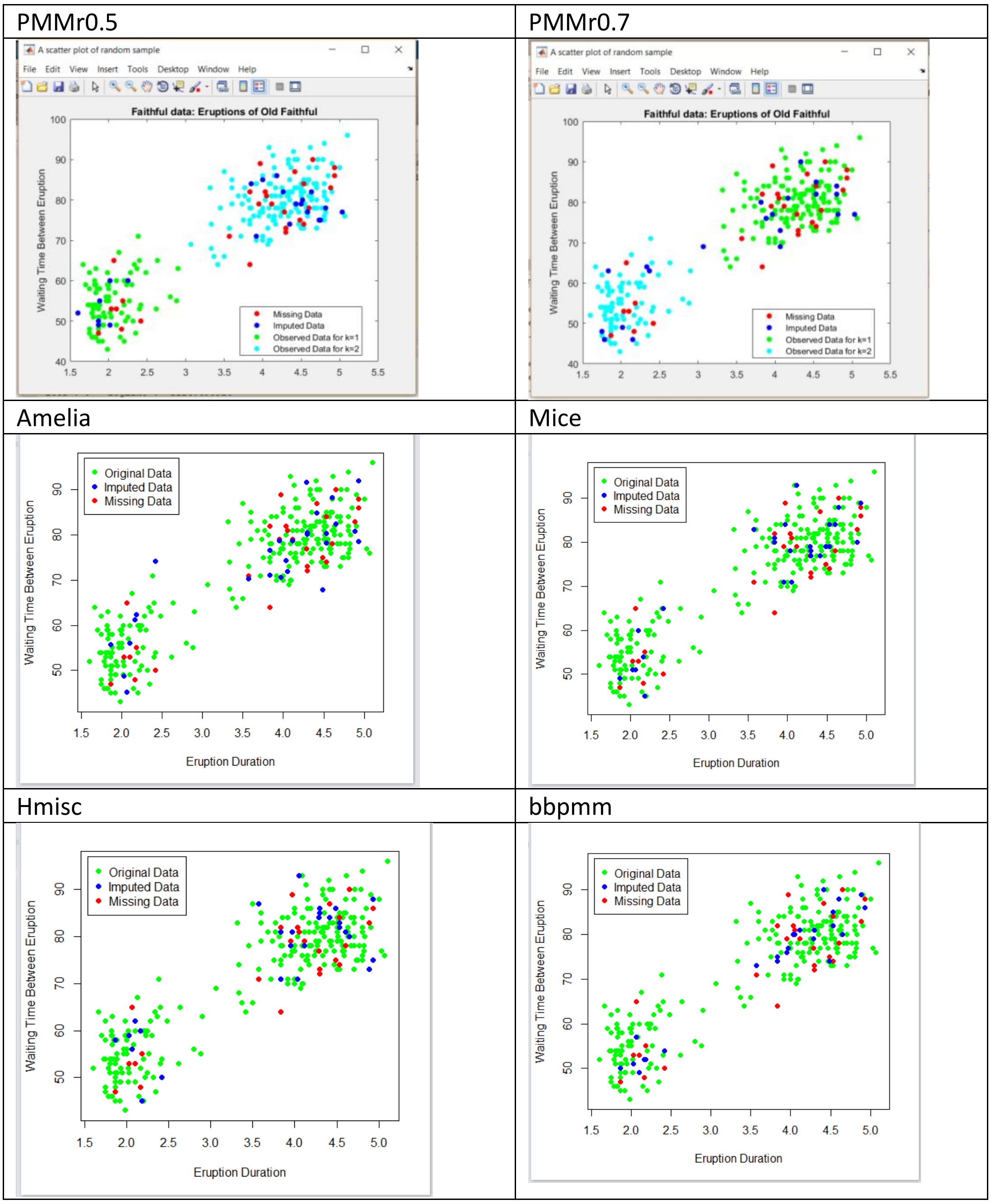


Figure 4.6 : The scatter plot of the comparison of proposed methods with other four packages in R software using different PMM – MI methods on the Fruitful Geyser data with 10% missing data percentage.

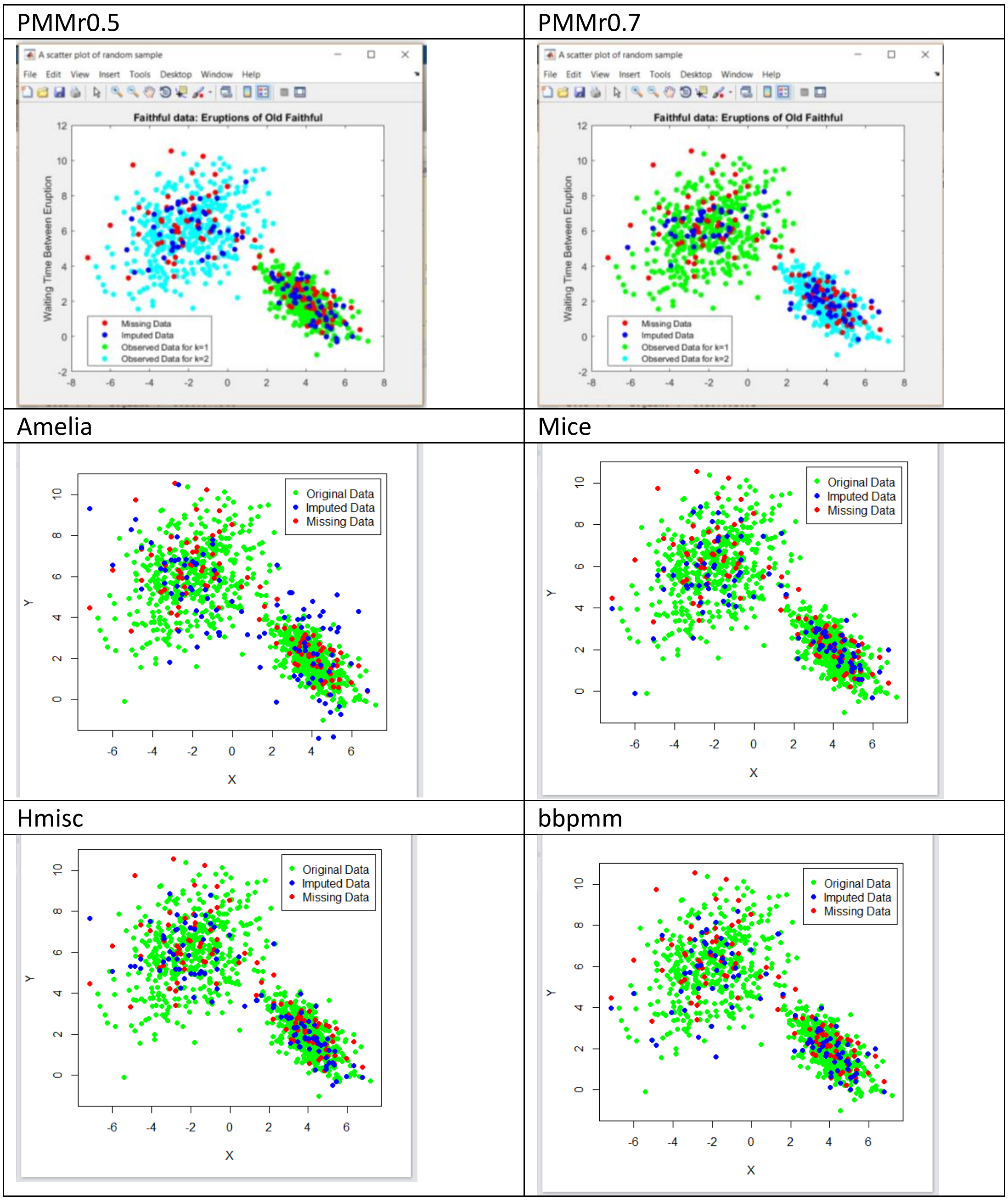


Figure 4.7 : The scatter plot of the comparison of proposed methods with other four packages in R software using different PMM – MI methods on the random data with 10% missing data percentage.

4.3 Chapter Summary

The proposed method in these simulation studies is to minimize and improve the error in the predicted values in the single imputation technique. The PMM technique in an extension of the regression imputation model. Although the PMM in the literature works proven to produce the good result when it collaborates with the MI technique, however, the MI itself deals with the uncertainty in the imputed data. The PMM with a fixed radius nearest neighbour could minimize the error and deviation between original data and imputed data as shown in the result. This also indicated that the imputed data from the PMMr are relatively close to the true value.

Chapter 5

Conclusion

This thesis provides two separate mechanisms to improve the existing solutions for handling missing data issues. The problem of missing data is ubiquitous in numerous domains. Inefficiency in providing an appropriate method for implementing missing data prediction will potentially leading to bias in analysis results.

The regression imputation is known as a single imputation technique. Without providing the random ε in a regression equation, this method, which is also known as conditional mean imputation, suffers with serious biased analysis results where the imputed values lie on the regression line. Thus, the analysis result such as R^2 and correlation will be too close to zero. Adding the random e as a residual to the regression model will imply to the uncertainty and high variance of the imputed data for which imputed data will be too far away from the regression line. This method is known as the stochastic regression model. Taking the idea of the wild bootstrap and applying the heteroscedastic data in prediction, this method may improve and will create a balanced variance in imputed data. Thus, the wild bootstrap is proposed combined with the ordinary regression model to impute the missing data with continuous data type. Considering preserving the data distribution, the prediction of missing values is performed in the Gaussian mixture model framework.

The second proposed method still takes account of the variability of the imputed data in a different way, as applied in the first method demonstrated in the Chapter 3. Chapter 3 presented the PMM method where this proposed method implements the imputation procedure through a combination of the nearest neighbour and hot-deck techniques. This implementation is to answer the research question about a plausible way to determine an appropriate number of donors d. In the existing research, few research works have introduced and applied $d = 1,3,5\ ,10$ as the number of data points in the donor pool. Missing values will be predicted by randomly imputing any one of the candidate donors from the donor pool d . The number of d is fixed and quite a small number. If the missing values occur in the high-density distributed data, the candidate of the imputed values might be too close to the regression line. Consequently, the predicted values might suffer from a biased analysis result and imply to overestimate the correlation and R^2 values. The third research question relates to the fact that the PMM procedure executed in the Gaussian mixture model framework will circumvent the nearest data points within the specific radius values taken from a different data cluster. These combinations are relatively good, may reduce the differentiation between original data and imputed data and hence may improve the prediction accuracy and performance.

Although the proposed method in this research work offers promising results, some limitations have been identified:

a) The combination of bootstrap resampling in the Gaussian mixture model framework is a cumbersome procedure and computationally expensive. This is due to the fact that the bootstrap resampling technique is a repetitive process and the sampling will be repeated and end up with a B number of process. Likewise, the Gaussian mixture model uses maximum likelihood via the EM algorithm to accomplish the parameter estimation. However, the EM algorithm is also a repetitive process and might suffer with a convergence issue where, in order to stop the iteration process, the current log-likelihood needs to meet a certain stopping criterion. In addition, due to the fact that the wild bootstrap works with the residual values, the author needs to generate the residuals that belong to the k^{th} Gaussian component prior to the bootstrap resampling procedure. This will require the process of fitting the mixture model to be carried out twice for this proposed model.

b) The implementation of the Gaussian mixture model on two Gaussian components might create a chance for any overlapping data point to be appointed to the non-actual Gaussian component. Fitting the mixture model on the two-mixture distributions is difficult for the overlapping components.

If any data points lying on the overlapping components are missing, the prediction for these data points will utilise the wrong parameter θ_k in the prediction process. As a result, the scenario will not only increase variance between the original data and imputed data, the imputation outcome may produce false analysis results.

c) The nearest neighbour (NN) technique is combined with the PMM algorithm. There are several NN techniques in the literature. To find an appropriate NN technique with fixed radius values is troublesome because it involves a geometry concept. Employing the established black-box solution from the MATLAB function library is one of the options. The *rangesearch* function applies the KD-tree algorithm to find the NN data points. This KD-tree is only applicable for one variable up to nine-variates data. Although the KD-tree only supports up to nine-variates data, the recursive process in the KD-tree to find particular nodes for multivariate data in the Gaussian mixture model framework is computationally expensive.

Several research guidelines for future research directions follow:

a) Which techniques are appropriate to improve the convergence issue in the EM algorithm?
b) Which technique is appropriate for dealing with an overlapping mixture component that contains missing values?
c) What is the nearest neighbour algorithm that may be efficient and fast for finding the nearest data points?
d) What is the evaluation criterion that is appropriate for measuring the missing data prediction in the different clusters?

Since this research work only deals with two-variates that contain two-mixture data distributions, subsequent research work needs to be extended to the multivariate finite mixture model with an appropriate method for dealing with missing data values.

Bibliography

[1] J. L. Schafer and J. W. Graham, "Missing data: our view of the state of the art," *Psychol. Methods*, vol. 7, no. 2, pp. 147–177, 2002.

[2] D. B. Rubin, *Multiple imputation for nonresponse in surveys*, vol. 81. John Wiley & Sons, 2004.

[3] J. L. Schafer, "Analysis of incomplete multivariate data.," *Stat. Med.*, vol. 19, no. 7, pp. 1006–1008, 1997.

[4] R. J. A. Little and D. B. Rubin, *Statistical Analysis with Missing Data*. 2002.

[5] M. Di Zio, G. Ugo, and O. Luzi, "Imputation through finite Gaussian mixture models," *Comput. Stat. Data Anal.*, vol. 51, no. 11, pp. 5305–5316, 2007.

[6] D. B. Rubin, "Multiple imputations in sample surveys - A phenomenological Bayesian approach to nonresponse," in *Proceedings of the Survey Research Methods Section of the American Statistical Association*, 1978, pp. 20–34.

[7] D. B. Rubin, "Inference and Missing Data," *Biometrika*, vol. 63, no. 3, p. 581, 1976.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. R. Stat. Soc. Ser. B*, vol. 39, no. 1, pp. 1–38, 1977.

[9] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*, vol. 274. 1997.

[10] M. Jamshidian and P. M. Bentler, "ML Estimation of Mean and Covariance Structures with Missing Data Using Complete Data Routines," *J. Educ. Behav. Stat.*, vol. 24, no. 1, pp. 21–24, 1999.

[11] A. Gelman and J. Hill, "Missing-data imputation," in *Data Analysis Using Regression and Multilevel/Hierarchical Models*, Cambridge: Cambridge University Press, 2006, pp. 529–544.

[12] S. Van Buuren, *Flexible Imputation of Missing Data*. Chapman & Hall/CRC, Boca Raton, FL, 2012.

[13] C. K. Enders, *Applied missing data analysis*. 2010.

[14] E. Eirola, A. Lendasse, V. Vandewalle, and C. Biernacki, "Mixture of Gaussians for distance estimation with missing data," *Neurocomputing*, vol. 131. pp. 32–42, 2014.

[15] T. P. Morris, "Practical use of multiple imputation," 2014.

[16] J. Siddique and T. R. Belin, "Multiple imputation using an iterative hot-deck with distance-based donor selection," *Stat. Med.*, vol. 28, no. July 2006, pp. 221–239, 2008.

[17] J. Siddique and O. Harel, "MIDAS : A SAS Macro for Multiple Imputation Using Distance-Aided Selection of Donors," *J. Stat. Softw.*, vol. 29, no. 9, 2009.

[18] J. Siddique and T. R. Belin, "Using an Approximate Bayesian Bootstrap to Multiply Impute Nonignorable Missing Data," vol. 6, no. 2, pp. 247–253, 2008.

[19] M. Paik, "Fractional Imputation," 2009.

[20] M. S. Srivastava and M. Dolatabadi, "Multiple imputation and other resampling schemes for imputing missing observations," *J. Multivar. Anal.*, vol. 100, no. 9, pp. 1919–1937, 2009.

[21] A. Azzalini and A. W. Bowman, "A Look at Some Data on the Old Faithful Geyser," *J. R. Stat. Soc.*, vol. 39, no. 3, pp. 357–365, 1990.

[22] C. K. Enders, *Applied Missing Data Analysis*. 2010.

[23] K. Groothuis-oudshoorn, "Journal of Statistical Software MICE : Multivariate

Imputation by Chained," vol. VV, no. Ii.
[24] D. Mcneish, "Missing data methods for arbitrary missingness with small samples," *J. Appl. Stat.*, vol. 44, no. 1, pp. 24–39, 2017.
[25] Y. Dong and C.-Y. J. Peng, "Principled missing data methods for researchers," *SpringerPlus*, vol. 2. Cham, 2013.
[26] A. N. Baraldi and C. K. Enders, "An introduction to modern missing data analyses," *J. Sch. Psychol.*, vol. 48, no. 1, pp. 5–37, 2010.
[27] Y. Dong and C.-Y. J. Peng, "Principled missing data methods for researchers.," *Springerplus*, vol. 2, no. 1, p. 222, 2013.
[28] J. S. J.N.K. Rao, "Jackknife Variance Estimation with Survey Data Under Hot Deck Imputation," *Biometrika Trust*, vol. 79, no. 4, pp. 811–822, 1992.
[29] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to Linear Regression Analysis*, vol. 49. 2001.
[30] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to Linear Regression Analysis, Solutions Manual (Wiley Series in Probability and Statistics)*. New York, NY, USA: Wiley-Interscience, 2007.
[31] R. J. A. Little and D. B. Rubin, *Statistical analysis with missing data*. 1987.
[32] K. F. Edwards, C. a. Klausmeier, and E. Litchman, "Evidence for a three-way trade-off between nitrogen and phosphorus competitive abilities and cell size in phytoplankton," *Ecology*, vol. 92, no. 11, pp. 2085–2095, 2011.
[33] R. A. Fisher, "On the mathematical foundations of theoretical statistics," *Philos. Trans. R. Soc. London A Math. Phys. Eng. Sci.*, vol. 222, no. 594–604, pp. 309–368, 1922.
[34] P. D. Allison, "Handling Missing Data by Maximum Likelihood," in *SAS Global Forum 2012 Statistics and Data Analysis*, 2012, pp. 1–21.
[35] A. P. Dempster, A. P. Dempster, N. M. Laird, N. M. Laird, D. B. Rubin, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Stat. Soc. Ser. B*, vol. 39, no. 1, pp. 1–38, 1977.
[36] J. a. Bilmes, "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," *Int. Comput. Sci. Inst.*, vol. 4, no. 510, p. 126, 1998.
[37] Z. Ghahramani and M. I.Jordan, "Supervised learning from incomplete data via an em approach," *Adv. Neural Inf. Process. Syst.*, vol. XXXIII, no. 2, pp. 81–87, 1994.
[38] E. Eirola, A. Lendasse, V. Vandewalle, and C. Biernacki, "Mixture of Gaussians for distance estimation with missing data," *Neurocomputing*, vol. 131, pp. 32–42, 2014.
[39] E. Eirola, "Machine learning methods for incomplete data and variable selection," Aalto University, 2014.
[40] J. E. Gentle, G. J. McLachlan, and T. Krishnan, "The EM Algorithm and Extensions.," *Biometrics*, vol. 54, no. 1, p. 395, 1998.
[41] G. McLachlan and D. Peel, *Finite Mixture Models*. 2000.
[42] B. Efron, "Bootstrap Methods : Another Look at the Jackknife," *Ann. Stat.*, vol. 7, no. 1, pp. 1–26, 1979.
[43] R. Y. Liu, "Bootstrap procedures under some non-i.i.d. models," *Ann. Stat.*, vol. 16, no. 4, pp. 1696–1708, 1988.
[44] E. Hardle, W. and Mammen, "Comparing Nonparametric versus Parametric Regression Fits," *Ann. Stat.*, vol. 21, no. 1926–1947, 1993.
[45] M. Zadkarami, "Bootsrapping: A Nonparametric Approach to Identify the Effect of Sparsity of Data in the Binary Regression Models," *J. Appl. Sci.*, vol. 8, no.

17, pp. 2991–2997, 2008.
[46] A. Stein and P. Hall, “The Bootstrap and Edgeworth Expansion.,” *Stat.*, vol. 45, no. 4, p. 532, 1996.
[47] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*, no. 57. Boca Raton, Florida, USA: Chapman & Hall/CRC, 1993.
[48] A. D. W. K. and B. Moshe, “A Three-step Method for Choosing the Number of Bootstrap Repetitions,” *Econometrica*, vol. 68, no. 1, pp. 23–51.
[49] F. Cribari-Neto and S. G. Zarkos, “Bootstrap methods for heteroskedastic regression models: evidence on estimation and testing,” *Econom. Rev.*, vol. 18, no. 2, pp. 211–228, 1999.
[50] C. F. J. Wu, “Jackknife , Bootstrap and Other Resampling Methods in Regression Analysis,” *Ann. Stat.*, vol. 14, no. 4, pp. 1261–1295, 1986.
[51] C. F. J. Wu, “Jackknife, Bootstrap and Other Resampling Methods in Regression Analysis,” *Ann. Stat.*, vol. 14, no. 4, pp. 1261–1295, 1986.
[52] T. Masayoshi, “Multiple Ratio Imputation by the EMB Algorithm,” 2015.
[53] B. Grün and F. Leisch, “Finite mixture model diagnostics using the parametric bootstrap,” *Proc. Jr. Sci. Conf.*, 2006.
[54] B. Gr and F. Leisch, “Finite Mixture Model Diagnostics Using Resampling Methods,” *Vignette R Packag. flexmix*, 2010.
[55] C. <<Algorithms for A. I. Huddersfield, “A bootstrap algorithm for mixture models and interval data in inter-comparisons,” in *Proceedings 2002*, 2001, pp. 138–145.
[56] A. O’Hagan, T. B. Murphy, and I. C. Gormley, “Investigation of Parameter Uncertainty in Clustering Using a Gaussian Mixture Model Via Jackknife, Bootstrap and Weighted Likelihood Bootstrap,” pp. 1–25, 2015.
[57] S. D. Foster, N. A. Hill, and M. Lyons, “Ecological grouping of survey sites when sampling artefacts are present,” *J. R. Stat. Soc. Ser. C Appl. Stat.*, vol. 66, no. 5, pp. 1031–1047, 2017.
[58] P. Schlattmann, “On bootstrapping the number of components in finite mixtures of Poisson distributions,” *Stat. Comput.*, vol. 15, no. 3, pp. 179–188, 2005.
[59] A. Colin Cameron, J. B. Gelbach, and D. L. Miller, “Bootstrap-based improvements for inference with clustered errors,” *Rev. Econ. Stat.*, vol. 90, no. 3, pp. 414–427, 2008.
[60] R. Davidson and J. G. MacKinnon, “Bootstrap Confidence Sets with Weak Instruments,” *Econom. Rev.*, vol. 33, no. 5–6, pp. 651–675, 2014.
[61] J. G. Mackinnon and M. D. Webb, “Wild Bootstrap Inference forWildly Different Cluster Sizes,” 2015.
[62] K. Manavi, S. Tashakkori, and L. Tapia, “Gaussian Mixture Models with Constrained Flexibility for Fitting Tomographic Tilt Series,” *Proc. 8th ACM Int. Conf. Bioinformatics, Comput. Biol. Heal. Informatics*, pp. 710–715, 2017.
[63] D. B. Rubin, “Statistical matching using file concatenation with adjusted weights and multiple imputations,” *J. Bus. Econ. Stat.*, vol. 4, no. 1, pp. 87–94, 1986.
[64] R. J. A. Little, “Missing-Data Adjustments in Large Surveys,” *J. Bus. Econ. Stat.*, vol. 6, no. 3, p. 300, 1988.
[65] P. Gaffert, F. Meinfelder, and V. Bosch, “Towards an MI-proper predictive mean matching,” 2016.
[66] E. Fix and J. L. Hodges, *Discriminatory Analysis: Nonparametric Discrimination, Consistency Properties*, no. v. 1-2. USAF School of Aviation Medicine, 1951.
[67] T. Cover and P. Hart, “Nearest neighbor pattern classification,” *IEEE Trans. Inf. Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[68] E. M. Burns, “Multiple and Replicate Item Imputation in a Complex Sample Survey,” in *Sixth Annual Research Conference*, 1990, pp. 655–665.
[69] G. Vink, L. E. Frank, J. Pannekoek, and S. van Buuren, “Predictive mean matching imputation of semicontinuous variables,” *Stat. Neerl.*, vol. 68, no. 1, pp. 61–90, 2014.
[70] R. R. Andridge, “A Review of Hot Deck Imputation for Survey Non-response,” *Int Stat Rev*, vol. 78, no. 1, pp. 40–64, 2011.
[71] D. F. Heitjan and R. J. A. Little, “Multiple Imputation for the Fatal Accident Reporting System,” *Wiley R. Stat. Soc.*, vol. 40, no. 1, pp. 13–29, 1991.
[72] S. van Buuren and K. Groothuis-Oudshoorn, “**mice** : Multivariate Imputation by Chained Equations in *R*,” *J. Stat. Softw.*, vol. 45, no. 3, 2011.
[73] F. E. H. Jr and C. Dupont, “Hmisc: Harrell Miscellaneous,” 2015.
[74] F. Meinfelder and T. Schnapp, “BaBooN: Bayesian Bootstrap Predictive Mean Matching - Multiple and Single Imputation for Discrete Data,” 2015.
[75] D. McNeish, “Exploratory Factor Analysis With Small Samples and Missing Data,” *J. Pers. Assess.*, vol. 99, no. 6, pp. 637–652, 2017.
[76] G. Vink, G. Lazendic, and S. Van Buuren, “Partitioned predictive mean matching as a multilevel imputation technique,” *Psychol. Test Assess. Model.*, vol. 5, no. 4, pp. 1–16, 2015.
[77] T. P. Morris, I. R. White, and P. Royston, “Tuning multiple imputation by predictive mean matching and local residual draws,” *BMC Med. Res. Methodol.*, vol. 14, no. 1, 2014.
[78] G. B. Durrant and C. Skinner, “Using data augmentation to correct for non-ignorable non-response when surrogate data are available: An application to the distribution of hourly pay,” *J. R. Stat. Soc. Ser. A Stat. Soc.*, vol. 169, no. 3, pp. 605–623, 2006.
[79] K. J. Lee and J. B. Carlin, “Multiple imputation in the presence of non-normal data,” *Stat. Med.*, vol. 36, no. 4, pp. 606–617, 2017.
[80] M. Di Zio and U. Guarnera, “Semiparametric predictive mean matching,” *AStA Adv. Stat. Anal.*, vol. 93, no. 2, pp. 175–186, 2009.
[81] J. L. Bentley, “A Survey of Techniques for Fixed Radius Near Neighbor Searching.,” Stanford University, Stanford, CA, USA, 1975.
[82] “Classification Using Nearest Neighbors,” *The MathWorks, Inc*, 2018. [Online]. Available: https://uk.mathworks.com/help/stats/classification-using-nearest-neighbors.html#bsfsm9c. [Accessed: 24-Aug-2018].
[83] J. H. Freidman, J. L. Bentley, and R. A. Finkel, “An Algorithm for Finding Best Matches in Logarithmic Expected Time,” *ACM Trans. Math. Softw.*, vol. 3, no. 3, pp. 209–226, 1977.
[84] H. Crc, “Flexible Imputation of Missing Data Multiple Imputation in Practice Your Second Course Why a book ? Rising popularity of multiple imputation Why a book ? Mean imputation Estimand Goal of multiple imputation Main steps used in multiple imputation Pooled es,” 2012.