Markov Modelling of Disease Progression in the Presence of Missing Covariates

Loamie Kotzé

Thesis presented in partial fulfilment
of the requirements for the degree of
Master of Commerce
at the University of Stellenbosch

department of Statistics and Actuarial Science

Supervisor: Prof PJ Mostert

April 2019

DECLARATION

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: April 2019

ABSTRACT

Breast cancer is a very prevalent cancer amongst women. The stages of breast cancer are influenced by characteristics such as age, hormone receptor statuses, HER2 status and staging information (TNM staging). This study aims to model the progression of breast cancer using a multi-state model which evaluates three pre-defined stages of the disease. A secondary aim is to determine an appropriate technique to impute missing data in the covariates.

The disease progression can be modelled by using multi-state models and it is of interest to analyse the effect of different risk factors on the transitions between the states. The variable of interest can be seen as the state of the individual at that time point. The transition intensities of the multi-state model provides the hazards of moving from one state to another and can be used to calculate the mean sojourn time in any given state.

A combination of claims data and authorisation treatment request data were obtained from Isimo Health for 393 breast cancer patients. Based on this, a dataset was simulated using the *TPmsm* package in R statistical programming. The simulated data were used to test two imputation techniques, one based on chained equations and one based on random forests, for the missing data present in the covariates. The latter technique performed the best based on several performance measures, and was used to impute the dataset from Isimo Health. Thereafter, a multi-state Markov model was fitted to the imputed data with three pre-defined states including curative (receive treatment with the intent to cure), non-curative (receive treatment with the intent to provide improved survival or symptom control) and death. It was observed that the Markov assumption does not hold and, therefore a semi-Markov model was fitted to the data.

The findings showed that only one of the covariates, namely staging, had a significant effect on the transition probabilities. This is only the case for the transition between the non-curative and death state. Covariates as a whole, did have a significant effect on the transitions from curative to non-curative and non-curative to death. However, there was no significant effect on the transition from curative to death.

It can be concluded, based on statistical measures, that the *missForest* package efficiently imputes missing covariates before modelling disease progression with multi-state models using the *p3state.msm* package.

OPSOMMING

Borskanker is ’n hoogs prevalente kanker onder vrouens. Die graad van borskanker word beïnvloed deur eienskappe soos hormoon reseptor statusse, HER2 status en die graad van die kanker (TNM gradering). Die studie beoog om die progressie van borskanker te modelleer deur gebruik te maak van ’n multi-staat model met drie voorafgedefinieerde state. Dit word ook verlang om ’n geskikte tegniek te verkry om ontbrekende data van die kovariate te verkry.

Multi-staat modelle word gebruik om die progressie van die borskanker te modelleer en dit is wenslik om die effek van verskillende risiko faktore op die oorgangsintensiteite tussen state te analiseer. Die veranderlike van belang kan gesien word as die staat waarin die individu op daardie oomblik bevind is. Die oorgangsintensiteite van multi-staat modelle verskaf die gevaarkoers om van een staat na die volgende te beweeg. Die oorgangsintensiteite kan ook gebruik word om die gemiddelde verblyftyd in enige gegewe staat te bereken.

’n Kombinasie van eise-data en magtigingsbehandeling versoek-data was verkry vanaf Isimo Health vir 393 borskanker pasiënte. Die *TPmsm* pakket in R was gebruik om ’n datastel te simuleer gebasseer op die Isimo Health data. Die gesimuleerde data was gebruik om verskillende imputeringstegnieke te toets om die ontbreekte data in die kovariate in te vul. Die imputeringstegniek gebaseer op *Random Forests* het die beste gevaar en was dus gebruik om die Isimo Health datastel te imputeer. Die *missForest* pakket in R was gebruik om die imputering te doen. Na die imputering, is ’n multi-staat Markov model gepas met drie voorafgedefinieerde state naamlik genesend (ontvang behandeling met die doel om te genees), nie-genesend (ontvang behandeling met die doel om oorlewing te verbeter of simptoombeheer) en afsterwing. Die Markov aanname geld nie en dus word ’n semi-Markov model aan die data gepas.

Die bevindings wys dat die graad van die kanker die enigste kovariaat is wat ’n statisties betekenisvolle effek op die oorgangswaarskynlikhede het. Dit is slegs die geval vir die oorgang tussen die nie-genesende en afsterwing staat. Die kovariate in geheel het ’n statisties betekenisvolle effek op die oorgangswaarskynlikhede van genesend na nie-genesend en nie-genesend na afsterwing. Dit het nie ’n statisties betekenisvolle effek op die oorgang van genesend na afsterwing nie.

Die *missForest* pakket is die mees geskikte pakket om kovariate met ontbrekende waardes te imputeer. Hierdie gevolgtrekking is gebaseer op verskillende statistiese maatstawwe. Daarna kan die *p3state.msm* pakket gebruik word om die progressie van borskanker te modelleer.

ACKNOWLEDGEMENTS

The author of this research assignment would like to acknowledge a few individuals whom were fundamental to the realisation of this thesis.

Firstly, the author acknowledges Isimo Health and MSH for giving her the opportunity to do her Masters degree as well as generously providing a dataset for the purposes of this thesis.

Secondly, the author deeply appreciates the knowledge and guidance given by the author's supervisor Prof PJ Mostert from the Actuarial and Mathematical Sciences Department of the University of Stellenbosch.

Thirdly, the author would like to thank her colleagues at Isimo Health for the inspiration, support and encouragement provided by them as well as her family and close friends for all the support, love and encouragement provided by them during the process of this thesis. It would not have been possible without them.

Lastly, the author was deeply inspired by the life of Esther Venter, her second cousin. She was diagnosed with metastatic breast cancer early in 2017 and peacefully passed on the 7th of September of the same year.

Dedicated to:
The person I am now one step closer to being.

NOTATION

The various notations and symbols used throughout the thesis document are shown and defined below.

$X(t)$	State occupied by stochastic process at time $t \geq 0$
p_{ij}	Transition probability for transition from state i to j
$D(x, i, t)$	Number of persons aged x with breast cancer i at time t
$N(x, t)$	Total projected population aged x at time t
$P(x, i, t)$	Breast cancer prevalence rate of level i, aged x and projected at time t
q_{ij}	Transition intensity of moving from state i to state j
Q	Transition intensity matrix
P	Transition probability matrix
X_t	Observation history of the process up to time t
$z(t)$	Time-varying explanatory variables
X_{obs}	Observed data
X_{mis}	Missing data
$p(Y\|\theta)$	Density function of complete dataset

ACRONYMS

The acronyms used throughout the thesis document are shown below.

AI	Aromatase inhibitors
DCIS	Dual carcinoma in situ
EM	Expectation-Maximisation
ER	Estrogen receptor
FIML	Full information maximum likelihood
HER2	Human epidermal growth factor receptor 2
KNN	K-nearest neighbor
MAR	Missing at random
MCAR	Missing completely at random
MCMC	Monte Carlo Markov Chain
MI	Multiple imputation
MNAR	Missing not at random
MSM	Multi-state model
NI	Non-ignorable
NRMSE	Normalised root mean squared error
PFC	Proportion of falsely classified
PR	Progesterone receptor
RF	Random Forest
SIR	Sampling importance resampling

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 Introduction

Breast cancer is the second most common form of cancer in the United States (Grayson, 2012). The Oxford English Dictionary (2017) defines breast cancer as a cancer arising in the mammary gland. Usually it occurs in the mammary gland in females, but occasionally can occur in the rudimentary tissue of the male (The Oxford English Dictionary, 2017). The incidence of breast cancer based on the Isimo Health data was 704 and 685 per 100 000 people for 2016 and 2017, respectively. The prevalence was 1 173 and 1 194 per 100 000 people for 2016 and 2017, respectively.

Although breast cancer is often seen as one disease, there are many different types of breast cancers. The commonality between these cancers is that they typically start in the breast. Breast tumours can be invasive or non-invasive and the prognosis is often affected by characteristics such as the hormone receptor and human epidermal growth factor receptor 2 (HER2) status. According to Grayson (2012), women with different types of breast cancer react differently to treatment. The worst breast cancer prognosis is when the cancer has already metastasised[1] at the time of diagnosis (Grayson, 2012).

According to Komen (2017), dual carcinoma in situ (DCIS) is a non-invasive breast cancer. This is the case when the milk ducts have not spread to nearby breast tissue. This non-invasive breast cancer can develop into invasive breast cancer over time if it is not treated. Invasive breast cancer is cancer that has spread from the original location into another part of the breast tissue

[1] The definition of metastasise per the Cambridge English Dictionary (2017): If cancer cells metastasise, they spread to other parts of the body and cause tumours to grow there.

as well as to the lymph nodes (National Cancer Institute, 2018). Consequently, invasive breast cancer has a poorer prognosis than DCIS.

Hormone receptors are breast cancer cells that have special proteins inside which needs estrogen and/or progesterone to grow (Komen, 2017). When breast cancers have many hormone receptors, the cancers are called hormone receptor positive cancers. Hormone receptor positive can mean either estrogen receptor (ER) positive or progesterone receptor (PR) positive. These statuses strongly influence the course of treatment and therefore the cost of treatment.

Almost 70 percent of breast cancers are hormone receptor positive. Breast cancers can be treated with hormone therapies if they are hormone receptor positive. Hormone therapies include tamoxifen and the aromatase inhibitors (AI), namely anastrozole (Arimidex), letrozole (Femara) and exemestane (Aromasin) (Komen, 2017). Most breast cancers that are ER positive also tend to be PR positive. In addition, breast cancers that are ER negative tend to be PR negative. A breast cancer that is ER positive can be PR negative, although this is uncommon (Komen, 2017).

Hormone therapies slow the growth of hormone receptor positive tumours by preventing the cancer cells from getting the hormones they need to grow. Tamoxifen and some other hormone therapies attach to the receptor in the cancer cell and block the estrogen from attaching to the receptor. Other hormone therapies such as AI, lower the level of estrogen in the body so that the cancer cells cannot get the estrogen they need to grow (Komen, 2017).

According to Komen (2017), the hormone receptor status is related to the chance of breast cancer recurrence. Hormone receptor positive tumours have a lower chance of breast cancer recurrence than hormone receptor negative tumours in the first five years after diagnosis.

HER2 is a protein that appears on the surface of some breast cancer cells. HER2/neu and ErbB2 are alternative names for HER2. When a breast cancer is HER2 positive the cancer has numerous HER2 protein. In this case it is referred to HER1 over expression, while HER2 negative has little or no HER2 protein (Komen, 2017). Almost 15 percent of newly diagnosed breast cancers are HER2 positive. The status of HER2 also effects the appropriate course of treatment.

The aim of the study is to model the progression of breast cancer by using multi-state models and to determine an appropriate technique to impute missing data present in the covariates. Missing data is frequently present in the covariates when analysing clinical datasets. The disease progression can be modelled using multi-state models and it is of interest to determine the effect of different covariates on the transition intensities. A dataset obtained from Isimo Health, containing 393 breast cancer patients, was used to simulate a dataset to test imputation techniques on the covariates. Thereafter, the best performing imputation techniques were used to impute the Isimo Health dataset. The imputed dataset was used to fit a multi-state Markov model for the progression of breast cancer.

1.2 Problem Statement

Multi-state models can be used when confronted with panel data. Panel data are also referred to as longitudinal or cross-sectional time-series data. Multi-state models are used in medical studies where the disease status of patients is documented over time. All the information included in the dataset is anonymous. The data are a combination of authorisation data from eAuth and claims data, from cancer patients treated by providers belonging to the Independent Clinical Oncology Network (ICON). eAuth is an authorisation system developed by ICON.

Panel data are simulated based on the data obtained from Isimo Health. The simulated data are used to investigate missingness and to choose an appropriate imputation technique. Different imputation techniques will be considered and two imputation techniques will be tested. The imputation technique performing the best will then be chosen to impute the Isimo Health dataset.

The imputed Isimo Health data will then be used to fit a multi-state Markov model for disease progression of breast cancer over time. The researcher will be looking at three pre-defined disease states namely curative, non-curative and death. The curative state is defined to be when the patient is treated with the intent to cure the cancer. The non-curative state is defined as being the state when the patient is treated without the intent to cure, but rather for improved survival or symptom control. The death state is entered when the patient is deceased.

1.3 Importance of the Study

This study will be beneficial to both funders and clinicians. For funders it will be useful to predict how patients progress from diagnosis to death. For clinicians it will be beneficial since the clinicians will be able to see how patients progress from certain states and in which way clinical factors such as HER2 status, hormone receptor status, age and other demographics influence the disease progression. Ultimately, the importance of this study is to build a platform to be able to do further research to enable building a forecasting tool to predict the total cost of cancer.

1.4 Research Design and Methodology

1.4.1 Sampling and data collection

A dataset is collected from ICON through Isimo Health. The dataset is a combination of authorisation data from eAuth and claims data provided by medical schemes belonging to the ICON network. Only patients who matched between the two data sources were included, since the model requires both accurate cost data and more granular clinical data.

1.4.2 Data analysis

Structured Query Language (SQL) was used to extract data from the data sources, while R programming was used to perform statistical analysis on the extracted data. The R packages *TPmsm*, *Metrics*, *mice*, *missForest, msm* and *p3state.msm* were used in the statistical analysis in R.

The disease progression is modelled by using multi-state models. It is often of interest to analyse the effect of different risk factors on the transition rates. A dataset was simulated using the *TPmsm* package. The *mice* and *missForest* packages were used to test the two different imputation techniques. Lastly, the *p3state.msm* package were used to fit a multi-state model to the data acquired from Isimo Health.

1.5 Chapter Outline

Chapter Two provides a literature review of the literature that is available on multi-state Markov models and imputation techniques. Chapter Three gives a thorough description of the data received from Isimo Health as well as the process of transforming and cleaning the dataset. Chapter Four describes the simulation approach used to simulate the data with the R package *TPmsm*.

In Chapter Five, the R packages available for imputation techniques are described. Thereafter two of the techniques are applied to impute the simulated dataset and the imputation technique with the best performance is selected to be used in Chapter Six on the real-world dataset from Isimo Health.

The chosen imputation technique from Chapter Five is used to impute the dataset from Isimo Health and the multi-state Markov model is fitted to the imputed dataset in Chapter Six. Chapter Six also gives a summary of the findings. Chapter Seven provides a conclusion, the limitations and the future opportunities for research to build on this thesis.

CHAPTER 2

LITERATURE REVIEW ON MULTI-STATE MODELS AND IMPUTATION

2.1 Introduction

The concept of a multi-state model as well as multi-state Markov models is discussed in detail in this chapter. Thereafter, the idea of missing data, the different types of missing data and ways of handling missing data are discussed.

In order to estimate the total cost of care required by breast cancer patients, it is necessary to project the cancer patient population to the year for which the forecast is needed. Several methods exist, amongst them a most frequently used method, namely the projection of the prevalence rates (Siegel, 2002). The projected prevalence rates are applied to the total projected population in this method. Prevalence rates indicate the proportion of persons who have breast cancer at a given time with respect to the total population. In this thesis it will be with respect to the total insured population. The number of persons aged x, with breast cancer i at time t is given as

$$D(x, i, t) = N(x, t) \times P(x, i, t), \tag{2.1}$$

where $N(x, t)$ is the total projected population aged x at time t and $P(x, i, t)$ is the breast cancer prevalence rate of level i (the specific type of breast cancer including the severity), aged x and projected at time t. The projected prevalence rates can be either static or dynamic (varying) prevalence rates.

This method has been widely used but is not necessarily suitable to model the disease progression since a more flexible model taking into account different states of health and the dependency with other factors would be more appropriate. Multi-state models are alternative but more comprehensive model types to consider. Multi-state models are the most common choice of model to analyse longitudinal survival data (Amorim et al., 2011). This technique is widely used in various fields such as medicine, physics, biology, economics and others.

A multi-state model is a stochastic process which occupies one of a set of discrete states, at any time point (Hougaard, 1999). Different health states can be defined in its simplest form as healthy, sick or diseased. The states may represent different health situations of the subject (Amorim et al., 2011). A transition or event refers to a change of state which corresponds for example to an outbreak of disease or even death. The state structure and the form of the hazard function for each possible transition is specified in the full statistical model (Hougaard, 1999).

The possibility of projecting the number of persons who will be in a certain state of cancer, based on transition probabilities or intensity rates between states, is the greatest utility of these models when dealing with cancer.

There are a few requirements when building a projection model:

- Baseline estimates of the level of cancer of the current population will need to be estimated.
- Transition rates between states need to be determined.
- Assumptions need to be formulated regarding transition rates.
- Projecting the number of persons with cancer with the need of treatment under different scenarios.

Let $X(t)$ denote the state occupied by the stochastic process at a specific time, $t \geq 0$. According to Amorim et al. (2011), the transition probability for the two states i and j where $s < t$, is represented by

$$p_{ij}(s,t) = P(X(t) = j|X(s) = i).$$

The estimation of the transition probability $p_{ij}(s,t)$ attracted much interest since it allows for the long-term prediction of the process (Amorim et al., 2011). A non-parametric estimator of $p_{ij}(s,t)$ for Markov models was introduced by Aalen and Johansen (1978). The Markov assumption requires that the future evolution of the process is independent of the state previously visited as well as independent of the times of the transition amongst the states given the present state of the process (Amorim et al., 2011).

Ideally, the transition probabilities should be obtained directly from the data. An alternative way of calculation the transition probabilities is by using the Markov model approach proposed by Sullivan (1971). The structure of the multi-state Markov model is given in Figure 2.1. The multi-state Markov model will thoroughly be discussed in subsequent sections.

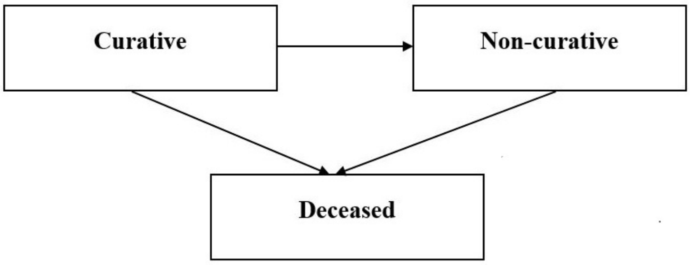


Figure 2.1 Transitions in the three-state Markov model

2.2 Multi-state Models

According to Mafu (2014), a multi-state model (MSM) is modelling time for event data where all the individuals start in one or more states, and eventually may end up in one or several absorbing state(s). It has also been defined as a process in which an individual move through a series of states in continuous time. A longitudinal dataset or panel dataset can be observed and investigated with a MSM. A panel dataset is defined when a sample of n subjects are followed over time and multiple observations on each subject is made (Mafu, 2014). Some of the individuals may also be censored before they reach an absorbing state. Censored observations cause some model difficulties and therefore need to be accounted for.

Mafu (2014) states that when considering MSMs, it is desired to investigate the effect of different risk factors. Therefore, in a MSM, the relationship between different predictors and the outcome or variable of interest is studied. The variable of interest can be seen as the state that each individual occupies at each point in time. The transition intensities, in MSMs, provide the hazards of moving from one state to another (Mafu, 2014). The transition intensities can also be used to calculate the mean sojourn time in any given state. In this section, the Markov process, the transition probability matrix, the transition intensity matrix, sojourn time and Markov chain properties are thoroughly discussed.

2.2.1 Markov process

The Markov process, $X(t)$, has by definition no after-effect properties. Zhang and Zhang (2009) explains that the after-effect properties imply that the state of the subject at time $t > t_m$ is only dependent on the state at time t_m in some process given that the state is known at time t_m, but is independent of the state before time t_m. Therefore, a Markov process is a stochastic process in which the future knowledge of the process is only provided by the current state of the process (Mafu, 2014).

Andrey Markov (1906) first introduced the Markov chain model. This type of model has been applied in various fields including physics, economics, finance and social sciences (Cong, 2010). According to Cong (2010), this model provides an efficient way of describing a process in which an individual move through a series of states in continuous time. Consequently, it has also been used extensively in the field of healthcare, where the progression of disease is of importance to both patients and clinicians (Cong, 2010).

Cong (2010) explains that the Markov chain model describes a finite or infinite random process

$$\mathbf{X} = \{X_t\}_{t\geq 1} = \{X_1, X_2, ...\}.$$

The Markov model considers the dependencies between the $X_i's$. This is the greatest difference between the independent and identically distributed (i.i.d.) model, which assumes the independency of the sequence of events $X_i's$, and the Markov model (Cong, 2010).

Let $\mathbf{X} = \{X_1, X_2, ..., X_N\}$ be a random process of random variables taking on values in a discrete state space $E = \{1, 2, ..., e\}$ and X_t be the state of the process of an individual at time t. Now, let the realisation of the entire history of the process up to and including time t be

$$\{X_t = x_t, X_{t-1} = x_{t-1}, ..., X_1 = x_1\},$$

where $x_t, x_{t-1}, ..., x_1$ is a sequence of states at different time points. A random process is classified as a Markov Chain if it satisfies the following condition:

$$P(X_{t+1} = x_{t+1}|X_t = x_t, X_{t-1} = x_{t-1}, ..., X_1 = x_1) = P(X_{t+1} = x_{t+1}|X_t = x_t), \qquad (2.2)$$

for every sequence $x_1, ..., x_t, x_{t+1}$ of the elements in E and every time point $t \geq 1$ (Cong, 2010).

In the stochastic process, the system will enter a state, spend time in the state (referred to as the sojourn time) and then move to another state where it will spend another sojourn time in that state (Mafu, 2014).

2.2.2 Transition probability matrix

Let p_{ij} be the transition probability of the system moving from state i to state j. The transition probability of moving from state i to state j at time t is defined as

$$p_{ij}(t) = p(X_{t+1} = j|X_t = i). \qquad (2.3)$$

In the case where the transition probabilities are independent of time, $p_{ij}(t)$ can be written as p_{ij} and then the Markov chain is referred to as time-homogeneous (Cong, 2010).

The transition probability matrix of a multi-state process at time t, is an $e \times e$ matrix and can be expressed as

$$P = P(t) = \begin{bmatrix} p_{11}(t) & p_{12}(t) & ... & p_{1e}(t) \\ p_{21}(t) & p_{22}(t) & ... & p_{2e}(t) \\ ... & ... & ... & ... \\ p_{e1}(t) & p_{e2}(t) & ... & p_{ee}(t) \end{bmatrix}, \qquad (2.4)$$

where E is the discrete state space $E = \{1, 2, ..., e\}$.

The transition probability matrix (2.4), is classified as a stochastic matrix since for any row i, $\sum_j p_{ij} = 1$ is true (Mafu, 2014). Therefore, the probabilities in each of the rows of the transition probability matrix add up to one (Cong, 2010). The entries of the probability transition matrix have been defined in (2.3) and these entries define the transition or movement probabilities of individuals through states (Mafu, 2014). The matrix defined in (2.4), is the transition probability matrix with its elements providing the probability of being in state j at time $t + 1$, conditional on being in state i at time t. The transition probability matrix is time dependent and is therefore denoted as $P(t)$ instead of P (Mafu, 2014). In time homogeneous Markov models, the dependency of t is omitted.

All the probabilities in the transition probability matrix must be greater than or equal to zero, that is $p_{ij} \geq 0, \forall j, i\epsilon\{1, ..., E\}$, and each row must sum to one $\sum_{j}^{e} p_{ij} = 1, \forall i, j\ \epsilon\{1, ..., e\}$ (Mafu, 2014).

For illustration purposes, consider a 3-state model with transition probability matrix

$$P(t) = \begin{bmatrix} p_{11}(t) & p_{12}(t) & p_{13}(t) \\ p_{21}(t) & p_{22}(t) & p_{23}(t) \\ p_{31}(t) & p_{32}(t) & p_{33}(t) \end{bmatrix}.$$

Since each row must sum to one, i.e. $p_{11}(t) + p_{12}(t) + p_{13}(t)) = 1$ and each probability must be greater than or equal to zero, e.g. $p_{12}(t) \geq 0$.

For an n-step state transition probability matrix, let $p_{ij}(n)$ be the conditional probability that the process will be in state j after precisely n transitions, given that it is in state i at present (Ibe, 2009). Therefore,

$$\begin{array}{c} p_{ij}(n) = P[X_{m+n} = j | X_m = i] \\ p_{ij}(0) = \begin{cases} 1, i = j \\ 0, i \neq j \end{cases} \\ p_{ij}(1) = p_{ij} \end{array}.$$

For illustration purposes, consider a two-step transition probability $p_{ij}(2)$, which is defined as

$$p_{ij}(2) = P[X_{m+2} = j | X_m = i].$$

If $m = 0$, $p_{ij}(2) = \sum_{k} p_{kj} p_{ik} = \sum_{k} p_{ik} p_{kj}$, where the summation is taken over all possible intermediate states k. Therefore, the probability of starting in state i and being in state j after the second transition is the probability that the individual first goes from state i to an intermediate state k and then to state j. The probability $p_{ij}(n)$ is the ij^{th} entry in the probability matrix P^n. This probability matrix is given as

$$P^n = \begin{bmatrix} p_{11}(n) & p_{12}(n) & ... & p_{1N}(n) \\ p_{21}(n) & p_{22}(n) & ... & p_{2N}(n) \\ ... & ... & ... & ... \\ p_{N1}(n) & p_{N2}(n) & ... & p_{NN}(n) \end{bmatrix},$$

with N representing the number of states. If $n = 1$, this matrix is referred to as the one-step probability matrix. The n-step probability matrix is obtained by multiplying the transition probability matrix by itself n times (Mafu, 2014). As $n \longrightarrow \infty$, the transition probability matrix

$p_{ij}(n)$ does not depend on i anymore (Mafu, 2014) and consequently, $P(X(n) = j)$ approaches a constant. In the Markov chain, if the limit exists, the limiting-state probabilities is defined as

$$\lim_{n \longrightarrow \infty} P(X(n) = j) = \pi_j, j = 1, 2, ..., N. \tag{2.5}$$

If the limiting-state probabilities exist but are independent of the initial state, (2.5) simplifies to

$$\lim_{n \longrightarrow \infty} p_{ij}(n) = \pi_j = \lim_{n \longrightarrow \infty} \sum_k p_{ik}(n-1)p_{kj} = \sum_k \pi_k p_{kj}.$$

According to Mafu (2014), the limiting-state probability vector $\underline{\pi}= (\pi_1, \pi_2, ..., \pi_N)$ will result in $\pi_j = \sum \pi_k p_{kj}$ where $j = 1, ..., N$, $\underline{\pi}= \underline{\pi}P$ and $\sum_{j=1}^{N} \pi_j = 1$.

The transition probability matrix must follow the same operation rules as the conventional matrix and will therefore satisfy the property $P^k = P^{(k-1)} * P = P^k$ (Zhang and Zhang, 2009).

The average transition process of the Markov chain is only dependent on the system's initial state and the transition matrix. The initial state of the process can be represented by

$$X^{(0)} = [X_{ij}^{(0)}]_{1 \times n}.$$

Let the process in a state k be $X^{(k)}$ after the k^{th} transition. According to the Chapman-Kolmogorov equation (Zhang and Zhang, 2009), $X^{(k+1)} = X^{(k)} * P$. The following recursive formula can then be obtained:

$$\begin{gathered} X^{(1)} = X^{(0)} * P, \\ X^{(2)} = X^{(1)} * P = X^{(0)} * P^2, \\ ..., \\ X^{(k)} = X^{(k-1)} * P = ... = X^{(0)} * P^k \end{gathered}$$

Therefore,

$$X^{(k+1)} = X^{(0)} * P^{k+1}.$$

2.2.3 Transition intensity matrix

The intensity between two states i and j, can be defined as the rate of change of the probability p_{ij} in a small time interval Δt (Mafu, 2014). The transition intensity is defined as

$$q_{ij}(t) = \lim_{\Delta t \longrightarrow 0} \frac{P(X(t + \Delta t) = j | X(t) = i)}{\Delta t}.$$

All possible intensities between possible states are collected in the transition intensity matrix denoted by Q (Mafu, 2014) and given by

$$Q = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1e} \\ q_{21} & q_{22} & \cdots & q_{2e} \\ \cdots & \cdots & \cdots & \cdots \\ q_{e1} & q_{e2} & \cdots & q_{ee} \end{bmatrix}.$$

The transition intensity matrix is used to define the multi-state model and used to calculate the transition probability matrix in (2.4). The elements in each of the rows of the transition intensity matrix must also sum to zero, $\sum_{j}^{e} q_{ij} = 0$, and the off-diagonal elements of Q must be non-negative $q_{ij} \geq 0, i \neq j$. The diagonal elements must be negative for all values where i is not equal to j, $q_{ii} = -\sum_{i \neq j} q_{ij}$ for $i = 1, ..., e$ (Mafu, 2014). Therefore, the rates on the diagonal represent states that subjects remain stationary and the off-diagonal values contain rates in which the subject moves to other states (Mafu, 2014).

As an example, for $e = 3$, the transition intensity matrix for a 3-state model is given by

$$Q(q) = \begin{bmatrix} -(q_{12} + q_{13}) & q_{12} & q_{13} \\ q_{21} & -(q_{21} + q_{23}) & q_{23} \\ q_{31} & q_{32} & -(q_{31} + q_{32}) \end{bmatrix}.$$

The off-diagonals in this matrix are rates at which the subjects move into other states and the diagonal elements are rates at which the subjects remain in their states (Mafu, 2014).

The transition probability matrix can be obtained by taking the matrix exponential of the scaled transition intensity matrix $P(t) = \exp(tQ)$. The exponential of a matrix C can be defined as $\exp(C) = 1 + \frac{C^2}{2!} + \frac{C^3}{3!} + ...$ using Taylor's Theorem.

The transition intensity matrix Q and transition probability matrix P can be obtained by maximising the likelihood, $L(Q)$. For an individual, let a series of times be $(t_1, t_2, ..., t_n)$ with corresponding states $(x_1, x_2, ..., x_n)$. A pair of successive states are observed to be i and j at time t_i and t_j. Three scenarios should be considered:

i. The information of the individual is obtained at arbitrary observation times and therefore the exact time of the transition of stages is unknown. Then, the contribution to the likelihood from this pair of states is calculated as $L_{ij} = p_{ij}(t_j - t_i)$.

ii. The exact times of the transitions between states are recorded and there are no transitions between the observed times. Then, the contribution to the likelihood from this pair of states is $L_{ij} = p_{ij}(t_j - t_i)q_{ij}$.

iii. The time of death (j) is known but the state on the previous instant (k) just before death is unknown. The contribution to the likelihood from this pair of states is $L_{ij} = \sum\limits_{k \neq j} p_{ik}(t_j - t_i)q_{kj}$.

After the construction of $L(Q)$, the estimated intensity and transition probabilities will maximise $L(Q)$(Cong, 2010).

2.2.4 Sojourn time

Rubino and Sericola (1988) explains that the sojourn time of a process X in a subset of states, is an integer valued random variable. It is the length of time that the process X remains in the state being occupied at time t.

The sojourn time of a continuous Markov process that is in state i is an independent and exponentially distributed random variable with mean $-\frac{1}{q_{ii}}$ (Cinlar, 1975). The remaining elements in the i^{th} row of the transition intensity matrix is proportional to the probabilities that govern the next state after state i to which the individual makes a transition. The probability that the next transition is from state i to state j is $-\frac{q_{ij}}{q_{ii}}$ (Mafu, 2014). The new state and the sojourn time are only dependent on state i and not on the history of the process prior to time t. Therefore, the sojourn time and the new state are independent of each other, given that the current state is state i. The mean sojourn time describes the average time period in a single stay in a state (Mafu, 2014).

2.2.5 Markov chain properties

2.2.5.1 No after effect property

It is seen from the above that the state of random variables with the Markov properties is only dependent on the state of the random variable and not on the previous states of the random variable (Zhang and Zhang, 2009).

2.2.5.2 Stationary distribution

According to Zhang and Zhang (2009), the state probability distribution $\{\pi_{(i)}, i\epsilon E\}$ with the Markov chain must satisfy $\pi_{(i)} = \sum\limits_{j\epsilon I} \pi_{(j)} P_{ij}$ with P_{ij} the state transition matrix of the random process and E the set of states.

2.2.5.3 Ergodic property

The probability of state j must stabilise in $\pi_{(j)}, j = 0, 1, ..., S$ after a sufficiently long time, independent of the state the process originates, hence, $\lim\limits_{n \longrightarrow \infty} P_{ij} = \pi_{(j)}$. Consequently, irrespective in which state the process originates, if the transition step number is sufficiently large, the probability of transitioning to state j approach a constant equal to $\pi_{(j)}$. This property states that the transition probability $\pi_{(j)}$ is an unique solution when the equations satisfy $\pi_{(j)} > 0$, $\sum\limits_{j=0}^{s} \eta_{(j)} = 1$ (Zhang and Zhang, 2009).

2.2.5.4 Interlinked property of state

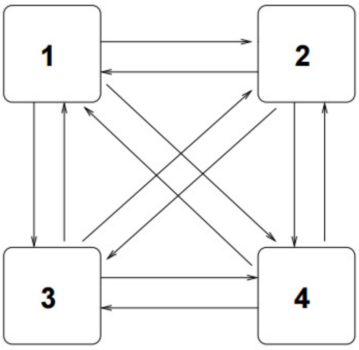
A stochastic process with the Markov property will reach a state k through a limited transition step regardless of the initial state being either i or j after certain transition steps (Zhang and Zhang, 2009).

2.3 Multi-state Markov Model

2.3.1 Introduction

A multi-state Markov model describes the process in which a patient moves through a series of states (Jackson, 2011). Fortunately, the *msm* package in R is one of the simpler packages that can be used to fit a multi-state model to a longitudinal dataset (Jackson, 2011). A longitudinal dataset consists of repeated measurements of the process at arbitrary times. The exact times of the state changes are unobserved and therefore unknown. For example, the state of a breast cancer patient may only be known when the patient consults with the oncologist.

The features of the *msm* package includes the ability to model transition rates and to include covariates in the models. It can also model data with censored states. Figure 2.2 gives an illustration of a general multi-state model.



$$Q = \begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ q_{41} & q_{42} & q_{43} & q_{44} \end{pmatrix}$$

Figure 2.2 General multi-state model (Source: Jackson, 2016: 3)

Figure 2.2 illustrates a multi-state model in continuous time. Its four states are labelled $1, 2, 3$ and 4. At a time t, the individual is in state $X(t)$. The arrows show which transitions are possible between states. The next state to which the individual moves, and the time of the change, are governed by a set of transition intensities $q_{ij}(t, z(t))$ for each pair of states i and j. The intensities may also depend on the time of the process t, or more generally a set of individual-specific or time-varying explanatory variables $z(t)$. The intensity represents the instantaneous risk of moving from state i to state j and is given by

$$q_{ij}(t, z(t)) = \lim_{\Delta t \longrightarrow 0} \frac{P(X(t+\Delta t) = j | X(t) = i)}{\Delta t}. \tag{2.6}$$

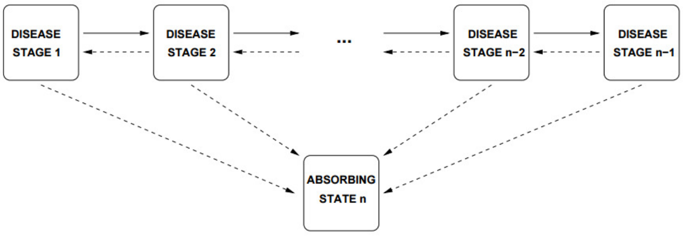
The intensities (2.6) form a matrix Q in which the rows sum to zero, such that the diagonal entries are defined by

$$q_{ii} = -\sum_{i \neq j} q_{ij}.$$

To fit a multi-state model to data, the transition intensity matrix must be estimated. This thesis concentrates on Markov models, which was explained in section 2.2.1, whereby the Markov assumption requires the future evolution only to be dependend on the current state. That is,

$q_{ij}(t, z(t), F_t)$ is independent of F_t, the observation history of the process up to the time preceding t.

Cox and Miller (1965) gives a thorough introduction into the theory of continuous-time Markov chains. A single period of occupancy in state i has an exponential distribution with rate $-q_{ii}$ (or mean $-1/q_{ii}$) in a time-homogeneous continuous-time Markov model (Jackson, 2011). The elements that remain in the i^{th} row of Q is proportional to the probabilities that govern the next state after i to which the individual transitions. The probability given by $-q_{ij}/q_{ii}$, is the probability of the individual's next move being from state i to state j (Jackson, 2011).



$$Q = \begin{pmatrix} q_{11} & q_{12} & 0 & 0 & \dots & q_{1n} \\ q_{21} & q_{22} & q_{23} & 0 & \dots & q_{2n} \\ 0 & q_{32} & q_{33} & q_{34} & \ddots & q_{3n} \\ 0 & 0 & q_{43} & q_{44} & \ddots & q_{4n} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 \end{pmatrix}$$

Figure 2.3 General model for disease progression (Source: Jackson, 2016: 3)

2.3.2 Disease progression models

The *msm* package was motivated by the broad applications to modelling of diseases (Jackson, 2011). As previously mentioned, multi-state Markov models in continuous time are often used in the progression of diseases. Figure 2.3 contains a model that is very commonly used. It represents a series of successively but more severe disease stages and then eventual death, which is regarded as an absorbing state (Jackson, 2011). From the illustration it is seen that a patient may move from one state to another and back again or die at any stage. Observations of the state $X_i(t)$ are made on several individuals i at different time points t. These time points may vary between individuals.

A homogeneous continuous-time Markov process can be used to model the stages of the disease with a transition matrix Q, as given in Figure 2.3. The illness-death model is commonly used with only three states representing health, illness and death. This model is illustrated in Figure 2.4. In this model, transitions are allowed from health to illness, illness to death and health to death. Sometimes recovery from illness to health may be considered.

Multi-state modelling has been used in a wide range of cancer applications, for example, Kay (1986) used it in hepatic cancer, Duffy and Chen (1995) and Chen et al. (1996) used it in breast cancer screening and Kirby and Spiegelhalter (1994) used it in cervical cancer screening.

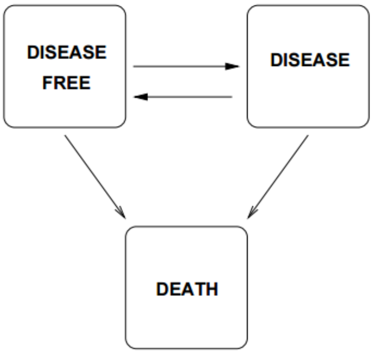


Figure 2.4 Illness-death model (Source: Jackson, 2016: 5)

2.3.3 Arbitrary observation times

Panel data are data with multiple dimensions that involve measurements over time. The panel data from monitoring the disease progression are often incomplete. Patients are usually seen at intermittent follow-up times at which information is collected, but the information from the periods between the visits are unavailable (Jackson, 2011). The exact time of the start of the disease is often unknown. Therefore, the state changes in a multi-state model and usually occur at unknown times whereby death times are mostly recorded within a day. Figure 2.5 illustrates a typical sampling situation and this specific individual is observed at four times over ten months. The final time is the death date which is recorded within a day. The only other information that is available is the occupancy of states 2, 2 and 1 and times 1.5, 3.5 and 5. It is unknown when the movement between states took place. For example, although the patient was in state 3 between times 7 and 9 months, it was not observed.

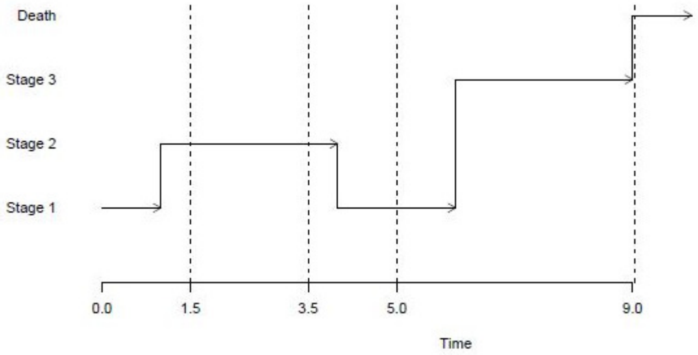


Figure 2.5 The evolution of a multi-state model (Source: Jackson, 2016: 5)

The reasons for observations made at given times must be considered when fitting a model to longitudinal data with arbitrary sampling times (Jackson, 2011). As in the case with missing data, a particular observation that is missing may implicitly give information about the value of that observation (Jackson, 2011). There are four different observation schemes listed below.

i. Fixed - patients observed at fixed intervals specified in advance.
ii. Random - the sampling time vary at random and independent of the current state of the disease.
iii. Doctor's care - the more ill a patient, the more closely the patient is observed and therefore, the next sampling time is chosen based on the current state of the disease.
iv. Patient self-selection - the patient decides on which occasions to visit the doctor e.g. when in poor condition.

Conditions under which sampling times are informative was discussed by Grüger et al. (1991). The inference made may be biased if a multi-state model is fitted while ignoring the information available in the sampling times (Grüger et al., 1991). The sampling times should be modelled along with the observation process $X(t)$, since the sampling times are often random themselves. The ideal situation, however, is when the joint likelihood for the times and the process is proportional to the likelihood obtained when the sampling times are fixed in advance (Jackson, 2011). If this is the case, parameters of the process can be estimated independently of the parameters of the sampling scheme. Grüger et al. (1991) showed that patient self-selection is informative whereas fixed, random and doctor's care observation policies are not informative.

2.3.4 Likelihood for the multi-state model

A general method for evaluating the likelihood for a general multi-state model in continuous time was described by Kalbfleisch and Lawless (1985) and at a later stage by Kay (1986). This method is applicable to all forms of the transition matrix. Here, the sampling times are assumed to be non-informative and the only available information is the observed state at a set of times. This can be seen in Figure 2.5.

According to Jackson (2011) and as mentioned in the transition probability matrix section, the transition probability matrix $P(t)$ is used to calculate the likelihood. The (i, j) entry of $P(t)$, $p_{ij}(t)$ is the probability of being in state i at time $t + u$, given the state is j at time u (for a time-homogeneous process). This does not give any information about the time of transition from state i to j. The process may have also entered other states between times u and $t + u$. The matrix exponential of the scaled transition intensity matrix can be taken to calculate $P(t)$. Therefore, $P(t) = \exp(tQ)$. This can be quite a difficult task and it is acceptable for simpler models to calculate an analytic expression for each element of $P(t)$ in terms of Q. This is generally a faster process and avoids the potential of having numerical instability of calculating the matrix exponential.

The three-state illness-death model, as described in section 2.3.2, where state one is disease free, state two is disease and state 3 is death, with no recovery, has a transition intensity matrix of the form

$$Q = \begin{bmatrix} -(q_{12} + q_{13}) & q_{12} & q_{13} \\ 0 & -q_{23} & q_{23} \\ 0 & 0 & 0 \end{bmatrix}.$$

The transition probabilities at time t that correspond to the transition intensity matrix Q are

$$\begin{array}{rcl}
p_{11}(t) & = & e^{-(q_{12}+q_{13})t} \\
p_{12}(t) & = & \left\{ \begin{array}{cc} \frac{q_{12}}{q_{12}+q_{13}-q_{23}}(e^{-q_{23}t} - e^{-(q_{12}+q_{13})t}) & (q_{12}+q_{13} \neq q_{23}) \\ q_{12}te^{-(q_{12}+q_{13})t} & (q_{12}+q_{13} = q_{23}) \end{array} \right. \\
p_{13}(t) & = & \left\{ \begin{array}{cc} 1 - e^{-(q_{12}+q_{13})t} - \frac{q_{12}}{q_{12}+q_{13}-q_{23}}(e^{-q_{23}t} - e^{-(q_{12}+q_{13})t}) & (q_{12}+q_{13} \neq q_{23}) \\ (-1 + e^{(q_{12}+q_{13})t} - q_{12}t)e^{-(q_{12}+q_{13})t} & (q_{12}+q_{13} = q_{23}) \end{array} \right. \\
p_{21}(t) & = & 0 \\
p_{22}(t) & = & e^{-q_{23}t} \\
p_{23}(t) & = & 1 - e^{-q_{23}t} \\
p_{31}(t) & = & 0 \\
p_{32}(t) & = & 0 \\
p_{33}(t) & = & 1
\end{array}.$$

According to Jackson (2011), the *msm* package calculates the transition probability matrix $P(t)$ analytically for selected models with two, three, four and five states. The framework of the model of special interest in this thesis can be found in Figure 2.1.

2.3.4.1 The likelihood for intermittently-observed processes

Suppose that the data for an individual n consist of a series of times $(t_{n1}, t_{n2}, ...t_{ni_n})$ and corresponding observed disease states $(X(t_{n1}), ..., X(t_{ni_n}))$. A general multi-state model is considered, with a pair of successive observed disease states $X(t_j), X(t_{j+1})$ at times t_j, t_{j+1}. The contribution to the likelihood of this pair of states can be expressed as

$$L_{i,j} = p_{X(t_j),X(t_{j+1})}(t_{j+1} - t_j). \tag{2.7}$$

This expression is also the entry of the transition matrix $P(t)$ at the $X^{th}_{(t_j)}$ row and $X^{th}_{(t_{j+1})}$ column evaluated at time $t = t_{j+1} - t_j$. The product of all such terms $L_{n,j}$ over all the individuals n and all the transitions, is then equal to the full likelihood $L(Q)$. The likelihood therefore depends on the unknown transition matrix Q, which was used to determine $P(t)$ (Jackson, 2011).

2.3.4.2 Exactly observed death times

It is commonly found, in observational studies of chronic diseases, that the time of death is known but the state is unknown the instant prior to death. If $X(t_{j+1}) = D$ is such a death state, the contribution to the likelihood is summed over the unknown state m on the instant just before death. Then the expression for the likelihood is given by

$$L_{i,j} = \sum_{m \neq D} p_{X(t_j),m}(t_{j+1} - t_j) q_{m,D}.$$

All the possible states m which can be visited between $X(t_j)$ and D are summed over (Jackson, 2011).

2.3.4.3 Exactly observed transition times

According to Jackson (2011), when the times $(t_{i1}, t_{i2}, ...t_{in_i})$ are the exact transition times between states, with no transitions between the observation times, the contributions can be expressed as

$$L_{i,j} = \exp(q_{X(t_j),X(t_j)}(t_{j+1} - t_j))q_{X(t_j),X(t_{j+1})},$$

since the state is assumed to be $X(t_j)$ throughout the interval between time t_j and time t_{j+1}, with a known transition to state $X(t_{j+1})$ at time t_{j+1}.

2.3.4.4 Censored states

A quantity with the exact value unknown, but known to be in a certain interval, is referred to as a censored quantity (Jackson, 2011). For intermittently-observed processes in multi-state models, the times of changes of states are usually interval censored, because it is known to be within bounded intervals,with the likelihood in (2.7). There are certain circumstances in which states or event times may be censored, for example at the end of a chronic disease, study patients are known to be alive but in an unknown state. For a censored observation $X(t_{j+1})$ that is known only to be in a state in the set E, have contribution to the likelihood expressed as

$$L_{i,j} = \sum_{m \in E} p_{X(t_j),m}(t_{j+1} - t_j).$$

This likelihood is not necessary if the state is known at the end of the study, for such a case (2.7) applies.

The *msm* package allows multi-state models to be fitted to data from processes with arbitrary observation times, exactly observed transition times, exact death times and censored, or a mixture of the above-mentioned schemes (Jackson, 2011).

2.3.5 Covariates

It is often of interest, the relationship of fixed or time-varying characteristics of individuals to their transition rates (Jackson, 2011). The explanatory variables for a particular transition intensity can be investigated by modelling the intensity as a function of the variables. A variation of the proportional hazards model was described by Marshall and Jones (1995), where the transition intensity matrix elements q_{ij} which are of interest can be replaced by

$$q_{ij}(z(t)) = q_{ij}^{(0)} \exp(\beta_{ij}^T z(t)).$$

The new transition intensity matrix Q can then be used to determine the likelihood. The contributions to the likelihood of the form $p_{ij}(t-u)$ can be replaced by $p_{ij}(t-u, z(u))$, if the covariates $z(t)$ are time dependent. This expression requires that the value of the covariate is known at every observation time u. The covariates are sometimes observed at different times to the main responses. It could then sometimes be assumed that the covariate is a step function, which remains constant between observation times (Marshall and Jones, 1995).

The *msm* package accounts for individual-specific or time-dependent covariates. Time-dependent covariates are assumed to be piecewise-constant in order to calculate the transition probabilities $P(t)$ on which the likelihood depends. Time-homogeneous models refer to models whose intensities change with time. Marshall and Jones (1995) also described the likelihood ratio and Wald tests for selection of covariates and testing hypotheses.

2.3.6 Semi-Markov process

The Markov assumption imply that the future movement of the process only depend on the current state and not on the past states (Mafu, 2014). The Markov assumption however imposes restrictions on the distribution of the sojourn time in a state. The sojourn time in a state should be exponentially distributed in continuous Markov processes and geometrically distributed in discrete Markov processes. The Markov assumption can be relaxed to overcome this problem, to allow arbitrarily distributed sojourn times in any state that still have the Markov assumption without being so restrictive (Mafu, 2014). Such a process is referred to as a semi-Markov process and is concerned with the random variables describing the state of the process. It is a generalisation of the Markov process, which makes transitions from state to state, such as a Markov process, but the amount of time spent in each state before the next transition is an arbitrary random variable that is dependent on the next state of the process (Ibe, 2009).

2.4 Missing Data

2.4.1 Introduction

The dataset supplied by Isimo Health contains missing data within the covariates. In order to handle missing data, the choice is either to delete incomplete observations or impute the missing values. To simply discard observations with missing data is not a reasonable solution, since valuable information is lost and the inferential power is compromised when doing the analysis after deleting incomplete data (Tang and Ishwaran, 2017). Therefore, it is better practice to rather impute the missing data. The dataset simulated in Chapter Four, is used in Chapter Five to test different imputation techniques to complete the missing data.

The three major problems with missing data, or otherwise known as incomplete data, are described by Barnard and Meng (1999) as:

i. The loss of information and the loss of efficiency or power due to the loss of data.

ii. The complication of handling the data as well as complications in the computation and analysis due to the irregularities in the patterns of the data.

iii. The potential bias due to the systematic differences between the observed data and the unobserved data.

According to Little and Rubin (1987), some of the techniques to handle missing data include deleting an entire case that have one or more missing values or replacing the missing values with a mean value of the missing data. Deleting cases with missing data can produce biased parameter estimates whereas using the mean values decrease the variability of the parameter estimates (Little and Rubin, 1987).

Imputation is an alternative approach to handling missing data. Imputation is defined as the process, where missing values are estimated from all the data available (Little and Rubin, 1987).

Andridge and Little (2010) also described missing value imputation as the replacement of missing data with acceptable values, by using the data in the recorded covariates, to unveil the information in the incomplete cases and also make inferences on the population parameters. The advantage of using imputation techniques is that once the missing data have been imputed, standard complete-data methods can be used to produce statistical results (Barnard and Meng, 1999). Much interest has been shown in using machine learning techniques to impute missing data. One of the approaches, based on Random Forests (RF), developed by Breiman (2001), will also be tested in Chapter Five together with another imputation technique.

2.4.2 Types of missing data

According to Rubin (1976), ignorability is an important concept in the literature of imputation techniques. Ignorability is the extent to which researchers have theoretical knowledge of the causes of data being missing.

In deciding how to handle missing data, it is helpful to know the reasons for the data being missing (Gelman and Hill, 2006). Missing data are categorised into four general missingness mechanisms. The matrix representation of the dataset which include the observed and missing values is denoted by $X = (X_{obs}, X_{mis})$, with X_{obs} the data that is observed and X_{mis} the data that is missing. This notation was introduced by Vargas-Chanes (2000).

2.4.2.1 Missingness at random

In the case where the probability of recording a value X depend on the observed variable Z and the probability do not depend on the missing values, the data can be regarded as missing at random (MAR). Therefore, for MAR, the probability that an observation is missing depends on what is actually observed. In principle, one can use the data to predict the missing values (Rubin, 1976). MAR assumes that the probability of an observation being missing depends only on the information that is available. The MAR assumption is often referred to as the ignorability assumption (Gelman and Hill, 2006). Gelman and Hill (2006) mentions that missingness at random is relatively easy to handle since all variables that affect the probability of missingness can be included as regression inputs. In summary, MAR is when the observation probability is independent of X_{mis} given the covariates Z and the observed responses X_{obs} (Spagnoli et al., 2011).

2.4.2.2 Missingness completely at random

Missing completely at random (MCAR) is a less restrictive condition and occurs when there is no particular reason for a value being missing. Such missing data happened by chance and therefore the mechanism of missing data is ignorable. Basically, the missing data are independent of the data values. In the MCAR case, the use of only the complete data (observations without any missing values) and therefore deleting cases, will give an unbiased result (Gelman and Hill, 2006). This is however only the case where the proportion of observations with missing values are rather small. A variable is considered MCAR when the probability of data being missing is the same for all of the units (Gelman and Hill, 2006). According to Spagnoli et al. (2001), MCAR can be summarised as, conditional on the covariates Z, the probability of the observation is independent of $X = (X_{obs}, X_{mis})$.

2.4.2.3 Missingness not at random or Non-ignorable missing data

As soon as the missing information depends on the information that has not been recorded (unobserved variables), missingness is no longer at random and therefore referred to as missing not at random (MNAR). Such missing cases must be explicitly modelled or it must be accepted that some bias will be included in the inferences made from the data (Gelman and Hill, 2006). This phenomenon occurs when the missing data depend on the unobserved variables. It is referred to as non-ignorable (NI) since the mechanism explaining the missing data is not observed or not accessible. Schafer (1997) addressed the point of transforming NI missing data to MAR. This will happen when missing data are not ignorable and the MAR conditions are not met.

2.4.2.4 Missingness dependent on the missing values

When the probability of the missingness depends on the variable itself, it is referred to as missingness dependent on the missing values (Gelman and Hill, 2006).

2.4.3 Notation of imputation techniques

The density function of the complete dataset can be expressed as

$$p(X|\theta) = \prod_{i=1}^{n} p(X_i|\theta), \tag{2.8}$$

where θ denotes the parameter governing the underlying distribution of X.

Suppose R is an indicator matrix with 1 if observed and 0 if the data is missing. Assuming R has the same dimensions as X, the joint conditional probability is expressed as

$$p(X, R|\theta, \phi) = p(X|\theta)p(R|X, \phi), \tag{2.9}$$

with ϕ denoting the conditional distribution of R given the complete dataset X. The complete dataset in (2.9) can be replaced by the observed data, which implies that the missing portion is integrated over, delivering the expression

$$p(X_{obs}, R|\theta, \phi) = \int p(X_{obs}, X_{mis}|\theta)p(R|X_{obs}, X_{mis}, \phi)dX_{mis}. \tag{2.10}$$

The distribution of the indicator matrix R is independent of the observed and the missing data if the missing data mechanism is missing completely at random. Rubin (1976) consequently defines MCAR as

$$p(R|X_{obs}, X_{mis}, \phi) = p(R|\phi). \tag{2.11}$$

This means that the distribution of the indicators in R of the observed and missing variables are independent on what is observed or missed. If the distribution of the missing data mechanism is independent of the missing values, but dependent on what is observed, i.e. the data is MAR, the density function can be expressed as

$$p(R|X_{obs}, X_{mis}, \phi) = p(R|X_{obs}, \phi). \tag{2.12}$$

Therefore, the missing mechanism is found in the data itself. In the case of the distribution of the observed values being unaffected by what is missing and taking only what is observed as being relevant, the substitution of (2.12) into (2.10) will lead to

$$p(X_{obs}, R|\theta, \phi) = p(R|X_{obs}, \phi)\int p(X_{obs}, X_{mis}|\theta, \phi)dX_{mis} = p(R|X_{obs}, \phi)p(X_{obs}|\theta). \tag{2.13}$$

Consequently, the joint distribution of the parameter space (θ, ϕ) can be divided into the product of the parameter space θ and ϕ, since the missing mechanism ϕ is independent of the observed data θ . This is valid under the MAR conditions.

David et al. (1986) stated that it is acceptable to impute by using the MAR assumption whenever the missing mechanism is NI, with the condition that there are covariates available for analysis.

2.4.4 Missing data techniques discarding data

According to Gelman and Hill (2006), many of the approaches to handle missing data simply ignores some of the data. Gelman and Hill (2006) discussed these approaches and showed that many of them lead to biased estimates. Therefore, larger standard deviations may be obtained due to sample sizes being reduced. The approaches discussed by Gelman and Hill (2006) include complete-case analysis (excluding all units with the outcome or any inputs missing), available-case analysis and non-response weighting.

2.4.5 Missing data techniques retaining all data - imputation techniques

Instead of discarding data with missing values, the missing values can be filled-in or imputed (Gelman and Hill, 2006). Imputation methods keep the full sample size. Additional to the simple missing data imputation techniques, three imputation methods will be discussed that includes the Expectation-Maximisation (EM) algorithm, multiple imputation (MI) and Full Information Maximum Likelihood (FIML) methods. The first two methods produce complete datasets with imputed values with the advantage being that the datasets generated can be used for analyses per usual, including structural equation models. The FIML method is a maximum likelihood approach for handling missing data, specifically in the context of structural equations. Thereafter, the use of Random Forests to impute missing data will also be discussed.

2.4.5.1 Simple missing data imputation techniques

Mean imputation is one of the easiest ways to impute missing data. It replaces each of the missing values with the mean of the observed values for that variable. According to Gelman and Hill (2006), this method can lead to underestimates of the standard deviation and it distorts the relationship between variables by basically pulling the estimates of the correlation towards zero (Gelman and Hill, 2006). Other methods include last value carried forward, using the information from related observations, indicator variables for missingness of categorical predictors, indicator variables for missingness of continuous predictors and imputation based on logical rules (Gelman and Hill, 2006).

2.4.5.2 Expectation Maximisation

Dempster et al. (1977) proposed the first idea for data imputation methods. The EM method provided a new perspective to maximum likelihood methods, when dealing with missing data (Dempster et al., 1977). Dempster et al. (1977) showed that filling in missing values should receive special attention and that deleting the data with missing values is an insufficient way of handling incomplete data.

Susianto et al. (2017) explains that the EM algorithm is a parametric method that imputes missing values based on the maximum likelihood estimation. The EM algorithm uses an iterative procedure to find the maximum likelihood estimators of a parameter vector through a two step algorithm (Susianto et al., 2017). The EM algorithm consists of two steps being the Expectation step (E-step) and the Maximisation step (M-step) (Dempster et al., 1977).

The conditional expected value of the full data of the log likelihood function $l(\theta|X)$ given the observed data is determined in the E-step (Susianto et al., 2017). Therefore, the expected values of the incomplete observations are computed in the E-step, given the observed data and current parameter estimates. In other words, in this step the missing data is replaced by estimated values and the model parameters are estimated. Suppose, that for any incomplete dataset, the distribution of the complete dataset X can be expressed as

$$\begin{aligned} f(X|\theta) &= f(X_{mis}, X_{obs}|\theta) \\ &= f(X_{obs}|\theta) f(X_{mis}|X_{obs}, \theta), \end{aligned} \tag{2.14}$$

where $f(X_{obs}|\theta)$ is the distribution of the observed data X_{obs} and $f(X_{mis}|X_{obs}, \theta)$ is the distribution of the missing dataset given the observed data. From (2.14), the log likelihood function can be obtained and expressed as

$$l(\theta|X) = l(\theta|X_{obs}) + \log f(X_{mis}|X_{obs}, \theta) \tag{2.15}$$

where $l(\theta|X)$ is the log likelihood function of the complete dataset, $l(\theta|X_{obs})$ is the log likelihood function of the observed dataset and $f(X_{mis}|X_{obs}, \theta)$ is the predictive distribution of the missing data given θ. By maximising the log likelihood function (2.15), θ is estimated. The right side of (2.15) can not be calculated since X_{mis} is unknown. The value of $l(\theta|X)$ is calculated based on the average value $\log f(X_{mis}|X_{obs}, \theta)$. This is calculated using the predictive distribution $f(X_{mis}|X_{obs}, \theta^{(t)})$ where $\theta^{(t)}$ is the temporary estimation of unknown parameters. The complete case analysis can be used to calculate an initial esimation $\theta^{(0)}$. Using this approach, the mean value of (2.15) can be expressed as

$$\begin{aligned} Q(\theta|\theta^{(t)}) &= l(\theta|X_{obs}) + \int \log f(X_{mis}|X_{obs}, \theta) f(X_{mis}|X_{obs}, \theta^{(t)}) \partial X_{mis} \qquad (2.16) \\ &= \int [l(\theta|X_{obs}) + \int \log f(X_{mis}|X_{obs}, \theta)] f(X_{mis}|X_{obs}, \theta^{(t)}) \partial X_{mis} \\ &= \int l(\theta|X_{obs}) f(X_{mis}|X_{obs}, \theta^{(t)}) \partial X_{mis}. \end{aligned}$$

The expression given in (2.16) gives a conditional expected value of the log likelihood function for the complete dataset $l(\theta|X)$, given the observed dataset and the inital estimate of the unknown parameter. (Susianto et al., 2017).

In the M-step, the missing data is replaced by the expected conditional value and the parameter estimates are computed by making use of the maximum likelihood method (Susianto et al., 2017). The M-step is done by iteratively estimating $\theta^{(t+1)}$ which maximises $Q(\theta|\theta^{(t)})$ as

$$Q(\theta^{(t+1)}|\theta^{(t)}) \geq Q(\theta|\theta^{(t)}).$$

The E-step and M-step are iterated until a pre-specified convergence criterion is met (Dempster et al., 1977).

The missing values provide the information required to generate parameter estimates. Reciprocally the estimates are generated and used to fill in the missing values (Schafer, 1997). This algorithm substitutes missing values using an initial value based on θ. It then uses the initial parameter to re-estimate the value of θ using the observed data and repeat the process until a specified criterion for convergence is met. Dempster et al. (1977) originally described the EM algorithm for non-ignorable models. Further details are also provided by Tanner (1993) and Schafer (1997).

2.4.5.3 Multiple Imputations - The MCMC Method

The EM method was extended by Rubin (1987). Rubin (1987) proposed a stochastic approach referred to as Multiple Imputations (MI), which include Monte Carlo Markov Chain (MCMC) techniques to improve the estimators' efficiency. According to Rubin (1987), Schafer (1997) and Tanner (1993), simulation techniques such as Gibbs sampling, the Metropolis algorithm, data augmentation and sampling importance resampling (SIR) are only some of the simulation techniques included in the MCMC methods.

According to Susianto et al. (2017), the MCMC method generates pseudo random variables from probability distributions via Markov chains (Markov processes was discussed in section 2.2.1). MCMC is a MI method that is used to imputate missing values of a continuous dataset. The MCMC algorithm assume that the data have a multivariate normal distribution, that the data are MCAR or MAR, and that the pattern of the missing data are monotone or arbitrary (Susianto et al., 2017). If the number of missing values are not too large, the inference of MCMC will be robust according to Susianto et al. (2017).

The Gibbs Sampling and Metropolis-Hastings algorithms are the two most popular MCMC methods. One draws from the conditional distribution of each component of a multivariate random variable given the other components in Gibbs sampling, whereas in Metropolis-Hastings, one draws from a probability distribution that approximate the distribution of interest and then accept or reject the drawn value with a specified probability (Susianto et al., 2017).

The EM algorithm discussed in section 2.4.5.2 provides a single dataset with data imputed by estimating the observations that are missing, whereas MI augments the data by simulating a possible set of values which delivers several sets of data with complete information. This is the most distinct difference between the EM method and the MI method.

MI essentially simulates data when missing data are present and therefore generates complete datasets by imputing the missing data which is a similar procedure to the EM algorithm (Rubin, 1987). From a Bayesian perspective, the information about the known parameters is expressed via a posterior probability distribution. Alternatively to maximum likelihood, a prior distribution is added for the parameters and the posterior distribution of the parameters of interest, is computed (Susianto et al., 2017). Again, X_{mis} and X_{obs} represent the missing values and the observed values, respectively. The observed data posterior can then be expressed as

$$p(\theta|X_{obs}) \propto p(\theta)p(X_{obs}|\theta) \tag{2.17}$$

where $p(\theta)$ is the prior distribution and $p(X_{obs}|\theta)$ the observed likelihood function. Since the data are incomplete, the observed data posterior $p(X_{obs}|\theta)$ cannot be easily simulated. Therefore, X_{obs} is augmented by an assumed value of X_{mis} which makes the resulting complete-data posterior $p(\theta|X_{obs}, X_{mis})$ much easier to handle. If the missing data X_{mis} has been observed the observed data posterior is related to the complete-data posterior distribution that would have been obtained, namely

$$p(\theta|X_{obs}, X_{mis}) \propto p(\theta)p(X_{obs}, X_{mis}|\theta). \tag{2.18}$$

From (2.17) and (2.18), the observed data posterior can be obtained as

$$\begin{aligned} p(\theta|X_{obs}) &= \int p(\theta, X_{mis}|X_{obs})dX_{mis} \qquad (2.19) \\ &= \int p(\theta|X_{obs}, X_{mis})p(X_{mis}|X_{obs})dX_{mis}. \end{aligned}$$

The posterior predictive distribution $p(X_{mis}|X_{obs})$ cannot be simulated directly in (2.19). It is however possible to create random draws of X_{mis} from $p(X_{mis}|X_{obs})$ using techniques of MCMC. The Gibbs sampling algorithm (as an example) can be used to draw the missing values X_{mis} from $p(X_{mis}|X_{obs})$. Assuming the data have a multivariate normal distribution allows data augmentation to be applied to Bayesian inference with missing data by repeating two steps. (Susianto et al., 2017).

The data augmentation algorithm, using MCMC, has two steps referred to as the I-step and the P-step. Initial estimates of the missing values are generated in the I-step. These are estimated given the conditional distribution of the observed values and initial parameter estimates of the distribution. In notation, given a current guess $\theta^{(t)}$ of the parameter, random draws of missing values X_{mis} is made from the posterior predictive distribution $p(X_{mis}|X_{obs})$ delivering

$$X_{i(mis)}^{(t+1)}\tilde{}\,p(X_{i(mis)}|X_{obs},\theta^{(t)}).$$

Thereafter, the P-step generates the parameters' starting values. These are estimated given the joint distribution of the observed and the initial imputation in the I-step. A new value of θ is therefore drawn from the complete data posterior conditional to $X_{i(mis)}^{(t+1)}$ delivering

$$\theta^{(t+1)}\tilde{}\,p(\theta|X_{obs,}X_{i(mis)}^{(t+1)})$$

(Susianto et al., 2017).

Starting from the intitial values $\theta^{(0)}$ and $X_{mis}^{(0)}$, these two steps define a Gibbs sampler. A stochastic Markov chain is generated in the two steps that converges in distribution to a certain value and produces various imputations. The stochastic sequences are $\{\theta^{(t)}\}$ and $\{X_{mis}^{(t)}\}$ with stationary distributions $p(\theta|X_{obs})$ and $p(X_{mis}|X_{obs})$, respectively (Susianto et al., 2017). Therefore, the MI algorithm generates several complete datasets. These datasets are sufficient to capture the variability averaged over the simulated parameter estimates to obtain a single estimate to represent the model (Rubin, 1987).

According to Rubin (1987), the motivation behind the MI method is the fact that one imputed dataset might not represent the original variation, but multiple observations based on simulated data could represent the outcome more efficiently.

2.4.5.4 Full Information Maximum Likelihood

Another approach to data imputation was proposed by Muthén et al. (1987). It was proposed to use a regression model to predict the missing data from the information available. Another method was proposed in the structural equation's context known as the Full Information Maximum Likelihood (FIML) method (Arbuckle, 1996; Little and Rubin, 1987). FIML model parameters and standard errors are estimated directy from the data available (Li, 2010). Therefore, no data preparation is required for FIML and no missing values are imputed. A log likelihood function is calculated and maximised for each individual when assuming a multivariate normal distribution and a MAR missingness mechanism. The log likelihood function measures the discrepancy between the observed data and the current parameter estimates by using all the data available from the variables that are modeled. Therefore, the log likelihood function being maximised for a subject i is given as

$$\log L_i = K_i - \frac{1}{2}\log|\sum_i| - \frac{1}{2}(\underline{x}_i - \underline{\mu}_i)'(\sum_i)^{-1}(\underline{x}_i - \underline{\mu}_i),$$

where $\underline{x}_i$ is the raw data vector for a subject i, and $\underline{\mu}_i$ and $\sum_i$ are the parameter mean vector and covariance matrix. The subscript i indicates that the sizes of the vectors and matrices differ because the number of complete observations for a given subject may differ. The N subject-wise discrepancy functions are then summed for the entire sample as

$$\log L(\underline{\mu}, \sum) = \sum_{i=1}^{N} \log L_i.$$

The FIML estimates are obtained by means of an iteration approach (Li, 2010).

In other words, this approach first uses maximum likelihood estimates for subsets of data consisting of complete data and thereafter generates several covariance matrices with their respective likelihood functions. Therefore, a combined likelihood function which incorporates all possible subsets of likelihood functions, that is based on the subsets of complete data, is generated. Unlike the other two approaches, no actual data imputation is used in the FIML method. The available data is used to estimate the parameters using a maximum likelihood function. This method can however only be used for structural equations.

Many covariance matrices are computed by the FIML algorithm. The number of covariance matrices depend on the number of complete patterns in the dataset. A pattern is seen as complete if it has a subset of variables from the original data without any missing values. Finally, a maximum likelihood estimation procedure is performed over all possible covariance matrices and this generates a unique set of parameter estimates for the model (Muthén et al., 1987).

2.4.5.5 Random Forests

Random Forest (RF) was first introduced by Breiman (2001). In RF, the base learner is a binary recursive tree that is grown using random input selection (Tang, 2017). Its random feature is formed by selecting a small group of input variables at random to split on at each node, and bootstrapping of the original dataset. The bootstrapped sample of each tree is referred to as in-bag data whereas the data not sampled are called out of bag (OOB) data. The OOB data are used to assess the predicting accuracy of the random forest.

Random forest (RF) missing data algorithms have become more attractive as an approach of handling missing data (Tang and Ishwaran, 2017). The RF techniques can handle mixed types of missing data, can adapt to interactions and nonlinearity and can potentially scale to big data settings. Tang and Ishwaran (2017) showed that the RF techniques perform good under moderate to high missingness and can even deal with data that is MNAR. It was also shown that the RF technique, *missForest*, outperform the K-nearest neighbour (KNN) method as well as an alternative method proposed by Davila and Rosado (2017).

This imputation method, for each variable in turn, will predict the missing values by using a random forest using the other variables as the targets. This process will be iterated until there is no further change. The imputed data will thereafter be used to construct a predictor. The trees cope with missing values since when the splits are considered, only the splits of the form $X < c$ is considered where c is one of the non-missing values of X. The splitting criterion is evaluated with the missing values ignored and for each split, the algorithm identifies splits using different variables that result in similar partitions of the feature space. These splits are used if a case has a missing value in the primary split. The missing values in the target are ignored when calculating the value of a tree in a region. In the same way trees handle missing values, so does random forests.

According to Tang (2017), the RF approach works as follows. The data are roughly imputed by replacing the missing values for continuous variables, with the median of the non-missing values. And replacing the missing values for categorical variables with the most frequent occuring non-missing value. Thereafter, a RF is fitted to the roughly imputed data and a proximity matrix is calculated from the fitted RF. The proximity matrix is a symmetric $n \times n$ matrix with entries (i, j) recording the frequency that subject i and j occur within the same terminal node. The proximity matrix is used to impute the data. The proximity weighted average of the non-missing data is used to imputed continuous variables. For integer variables, the integer value having the largest average proximity over non-missing data is used to impute the missing values. Thereafter, the updated data are used as an input in the RF and the procedure is iterated until a stable solution is reached. (Tang, 2017).

According to Tang (2017), RF algorithm groups variables and runs a multivariate forest using each group in turn as a set of dependant variables which replaces p regressions where p is the number of variables. The missing data problem is recasted as a prediction problem. The missing data is imputed by regressing each variable in turn against all other variables and then predicting missing data for the dependent variable using the fitted forest. Therefore, p forests are fitted at each iteration since there are p variables. Let X be the $n \times p$ matrix with missing values X_{mis}, and the stopping criteria ς and grouping factor $\alpha, 0 < \alpha \leq 1$. Firstly, it is recorded which variables and which positions have missing values in X denoting p_0 the number of variables that have missing values and X_{imp} the quick and rough imputation. Set $diff = \infty$, and while $diff \geq \varsigma$ let $X_{old.imp} \leftarrow X_{imp}$. Thereafter, randomly separate the p_0 variables into $K = K(\alpha)$ groups of approximately the same size. Then, for $i = 1, ..., K$, let X_i be the columns of X corresponding to group i and $X_{(-i)}$ the columns of X excluding group i. Thereafter, for $i = 1, ..., K$, set the values in X_i which were missing back to NA. For $i = 1, ..., K$, fit a multivariate random forest using the variables in groups i as response variables and the rest of the variables as predicting variables and calculate X_{imp} as the final summary inputed value using the terminal average for continuous variables and using the maximal terminal node class rule for categorical variables. Now, set $diff = \xi(X_{old.imp}, X_{imp})$ and return to the imputed matrix X_{imp} (Tang, 2017).

2.5 Summary

Multi-state Markov models was thoroughly discussed in this chapter. The concept of missing data, types of missing data as well as different imputation techniques were also discussed in detail. These techniques will be applied in subsequent chapters. The imputation techniques will be compared and used to impute the dataset whereafter the multi-state model will be fitted to the complete dataset.

CHAPTER 3 DESCRIPTION OF CLAIMS AND AUTHORISATION DATASET FOR BREAST CANCER

3.1 Introduction

The data obtained from Isimo Health will be discussed in this chapter. The process of retrieving, extracting transforming and cleaning the data is included in this discussion.

3.2 Data Source

As mentioned before, the dataset was collected from ICON. The data consisted of a combination of claims data from medical schemes and authorisation data from ICON's authorisation system eAuth.

The claims data from medical schemes provided funding information. All claims for the three-year period 2014-01-01 to 2016-12-31 were extracted. The claims data also contained provider information and a diagnosis date as per medical scheme records. The claims data were linked to a practice through a practice number and therefore having a practice location. More detail about the item in the claim was also available. The claims were grouped into claims regarding chemotherapy (actually medical therapy, which includes chemotherapy and hormone therapy), radiotherapy, hospitalisation, radiology, pathology and any other types of claims.

A patient details file that is collected by ICON together with the claims data from medical schemes contained information such as the date of birth, the date of death, the cancer registry registration date. Other information such as whether the patient was on a program for Best Supportive Care (BSC) and the date that the cancer was diagnosed as being metastatic was also included in the patient details file. Patients who becomes metastatic, resulted in the treatment intent that changed to being non-curative.

The authorisation data from eAuth contained the clinical information. This information is entered into the system when providers obtain authorisation for a course of treatment. The TNM staging factors being tumour size, node size and metastasis size, are included in the authorisation data. A derived cancer staging, called r_stage, is also available from the eAuth data.

The last source of information was a source that contained all the risk clinical attributes that were mainly collected from the eAuth system. A list of the 22 different clinical attributes relevant to breast cancer is given below.

Table 3.1 Risk clinical attribute names with descriptions

Risk clinical attribute	Description
BMI	Body Mass Index is a value derived from the mass (weight) and height of an individual in kg/m^2(real number $0 < BMI < 100$).
BSA	Body Surface Area is the surface area of a human body (real number $0 < BSA \leq 3$).
CA-15-3	Monitor response to breast cancer treatment and disease recurrence. Increase could indicate treatment failure, but levels can rise during initial 4-6 weeks of therapy. (Any real number).
CA125	Protein found on surface of many ovarian cancer cells; if levels go down the treatment is working. (Any real number).
comorbidity asthma	Indicator indicating whether patient has asthma.
comorbidity diabetes	Indicator indicating whether patient has diabetes.
comorbidity HIV	Indicator indicating whether patient has HIV.
comorbidity hypertension	Indicator indicating whether patient has hypertension.
ECOG	Performance status; 0 is fully active, 5 is dead, 4 completely disabled (values 0 to 4).
estrogen receptor	Has receptors for estrogen; if positive hormone therapy will most likely work (indicator indicating whether ER positive).
height	Patient height in cm.
HER2 FISH	Her2 is a gene that can play a role in the development of breast cancer; her2 positive cancer cells divide and multiply quickly leading to aggressive tumor growth.
HER2 ICH	The three different tests are done in sequence. With the ICON Head of Clinical Services an indicator was created to identify patients that are HER2 positive.
HER2 ISH	Indicator whether positive (0=negative).
KI 67	
KI 67 (tissue)	Percentage protein in cell increases as prepare to divide into new cells. <10% low 10-20% borderline and >20% high.
metastasis size	Metastatic indicator in TNM staging.
node size	Node size 0 to 3 in TNM staging.
progesterone receptors	Indicator whether PR positive.
r_stage level	Derived cancer staging 0-4.
tumor size	tumour size 0-4 in TNM staging
weight	patient weight in kilogram

3.3 Data Extraction and Transformation

The eAuth data are valuable because of the demographical and clinical detail it contains. It gives an indication of the planned course of treatment. The claims data were then used to confirm whether the planned treatment took place. Both these sets of data had to match with their respective ID numbers, making this process difficult. Put simply, an internal number for each patient with a claim was identified and matched to an ID number. Then, for each authorisation request from eAuth, the patient key was also matched to an ID number. Thereafter, the ID numbers between the two sources needed to be matched.

For this thesis, only data that had been matched were used. A total of 393 distinct patients could be matched in the claims and authorisation data. That included 142 733 claim lines, 5 500 authorisation requests and 4 370 lines of clinical attributes data. All patients that were deceased before 2014 were omitted from the analysis.

After the matching process, all the data were de-identified for confidentiality purposes. The matched data extracted were in the form of a longitudinal dataset or otherwise known as panel data. A panel dataset consists of repeated measurements of a state of a patient at different time points over several years. The matching and extraction of the data was done in SQL. Excel was then used to do some data clean-up. Thereafter, R statistical programming was used to perform the statistical analysis.

The concept of a treatment episode was to be clearly understood in order to extract the data correctly. An episode of care is defined as all services provided to a patient with a medical problem within a specific period of time across a continuum of care in an integrated system (Farlex Partner Medical Dictionary, 2012). A longitudinal record of all treatment episodes for all breast cancer patients over a period of three years was compiled. The treatment episode (per data line) indicated what type of treatment was received grouped in chemotherapy (medical therapy including chemotherapy and hormone therapy), radiotherapy or combination therapy (a combination of chemotherapy and radiotherapy). Thereafter, all the different clinical attributes, described above in the table in the data source section, were added to the dataset.

The dataset containing the treatment episodes eventually had 37 columns of information. The descriptions for each of the columns are given below. Not included in this list (but forming part of the 37 columns) is the 22 clinical attributes described above. The status of each of the clinical attributes is taken at the start of each treatment episode.

Table 3.2 Column names and descriptions of treatment episode dataset

Column name	Description
patient key	Unique key to identify a patient
is_chemo	A variable indicating whether the patient received chemotherapy (1) or not (0).
is_radiotherapy	A variable indicating whether the patient received radiotherapy (1) or not (0).
benefit_paid	The total cost of the treatment episode.
episode_start_dt	The start date of the treatment episode.
episode_end_dt	The end date of the treatment episode.
episode_duration_in_days	The difference in days between the start and end date of the treatment episode.
number_of_episodes	The total number of treatment episodes for the patient.
state	The state of a patient in the episode of care. (1,2 or 3)
birth_dt	The date of birth of the patient.
cancer_registry_dt	The first date of registration on the cancer registry.
diagnosis_dt	The diagnosis date of the breast cancer.
gender	The gender of the patient. 0=female, 1=male
age_at_diagnosis	The patient age at the time of diagnosis.

After creating a longitudinal record of all the treatment episodes (with the variables described above) it was seen that it was unnecessary to have all the treatment episodes separately, since the main interest was to investigate how the patient progresses from being treated curatively to non-curatively, and eventually being deceased. Therefore, all the treatment episodes for an individual, with all covariates equal, were combined into a single data line. An indicator showing whether the patient had hospital claims, radiology claims and/or pathology claims was also added to the dataset. Other demographical detail, such as the death date, registration on a Best Supportive Care (BSC) programme and the location of the practice that the patients was treated at, was also added. Due to the very high percentage of missing data in the last two of the three demographical details just mentioned, these were eventually ignored. Oncologists at Isimo Health were consulted to fill in some of the gaps in the data. The comorbidity indicators such as asthma and HIV were eliminated from the dataset since these indicators are not reliably captured in the authorisation system.

The next stage of data cleaning was to get a single record per patient. At this stage, each state was represented in a separate line. The dataset then contained the columns provided in Table 3.3.

Table 3.3 Column names and description of dataset containing one record per patient

Column name	Description
patient_key	Unique key to identify a patient
gender	The gender of the patient. 0=female, 1=male
weight_at_diagnosis	The weight of the patient at the time of diagnosis.
height_at_diagnosis	The height of the patient at the time of diagnosis.
age	The patient age at the time of diagnosis.
start(t)	The starting time of the time period t. 0 is defined as the diagnosis date.
end(t)	The end time of the time period t.
s(t)	The state during the time period t. (1=curative; 2=non-curative; 3=death)
cost(t)	The cost of the time period t.
HER2(t)	The HER2 status during time period t.
ER(t)	The ER status during time period t.
PR(t)	The PR status during time period t.
eps_count	The total number of treatment episodes for the patient during time period t.
node_size	The node size of the TNM staging during time period t.
treatment(t)	The treatment used during time period t. (1=Chemotherapy; 2=Radiotherapy; 3=Combination therapy)
r_stage	Cancer staging 0,1,2,3,4.

The last ten variables were done for time periods $t = 1, 2, 3$. It was done for all three time periods since the time periods represent the time periods in the three respective states.

The final stage of the data transformation and clean-up was to get the data into a format that is accepted by the *TPmsm* package. The final dataset contained 13 variables as described in Table 3.4.

Table 3.4 Column names and descriptions of the final dataset

Column name	Description
time1	The total time spent in state 1
event1	An indicator variable indicating whether the patient left state 1.
Stime	The total survival time. Therefore, the total time spent in state 1 and 2.
event	An indicator variable indicating whether the patient moved into state 3.
gender	The gender of the patient. (0=female; 1=male)
weight	The weight of the patient. (kg)
height	The height of the patient. (cm)
r_stage	Cancer staging (0,1,2,3,4).
age	The age of the patient at diagnosis.
HER2	The HER2 status of the patient. (1=positive, 0=negative)
ER	The ER status of the patient. (1=positive, 0=negative)
PR	The PR status of the patient. (1=positive, 0=negative)
node	The node size of the patient's cancer. (0-3)

3.4 Conclusion

The process of extracting the data from the data sources and transforming the data into a useful form was a tedeous and yet exciting process. It took many computer hours to transform data into different forms that were not used in the end. This process gave the researcher a much deeper understanding of the information contained in the Isimo Health dataset. This dataset will be used in Chapter Four to simulate a new dataset, to test the imputation techniques on which will be used at a later stage to impute this dataset.

CHAPTER 4

SIMULATION STUDY

4.1 Introduction

The R package, *TPmsm*, was used to simulate the data for the rest of the thesis. The package was formulated by authors Artur Araújo, Luís Meira-Machado and Javier Roca-Pardiñas (2014). According to Araújo et al. (2014), the *TPmsm* package provide seven different approaches to model three-state illness-death models. Three covariates were simulated from different distributions, including the normal distribution, Bernoulli distribution and multinomial distribution. The aim of the data simulation was to test different imputation techniques.

4.2 Introduction to the *TPmsm* Package

Referring back to previous explanations, a stochastic process $(X(t), t\epsilon T)$ with a finite state space. $X(t)$ represents the state that is occupied by the process at time $t \geq 0$. The future state transitions of MSMs may be dependent on past events.

A non-parametric estimator for quantities in the non-homogeneous Markov model was first introduced by Aalen and Johansen (1978). Aalen and Johansen (1978) extended the Kaplan-Meier estimator to Markov chains. The standard error of the Aalen-Johansen estimator is possibly large when a lot of censoring is present, especially in the case of a small sample size. A possible solution to this problem was introduced by Meira-Machado et al. (2006) by introducing a substitute for the Aalen-Johansen estimator in the case of a non-Markov illness-death model. The estimator introduced by Meira-Machado et al. (2006) performs better when the Markov assumption does not hold. The Kaplan-Meier weights relating to the distribution of the total survival time of the process is used to weight the data.

According to Araújo et al. (2014), the *TPmsm* package aims to implement non-parametric and semi-parametric estimators for the transition probabilities in three state models. Right censoring is dealt with by using inverse censoring probability reweighting. Such approaches lead to consistent estimators when dependent censoring is present.

4.3 Methodology Behind *TPmsm*

Araújo et al. (2014) considers the progressive illness-death model when describing the methodology behind the *TPmsm* package. The progressive illness-model can be seen in Figure 4.1.

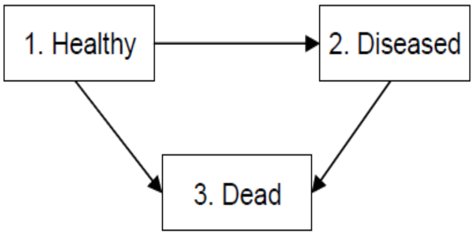


Figure 4.1 Illness-death model (Source: Araújo et al.,2014:4)

In this model, all subjects are assumed to be in state 1 at time $t = 0$. Please note, that this is not the case for the simulated data. The subjects may visit state 2 at some time point or go directly into the absorbent state, state 3, or remain in the first state. A random vector (T_{12}, T_{13}, T_{23}) can be used to describe the stochastic behaviour of the process in Figure 3.1 where T_{ij} is the potential transition from state i to state j with $1 \leq i < j \leq 3$. T_{23} represents the sojourn time spent in state 2. This model contains two competing transitions $1 \longrightarrow 2$ and $1 \longrightarrow 3$. The sojourn time in state 1 can be denoted by $Z = \min(T_{12}, T_{13})$ and the survival time of the subject is given by

$$T = I(T_{12} \leq T_{13})(T_{12} + T_{23}) + I(T_{12} > T_{13})T_{13}.$$

Due to censoring, $(\tilde{Z}, \tilde{T}, \Delta_1, \Delta)$ is observed where

$$\tilde{Z} = \min(Z, C), \tilde{T} = \min(T, C),$$

$$\Delta_1 = I(Z \leq C)$$

and

$$\Delta = I(T \leq C).$$

The potential censoring, time assumed to be independent of the process, is denoted by C. Therefore, C and (Z, T) are assumed to be independent.

Araújo et al. (2014) defines the transition probabilities between two time points $s < t$ as

$$p_{ij}(s,t) = P(X(t) = j | X(s) = i).$$

It can be seen from Figure 3.1, that five different transition probabilities need to be estimated. The five transition probabilities include $p_{11}(s,t)$, $p_{12}(s,t)$, $p_{13}(s,t)$, $p_{22}(s,t)$ and $p_{23}(s,t)$. Since three of the transition probabilities can be obtained from the relationships

$$p_{11}(s,t) + p_{12}(s,t) + p_{13}(s,t) = 1$$

and

$$p_{22}(s,t) + p_{23}(s,t) = 1,$$

only two of the transition probabilities need to be estimated.

According to Cox and Miller (1965), the Markov model transition probabilities can be calculated from the transition intensities. If one assumes that the transition intensities exist, the transition probabilities can be expressed as

$$q_{ij}(t) = \lim_{\Delta t \to 0} \frac{p_{ij}(t, t+\Delta t)}{\Delta t}$$

by solving the forward Kolmogorov differential equation. The illness-death model has explicit expressions for the transition probabilities,

$$p_{11}(s,t) = \exp(-Q_{12}(s,t) - Q_{13}(s,t)),$$

$$p_{22}(s,t) = \exp(-Q_{23}(s,t))$$

and

$$p_{12}(s,t) = \int_s^t p_{11}(s,u) q_{12}(u) p_{22}(u,t) du,$$

where

$$Q_{ij}(s,t) = \int_s^t q_{ij}(u) du$$

is the cumulative or integrated intensity between s and t.

The expressions for the transition probabilities in time-homogeneous Markov models is given by

$$p_{11}(s,t) = \exp(-q_{12}(s,t) - q_{13}(s,t)),$$
$$p_{22}(s,t) = \exp(-q_{23}(s,t))$$

and

$$p_{12}(s,t) = \frac{q_{12}}{q_{12}+q_{13}-q_{23}}[\exp(-q_{23}(t-s)) - \exp(-(q_{12}+q_{13})(t-s))].$$

These transition probabilities can also be estimated non-parametrically or semi-parametrically and the expressions are then given by

$$\begin{aligned} p_{11}(s,t) &= \tfrac{P(Z>t)}{P(Z>s)}, \\ p_{12}(s,t) &= \tfrac{P(s<Z\leq t, T>t)}{P(Z>s)}, \\ p_{13}(s,t) &= \tfrac{P(Z>s, T\leq t)}{P(Z>s)}, \\ p_{22}(s,t) &= \tfrac{P(Z\leq s, T>t)}{P(Z\leq s, T>s)} \end{aligned}$$

and

$$p_{23}(s,t) = \frac{P(Z \leq s, s < T \leq t)}{P(Z \leq s, T > s)}.$$

Araújo et al. (2014) explains that the transition probabilities mentioned above may be estimated non-parametrically using the Aalen-Johansen estimator. The Kaplan-Meier estimator is used as the Aalen-Johansen estimate of the transition probability $p_{11}(s,t)$ and is given by

$$\hat{p}_{11}^{AJ}(s,t) = \prod_{s<\tilde{Z}_i\leq t} [1 - \frac{\Delta_{1i}}{n\tilde{M}_{0n}(\tilde{Z}_i)}],$$

where

$$\tilde{M}_{0n}(y) = \frac{1}{n}\sum_{j=1}^{n} I(\tilde{Z}_j \geq y).$$

The Kaplan-Meier estimator for the transition probability $p_{22}(s,t)$ is given similarly as

$$\hat{p}_{22}^{AJ}(s,t) = \prod_{s<\tilde{T}_i\leq t, \tilde{Z}_i<\tilde{T}_i} [1 - \frac{\Delta_i}{n\tilde{M}_{1n}(\tilde{T}_i)}],$$

where

$$\tilde{M}_{1n}(y) = \frac{1}{n}\sum_{j=1}^{n} I(\tilde{Z}_j < y \leq \tilde{T}_j).$$

Similarly, the estimate for $p_{12}(s,t)$ is given as

$$\hat{p}_{12}^{AJ}(s,t) = \frac{1}{n}\sum_{i=1}^{n}\frac{\hat{p}_{11}^{AJ}(s,\tilde{Z}_i^{-})\hat{p}_{22}^{AJ}(\tilde{Z}_i,t)I(s<\tilde{Z}i\leq t,\tilde{Z}i<\tilde{T}i)}{n\tilde{M}_{0n}(\tilde{Z}_i)}, \tag{4.1}$$

where

$$\hat{p}_{11}^{AJ}(s,t^{-}) = \lim_{u\uparrow t}\hat{p}_{11}^{AJ}(s,u).$$

Different estimation methods will be thoroughly discussed in this chapter and can all be implemented using the *TPmsm* software package. The arguments required by the *TPmsm* package include the observed time in state 1 (*time1*), the corresponding censoring indicator (*event1*), the total survival time (*Stime*) and the final status of the subject (*event*). The *event* argument assumes the value 1, if the final event of interest (death) is observed.

4.3.1 Pre-smoothed Aalen-Johansen estimator

Araújo et al. (2014) further explains that the Aalen-Johansen estimator may have a larger standard error, in the presence of heavy censoring. This is especially evident when the sample size is small. Pre-smoothing may reduce the variance of the Aalen-Johansen estimator.

According to Moreira et al. (2013), this process referred to as pre-smoothing involves replacing the censoring indicators within the transition probabilities by a smooth fit using some sort of regression. The corresponding pre-smoothed Aalen-Johansen estimator of $p_{11}(s,t)$ is given by

$$\hat{p}_{11}^{PAJ}(s,t) = \prod_{s<\tilde{Z}_i\leq t}[1-\frac{m_{0n}(\tilde{Z}_i)}{n\tilde{M}_{0n}(\tilde{Z}_i)}], \tag{4.2}$$

where $m_{0n}(\tilde{Z}_i)$ is the estimator of the conditional probability of the event $\Delta_1 = 1$ given $\tilde{Z}$. The quantity $m_{0n}(\tilde{Z}_i)$ can be estimated using logistic regression.

The pre-smoothed version of the Aalen-Johansen estimator of $p_{22}(s,t)$ is given by

$$\hat{p}_{22}^{PAJ}(s,t) = \prod_{s<\tilde{T}_i\leq t,\tilde{Z}_i<\tilde{T}_i}[1-\frac{m_{1n}(\tilde{Z}_i,\tilde{T}_i)}{n\tilde{M}_{1n}(\tilde{T}_i)}], \tag{4.3}$$

where $m_{1n}(\tilde{Z},\tilde{T})$ is an estimator of the conditional probability for the event $\Delta = 1$ given $(\tilde{Z},\tilde{T})$ and provided that the transition from state 1 to state 2 was observed. The transition probability $p_{12}(s,t)$ can be obtained by substituting (4.2) and (4.3) into (4.1).

4.3.2 Kaplan-Meier weighted estimator

It was verified by Meira-Mochado et al. (2006) that the use of Aalen-Johansen estimators to empirically estimate the transition probabilities may not be appropriate in the non-Markov scenario. Meira-Mochado et al. (2006) proposes a Markov-free alternative to estimate the transition probabilities, which does not rely on the Markov assumption.

The Kaplan-Meier estimator, relating to the distribution of the total time to weight the bivariate data, should be used to estimate the transition probabilities. The Kaplan-Meier weighted estimators are therefore given by

$$\hat{p}_{11}^{KMW}(s,t) = \frac{\sum\limits_{i=1}^{n} W_{1i} I(\tilde{Z}_i > t)}{\sum\limits_{i=1}^{n} W_{1i} I(\tilde{Z}_i > s)},$$

$$\hat{p}_{12}^{KMW}(s,t) = \frac{\sum\limits_{i=1}^{n} W_i I(s < \tilde{Z}_i \leq t, \tilde{T}_i > t)}{\sum\limits_{i=1}^{n} W_{1i} I(\tilde{Z}_i > s)}$$

and

$$\hat{p}_{22}^{KMW}(s,t) = \frac{\sum\limits_{i=1}^{n} W_i I(\tilde{Z}_i \leq s, \tilde{T}_i > t)}{\sum\limits_{i=1}^{n} W_i I(\tilde{Z}_i \leq s, \tilde{T}_i > t)},$$

where W_i and W_{1i} are the Kaplan-Meier weights attached to $\tilde{T}_i$ and $\tilde{Z}_i$ when estimating the marginal distribution of T and Z from $(\tilde{T}_i, \Delta_i)$ and $(\tilde{Z}_i, \Delta_{1i})$. The Kaplan-Meier weights are given by the expression

$$W_i = \frac{\Delta_i}{n-i+1} \prod_{j=1}^{i-1} [1 - \frac{\Delta_j}{n-j+1}].$$

4.3.3 Kaplan-Meier pre-smoothed weighted estimator

A modification of the Kaplan-Meier weighted estimator was proposed by Amorim et al. (2011). This modification is based on pre-smoothing and allows for variance reduction when censoring is present. The censoring indicator variables are replaced by Kaplan-Meier weights by a smooth fit of a binary regression. The pre-smoothed Kaplan-Meier weights are given by

$$W_i^* = \frac{m(T_{1i}, \tilde{T}_i)}{n - R_i + 1} \prod_{j=1}^{i-1} [1 - \frac{m(\tilde{T}_{1j}, \tilde{T}_j)}{n - R_j + 1}].$$

In this expression,

$$m(x, y) = P(\Delta_2 = 1 | \tilde{T}_1 = x, \tilde{T} = y, \Delta_1 = 1)$$

and $m(\tilde{T}_1, \tilde{T})$ belongs to a parametric family of binary regression curves such as the logistic regression curve.

It can be assumed in practice that

$$m(x, y) = m(x, y; \underline{\beta}),$$

where $\underline{\beta}$ is a vector of parameters computed by maximising the conditional likelihood of the $\Delta_2's$ given the $(\tilde{T}_1, \tilde{T})$ for those with $\Delta_1 = 1$. Where no pre-smoothing is present, the Kaplan-Meier pre-smoothed weighted estimator reduces to the Kaplan-Meier weighted estimator. It was shown by Amorim et al. (2011) that the pre-smoothed estimator gains efficiency.

4.3.4 Accounting for covariates

Estimation methods for the transition probabilities conditional on current or past measures are introduced by Meira-Machado et al. (2012) to account for the influence of covariates. Meira-Machado et al. (2012) provide two non-parametric regression estimators for the conditional transition probabilities $p_{hj}(s, t|X)$, where X represent the current or past measure referred to above. Both these estimators are valid when the system is either Markovian or non-Markovian. Inverse censoring probability reweighting are used in both these estimators to deal with right censoring. Local smoothing is done by introducing regression weights based on local constant such as the Nadaraya-Watson or on local linear regression.

Meira-Machado et al. (2012) uses the following notation. The conditional distribution function of C given $X = X_i$ is denoted by G_{X_i} and $\hat{G}_{X_i}$ is its estimator. An estimator was introduced by Beran (1981) and is given by

$$\hat{G}_x(t) = \prod_{T_i \leq t, \Delta_i = 0} [1 - \frac{NW_{0i}(x, a_n)}{\sum_{j=1}^{n} I(T_j \geq T_i) NW_{0j}(x, a_n)}],$$

with

$$W_{0i}(x, a_n) = \frac{K_0((x - X_i)/a_n}{\sum_{j=1}^{n} K_0((x - X_i)/a_n},$$

where $NW_{0i}(x, a_n)$ is the Nadaraya-Watson weights, K_0 is a known probability density function and a_n is a sequence of bandwidths. When all the weights are equal, this estimator reduces to the Kaplan-Meier estimator (Kaplan and Meier, 1958). Consequently, the inverse probability censoring weighted estimators are given by

$$\hat{p}_{11}^{IPCW}(s, t|X = x) = \frac{\sum_{i=1}^{n} NW_{1i}(x, b_n) \frac{I(\tilde{Z}_i > t)\Delta_i}{1 - \hat{G}_{X_i}(T_i^-)}}{\sum_{i=1}^{n} NW_{1i}(x, b_n) \frac{I(\tilde{Z}_i > s)\Delta_i}{1 - \hat{G}_{X_i}(T_i^-)}},$$

$$\hat{p}_{12}^{IPCW}(s, t|X = x) = \frac{\sum_{i=1}^{n} NW_{1i}(x, b_n) \frac{I(s < \tilde{Z}_i \leq t, \tilde{T}_i > t)\Delta_i}{1 - \hat{G}_{X_i}(T_i^-)}}{\sum_{i=1}^{n} NW_{1i}(x, b_n) \frac{I(\tilde{Z}_i > s)\Delta_i}{1 - \hat{G}_{X_i}(T_i^-)}}$$

and

$$\hat{p}_{22}^{IPCW}(s, t|X = x) = \frac{\sum_{i=1}^{n} NW_{1i}(x, b_n) \frac{I(\tilde{Z}_i \leq s, \tilde{T}_i > t)\Delta_i}{1 - \hat{G}_{X_i}(T_i^-)}}{\sum_{i=1}^{n} NW_{1i}(x, b_n) \frac{I(\tilde{Z}_i \leq s, \tilde{T}_i > s)\Delta_i}{1 - \hat{G}_{X_i}(T_i^-)}},$$

where $NW_{1i}(x, b_n)$ is the Nadaraya-Watson weight and $\hat{G}_{X_i}(T_i^-) = \hat{G}_{x=X_i}(T_i^-)$.

Lin et al. (1999) introduced an approach for the bivariate distribution function which also accounts for the influence of covariates. A different set of estimators is obtained and given by

$$\hat{p}_{11}^{LIN}(s, t|X = x) = \frac{\sum_{i=1}^{n} NW_{1i}(x, b_n) \frac{I(\tilde{Z}_i > t)}{1 - \hat{H}_{X_i}(t^-)}}{\sum_{i=1}^{n} NW_{1i}(x, b_n) \frac{I(\tilde{Z}_i > s)}{1 - \hat{H}_{X_i}(s^-)}},$$

$$\hat{p}_{12}^{LIN}(s, t|X = x) = \frac{\sum_{i=1}^{n} NW_{1i}(x, b_n) \frac{I(s < \tilde{Z}_i \leq t, \tilde{T}_i > t)}{1 - \hat{G}_{X_i}(t^-)}}{\sum_{i=1}^{n} NW_{1i}(x, b_n) \frac{I(\tilde{Z}_i > s)}{1 - \hat{G}_{X_i}(s^-)}}$$

and

$$\hat{p}_{22}^{LIN}(s, t|X = x) = \frac{\sum_{i=1}^{n} NW_{1i}(x, b_n) \frac{I(\tilde{Z}_i \leq s, \tilde{T}_i > t)}{1 - \hat{G}_{X_i}(t^-)}}{\sum_{i=1}^{n} NW_{1i}(x, b_n) \frac{I(\tilde{Z}_i \leq s, \tilde{T}_i > t)}{1 - \hat{G}_{X_i}(s^-)}},$$

where $\hat{H}_X$ is the Kaplan-Meier estimator of the conditional distribution of C given X based on the $(\tilde{Z}_i, 1 - \Delta_{i1})'s$, which is defined similarly to $\hat{G}_x$. C is assumed to be independent of $(z, T)|X$ with the assumption not excluding the possibility of dependent censoring. The approach by Lin et al. (1999) has the disadvantage of occasionally providing non-monotone curves for the transition probabilities which makes the first approach more recommendable according to Araújo et al. (2014).

4.3.5 Location-scale estimator

Keilegom et al. (2011) proposed another estimator of the transition probabilities. This estimator assumes that the vector of gap times $(Z, Y = T - Z)$ satisfies the non-parametric location-scale regression model which allows for the transfer of tail information from lightly censored areas to heavily censored areas.

Meira-Machade et al. (2013) introduces an automatic bandwidth procedure. The non-parametric location-scale regression model

$$Y = m(Z) + \sigma(Z)\varepsilon$$

is considered where the functions m and σ are smooth functions and ε is independent of Z. A non-parametric estimator of the distribution of the error variable F_ε is proposed by Meira-Machade et al. (2013). A Kaplan-Meier estimator of F_ε is based on the $(\hat{E}_i, \Delta_i)'s$ where

$$\hat{E}_i = \frac{\hat{Y}_i - \hat{m}(\tilde{Z}_i)}{\hat{\sigma}(\tilde{Z}_i)},$$

which is used to construct the estimator for the conditional distribution of the second gap time

$$\hat{F}(y|x) = \hat{F}_\varepsilon(\frac{y - \hat{m}(x)}{\hat{\sigma}(x)}).$$

An extension of the estimator given by Beran (1981) is used to estimate the location and scale functionals. This estimator functions well with censoring in the first gap time. The estimators for the transition probabilities is given by the expressions

$$\hat{p}_{11}^{LS}(s,t) = \frac{1 - \hat{F}_1(t)}{1 - \hat{F}_1(s)},$$

$$\hat{p}_{12}^{LS}(s,t) = \frac{1}{1 - \hat{F}_1(s)} \int_s^t [1 - \hat{F}(t - u|u)]\hat{F}_1(du)$$

and

$$\hat{p}_{22}^{LS}(s,t) = \frac{\int\limits_0^s [1 - \hat{F}(t-u|u)]\hat{F}_1(du)}{\int\limits_0^s [1 - \hat{F}(s-u|u)]\hat{F}_1(du)},$$

where $F_1(.)$ represent the marginal distribution of the first gap time, which is estimated by the Kaplan-Meier estimator based on the $(\tilde{Z}_i, \Delta_{1i})'s$. This transfer of tail information improves the estimate of the transition probabilities specifically in points where the uncensored information is scarce (Meira-Machado et al., 2013). This location-scale method was shown to outperform the Kaplan-Meier weighted estimator. This was shown by Meira-Machado et al. (2006). The Kaplan-Meier weighted estimator however becomes better when the model deviates a lot from a location-scale model. A disadvantage of the location-scale method is the fact that it can only be used for modelling of the progressive three state model.

4.3.6 State occupation probabilities

The estimation of the state occupation probabilities is another important capability of multi-state modelling. Three state occupation probabilities must be estimated for the illness-death model. These three state occupation probabilities include $p_{11}(0,t)$, $p_{12}(0,t)$ and $p_{13}(0,t)$. It was shown by Datta and Satten (2001) that these probabilities can be estimated by the Aalen-Johansen estimates without the process being Markovian. Araújo et al. (2014) recommends the two Markovian approaches discussed, namely the Aalen-Johansen estimator and the Pre-smoothed Aalen-Johansen estimator.

4.4 Data Simulation Using the *TPmsm* Package

The function *dgpTP* can be used to generate data from the illness-death model. For this model, all individuals are assumed to be in state 1 at time $t = 0$. The subject's history can be divided into two groups according to whether state 2 was entered $(1 \longrightarrow 2 \longrightarrow 3)$ or $(1 \longrightarrow 3)$.

For the $(1 \longrightarrow 2 \longrightarrow 3)$ subgroup of subjects, the successive gap times $(Z, T - Z)$ can be simulated from two of the well-known copula functions including Farlie-Gumbel-Morgenstern copula with exponential marginals or the bivariate Weibull distribution. The *dgpTP* function simulated data from the illness-death model using Gumbel's bivariate exponential distribution

$$F_{12}(x,y) = F_1(x)F_2(y)[1 + \theta\{1 - F_1(x)\}\{1 - F_2(y)\}]$$

with unit exponential margins. The amount of dependency between the gap times $(Z, T - Z)$ is controlled by the parameter θ. The *corrTP* function can be used to obtain the theoretical correlation between the gap times.

For the $(1 \longrightarrow 3)$ subgroup of subjects, the survival time is simulated according to an exponential distribution with rate parameter 1.

4.5 Implementation of Data Simulation

The data simulation was divided into two parts. The subjects that entered at state 1 (referred to as part 1) was simulated separately from those subjects that entered at state 2 (referred to as part 2). Based on the Isimo Health dataset, 80% of subjects start in the curative state (state 1) and the remaining 20% start in non-curative state (state 2). The desired sample size of the simulation is 10 000.

In part 1 of the simulation, the *dgpTP* function was used to generate a sample of size 8 000, contributing 80% of the 10 000, assuming no correlation in Gumbel's bivariate exponential distribution, using an independent uniform censoring time, according to model $U(0, 3)$. Markov data were simulated by using corr=0 since a correlation of zero in Gumbel's bivariate exponential distribution leads to independent gap times. Based on the Isimo Health dataset, the proportion of transitions into state 2 (from state 1) is 75%. A value of one would have led to the progressive three state model where all subjects pass through state 2. This is not the desired outcome of this simulation. The first 10 observations of the simulated data are given in Table 4.1.

Table 4.1 First ten observations of the first part of the simulated data

	time1	event1	Stime	event
1	1.0816105152	0	1.0816105152	0
2	2.0822430096	1	2.0822430096	1
3	1.2925065755	1	1.8331302023	1
4	1.6612869111	0	1.6612869111	0
5	1.4274986331	1	1.6332586317	1
6	1.4527274890	1	1.8421824081	1
7	0.7698157828	1	1.9346533519	0
8	0.3889267237	1	2.7234675501	0
9	0.4012843777	0	0.4012843777	0
10	0.8713118535	0	0.8713118535	0

The *transAJ* command gives the Aalen-Johansen estimate of the transition probabilities. This command provides a 95% pointwise confidence interval using 10 000 bootstrap replicates. The pointwise confidence interval is constructed by random sampling the items from the simulated (original) dataset with replacement. The starting time is specified as zero and the ending time three representing three years. The Aalen-Johansen transition probabilities are given as

$$P = \begin{bmatrix} 0.4757012 & 0.2627678 & 0.2615309 \\ 0 & 0.4785433 & 0.5214567 \\ 0 & 0 & 1 \end{bmatrix},$$

with confidence bands

$$\begin{bmatrix} 0.4518965 & 0.2383743 & 0.2397261 \\ 0 & 0.4322844 & 0.4774942 \\ 0 & 0 & 1 \end{bmatrix}$$

and

$$\begin{bmatrix} 0.4982338 & 0.2874388 & 0.2853586 \\ 0 & 0.5225058 & 0.5677156 \\ 0 & 0 & 1 \end{bmatrix}.$$

Similarly, the *transPAJ* command gives the pre-smoothed Aalen-Johansen estimate of the transition probabilities and provides a 95% pointwise confidence interval using 10 000 bootstrap replicates. The pre-smoothed Aalen-Johansen transition probabilities are given as

$$P = \begin{bmatrix} 0.2470805 & 0.1711035 & 0.5818160 \\ 0 & 0.3046237 & 0.6953763 \\ 0 & 0 & 1 \end{bmatrix},$$

with confidence bands

$$\begin{bmatrix} 0.2212681 & 0.1439576 & 0.5496157 \\ 0 & 0.2577273 & 0.6488574 \\ 0 & 0 & 1 \end{bmatrix}$$

and

$$\begin{bmatrix} 0.2745121 & 0.1975188 & 0.6143271 \\ 0 & 0.3511426 & 0.7422727 \\ 0 & 0 & 1 \end{bmatrix}.$$

Part 2 of the simulation involved the subjects that enter at state 2 and either stay in state 2 or move to the absorbing state 3. The *dgpTP* function was used once again to generate a sample of size 30 000 using the same model parameters as the previous simulation. The sample size was required to be large in order to compensate for removing observations with event1=0 (subjects left state 1). The second filter would be applicable to subjects that had time1 unequal to Stime. If time1=Stime, it would mean that the subject moved from state 1 to state 3 (instead of moving to state 2). After both the filters were applied, the first 2 000 subjects were extracted and the variable time1 was given the value 0 to imply that subjects spent 0 time in state 1 and the variable event1 was given the value 1 to state that the subject left state 1 and therefore entered at state 2. The Stime variable was formed by taking the Stime value minus the time1 value to extract the time that the subject spent in state 2. This dataset then formed the second part of the simulation. The first 10 observations of the simulated data are given in Table 4.2.

Table 4.2 First ten observations of the second part of the simulated data

	time1	event1	Stime	event
1	0	1	1.51079296	0
2	0	1	0.05436288	1
3	0	1	1.41029859	0
4	0	1	1.76382468	0
5	0	1	0.69662589	1
6	0	1	2.12838743	0
7	0	1	0.95875733	0
8	0	1	1.14429681	0
9	0	1	0.08896630	0
10	0	1	2.50868498	0

The two datasets were then combined to form the total simulated dataset consisting of 10 000 subjects.

Thereafter, the researcher distinguished between six different possible scenarios. Scenario A being that the subject enters at state 1 and remains in state 1 for the entire duration. Scenario B is when the subject enters at state 1 and moves to state 2. Scenario C is when a subject enters at state 2 and remains in state 2. Scenario D is when a subject enters at state 1, moves to state 2 and ends in the absorbing state 3. Scenario E is when the subject enters at state 1 and moves to state 3 directly. The last scenario, scenario F, is when a subject enters at state 2 and moves to state 3. These scenarios are ordered from best to worst case scenario in terms of prognosis of the disease. These six scenarios were used to simulate the covariates.

Three covariates were simulated representing different types of real-world data. The underlying distributions from which these covariates were simulated are given below. Three different covariates were desired and three different distributions was chosen to represent different types of variables. Thereafter, it was decided on which variables from the Isimo dataset, these variables should be simulated from.

Covariate1 was simulated from a normal distribution. The probability density function for a normal distribution is given by the expression

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2},$$

where μ is the population mean and σ^2 the variance . $X \sim N(\mu, \sigma^2)$ represents a random variable X that follows the normal distribution with mean μ and standard deviation σ. The *rnorm* function in the *stats* package in R was used to simulate the data from a normal distribution.

The age variable from the dataset obtained from Isimo Health was used to determine the mean and standard deviation for the normal distribution within each of these six scenarios defined above. The respective means and standard deviations used for the normal distributions are given in Table 4.3 below.

Table 4.3 Mean and standard deviations used to simulate Covariate1

Scenario	mean (μ)	standard deviation (σ)
A	55.85906	14.32430
B	50.14286	15.43651
C	54.88636	15.08613
D	56.01613	15.60842
E	56.30894	14.54591
F	57.72917	15.39514

The second covariate, Covariate2, was generated from a Bernoulli distribution. The Bernoulli distribution is a discrete distribution with two possible outcomes being either that a success occurs with probability p or a failure occurs with probability $q = 1 - p$. The probability mass function for the Bernoulli distribution is given by the expression

$$P(n) = p^n(1-p)^{1-n}.$$

The binomial distribution gives the probability of obtaining n successes out of N Bernoulli trials. Therefore, the binomial distribution can be used to simulate the Bernoulli distribution with only $N = 1$ trial. The *rbinom* function in the *stats* package in R can be used to simulate the data from a binomial distribution.

The probability parameter is based on the probability of being HER2 positive in the Isimo Health dataset within each of the six scenarios. The respective probabilities are given in Table 4.4.

Table 4.4 Probability of success used to simulate Covariate2

Scenario	Probability of Success
A	0.200935
B	0.250000
C	0.193548
D	0.333333
E	0.216000
F	0.370370

The third and last covariate, Covariate3, is generated from a multinomial distribution. The multinomial distribution is another generalisation of the binomial distribution. The multinomial distribution models the outcome of n experiments where each of the n trials has an outcome with a categorical distribution. The probability mass function for the multinomial distribution is given by the expression

$$
\begin{aligned}
f(x_1, x_2, ..., x_k, p_1, p_2, ..., p_k) &= P(X_1 = x_1, ..., X_k = x_k) \\
&= \begin{cases} \frac{n!}{x_1! x_2! ... x_k!} p_1^{x_1} ... p_k^{x_k} & \sum\limits_{i=1}^{k} x_i = n \\ 0 & otherwise \end{cases},
\end{aligned}
$$

for non-negative integers $x_1, x_2, ..., x_k$.

The node size from the Isimo Health dataset was used to get to the probabilities. The probabilities of belonging to the groups 0, 1, 2 or 3 are given in Table 4.5 below.

Table 4.5 Probabilities used to simulate Covariate3

Scenario	Group=0	Group=1	Group=2	Group=3
A	0.486692	0.311787	0.144487	0.057034
B	0.400000	0.200000	0.300000	0.100000
C	0.322581	0.258065	0.354839	0.064516
D	0.282609	0.347826	0.239130	0.130435
E	0.451104	0.315457	0.160883	0.072555
F	0.250000	0.388889	0.222222	0.138889

The six datasets from the six different scenarios were combined to form a complete simulated dataset with 10 000 rows (subjects) and 7 columns. A preview of the data is given in Table 4.6.

Table 4.6 First ten observations of the complete dataset

	time1	event1	Stime	event	Covariate1	Covariate2	Covariate3
1	1.0816105	0	1.0816105	0	46.88555	0	0
2	1.6612869	0	1.6612869	0	58.48962	0	1
3	0.4012844	0	0.4012844	0	43.88927	1	0
4	0.8713119	0	0.8713119	0	78.71034	0	1
5	0.1311183	0	0.1311183	0	60.57903	0	1
6	0.8373290	0	0.8373290	0	44.10642	0	0
7	0.7654208	0	0.7654208	0	62.84114	0	1
8	0.7800595	0	0.7800595	0	66.43504	1	0
9	0.2040727	0	0.2040727	0	64.10672	0	1
10	1.5666755	0	1.5666755	0	51.48459	0	0

The last step was to change the variable type of Covariate2 and Covariate3 to a factor variable since this will have an impact when applying the imputation techniques.

4.6 Conclusion

A dataset representing the Isimo Health dataset was successfully simulated in this chapter. The simulated dataset will be used in Chapter Five to test two different imputation techniques to impute missing values obtained in covariates. Thereafter, the imputation technique performing the best, based on statistical measures, will be used to impute the Isimo Health dataset.

CHAPTER 5

IMPUTATION OF MISSING DATA

5.1 Chapter Overview

Two of the imputation techniques discussed in the Literature Review chapter (Chapter Two) were applied to the data simulated in Chapter Four. Based on the literature review, two imputation techniques, one based on chained equations and the other based on random forests, were chosen for comparison to identify the best performing imputation technique. The R packages chosen to perform these imputation techniques were *mice* and *missForest*.

According to Davila and Rosado (2017), the performance of an imputation technique is dependent on the percentage of missing data in the dataset. The dataset of Chapter Four was modified to reflect different ratios of missing values. R was used to eliminate data at random from the dataset simulated in Chapter Four of this thesis. Three sets of data containing missing data were created. The datasets contained 5% (missing.05), 10% (missing.10) and 15% (missing.15) of missing data in each covariate, respectively. The missing data created can be seen as completely at random, since the process of deleting the entries was not influenced by the data or the data generating process. This also means that there should be an evenly distributed amount of missing values over the variables in the dataset.

5.2 Patterns of Missing Data

The pattern of missing data can be determined visually by making a bar chart of the missing value proportions or through the so called *md.pattern* function, which provides a summary of the missing values. This study makes use of both techniques. The patterns of the missing data in the missing.05 dataset, using the *md.pattern* function in R, can be seen in Table 5.1.

Table 5.1 Missing data pattern for missing.05 dataset

	time1	event1	Stime	event	Covariate1	Covariate2	Covariate3	Number of missing variables
8576	1	1	1	1	1	1	1	0
465	1	1	1	1	1	1	0	1
451	1	1	1	1	1	0	1	1
26	1	1	1	1	1	0	0	2
432	1	1	1	1	0	1	1	1
29	1	1	1	1	0	1	0	2
21	1	1	1	1	0	0	1	2
	0	0	0	0	482	498	520	1500

In Table 5.1, the column to the left gives the number of observations containing the missing data pattern, indicated to the right of this column, where the zero indicates missing values within a variable. The column to the right gives the number of variables containing missing values within that combination. To clarify, in the top row, there are 8576 observations that contain no missing values. As another example, in the second row, there are 465 observations with missing values only in Covariate3. The last row gives the total number of observations missing within each covariate. Table 5.2 and Table 5.3 gives the missing data pattern for the 10% and 15% missing datasets, respectively.

Table 5.2 Missing data pattern for missing.10 dataset

	time1	event1	Stime	event	Covariate1	Covariate2	Covariate3	Number of missing variables
7293	1	1	1	1	1	1	1	0
859	1	1	1	1	1	1	0	1
796	1	1	1	1	1	0	1	1
91	1	1	1	1	1	0	0	2
769	1	1	1	1	0	1	1	1
97	1	1	1	1	0	1	0	2
85	1	1	1	1	0	0	1	2
10	1	1	1	1	0	0	0	3
	0	0	0	0	961	982	1057	3000

Table 5.3 Missing data pattern for missing.15 dataset

	time1	event1	Stime	event	Covariate1	Covariate2	Covariate3	Number of missing variables
6179	1	1	1	1	1	1	1	0
1098	1	1	1	1	1	1	0	1
1071	1	1	1	1	1	0	1	1
202	1	1	1	1	1	0	0	2
1006	1	1	1	1	0	1	1	1
199	1	1	1	1	0	1	0	2
212	1	1	1	1	0	0	1	2
33	1	1	1	1	0	0	0	3
	0	0	0	0	1450	1518	1532	4500

Figure 5.1 contains a visual representation of the missing data pattern of the dataset with 5% of missing values, another output provided by the *md.pattern* function. Figure 5.1 therefore gives the results in Table 5.1 visually.

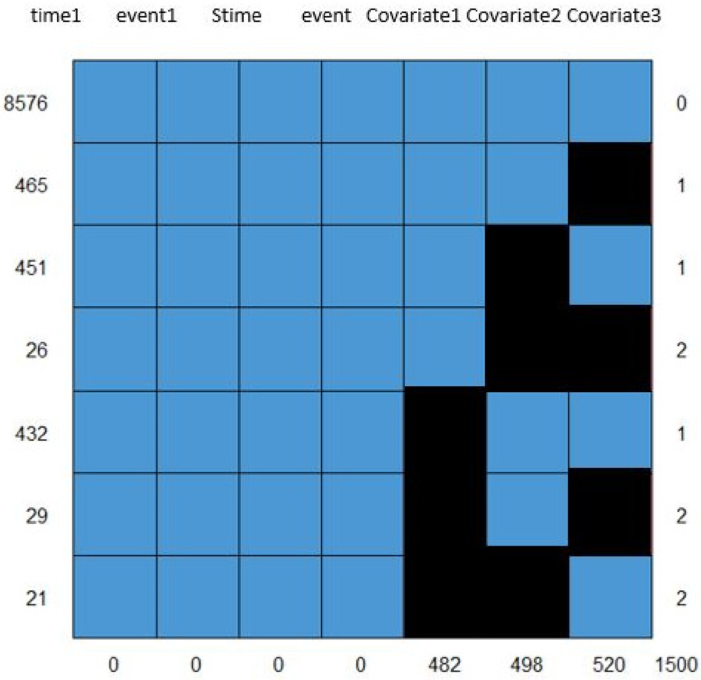


Figure 5.1 Missing data pattern of missing.05 dataset display

Figures 5.2 and 5.3 gives the graphical representation of the missing data patterns for the missing.10 and missing.15 datasets.

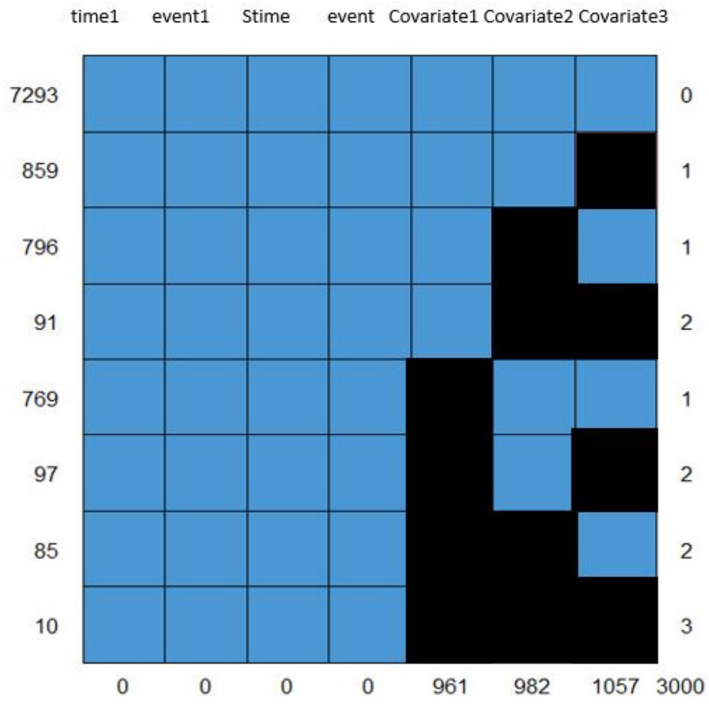


Figure 5.2 Missing data pattern of missing.10 dataset display

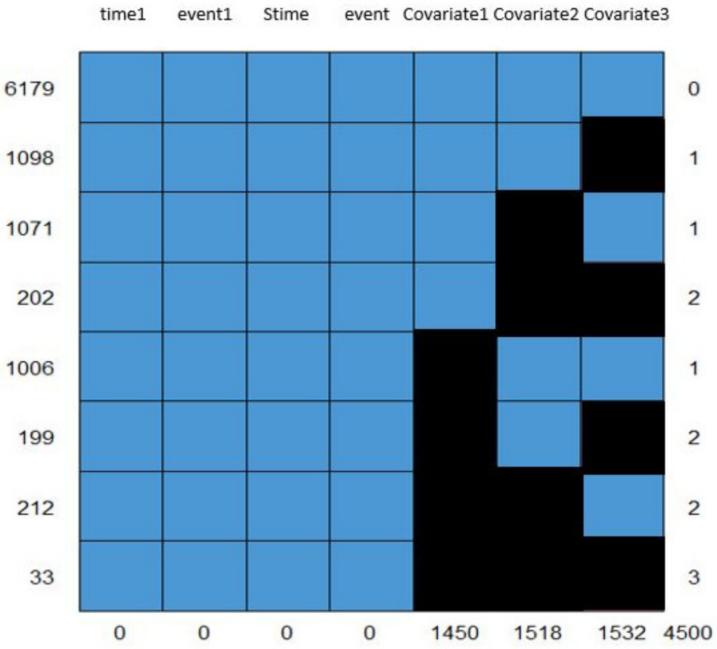


Figure 5.3 Missing data pattern of missing.15 dataset display

The bar chart of the proportion of missing values are given in Figure 5.4. Once again, this bar chart gives the same results as in Table 5.1 and Figure 5.1.

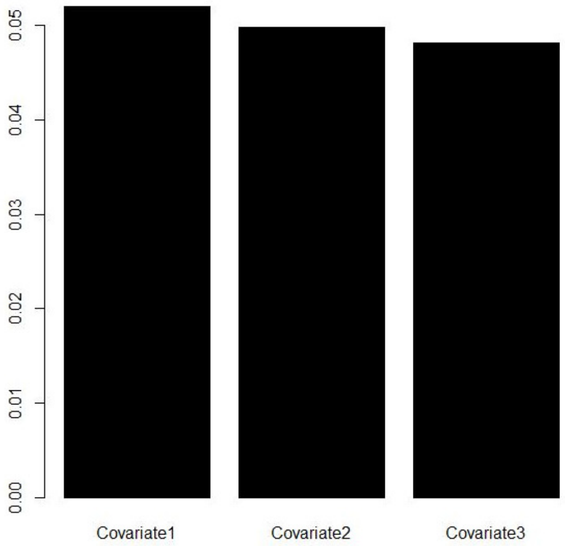


Figure 5.4 Proportion of missing data in the covariates of missing.05 dataset

Figures 5.5 and 5.6 gives the bar charts of the proportion of missing values in the missing.10 and missing.15 datasets.

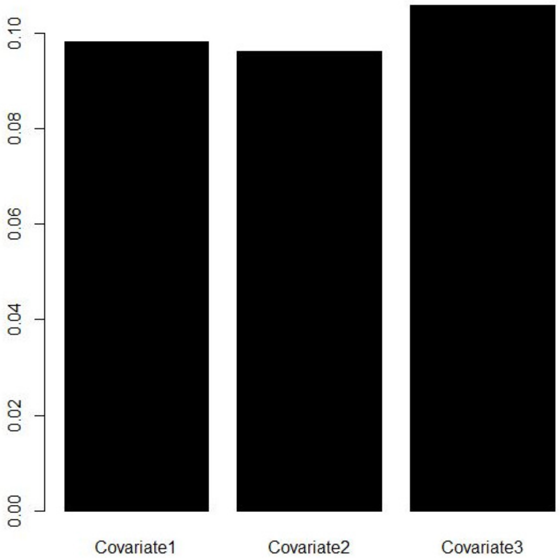


Figure 5.5 Proportion of missing data in the covariates of missing.10 dataset

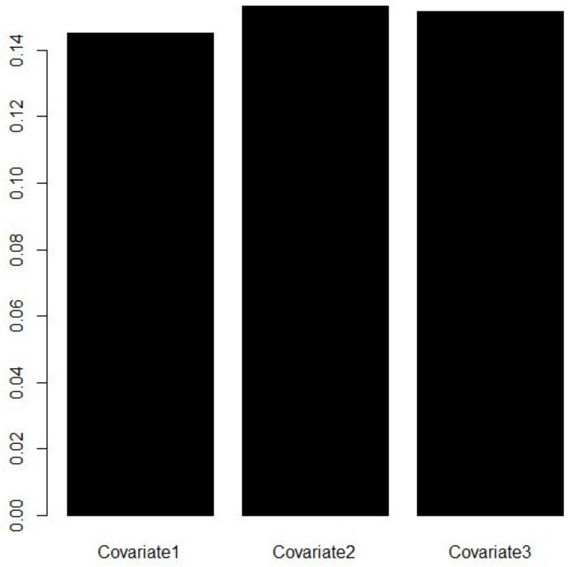


Figure 5.6 Proportion of missing data in the covariates of missing.15 dataset

5.3 R Packages for Imputation

R has several robust packages for imputing missing values. The *mice* package creates multiple imputations instead of a single imputation. The *missForest* package treats the missing data problem as a prediction problem. The data is imputed by regressing each of the variables against all of the other variables and then predicting the missing data for the dependent variable by using the fitted forest. The *missForest* approach was chosen since it was shown by Waljee et al. (2013), that it outperforms other well-known methods such as the k-nearest neighbours (KNN).

5.3.1 The *mice* package

The Multivariate Imputation using Chained Equations (*mice*) package is one of the packages most frequently used in R for imputation (Van Buuren and Groothuis-Oudshoorn, 2011). This package creates multiple imputations instead of a single imputation, such as taking the average. This accounts for the uncertainty in missing values. The *mice* package assumes that the missing data are MAR, which implies that the probability of a value being missing is only dependent on the observed value and can therefore be predicted by using those values. This package specifies an imputation model per variable and therefore imputes data on a variable by variable basis.

Let the variables of a dataset be X_1,X_2,...,X_k. If the variable X_1 contains missing values, it will be regressed on the other variables X_2,...,X_k. The predicted values that are obtained will replace the missing values in variable X_1. Linear regression is the default method to predict continuous missing values and for categorical missing values, logistic regression is used. Once the imputation process is complete, multiple datasets are generated only differing by the imputed missing values.

The imputation methods used by this package are summarised in Table 5.4.

Table 5.4 Imputation methods used by the *mice* package

Method	Application
Predictive Mean Matching (PMM)	Numeric variables.
Logistic Regression (logreg)	Binary variables with 2 levels.
Bayesian Polytomous Regression (polyreg)	Factor variables with $\geq$ 2 levels.
Proportional Odds Model	Ordered variables with $\geq$ 2 levels.

The missing values were imputed for each of the three covariates separately. The *mice* function contains several parameters of which an explanation of the most important ones is given. The argument *m* refers to the number of imputed datasets. Each of the three missing datasets (5%, 10% and 10%) was imputed 1 000 (*m*=1 000) times in order to ensure accuracy of the measures. The argument *maxit* refers to the number of iterations taken to impute the missing values. In this imputation, the number of iterations was taken as 5 since it is just too computer intensive to take any more than that. Finally, *method* refers to the method used in imputation. In this imputation, predictive mean matching was used for Covariate1, Logistic Regression for Covariate2 and the Proportional Odds Model for Covariate3.

5.3.2 R package *missForest*

5.3.2.1 Background on *missForest* package

The selection of arguments with respect to feasibility and accuracy issues are discussed in the user guide by Stekhoven (2011). The *missForest* package provides a non-parametric imputation method that can be used for a wide range of different datasets (Stekhoven, 2011). A non-parametric method does not make explicit assumptions about the functional form of an arbitrary function f. It rather attempts to estimate f in such a manner that it can be close to the data points without seeming impractical. The only requirement for this algorithm to work is that the observations must be mutually independent. Stekhoven (2011) states that the *missForest* algorithm is based on the random forest algorithm, developed by Breiman (2001), and is therefore dependent on the R implementation of Random Forest by Liaw and Matthew (2002).

Basically, the *missForest* algorithm fits a random forest on the observed part and then predict the missing information. According to Stekhoven (2011), these two steps are repeated until a stopping criterion is met or the specified number of maximum iterations is reached, whichever comes first. During the iterative process, the imputed matrix is updated continuously, variable by variable. The performance is assessed between iterations. The assessment of the performance between iterations is done by considering the difference in results between the previous imputation result and the new imputation result. The algorithm stops as soon as the difference increases, meaning there is no more improvement.

This algorithm yields an out of bag (OOB) imputation error estimate and provides a high level of control over the imputation process (Stekhoven, 2011). This algorithm can also account for categorical variables and therefore, the *missForest* package can be used for datasets with different types of variables.

5.3.2.2 Function arguments

The *maxiter* argument controls the number of iterations that are allowed. It might be required by the data that more than the usual five (default) iterations are required until the stopping criteria kicks in. It is the ultimate goal for the algorithm to stop due to the stopping criterion and not due to the maximum number of iterations being reached. The difference in improvement might in some cases be so marginal that is reasonable to limit the number of iterations. The number of iterations will also influence the run time of the algorithm and therefore it is necessary to have a maximum number of iterations. (Stekhoven, 2011).

There is a speed versus accuracy trade-off to be made by manipulating the arguments *ntree* and *mtry*. The *missForest* package grows, in each of the iterations for each of the variables, a random forest to impute the missing values within that variable. A large number of variables can lead to an undesired long computational time. The computational time of the imputation process can be lowered by either reducing the number of trees that is grown in each forest (*ntree*) or by reducing the number of variables that are randomly sampled at each split (*mtry*). The reduction of either of these arguments will however have a reduction effect on the accuracy of the process as well. (Stekhoven, 2011).

According to Stekhoven (2011), *ntree* have a linear effect on the computation time. Therefore, when halving the *ntree* argument, the computation time will also be halved. The default value of *ntree* is 100. This is quite a large number of trees and it can be shown that a smaller *ntree* value can also produce appropriate results.

The change in the *mtry* argument have a larger effect in high dimensional cases. The default for the *missForest* is the square-root of the number of dimensions. This delivers a good trade-off between imputation error and computation time.

5.3.2.3 Function output

The imputed data matrix is given as output with the name *ximp*. The estimated OOB imputation error is given by *OOBerror*. The Normalised Root Mean Squared Error (NRMSE) is returned for continuous variables and the proportion of falsely classified (PFC) entries is returned for categorical variables (Stekhoven, 2013).

The NRMSE is defined as

$$\sqrt{\frac{mean((X_{true} - X_{imp})^2)}{var(X_{true})}},$$

where X_{true} is the complete dataset, X_{imp} is the imputed dataset and the $mean$/var are used as a short notation for the empirical mean and variance computed over the continuous missing values (Stekhoven, 2013). Values closer to zero are preferred for the NRMSE measurement (Dávila and Rosado, 2017).

According to Stekhoven (2013), since the simulated dataset contains a mixed type of variables, the *mixError* function in the *missForest* package can be used to compute the imputation error for mixed-type data.

5.4 Measurement of Imputation Technique Performance

Seven measures of performance were chosen to adequately choose the best imputation technique. The *Metrics* (Hamner and Frasco, 2018) package in R was used to calculate these seven measures of performance. The seven measures are listed and described below. The mathematical formulas for each of the measures are also given with X_i being the actual value of observation i, $\hat{X}_i$ the imputed value of observation i and $\bar{X}$ the mean value of the actual values.

5.4.1 Mean Squared Error

The Mean Squared Error (MSE) is the average squared difference between the actual and imputed values. It is an overall measure of the size of the imputation error (Rice, 2007, p.136). The function in the *Metrics* package for the MSE is called with *mse*(). The MSE can be calculated by using

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{X}_i)^2.$$

5.4.2 Accuracy

The accuracy measure is the measurement of the proportion of elements in the actual data that are equal to the corresponding element in the imputed data. This is the only performance measure that a higher value is desired, where all the other measures mentioned requires small values. The accuracy is calculated in R through the *Metrics* package by calling the function *accuracy*().

5.4.3 Mean Absolute Error

The Mean Absolute Error (MAE) is the average absolute difference between the actual and imputed values. The function in the *Metrics* package for the MAE is called with *mae*(). Similar to the MSE, a lower value of the MAE is better. The MAE can be calculated by using

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|X_i - \hat{X}_i|.$$

5.4.4 Relative Absolute Error

The Relative Absolute Error (RAE) is the relative absolute error between the actual and imputed values. The function in the *Metrics* package for the RAE is called with *rae*(). Similar to the MSE and MAE a lower value of the RAE is better. The REA can be calculated by using the formula

$$RAE = \frac{\sum\limits_{i=1}^{n} |X_i - \hat{X}_i|}{\sum\limits_{i=1}^{n} |X_i - \bar{X}|}.$$

5.4.5 Root Mean Square Error

The Root Mean Square Error (RMSE) is the root mean squared difference between the actual and imputed values. It measures the difference between the actual values and the imputed values (Schmitt et al., 2015). According to Schmitt et al. (2015), this measure basically represents the sample standard deviation of the difference. The function to calculate the RMSE is called *rmse*() in the *Metrics* package. A lower value of the RMSE is better. The RMSE can be calculated by using

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{X}_i)^2}.$$

5.4.6 Sum of Squared Errors

The Sum of Squared Errors (SSE) is the sum of the squared differences between the actual and imputed values. The function in the *Metrics* package for the SSE is called with *sse*(). Similar to all the other measures, except for accuracy, a lower value of the SSE is better. The SSE can be calculated by using

$$SSE = \sum_{i=1}^{n}(X_i - \hat{X}_i)^2.$$

5.4.7 Bias

Bias is the average amount by which the actual is greater than the predicted. The *bias*() function in R is used to calculate the bias. The bias is calculated as

$$bias = \frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{X}_i)$$

and a lower value of bias is desired.

In the MAE and RMSE, the average difference between the actual and imputed values are compared. It is therefore related to the scale of the observations. In RAE, the differences of the actual and imputed values are divided by the variation of the actual values. Therefore, the RAE values now have a scale from zero to one. The denominator, $\sum_{i=1}^{n}|X_i - \bar{X}|$, gives an indication of how much the actual values differ from the mean value.

5.5 Imputation Results

The results for the two imputation techniques are given in the tables below. Separate tables are given for each covariate as well as for each of the three different percentages of missing data.

Table 5.5 Imputation of Covariate1 - missing.05

	mice	missForest
MSE	22.458	10.974
Accuracy	-	-
MAE	0.859	0.606
RAE	0.072	0.051
RMSE	4.737	3.313
SSE	224581.782	109374.290
Bias	**-0.033**	-0.030

Table 5.6 Imputation of Covariate1 - missing.10

	mice	missForest
MSE	43.320	12.032
Accuracy	-	-
MAE	1.654	0.661
RAE	0.139	0.056
RMSE	6.580	3.446
SSE	433195.530	120317.549
Bias	0.066	-0.018

Table 5.7 Imputation of Covariate1 - missing.15

	mice	missForest
MSE	66.596	13.205
Accuracy	-	-
MAE	2.538	0.724
RAE	0.214	0.061
RMSE	8.160	3.558
SSE	665962.67	132048.926
Bias	0.073	-0.021

Table 5.8 Imputation of Covariate2 - missing.05

	mice	missForest
MSE	0.017	0.014
Accuracy	0.983	0.986
MAE	0.017	0.014
RAE	0.051	0.042
RMSE	0.132	0.120
SSE	172	144
Bias	**-0.001**	0.001

Table 5.9 Imputation of Covariate2 - missing.10

	mice	missForest
MSE	0.034	0.016
Accuracy	0.966	0.984
MAE	0.034	0.016
RAE	0.098	0.047
RMSE	0.184	0.126
SSE	339	164
Bias	**0.000**	0.001

Table 5.10 Imputation of Covariate2 - missing.15

	mice	missForest
MSE	0.050	0.018
Accuracy	0.950	0.982
MAE	0.050	0.018
RAE	0.145	0.052
RMSE	0.224	0.131
SSE	501	181
Bias	**-0.002**	0.000

Table 5.11 Imputation of Covariate3 - missing.05

	mice	missForest
MSE	**0.087**	0.098
Accuracy	0.968	0.970
MAE	**0.049**	0.050
RAE	**0.062**	0.064
RMSE	**0.295**	0.313
SSE	**869**	977
Bias	0.000	-0.002

Table 5.12 Imputation of Covariate3 - missing.10

	mice	missForest
MSE	0.187	0.107
Accuracy	0.929	0.966
MAE	0.105	0.055
RAE	0.134	0.071
RMSE	0.433	0.325
SSE	1872	1073
Bias	-0.002	-0.004

Table 5.13 Imputation of Covariate3 - missing.15

	mice	missForest
MSE	0.268	0.117
Accuracy	0.897	0.963
MAE	0.153	0.061
RAE	0.195	0.077
RMSE	0.517	0.335
SSE	2678	1170
Bias	-0.001	-0.005

In order to determine the best performing technique, various statistical measures were evaluated. It can be seen from the tables above, that the MSE, MAE, RAE, RMSE and SSE are lower for the *missForest* imputation for all three sets of data for Covariate1 and Covariate2. For Covariate3 however, the *mice* imputation technique is more favourable in the missing.05 dataset although the *missForest* imputation remains in the lead for the other two datasets. The bolded values in the tables indicate the values where the performance of the *mice* package is better than the performance of *missForest*. The bias is relatively similar in all cases with the bias being slightly lower for *mice* in Covariate1 dataset missing.05, as well as Covariate2 for all three datasets. The accuracy measure is better for the *missForest* technique in both covariates 1 and 2, and for all three datasets. The various statistical measures demonstrated that the *missForest* package had better performance in the imputation process.

5.6 Conclusion

Based on the evidence of the statistical measures, the *missForest* package performs better in imputation of the covariates in the Isimo Health data. Therefore, in Chapter Six, the *missForest* package will be used to impute the Isimo Health dataset and thereafter the multi-state Markov model will be fitted to the data.

CHAPTER 6
DATA ANALYSIS ON ISIMO HEALTH DATASET

6.1 Introduction

In this chapter the Isimo Health dataset is analysed using the results from Chapter Five. The pattern of the missing data in the covariates of the Isimo Health dataset is investigated. It was shown in Chapter Five that the *missForest* imputation technique performs better for imputation of missing values in the covariates. Therefore, the *missForest* technique is applied to the Isimo Health dataset. Thereafter, a multi-state Markov model is fitted to the imputed dataset. It is seen that the Markov assumption does not hold and therefore a semi-Markov model is fitted to the data. The *p3state.msm* package is used to fit the Markov and the semi-Markov multi-state models.

6.2 Missing Data

6.2.1 Missing data patterns

The *md.pattern()* function in the *mice* package is used to obtain a summary of the missing data and a graphical representation of the missing data that is present in the Isimo Health dataset.

Table 6.1 Isimo Health dataset missing data pattern

	time1	event1	Stime	event	gender	age	r_stage	ER	PR	node	HER2	height	weight	
87	1	1	1	1	1	1	1	1	1	1	1	1	1	0
105	1	1	1	1	1	1	1	1	1	1	1	0	0	2
15	1	1	1	1	1	1	1	1	1	1	0	1	1	1
16	1	1	1	1	1	1	1	1	1	1	0	0	0	3
9	1	1	1	1	1	1	1	1	1	0	1	1	1	1
6	1	1	1	1	1	1	1	1	1	0	1	0	0	3
3	1	1	1	1	1	1	1	1	1	0	0	1	1	2
2	1	1	1	1	1	1	1	1	1	0	0	0	0	4
2	1	1	1	1	1	1	1	1	0	1	0	0	0	4
1	1	1	1	1	1	1	1	0	0	1	1	0	0	4
3	1	1	1	1	1	1	1	0	0	1	0	1	1	3
5	1	1	1	1	1	1	1	0	0	1	0	0	0	5
1	1	1	1	1	1	1	1	0	0	0	0	1	1	4
2	1	1	1	1	1	1	1	0	0	0	0	0	0	6
1	1	1	1	1	1	1	0	1	1	0	1	1	1	2
	0	0	0	0	0	0	1	12	14	24	49	139	139	378

This table summarises that there are 87 observations that contain no missing values. There are 105 observations with missing values in both weight and height. There are 15 observations with HER2 missing values. There are 16 observations with missing values in height, weight and HER2. There are nine observations that have missing values for node. There are six observations with missing values in height, weight and node. There are three observations with missing values in node and HER2. There are two observations with missing values in node, HER2, height and weight and another two with missing values in PR, HER2, height and weight. There is only one observation with missing values for ER, PR, height and weight. There are three observations with missing values HER2, PR and ER. Five observations however contain missing values in ER, PR, HER2, height and weight, and only one observation has missing values in HER2, ER, PR and node. There are two observations that have a missing value for ER, PR, node, HER2, height and weight. And finally, there is one observation with missing values for r_stage and node. The pattern of the missing data can be seen visually in Figure 6.1.

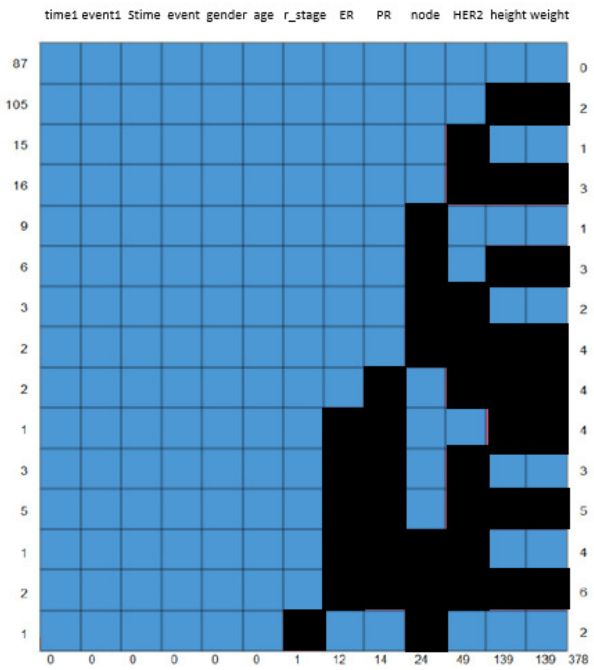


Figure 6.1 Missing data in the Isimo dataset

The proportion of missing data within each of the variables containing missing data can be seen in Figure 6.2.

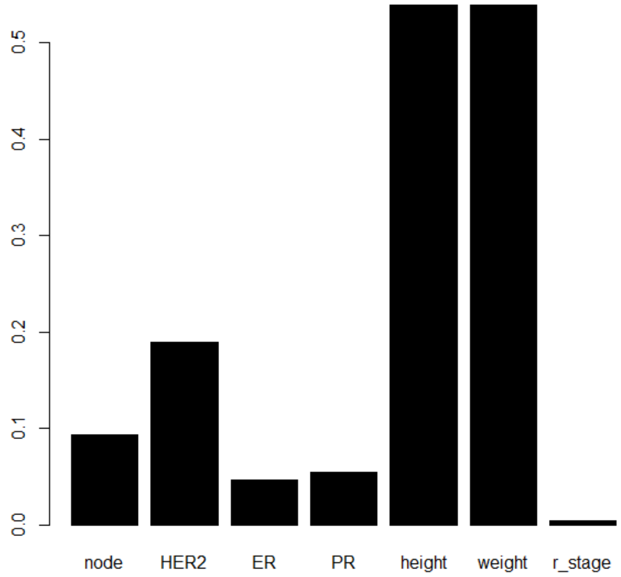


Figure 6.2 The proportion of missing values contained in the variables of the Isimo dataset

It can be seen from Figure 6.2, that the two variables height and weight have an extraordinary proportion of missing data. The author therefore decided to take out these two variables and rather work with the other variables that have less than 20% of missing data and can be successfully imputed.

It should be noted that the variables node and r_stage are ordinal factor variables, the variables HER2, ER and PR are binary variables and the two variables weight and height is numerical variables.

6.2.2 Missing data imputation

The *missForest* package in R is used to impute the missing data in the dataset. The dataset was imputed in only four iterations. From the output provided by the *missForest* function it is seen that the $NRMSE = 0.02094632$ and the $PFC = 0.18319743$. Therefore, the normalised root mean squared error is 2.09% and the proportion of falsely classified entries is 18.32%.

The final imputed dataset consists of 11 variables. Each of the 258 lines represent an individual with breast cancer. The variable time1 denotes the sojourn time spent in State 1 (curative) whereas the variable Stime is the total survival time of the individual. It should be noted that $time1 < Stime$ means that a transition occurred from State 1 (curative) to State 2 (non-curative).

6.3 Multi-state Markov Model Fitted to the Data

The multi-state Markov model will be fitted to the imputed Isimo Health dataset in this section. The three states of the multi-state Markov model include curative, non-curative and death. The death state is an absorbing state. The three-state Markov model can be seen in Figure 6.3.

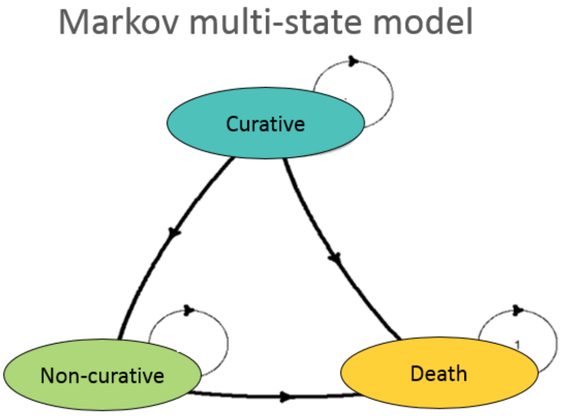


Figure 6.3 Three-state breast cancer Markov multi-state model

The *p3state.msm* package will be used to fit the multi-state model to the Isimo Health dataset.

6.3.1 The *p3state.msm* Package methodology

As mentioned in Chapter Two, the multi-state process is characterised through transition probabilities between two states i and j which can be expressed as

$$p_{ij}(s,t) = p(X(t) = j|X(s) = i, X_s), s \leq t,$$

where the history of the process is denoted by X_s (Meira-Machado and Roca-Pardiñas, 2011). The history, X_s, is generated and consists of the observation of the process over the interval $[0, s)$. According to Meira-Machado and Roca-Pardiñas (2011), it can also be generated through the transition intensities expressed as

$$q_{ij}(t) = \lim_{\Delta t \to 0} \frac{p_{ij}(t, t + \Delta t)}{\Delta t},$$

which represents the instantaneous hazard of progressing to state j conditional on occupying state i.

Let $\{X(t), t \geq 0\}$ denote the stochastic process where $X(t)$ denotes the state being occupied at time t for which all individuals are either in state 1 or 2 at time zero. A random vector (T_{12}, T_{13}, T_{23}) represent the stochastic behaviour of the process, where T_{ij} is the potential transition from state i to state j, $1 \leq i < j \leq 3$ in which T_{23} is the sojourn time spent in state 2. The expression

$$T = I(T_{12} \leq T_{13})(T_{12} + T_{23}) + I(T_{12} > T_{13})T_{13},$$

gives the survival time of the stochastic process.

The random vector expressed above may be subject to a random right-censoring variable denoted by C which is assumed to be independent of (T_{12}, T_{13}, T_{23}). Due to censoring, only the following are observed:

- sojourn time spent in state 1, $U = min(T_{12}.T_{13}, C)$
- sojourn time spent in state 2, $V = min(T_{23}, C - T_{12})$
- observed total time is expressed by $\tilde{T} = U + \delta V = min(T, C)(\delta = I(T_{12} \leq \min(T_{13}, C)))$
- indicator statuses $\Delta_1 = I(\min(T_{12}, T_{13}) \leq C)$ and $\Delta = I(T \leq C)$.

The transition probabilities are estimated by the joint distribution of (T_{12}, T_{13}, T_{23}). The estimation of $p_{11}(s,t)$ specifically require knowledge of the distribution of F of $min(T_{12}, T_{13})$. The estimators of the transition probabilities can be expressed as

$$\begin{aligned}\hat{p}_{11}(s,t) &= \frac{1 - \hat{X}(t)}{1 - \hat{X}(s)}, \\ \hat{p}_{12}(s,t) &= \frac{\sum_{i=1}^{n} W_i \phi_{s,t}(U_{[i]}, \tilde{T}_{(i)})}{1 - \hat{X}(s)}\end{aligned}$$

and

$$\hat{p}_{22}(s,t) = \frac{\sum_{i=1}^{n} W_i \tilde{\phi}_{s,t}(U_{[i]}, \tilde{T}_{(i)})}{\sum_{i=1}^{n} W_i \tilde{\phi}_{s,s}(U_{[i]}, \tilde{T}_{(i)})},$$

where W_i are the Kaplan-Meier weights attached to $\tilde{T}_{(i)}$, the Kaplan-Meier estimator based on the pairs (U_i, Δ_{1i}) is $\hat{X}$ and

$$\phi_{s,t}(u,v) = I(s < u \leq t, v > t)$$

and

$$\tilde{\phi}_{s,t}(u,v) = I(u \leq s, v > t).$$

In these expressions, the ordered sample of $\tilde{T}_i \prime s$ is denoted by $\tilde{T}_{(1)} \leq ... \leq \tilde{T}_{(n)}$ and $U_{[i]}$ denotes the pairs attached to the $Y_{(i)}$ values.

The transition probabilities that need to be estimated reduce to $p_{11}(s,t), p_{12}(s,t)$ and $p_{22}(s,t)$

since $p_{13}(s,t)$ and $p_{23}(s,t)$ can be estimated from the others by

$$p_{13}(s,t) = 1 - p_{11}(s,t) - p_{11}(s,t)$$

and

$$p_{23}(s,t) = 1 - p_{22}(s,t),$$

in the illness-death model.

In multi-state models, it is important to study the relationships between the different predictors and the outcome. According to Meira-Machado and Roca-Pardiñas (2011), several models have been used in literature, to relate the individual characteristics to the intensity rates. The transition intensities for the illness-death models, $q_{ij}(z(t))$, $1 \leq i < j \leq 3$, may be modelled by using a model of the form similar to Cox regression:

$$q_{ij}(z(t)) = q_{ij}^{(0)}(t)\exp(\beta_{ij}^T z(t)).$$

This model however assume that the process is Markovian and is known as Cox Markov models (CMM). As previously defined, the Markov assumption states that the future state depends only on the individual's current state. Several limitations are brought in when ignoring the disease history, therefore an alternative approach is to use Cox semi-Markov models (CSMM) in which the future of the process depends on the duration in the current state rather than the current time. Such models are also referred to as "clock reset" models since each time the individual enter a new state, the time is reset to zero. The only difference in estimating the transition intensities of the CMM and CSMM models, is that the q_{23} in CSMM is given by

$$q_{23}(z(t - T_{12})) = q_{23}^{(0)}(t - T_{12})\exp(\beta_{23}^T z(t)),$$

where the entry time into state 2 is denoted by T_{12} (Meira-Machado and Roca-Pardiñas, 2011).

6.3.2 Description of the *p3state.msm* package

The *p3state.msm* package compose of six functions that enables the user of the package to fit the proposed models and methods. The functions are summarised in Table 6.2.

Table 6.2 Summary of the functions in the *p3state.msm* package

Function	Description
p3state	The main function for fitting regression models and obtaining multi-state estimates including transition probabilities and bivariate distribution functions.
plot	This function provides the plots for transition probabilities.
summary	Summarise the objects of class p3state.
data.creation.reg	Provides the correct dataset for implementing regression models (TDCM, CMM and CSMM).
pLIDA	Provides estimates for the transition probabilities using methods in the paper by Meira-Machado, de Uña-Álvarez and Cadarso-Suárez (2006).
Biv	Provides estimates for the bivariate distribution function, using the paper by Uña-Álvarez and Meira-Machado (2008). This function is only available for progressive three-state models.

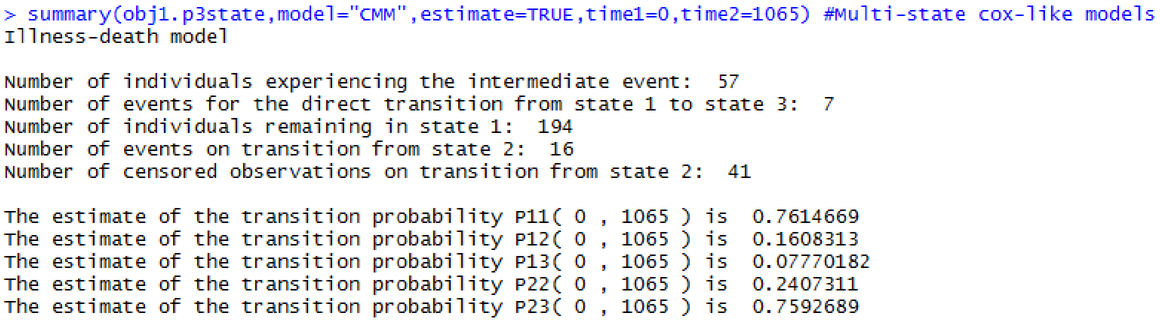
Source: Meira-Machado and Roca-Pardiñas (2011).

The data that are used for the functions in this package should contain variables *times1*, *delta*, *times2*, *time*, *status*, covariate 1, covariate 2,... with one line per individual. The *times1* variable represents the observed time in state 1, delta is an indicator variable indicating a transition to state 2, *times2* represent the observed time in state 2, time is the total observed time (times1+times2) and *status* is the final status of the individual with 1 indicating movement into state 3 and 0 otherwise (Meira-Machado and Roca-Pardiñas, 2011). The remaining variables are the covariates considered in the regression model.

6.3.3 Application to the Isimo Health dataset

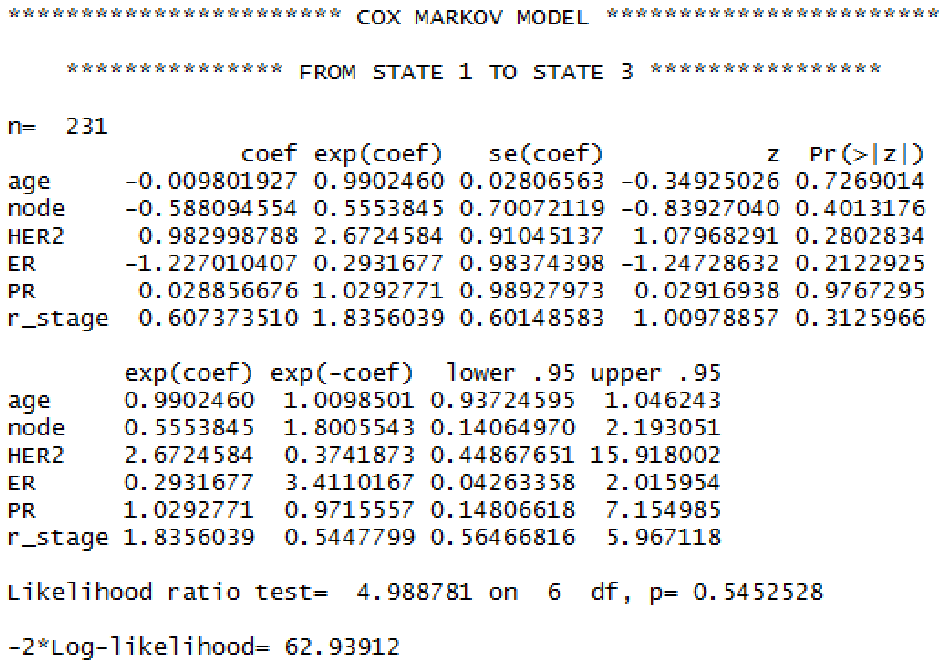
The *TPmsm* function in the *TPmsm* package was used to convert the dataset into the format required for the *p3state.msm* package. Since there is not enough males (gender=1) that have breast cancer in the Isimo Health dataset, this covariate has been taken out.

The multi-state Cox-like models (CMM and CSMM) is obtained by changing the *model* argument. The summary output for the CMM model is given below in Figures 6.4, 6.5, 6.6, 6.7 and 6.8.



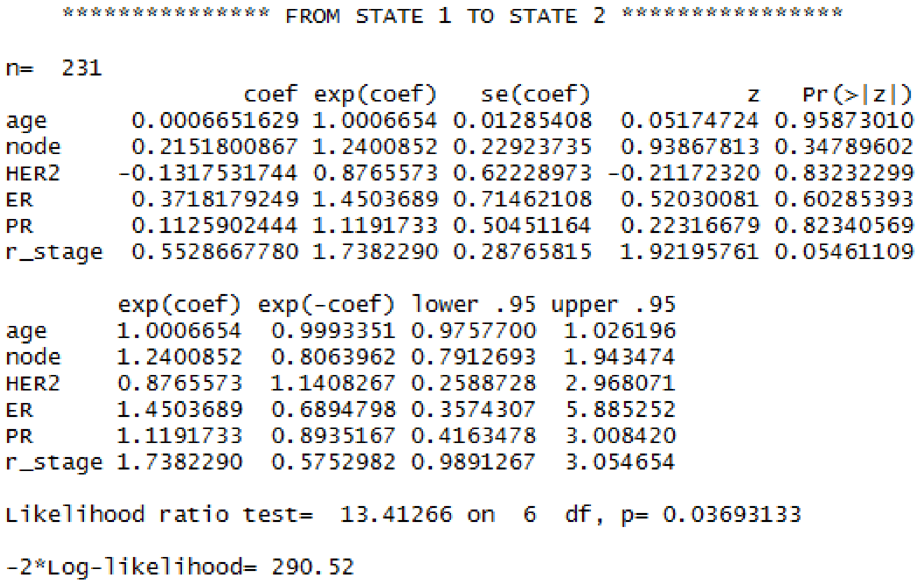
```
> summary(obj1.p3state,model="CMM",estimate=TRUE,time1=0,time2=1065) #Multi-state cox-like models
Illness-death model

Number of individuals experiencing the intermediate event:  57
Number of events for the direct transition from state 1 to state 3:  7
Number of individuals remaining in state 1:  194
Number of events on transition from state 2:  16
Number of censored observations on transition from state 2:  41

The estimate of the transition probability P11( 0 , 1065 ) is  0.7614669
The estimate of the transition probability P12( 0 , 1065 ) is  0.1608313
The estimate of the transition probability P13( 0 , 1065 ) is  0.07770182
The estimate of the transition probability P22( 0 , 1065 ) is  0.2407311
The estimate of the transition probability P23( 0 , 1065 ) is  0.7592689
```

Figure 6.4 CMM Output with transition probabilities



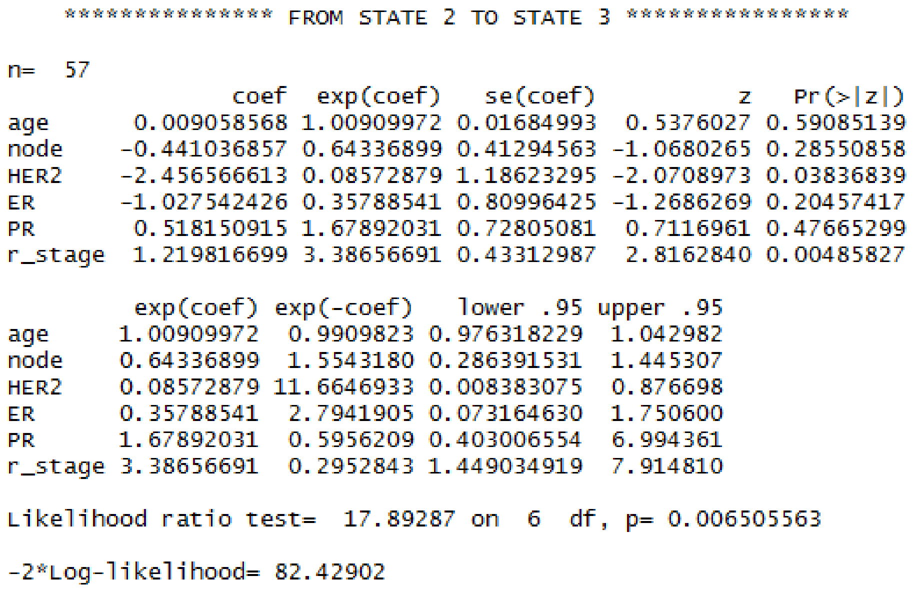
```
*********************** COX MARKOV MODEL ***********************

    *************** FROM STATE 1 TO STATE 3 ****************

n=  231
                coef exp(coef)   se(coef)            z  Pr(>|z|)
age     -0.009801927 0.9902460 0.02806563 -0.34925026 0.7269014
node    -0.588094554 0.5553845 0.70072119 -0.83927040 0.4013176
HER2     0.982998788 2.6724584 0.91045137  1.07968291 0.2802834
ER      -1.227010407 0.2931677 0.98374398 -1.24728632 0.2122925
PR       0.028856676 1.0292771 0.98927973  0.02916938 0.9767295
r_stage  0.607373510 1.8356039 0.60148583  1.00978857 0.3125966

        exp(coef) exp(-coef)  lower .95 upper .95
age     0.9902460  1.0098501 0.93724595  1.046243
node    0.5553845  1.8005543 0.14064970  2.193051
HER2    2.6724584  0.3741873 0.44867651 15.918002
ER      0.2931677  3.4110167 0.04263358  2.015954
PR      1.0292771  0.9715557 0.14806618  7.154985
r_stage 1.8356039  0.5447799 0.56466816  5.967118

Likelihood ratio test=  4.988781 on  6  df, p= 0.5452528

-2*Log-likelihood= 62.93912
```

Figure 6.5 CMM Output of Cox Markov Model from state 1 to state 3



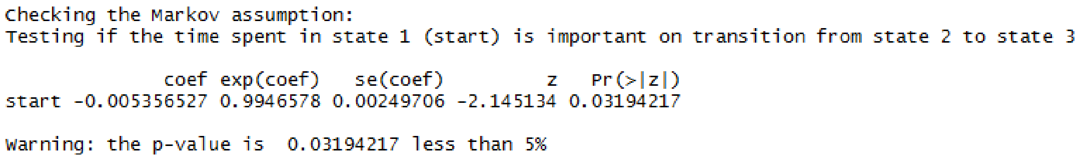
```
     **************** FROM STATE 1 TO STATE 2 *****************

 n=  231
                  coef exp(coef)   se(coef)           z   Pr(>|z|)
 age       0.0006651629 1.0006654 0.01285408  0.05174724 0.95873010
 node      0.2151800867 1.2400852 0.22923735  0.93867813 0.34789602
 HER2     -0.1317531744 0.8765573 0.62228973 -0.21172320 0.83232299
 ER        0.3718179249 1.4503689 0.71462108  0.52030081 0.60285393
 PR        0.1125902444 1.1191733 0.50451164  0.22316679 0.82340569
 r_stage   0.5528667780 1.7382290 0.28765815  1.92195761 0.05461109

         exp(coef) exp(-coef) lower .95 upper .95
 age      1.0006654  0.9993351 0.9757700  1.026196
 node     1.2400852  0.8063962 0.7912693  1.943474
 HER2     0.8765573  1.1408267 0.2588728  2.968071
 ER       1.4503689  0.6894798 0.3574307  5.885252
 PR       1.1191733  0.8935167 0.4163478  3.008420
 r_stage 1.7382290  0.5752982 0.9891267  3.054654

 Likelihood ratio test=  13.41266 on  6  df, p= 0.03693133

 -2*Log-likelihood= 290.52
```

Figure 6.6 CMM Output of Cox Markov Model from state 1 to state 2



```
     **************** FROM STATE 2 TO STATE 3 *****************

 n=  57
                 coef  exp(coef)   se(coef)          z   Pr(>|z|)
 age       0.009058568 1.00909972 0.01684993  0.5376027 0.59085139
 node     -0.441036857 0.64336899 0.41294563 -1.0680265 0.28550858
 HER2     -2.456566613 0.08572879 1.18623295 -2.0708973 0.03836839
 ER       -1.027542426 0.35788541 0.80996425 -1.2686269 0.20457417
 PR        0.518150915 1.67892031 0.72805081  0.7116961 0.47665299
 r_stage   1.219816699 3.38656691 0.43312987  2.8162840 0.00485827

          exp(coef) exp(-coef)   lower .95 upper .95
 age     1.00909972  0.9909823 0.976318229  1.042982
 node    0.64336899  1.5543180 0.286391531  1.445307
 HER2    0.08572879 11.6646933 0.008383075  0.876698
 ER      0.35788541  2.7941905 0.073164630  1.750600
 PR      1.67892031  0.5956209 0.403006554  6.994361
 r_stage 3.38656691  0.2952843 1.449034919  7.914810

 Likelihood ratio test=  17.89287 on  6  df, p= 0.006505563

 -2*Log-likelihood= 82.42902
```

Figure 6.7 CMM Output of Cox Markov Model from state 2 to state 3



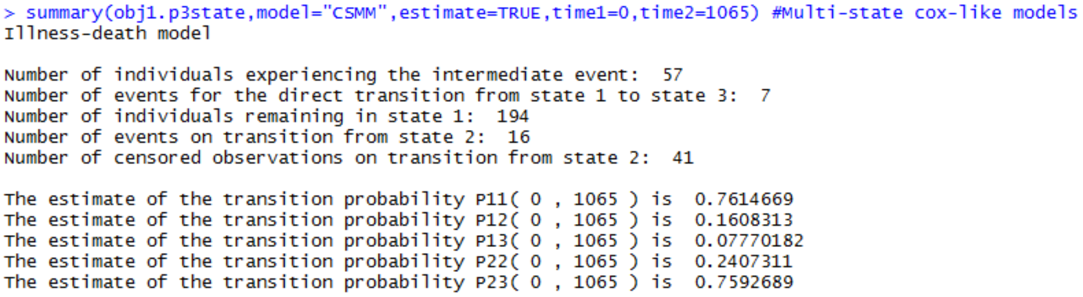
```
Checking the Markov assumption:
Testing if the time spent in state 1 (start) is important on transition from state 2 to state 3

               coef exp(coef)   se(coef)         z   Pr(>|z|)
start -0.005356527 0.9946578 0.00249706 -2.145134 0.03194217

Warning: the p-value is  0.03194217 less than 5%
```

Figure 6.8 CMM Output checking the Markov assumption

It can be seen from the last CMM output in Figure 6.8, that the Markov assumption is not valid.

This makes sense since the time spent in the curative (1) state does influence the transition from state 2 (non-curative) to state 3 (death).

The summary output for the Cox Semi-Markov regression is given below in Figures 6.9, 6.10, 6.11, 6.12 and 6.13.



```
> summary(obj1.p3state,model="CSMM",estimate=TRUE,time1=0,time2=1065) #Multi-state cox-like models
Illness-death model

Number of individuals experiencing the intermediate event:  57
Number of events for the direct transition from state 1 to state 3:  7
Number of individuals remaining in state 1:  194
Number of events on transition from state 2:  16
Number of censored observations on transition from state 2:  41

The estimate of the transition probability P11( 0 , 1065 ) is  0.7614669
The estimate of the transition probability P12( 0 , 1065 ) is  0.1608313
The estimate of the transition probability P13( 0 , 1065 ) is  0.07770182
The estimate of the transition probability P22( 0 , 1065 ) is  0.2407311
The estimate of the transition probability P23( 0 , 1065 ) is  0.7592689
```

Figure 6.9 CSMM Output of transition probabilities

The output provided in Figure 6.9 gives the transition probability matrix as

$$P = \begin{bmatrix} 0.7615 & 0.1608 & 0.0777 \\ 0 & 0.2407 & 0.7593 \\ 0 & 0 & 1 \end{bmatrix}.$$

This means that a patient that is in the curative state has a 76.15% probability of remaining in the curative state, a 16.08% probability of moving into the non-curative state (the cancer metastasising) and a 7.77% probability of dying. A patient that is currently in the non-curative state has a 24.07% chance of remaining in the non-curative state and a 75.93% chance of dying. The intensity matrix can also be calculated from this with $q_{23} = 0.0013337519$, $q_{12} = 0.000333958$ and $q_{13} = -0.000078082$.

Therefore, the intensity matrix is

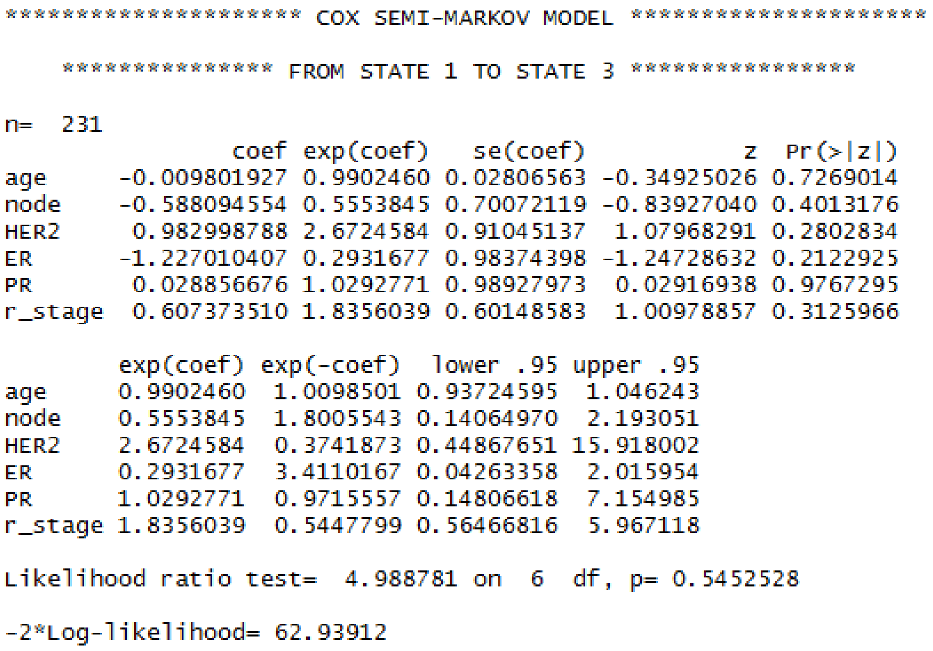
$$Q = \begin{bmatrix} -0.000255877 & 0.000333958 & -0.000078082 \\ 0 & -0.001337159 & 0.001337159 \\ 0 & 0 & 0 \end{bmatrix}.$$

From the intensity matrix, the sojourn time spent in each state can be calculated. The sojourn time the individual spends in the curative state before moving to the non-curative state or death state is

$$-\frac{1}{q_{11}} = -\frac{1}{-0.000255877} = 3908 days = 10.70 years.$$

The sojourn time spent by an individual in the non-curative state before moving to the death state is

$$-\frac{1}{q_{22}} = -\frac{1}{-0.001337159} = 748days = 2.05years.$$



```
********************* COX SEMI-MARKOV MODEL *********************

    *************** FROM STATE 1 TO STATE 3 ****************

n=  231
                 coef exp(coef)   se(coef)           z  Pr(>|z|)
age     -0.009801927 0.9902460 0.02806563 -0.34925026 0.7269014
node    -0.588094554 0.5553845 0.70072119 -0.83927040 0.4013176
HER2     0.982998788 2.6724584 0.91045137  1.07968291 0.2802834
ER      -1.227010407 0.2931677 0.98374398 -1.24728632 0.2122925
PR       0.028856676 1.0292771 0.98927973  0.02916938 0.9767295
r_stage  0.607373510 1.8356039 0.60148583  1.00978857 0.3125966

        exp(coef) exp(-coef)  lower .95 upper .95
age     0.9902460  1.0098501 0.93724595  1.046243
node    0.5553845  1.8005543 0.14064970  2.193051
HER2    2.6724584  0.3741873 0.44867651 15.918002
ER      0.2931677  3.4110167 0.04263358  2.015954
PR      1.0292771  0.9715557 0.14806618  7.154985
r_stage 1.8356039  0.5447799 0.56466816  5.967118

Likelihood ratio test=  4.988781 on  6  df, p= 0.5452528

-2*Log-likelihood= 62.93912
```

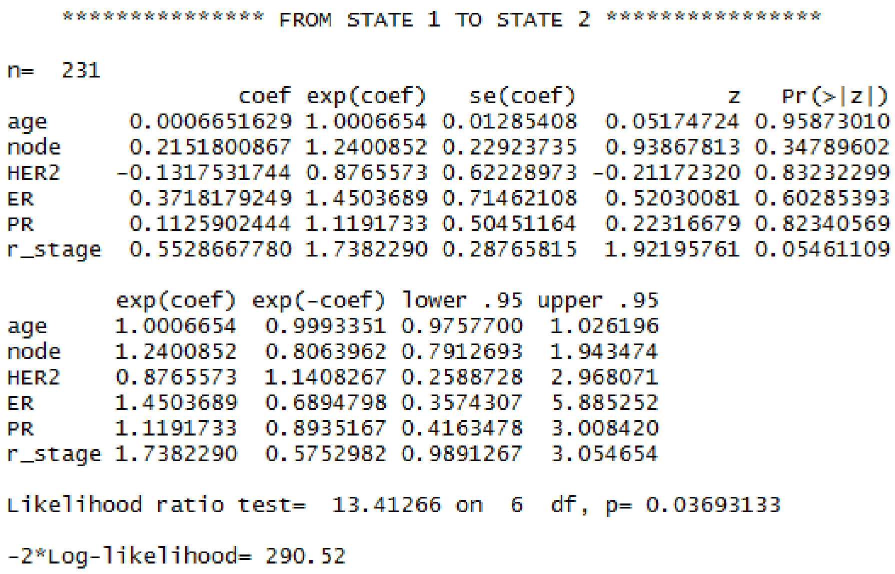
Figure 6.10 CSMM Output of Cox Semi-Markov Model from state 1 to state 3

In Figures 6.10, 6.11 and 6.12, the column marked "z" gives the Wald statistic value. This value corresponds to the ratio of each regression coefficient to its standard error. Therefore, $z = \frac{coef}{se(coef)}$. This statistic evaluates whether the coefficient (β) of a given variable is statistically significantly different from zero. It can be seen from Figure 6.10 that for state 1 to state 3, none of the variables are statistically significant.

The sign of the regression coefficients is also of importance. A positive sign implies that the hazard (risk of event) is higher and therefore the prognosis is worse, for individuals with higher variables for that specific variable. The hazard ratio (HR) in R is given as the second group relative to the first group therefore, for age, younger versus older. The beta coefficient for age $\beta_{age} = -0.0098$ (Figure 6.10) indicates that the younger individuals have higher risk of dying (state 3) while in the curative state (state 1), than the individuals that are older. The beta coefficient for node is $\beta_{node} = -0.5881$ which indicates that the individuals with a lower node status (more to zero) have a higher risk of dying while in the curative state than individuals that have a higher node status (more towards 3). $\beta_{HER2} = 0.9830$ is the beta coefficient for HER2. This indicates that the individuals that have a HER2 status = 1 and therefore being HER2 receptor positive have a higher risk of dying while in the curative state than individuals that have HER2 status = 0 which is negative. The beta coefficient for ER however is $\beta_{ER} = -1.2270$, which indicates that the individuals that are ER positive (ER status =1) have a lower risk of dying than those that are ER negative. The opposite is true for PR status. With a beta coefficient of $\beta_{PR} = 0.0289$, the risk of dying for an individual that is PR negative is lower than for an individual that is PR positive. The beta coefficient for r staging is $\beta_{r\_stage} = 0.6074$ which means that individuals with larger staging have an increased risk of dying when in the curative state.

The hazard ratio (HR) is the exponential of the coefficients. These coefficients provide the effect size of the covariates. Therefore, being one year older, reduces the hazard by a factor of 0.99 with $95\%CI = (0.94; 1.05)$, or 1%. Having a node size of 1 reduces the hazard by 0.56 (44%) with $95\%CI = (0.14; 2.19)$. Therefore, having a higher node size is a good prognostic factor when in the curative state at risk of dying. Being HER2 positive increases the hazard by 2.67 with $95\%CI = (0.45; 15.92)$, being ER positive reduces the hazard by 0.29 with $95\%CI = (0.04; 2.02)$, being PR positive increase the hazard by 1.03 with $95\%CI = (0.15; 7.15)$ and each r staging higher than 0 increase the hazard by 1.84 with $95\%CI = (0.56; 5.97)$.

Lastly, the Likelihood ratio test $= 4.9988781$ (Figure 6.10) with 6 degrees of freedom gives a p-value of $0.55 > 0.05$ and is therefore not significant. Therefore, the covariates does not have a statistically significant effect on the transition from the curative state to the death state.



```
       *************** FROM STATE 1 TO STATE 2 ****************

n=  231
                  coef exp(coef)   se(coef)           z   Pr(>|z|)
age       0.0006651629 1.0006654 0.01285408  0.05174724 0.95873010
node      0.2151800867 1.2400852 0.22923735  0.93867813 0.34789602
HER2     -0.1317531744 0.8765573 0.62228973 -0.21172320 0.83232299
ER        0.3718179249 1.4503689 0.71462108  0.52030081 0.60285393
PR        0.1125902444 1.1191733 0.50451164  0.22316679 0.82340569
r_stage   0.5528667780 1.7382290 0.28765815  1.92195761 0.05461109

        exp(coef) exp(-coef) lower .95 upper .95
age     1.0006654  0.9993351 0.9757700  1.026196
node    1.2400852  0.8063962 0.7912693  1.943474
HER2    0.8765573  1.1408267 0.2588728  2.968071
ER      1.4503689  0.6894798 0.3574307  5.885252
PR      1.1191733  0.8935167 0.4163478  3.008420
r_stage 1.7382290  0.5752982 0.9891267  3.054654

Likelihood ratio test=  13.41266 on  6  df, p= 0.03693133

-2*Log-likelihood= 290.52
```

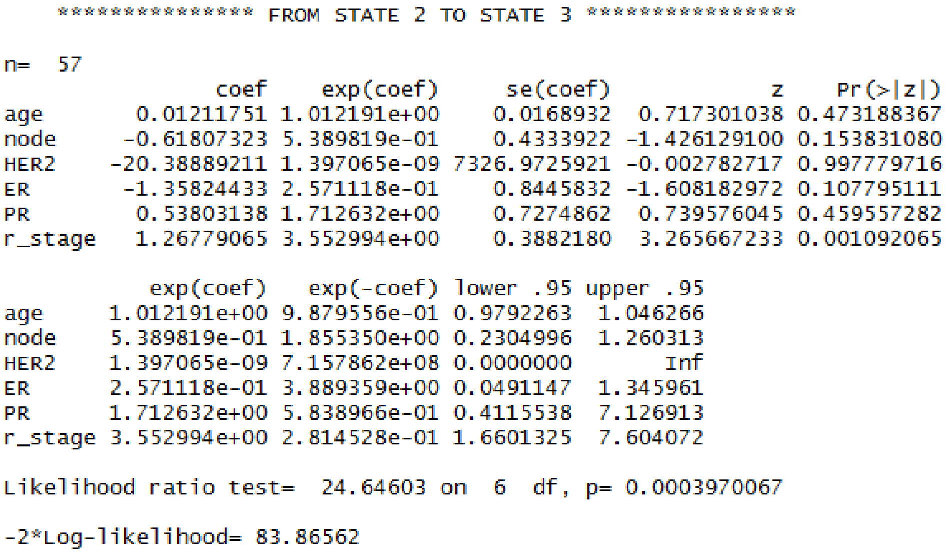
Figure 6.11 CSMM Output of Cox Semi-Markov Model from state 1 to state 2

Now, for Figure 6.11, it can be seen that for state 1 to state 2, none of the variables are statistically significant.

As for the beta coefficients for state 1 to state 2, the beta coefficient for age $\beta_{age} = 0.0007$ indicates that the younger individuals have lower risk of moving to the non-curative state (state 2) while in the curative state (state 1), than the individuals that are older. The beta coefficient for node is $\beta_{node} = 0.2152$ which indicates that the individuals with a lower node status (more to zero) have a lower risk of moving from the curative state to the non-curative state than individuals that have a higher node status (more towards 3). $\beta_{HER2} = -0.1318$ is the beta coefficent for HER2. This indicates that the individuals that have a HER2 status = 1 and therefore being HER2 receptor positive have a lower risk of moving into the non-curative state while in the curative state than individuals that have HER2 status = 0 which is negative. The beta coefficient for ER is $\beta_{ER} = 0.3718$, which indicates that the individuals that are ER positive (ER status =1) have a higher risk of becoming non-curative (state 2) than those that are ER negative. The same is true for PR status. With a beta coefficient of $\beta_{PR} = 0.1126$, the risk of progressing to state 2 for an individual that is PR negative is lower than for an individual that is PR positive. The beta coefficient for r staging is $\beta_{r\_stage} = 0.5529$ which means that individuals with higher staging have an increased risk of moving to the non-curative state when in the curative state.

The HR of age is 1.00, therefore, being one year older, increases the hazard by a factor of 1.00 with $95\%CI = (0.98; 1.03)$. Having a node size of 1 increases the hazard by 1.24 with $95\%CI = (0.79; 1.94)$. Therefore, having a higher node size is a bad prognostic factor when in the curative state to potentially move to the non-curative state. Being HER2 positive reduces the hazard by 0.88 with $95\%CI = (0.26; 2.97)$, being ER positive increases the hazard by 1.45 with $95\%CI = (0.36; 5.89)$, being PR positive increase the hazard by 1.12 with $95\%CI = (0.42; 3.01)$ and each r staging higher than 0 increase the hazard by 1.74 with $95\%CI = (0.99; 3.05)$.

Lastly, the Likelihood ratio test $= 13.41266$ (Figure 6.11) with 6 degrees of freedom gives a p-value of $0.04 < 0.05$ and is therefore it is statistically significant. This means that the covariates have a significant impact on the transition for the curative state to the non-curative state.



```
        **************** FROM STATE 2 TO STATE 3 ****************

n=  57
                 coef    exp(coef)     se(coef)            z    Pr(>|z|)
age        0.01211751 1.012191e+00    0.0168932  0.717301038 0.473188367
node      -0.61807323 5.389819e-01    0.4333922 -1.426129100 0.153831080
HER2     -20.38889211 1.397065e-09 7326.9725921 -0.002782717 0.997779716
ER        -1.35824433 2.571118e-01    0.8445832 -1.608182972 0.107795111
PR         0.53803138 1.712632e+00    0.7274862  0.739576045 0.459557282
r_stage    1.26779065 3.552994e+00    0.3882180  3.265667233 0.001092065

            exp(coef)   exp(-coef) lower .95 upper .95
age      1.012191e+00 9.879556e-01 0.9792263  1.046266
node     5.389819e-01 1.855350e+00 0.2304996  1.260313
HER2     1.397065e-09 7.157862e+08 0.0000000       Inf
ER       2.571118e-01 3.889359e+00 0.0491147  1.345961
PR       1.712632e+00 5.838966e-01 0.4115538  7.126913
r_stage 3.552994e+00 2.814528e-01 1.6601325  7.604072

Likelihood ratio test=  24.64603 on  6  df, p= 0.0003970067

-2*Log-likelihood= 83.86562
```

Figure 6.12 CSMM Output of Cox Semi-Markov Model from state 2 to state 3

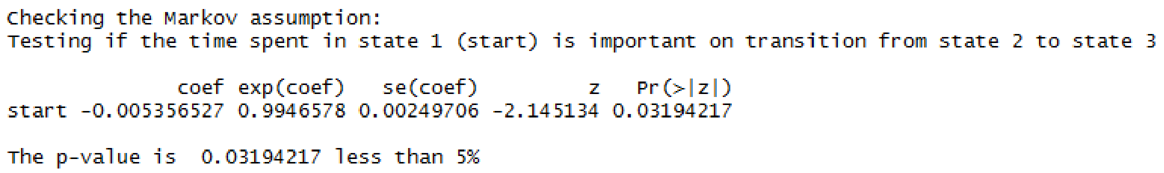
It should be noted in the results given in Figure 6.12, that the only difference between the CMM and the CSMM model since this is the only probability that is influence by the Markov assumption.

In Figure 6.12, it can be seen that for transitions between the non-curative state (state 2) and death (state 3), the variable r_stage is statistically significant.

As for the beta coefficients for state 2 to state 3, the beta coefficient for age $\beta_{age} = 0.0121$ indicates that the younger individuals have lower risk of dying (state 3) while in the non-curative state (state 2), than the individuals that are older. The beta coefficient for node is $\beta_{node} = -0.6181$ which indicates that the individuals with a lower node status (more to zero) have a higher risk of dying while in the non-curative state than individuals that have a higher node status (more towards 3). $\beta_{HER2} = -20.3889$ is the beta coefficient for HER2. This indicates that the individuals that have a HER2 status = 1 and therefore being HER2 receptor positive have a lower risk of dying while in the non-curative state than individuals that have HER2 status = 0 which is negative. The beta coefficient for ER however is $\beta_{ER} = -1.3582$ which indicates that the individuals that are ER positive (ER status =1) have a lower risk of dying than those that are ER negative. The opposite is true for PR status. With a beta coefficient of $\beta_{PR} = 0.5380$, the risk of dying for an individual that is PR negative is lower than for an individual that is PR positive. The beta coefficient for r staging is $\beta_{r\_stage} = 1.2678$ which means that individuals with higher staging have an increased risk of dying when in the non-curative state.

Adding one year of age, increase the hazard by a factor of 0.01 with $95\%CI = (0.98; 1.05)$. Having a node size of 1 reduces the hazard by 0.54 (46%) with $95\%CI = (0.23; 1.26)$. Therefore, having a higher node size is a good prognostic factor when in the non-curative state at risk of dying. Being HER2 positive reduces the hazard by 0.00 with $95\%CI = (0.00; \infty)$, being ER positive reduces the hazard by 0.26 with $95\%CI = (0.05; 1.35)$, being PR positive increase the hazard by 1.71 with $95\%CI = (0.41; 7.13)$ and each r staging higher than 0 increase the hazard by 3.55 with $95\%CI = (1.66; 7.60)$.

Lastly, the Likelihood ratio test $=$ 24.64603 (Figure 6.12) with 6 degrees of freedom gives a p-value of $0.0004 < 0.05$ and is therefore statistically significant. This implies that the covariates do have a significant effect on the transition between the non-curative state and the death state.



```
Checking the Markov assumption:
Testing if the time spent in state 1 (start) is important on transition from state 2 to state 3

              coef exp(coef)    se(coef)         z    Pr(>|z|)
start -0.005356527 0.9946578 0.00249706 -2.145134 0.03194217

The p-value is  0.03194217 less than 5%
```

Figure 6.13 CSMM Output of the testing of the Markov assumption

It can once again be seen from Figure 6.13 that the Markov assumption is violated, but this is not a problem since it is not a requirement for the Semi-Markov Cox regression.

6.4 Conclusion

In conclusion, the only covariate that has a significant effect on the transition probabilities is r_stage and it is only significant in the transition from the non-curative state to the death state. The mean sojourn time spent in the curative (non-metastatic) state is 10.70 years. It was also seen that the covariates only have a significant effect on the transitions from curative to non-curative and non-curative to death. The covariates do however not have a statistically significant effect on the transition from curative to death. The mean sojourn time spent in the non-curative (metastatic) state is 2.05 years.

CHAPTER 7
CONCLUSION

The aim of the study was to model the progression of breast cancer by using multi-state Markov models after determining an appropriate technique to impute missing data present in the covariates. A Literature Review was done on multi-state Markov models, the Markov process, the definition of missing data as well as the different types of missing data and the imputation techniques used. A thorough description of the claims and authorisation dataset for breast cancer obtained from Isimo Health followed as well as the process undertook in transforming and cleaning the dataset. The simulation approach used to simulate the data with the R package *TPmsm* was described and the available R packages for imputation techniques were described in detail whereafter two of the techniques were applied to impute the simulated dataset. The best performing imputation technique was thereafter used on the real-world dataset from Isimo Health. After imputation, a multi-state Markov model was fitted to the imputed data.

The Isimo Health dataset contained missing information within the covariates, which required imputation in order to fit the multi-state Markov model. Discarding observations with missing values would normally lead to valuable information being lost. After imputing missing data, standard complete-data methods could be used to produce statistical results.

Some of the imputation methods discussed in the thesis, include the Expectation-Maximisation (EM) algorithm, multiple imputation (MI) and Full Information Maximum Likelihood (FIML) methods. Complete datasets with imputed values are produced by the EM and MI methods with the advantage being that the generated datasets could be used for usual statistical analysis. The FIML method is a maximum likelihood approach for handling missing data. The use of Random Forests to impute missing data was also discussed.

Simulated datasets were used to test the different imputation techniques. The *gdpTP* function was used to simulate data from the Illness-death model. The simulation process were divided into two parts. The first part of the simulation were based on patients not entering state 2 whereafter the second part were simulated for patients entering state 2. Three covariates were simulated using distributions that resembles typical covariates. Six different possible scenarios were studied where different movement paths were permuted within the Illness-death model. These scenarios were ordered from best to worst case scenario in terms of prognosis of the disease. These six scenarios were then used to simulate the covariates.

The simulated dataset was modified to reflect different ratios of missingness and used to test two imputation techniques, one based on chained equations and the other based on Random Forests. The R packages chosen to perform these imputation techniques were *mice* and *missForest*. The *mice* package creates multiple imputations instead of a single imputation, whereas the *missForest* package treats the missing data problem as a prediction problem. In the *missForest* package, the data are imputed by regressing each of the variables against all of the other variables and then predicting the missing data for the dependent variable by using the fitted forest. A variety of performance measures were used in the assessment of the imputation technique. Based on these measures, the *missForest* imputation technique out performed the other imputation technique.

The *missForest* package was used as a result to impute the Isimo Health dataset, whereafter a multi-state Markov model was fit to the data.

As a consequence, the ultimate goal of this research is to build a forecasting tool predicting the progression of a breast cancer patient and to predict the costs associated with each of the states of the disease. The researcher would eventually be in a position predict the total cost of cancer for a medical scheme, using occupancy of the state space within a given model and the associated risk factors for each patient.

From the fitted multi-state Markov models, it was found that certain covariates had a significant effect on the transition probabilities and was only significant in the transition between certain states. Some of the covariates did however not have a statistically significant effect on certain transitions.

In conclusion, imputation using Random Forests was succesfully used in the presence of missing covariates before fitting a multi-state Markov model to the data.

Future study would include exploring Bayesian techniques for modelling the health states as well as exploring alternative ways of handling censoring in healthcare data. Lastly, it will be useful to model the costs within each of the healthcare states to result in the model predicting the total cost of cancer.

REFERENCES

Aalen, O. and Johansen, S. 1978. An Empirical Transition Matrix for Non-Homogeneous Markov and Chains Based on Censored Observations. *Scandinavian Journal of Statistics*, 5(3): 141-150.

Amorim, AP., de Uña-Álvarez, J. and Meira-Machado, L. 2011. Presmoothing the Transition Probabilities in the Illness-Death Model. *Statistics & Probability Letters*, 81(7): 797-806.

Analytics Vidhya. 2016. Tutorial on 5 Powerful R Packages used for imputing missing values. [Online]. Available: https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/ [2018, March 23].

Andridge, R.R. and Little, R.J.A. 2010. A review of Hot Deck Imputation for Survey Non-responses. *International Statistical Review*, 78(1): 40-64.

Araújo, A., Meira-Machado, L. and Roca-Pardiñas, J. 2014. TPmsm: Estimation of the Transition Probabilities in 3-State Models. *Journal of Statistical Software*, 62(4): 1-29.

Arbuckle, J.L. 1996. Full Information Estimation in the Presence of Incomplete Data. In G.A. Marcoulides and R.E. Schumacker (Eds.), *Advanced Structural Equation Modeling: Issues and Techniques*, 243-277. Mahwah, New Jersey. Lawrence Erlbaum Associates,.

Barnard, J. and Meng, X. 1999. Applications of multiple imputation in medical studies: from AIDS to NHANES. *Statistical Methods in Medical Research*, 8(1): 17-36.

Beran, R. 1981. Nonparametric *Regression with Randomly Censored Survival Data*. Technical report, University of California, Berkeley. [Online]. Available: https://www.researchgate.net/publication/316173118 [2018, September 1].

Breiman, L. 2001. Random Forests. *Machine Learning*, 45(1): 5–32.

REFERENCES

Chen, H.H., Duffy, S.W. and Tabar, L. 1996. A Markov chain method to estimate the tumour progression rate from preclinical to clinical phase, sensitivity and positive predictive value for mammography in breast cancer screening. *The Statistician*, 45(3): 307-317.

Cinlar, E. 1975. *Introduction to Stochastic Processes*. 1st ed. Englewood Cliffs, New York: Prentice Hall.

Cong, C. 2010. Statistical Analysis and Modeling of Breast Cancer and Lung Cancer. Unpublished doctor of philosophy dissertation. South Florida: University of South Florida.

Cox, D.R. and Miller, H.D. 1965. *The Theory of Stochastic Processes*. London: Chapman and Hall.

Datta, S. and Satten, GA. 2001. Validity of the Aalen-Johansen Estimators of Stage Occupation Probabilities and Nelson Aalen Integrated Transition Hazards for Non-Markov Models. *Statistics & Probability Letters*, 55(4): 403-411.

Dávila, S. and Rosado, H. 2017. Performance of missing value imputation schemes in health-related data. In Industrial and Systems Engineering Conference. Pittsburgh, Pennsylvania, USA, 20-23 May. Norcross, USA: Institute of Industrial & Systems Engineers. 2147-2152.

Dempster, A.P., Laird, N.M. and Rubin, D.B. 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, 39(B): 1-38.

Dinse, GE. and Larson, MG. 1986. A note on semi-Markov models for partially censored data. *Biometrika*, 73(2): 379-386.

Duffy, S.W. and Chen, H.H. 1995. Estimation of mean sojourn time in breast cancer screening using a Markov chain model of entry to and exit from preclinical detectable phase. *Statistics in Medicine*, 14(14): 1531-1543.

De Uña-Álvarez, J. and Meira-Machado, L.F. 2008. A Simple Estimation of Stage Occupation Probabilities and Nelson-Aalen Integrated Transition Hazards for Non-Markov Models. *Statistics and Probability Letters*, 55(4): 403-411.

REFERENCES

Farlex Partner Medical Dictionary. 2012. [Online].Available: https://medical-dictionary.thefreedictionary.com/ episode+of+care [2018, October 08].

Gelman, A. and Hill, J. 2006. *Missing-data imputation*. Cambridge: Cambridge University Press.529-544.

Grayson, M. 2012. Breast cancer. *Nature*, 485(7400): S49.

Grüger, J., Kay, R. and Schumacher, M. 1991. The validity of inferences based on incomplete observations in disease state models. *Biometrics*, 47: 595-605.

Hamner, B. and Frasco, M. 2018. Metrics: Evaluation Metrics for Machine Learning. R package version 0.1.4. [Online]. Available: https://CRAN.R-project.org/package=Metrics [2017, October 07]..

Hougaard, P. 1999. Multi-state Models: A Review. *Lifetime Data Analysis*, 5(3): 239-264.

Ibe, O.C. 2009. *Markov processes for stochastic modelling*. 2nd Ed. Lowell, USA: Elsevier Academic Press.

Jackson, C. 2011. Multi-state modelling with R: the msm package for R. *Journal of Statistical Software*, 38(8): 1-29.

Kalbfleisch, J.D. and Lawless, J.F. 1985. The analysis of panel data under a Markov assumption. *Journal of American Statistical Association*, 80(392): 863-871.

Kaplan, EL. and Meier, P. 1958. Nonparametric Estimation From Incomplete Observations. *Journal of the American Statistical Association*, 53(282): 457-481.

Kay, R. 1986. A Markov model for analysing cancer markers and disease states in survival studies. *Biometrics*, 42(4): 855-865.

Kirby, A.J. and Spiegelhalter, D.J. 1994. *Statistical modelling for the precursors of cervical cancer*. In Case Studies in Biometry. New York: Wiley.

Komen, S.G. 2017. [Online]. Available: http://ww5.komen.org/BreastCancer/WhatisBreastCancer.html [2017, March 13].

Li, J. 2010. Effects of Full Information Maximum Likelihood, Expectation Maximization, Multiple Imputation, and Similar Response Pattern Imputation on Structural Equation Modeling with Incomplete and Multivariate Nonnormal Data. Ohio: The Ohio State University.

Liaw, A. and Wiener, M. 2002. Classification and Regression by randomForest. *R News*, 2(3): 18-22.

Lin, DY., Sun, W. and Ying, Z. 1999. Nonparamteric Estimation of the Time Distributions for Serial Events with Censored Data. *Biometrika*, 86(1): 59-70.

Little, R.J.A. and Rubin, D.B. 1987. *Statistical Analysis with Missing Data*. New York: Wiley.

Mafu T.J. 2014. Modelling of Multi-State Panel Data: The Importance of the Model Assumptions. Doctors dissertation. Stellenbosch: University of Stellenbosch.

Marshall, G. and Jones, R.H. 1995. Multi-state Markov models and diabetic retinopathy. *Statistics in Medicine*, 14(18): 1975-83.

Meira-Machado, L., De Uña-Álvarez, J. and Cadarso-Suárez, C. 2006. Nonparametric Estimation of Transition Probabilities in a Non-Markov Illness-Death Model. Lifetime *Data Analysis*, 12(3): 325-344.

Meira-Machado, L., De Uña-Álvarez, J. and Somnath, D. 2012. *C*onditional Transition Probabilities in a Non-Markov Illness-Death Model. Discussion Papers in Statistics and Operation Research 12/05, Universidade de Vigo. [Online]. Available: http://depc05.webs.uvigo.es/reports/12_05.pdf [2018, September 2].

Meira-Machado, L., Roca-Pardiñas, J., Keilegom, IV. and Cadarso-Suárez, C. 2013. Bandwidth Selection for the Estimation of Transition Probabilities in the Location-Scale Progressive Three-State Model. *Computational Statistics*, 28(5): 2185-2210.

Meira-Machado, L. and Roca-Pardiñas, J. 2011. p3state.msm: Analyzing Survival Data from an Illness-Death Model. *Journal of Statistical Software*, 38(3): 1-18.

Moreira, AC., De Uña-Álvarez, J. and Meira-Machado, L. 2013. Presmoothing the Aalen-Johansen Estimator in the Illness-Death Model. *Electronic Journal of Statistics*, 7: 1491-1516.

Muthén, B.O., Kaplan, D. and Hollis, M. 1987. On Structural Equation Modeling with Data that are not Missing Completely at Random. *Psychometrika*, 52(3): 431-462.

National Cancer Institute. 2018. [Online]. Available: https://www.cancer.gov/publications/dictionaries/cancer-terms/def/cancer [2018, April 02].

R Core Team. 2013. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. [Online]. Available: http://www.R-project.org/.

Rice, J.A. 2007. *Mathematical Statistics and Data Analysis*, 3rd Ed, Brooks/Cole Berkeley: CENCAGE Learning.

Rubin, D.B. 1976. Inference and Missing Data. *Biometrika*, 63(3): 581-592.

Rubin, D.B. 1987. *Multiple Imputation for Nonresponse in Surveys*. New York: Wiley.

Schafer, J.L. 1997. *Analysis of Incomplete Multivariate Data.* New York: Chapman Hall.

Schmitt, P., Mandel, J. and Guedj, M. 2015. A Comparison of Six Methods for Missing Data Imputation. *Journal of Biometrics and Biostatistics* 6:224.

Spagnoli, A., Henderson, R., Boys, R.J. and Houwing-Duistermaat, J.J. 2011. A hidden Markov model for informative dropout in longitudinal response data with crisis states. *Statistics and Probability Letters*, 81(7): 730-738.

Stekhoven, D.J. 2011. Using the missForest Package. [Online]. Available: https://stat.ethz.ch/education/semesters/ss2012/ams/paper/missForest_1.2.pdf [2018, March 28].

Stekhoven, D.J. 2013. missForest: Nonparametric Missing Value Imputation using Random Forest. R package version 1.4.

Stekhoven, D.J. and Bühlman, P. 2012. MissForest - non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1), 112-118.

Susianto, Y., Notodiputro, K.A., Kurnia, A. and Wijayanto H. 2017. A Comparative Study of Imputation Methods for Estimation of Missing Values of Per Capita Expenditure in Central Java. IOP Conf. Series: *Earth and Environmental Science*, 58(2017): 012017.

Tang, F. 2017. Random Forest Missing Data Approaches. Miami: University of Miami.

Tang, F. and Ishwaran, H. 2017. Random forest missing data algorithms. Stat Anal Data Min: *The ASA Data Science Journal*, 10:3 63-377.

Tanner, M. 1993. Methods for the Exploration of Posterior Distributions and Likelihood Functions. *Tools for Statistical Inference* New York: Springer-Verlag.

The Cambridge English Dictionary. 2017. Cambridge University Press. [Online].

Available: http://dictionary.cambridge.org/dictionary/english/metastasize [2017, March 13].

The Oxford English Dictionary. 2017. Oxford University Press. [Online].

Available: https://en.oxforddictionaries.com/definition/breast_cancer [2017, March 13].

Van Buuren, S. and Groothuis-Oudshoorn, K. 2011. mice: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software*, 45(3): 1-67.

Van Keilegom, I., De Uña-Álvarez, J. and Meira-Machado, L. 2011. Nonparametric Location-Scale Models for Successive Survival Times Under Dependent Censoring. *Journal of Statistical Planning and Inference*, 141(3): 1118-1131.

Vargas-Chanes, D. 2000. Imputation Methods for Incomplete Panel Data with Applications to Latent Growth Curves. Ames, Iowa: Iowa State University.

Waljee, A.K., Mukherjee, A., Singal, A.G., Zhang, Y., Warren, J. Balis, U, Marrero, J. Zhu, J. and Higgins D.R. 2013. Comparison of imputation methods for missing laboratory data in medicine. *BMJ Open* 3(8): e002847.

REFERENCES

Zhang, D. and Zhang, X. 2009. Study on Forecasting the Stock Market Trend Based on Stochastic Analysis Method. International Journal of Business and Management, 4(6): 163-170.

APPENDICES

APPENDIX A

R code: Data Simulation

```
library(TPmsm) #Load required package

setThreadsTP(1)

seed=c(2718,3141,5436,6282,8154,9423)

setPackageSeedTP(seed)

temp=dgpTP(n=8000,corr=0,dist="exponential",dist.par=c(4,4),model.ce
ns="uniform", cens.par=3,state2.prob=0.75)

temp2=dgpTP(n=30000,corr=0,dist="exponential",dist.par=c(4,4),model.
cens="uniform", cens.par=3,state2.prob=0.75)

temp3=filter(temp2[[1]],event1==1) #state 2 to 3 data only

temp4=filter(temp3,time1!=Stime)

      col1=rep(0,2000)

      col2=rep(1,2000)

      col3=temp4[1:2000,3]-temp4[1:2000,1]

      col4=temp4[1:2000,4]

temp5=cbind(col1,col2,col3,col4)

colnames(temp5)=colnames(temp[[1]])

dataset=rbind(temp[[1]],temp5)

write.table(dataset, "location", sep="\t")

A=filter(dataset,event1==0 & event==0 & time1==Stime)

E=filter(dataset,event1==1 & event==1 & time1==Stime)

D=filter(dataset,event1==1 & event==1 & time1!=Stime & time1!=0)

F=filter(dataset,event1==1 & event==1 & time1!=Stime & time1==0)

B=filter(dataset,event1==1 & event==0 & time1!=Stime & time1!=0)

C=filter(dataset,event1==1 & event==0 & time1!=Stime & time1==0)

#Sample sizes

      nA=nrow(A)

      nB=nrow(B)

      nC=nrow(C)

      nD=nrow(D)

      nE=nrow(E)

      nF=nrow(F)

#Generate covariate1 from normal distribution
```

```
      Covariate1A=rnorm(nA,mean=55.85906,sd=14.3243)

      Covariate1B=rnorm(nB,mean=50.14286,sd=15.43651)

      Covariate1C=rnorm(nC,mean=54.88636,sd=15.08613)

      Covariate1D=rnorm(nD,mean=56.01613,sd=15.60842)

      Covariate1E=rnorm(nE,mean=56.30894,sd=14.54591)

      Covariate1F=rnorm(nF,mean=57.72917,sd=15.39514)

#Generate covariate2 from bernoulli distribution

           Covariate2A=rbinom(nA,1,prob=0.200935)

           Covariate2B=rbinom(nB,1,prob=0.25)

           Covariate2C=rbinom(nC,1,prob=0.193548)

           Covariate2D=rbinom(nD,1,prob=0.333333)

           Covariate2E=rbinom(nE,1,prob=0.216)

           Covariate2F=rbinom(nF,1,prob=0.37037)

#Generate covariate3 from multinomial distribution

      func<-function(x) which(x==1)

Covariate3A.1=apply(rmultinom(nA,1,c(0.486692,0.311787,0.144487,0.05
7034)),2,func)

Covariate3A=Covariate3A.1-1

Covariate3B.1=apply(rmultinom(nB,1,c(0.4,0.2,0.3,0.1)),2,func)

Covariate3B=Covariate3B.1-1

Covariate3C.1=apply(rmultinom(nC,1,c(0.322581,0.258065,0.354839,0.06
4516)),2,func)

Covariate3C=Covariate3C.1-1

Covariate3D.1=apply(rmultinom(nD,1,c(0.282609,0.347826,0.239130,0.13
0435)),2,func)

Covariate3D=Covariate3D.1-1

Covariate3E.1=apply(rmultinom(nE,1,c(0.451104,0.315457,0.160883,0.07
2555)),2,func)

Covariate3E=Covariate3E.1-1

Covariate3F.1=apply(rmultinom(nF,1,c(0.250000,0.388889,0.222222,0.13
8889)),2,func)

Covariate3F=Covariate3F.1-1

      newA=cbind(A,Covariate1A,Covariate2A,Covariate3A)

colnames(newA)=c("time1","event1","Stime","event","Covariate1","Cova
riate2", "Covariate3")

      newB=cbind(B,Covariate1B,Covariate2B,Covariate3B)
```

```
colnames(newB)=c("time1","event1","Stime","event","Covariate1","Cova
riate2", "Covariate3")

      newC=cbind(C,Covariate1C,Covariate2C,Covariate3C)

colnames(newC)=c("time1","event1","Stime","event","Covariate1","Cova
riate2", "Covariate3")

      newD=cbind(D,Covariate1D,Covariate2D,Covariate3D)

colnames(newD)=c("time1","event1","Stime","event","Covariate1","Cova
riate2", "Covariate3")

      newE=cbind(E,Covariate1E,Covariate2E,Covariate3E)

colnames(newE)=c("time1","event1","Stime","event","Covariate1","Cova
riate2", "Covariate3")

      newF=cbind(F,Covariate1F,Covariate2F,Covariate3F)

colnames(newF)=c("time1","event1","Stime","event","Covariate1","Cova
riate2", "Covariate3")

final=rbind(newA,newB,newC,newD,newE,newF)

final$Covariate2 = as.factor(final$Covariate2)

final$Covariate3 = as.factor(final$Covariate3)
```

APPENDIX B

R code: Creating Missingness in Simulated Dataset

```
#5% missingness:

    p=0.05

    missing.05=final

    nr=nrow(missing.05)

    nc=ncol(missing.05)

    ina=is.na(unlist(missing.05[,5:7])) #no NA's yet

    n2=floor(p*nr*3)-sum(ina)

    ina[sample(which(!is.na(ina)),n2)]=TRUE #replace some values


cbind.matrix=cbind(matrix(rep(FALSE,4*nr),nrow=nr),matrix(ina,nr=nr,
nc=3))

    missing.05[matrix(cbind.matrix,nr=nr,nc=nc)]=NA

#10% missingness:

    p=0.1

    missing.1=final
    nr=nrow(missing.1) #5
    nc=ncol(missing.1) #7
    ina=is.na(unlist(missing.1[,5:7])) #no NA's yet
    n2=floor(p*nr*3)-sum(ina) #3
    ina[sample(which(!is.na(ina)),n2)]=TRUE #replace some values

cbind.matrix=cbind(matrix(rep(FALSE,4*nr),nrow=nr),matrix(ina,nr=nr,
nc=3))
    missing.1[matrix(cbind.matrix,nr=nr,nc=nc)]=NA
#15% missingness:
    p=0.15
    missing.15=final
    nr=nrow(missing.15) #5
    nc=ncol(missing.15) #7
    ina=is.na(unlist(missing.15[,5:7])) #no NA's yet
    n2=floor(p*nr*3)-sum(ina) #3
    ina[sample(which(!is.na(ina)),n2)]=TRUE #replace some values

cbind.matrix=cbind(matrix(rep(FALSE,4*nr),nrow=nr),matrix(ina,nr=nr,
nc=3))
    missing.15[matrix(cbind.matrix,nr=nr,nc=nc)]=NA
```

APPENDIX C

R code: MICE Imputation

```
#missing.05
#MICE:
    install.packages("mice")
    library(mice)
    install.packages("VIM")
    library(VIM)
imputed_Data_mice = mice(missing.05[,5:7], m=1000, maxit = 5, method
= c('pmm','logreg','polr'), seed = 500)
    #Results tables:
impute.results.05.Covariate1=matrix(NA,nrow=7,ncol=2)
       colnames(impute.results.05.Covariate1)=c("MICE","missForest")
```

```
rownames(impute.results.05.Covariate1)=c("MSE","Accuracy","MAE","RAE
", "RMSE","SSE","Bias")
impute.results.05.Covariate2=matrix(NA,nrow=7,ncol=2)
       colnames(impute.results.05.Covariate2)=c("MICE","missForest")

rownames(impute.results.05.Covariate2)=c("MSE","Accuracy","MAE","RAE
", "RMSE","SSE","Bias")
    impute.results.05.Covariate3=matrix(NA,nrow=7,ncol=2)
       colnames(impute.results.05.Covariate3)=c("MICE","missForest")

rownames(impute.results.05.Covariate3)=c("MSE","Accuracy","MAE","RAE
", "RMSE","SSE","Bias")
    #MSE:
    MSE.mice=matrix(0,ncol=3,nrow=1000)
    for (i in 1:1000)
       {
       MSE.mice[i,1]=mse(final$Covariate1,cbind(missing.05[,1:4],
          complete(imputed_Data_mice,i))$Covariate1)
       MSE.mice[i,2]=mse(as.numeric(final$Covariate2),as.numeric(
          complete(imputed_Data_mice,i)$Covariate2))
       MSE.mice[i,3]=mse(as.numeric(final$Covariate3),as.numeric(
          complete(imputed_Data_mice,i)$Covariate3))
       }

MSE.mice.avg=c(mean(MSE.mice[,1]),mean(MSE.mice[,2]),mean(MSE.mice[,
3]))
         impute.results.05.Covariate1[1,1]=mean(MSE.mice[,1])
         impute.results.05.Covariate2[1,1]=mean(MSE.mice[,2])
         impute.results.05.Covariate3[1,1]=mean(MSE.mice[,3])
    #Accuracy:
    accuracy.mice=matrix(0,ncol=3,nrow=1000)
    for (i in 1:1000)
       {

accuracy.mice[i,1]=accuracy(final$Covariate1,cbind(missing.05[,1:4],
complete(imputed_Data_mice,i))$Covariate1)

accuracy.mice[i,2]=accuracy(as.numeric(final$Covariate2),as.numeric(
complete(imputed_Data_mice,i)$Covariate2))
accuracy.mice[i,3]=accuracy(as.numeric(final$Covariate3),as.numeric(
complete(imputed_Data_mice,i)$Covariate3))
       }

accuracy.mice.avg=c(mean(accuracy.mice[,1]),mean(accuracy.mice[,2]),
mean(accuracy.mice[,3]))
         impute.results.05.Covariate1[2,1]=mean(accuracy.mice[,1])
         impute.results.05.Covariate2[2,1]=mean(accuracy.mice[,2])
         impute.results.05.Covariate3[2,1]=mean(accuracy.mice[,3])
    #Mean Absolute Error:
    MAE.mice=matrix(0,ncol=3,nrow=1000)
    for (i in 1:1000)
       {

MAE.mice[i,1]=mae(final$Covariate1,cbind(missing.05[,1:4],complete(i
mputed_Data_mice,i))$Covariate1)
MAE.mice[i,2]=mae(as.numeric(final$Covariate2),as.numeric(complete(i
mputed_Data_mice,i)$Covariate2))
MAE.mice[i,3]=mae(as.numeric(final$Covariate3),as.numeric(complete(i
mputed_Data_mice,i)$Covariate3))
       }

MAE.mice.avg=c(mean(MAE.mice[,1]),mean(MAE.mice[,2]),mean(MAE.mice[,
3]))
         impute.results.05.Covariate1[3,1]=mean(MAE.mice[,1])
         impute.results.05.Covariate2[3,1]=mean(MAE.mice[,2])
         impute.results.05.Covariate3[3,1]=mean(MAE.mice[,3])
    #Relative Absolute Error:
    RAE.mice=matrix(0,ncol=3,nrow=1000)
    for (i in 1:1000)
       {

RAE.mice[i,1]=rae(final$Covariate1,cbind(missing.05[,1:4],complete(i
mputed_Data_mice,i))$Covariate1)
RAE.mice[i,2]=rae(as.numeric(final$Covariate2),as.numeric(complete(i
mputed_Data_mice,i)$Covariate2))
RAE.mice[i,3]=rae(as.numeric(final$Covariate3),as.numeric(complete(i
mputed_Data_mice,i)$Covariate3))
       }

RAE.mice.avg=c(mean(RAE.mice[,1]),mean(RAE.mice[,2]),mean(RAE.mice[,
3]))
```

```
          impute.results.05.Covariate1[4,1]=mean(RAE.mice[,1])
          impute.results.05.Covariate2[4,1]=mean(RAE.mice[,2])
          impute.results.05.Covariate3[4,1]=mean(RAE.mice[,3])
    #RMSE:
    RMSE.mice=matrix(0,ncol=3,nrow=1000)
    for (i in 1:1000)
       {
RMSE.mice[i,1]=rmse(final$Covariate1,cbind(missing.05[,1:4],complete
(imputed_Data_mice,i))$Covariate1)
RMSE.mice[i,2]=rmse(as.numeric(final$Covariate2),as.numeric(complete
(imputed_Data_mice,i)$Covariate2))
RMSE.mice[i,3]=rmse(as.numeric(final$Covariate3),as.numeric(complete
(imputed_Data_mice,i)$Covariate3))
       }
RMSE.mice.avg=c(mean(RMSE.mice[,1]),mean(RMSE.mice[,2]),mean(RMSE.mi
ce[,3]))
          impute.results.05.Covariate1[5,1]=mean(RMSE.mice[,1])
          impute.results.05.Covariate2[5,1]=mean(RMSE.mice[,2])
          impute.results.05.Covariate3[5,1]=mean(RMSE.mice[,3])
    #SSE:
    SSE.mice=matrix(0,ncol=3,nrow=1000)
    for (i in 1:1000)
       {
SSE.mice[i,1]=sse(final$Covariate1,cbind(missing.05[,1:4],complete(i
mputed_Data_mice,i))$Covariate1)
SSE.mice[i,2]=sse(as.numeric(final$Covariate2),as.numeric(complete(i
mputed_Data_mice,i)$Covariate2))
SSE.mice[i,3]=sse(as.numeric(final$Covariate3),as.numeric(complete(i
mputed_Data_mice,i)$Covariate3))
       }
SSE.mice.avg=c(mean(SSE.mice[,1]),mean(SSE.mice[,2]),mean(SSE.mice[,
3]))
          impute.results.05.Covariate1[6,1]=mean(SSE.mice[,1])
          impute.results.05.Covariate2[6,1]=mean(SSE.mice[,2])
          impute.results.05.Covariate3[6,1]=mean(SSE.mice[,3])
    #Bias
    Bias.mice=matrix(0,ncol=3,nrow=1000)
    for (i in 1:1000)
```

```
        {

Bias.mice[i,1]=bias(final$Covariate1,cbind(missing.05[,1:4],complete(imputed_Data_mice,i))$Covariate1)
Bias.mice[i,2]=bias(as.numeric(final$Covariate2),as.numeric(complete(imputed_Data_mice,i)$Covariate2))
Bias.mice[i,3]=bias(as.numeric(final$Covariate3),as.numeric(complete(imputed_Data_mice,i)$Covariate3))
        }

Bias.mice.avg=c(mean(Bias.mice[,1]),mean(Bias.mice[,2]),mean(Bias.mice[,3]))
          impute.results.05.Covariate1[7,1]=mean(Bias.mice[,1])
          impute.results.05.Covariate2[7,1]=mean(Bias.mice[,2])
          impute.results.05.Covariate3[7,1]=mean(Bias.mice[,3])
#missing.10
imputed_Data_mice = mice(missing.1[,5:7], m=1000, maxit = 5,method = c('pmm','logreg','polr'), seed = 500)
#Results tables:
impute.results.1.Covariate1=matrix(NA,nrow=7,ncol=2)
       colnames(impute.results.1.Covariate1)=c("MICE","missForest")

rownames(impute.results.1.Covariate1)=c("MSE","Accuracy","MAE","RAE","RMSE","SSE","Bias")
impute.results.1.Covariate2=matrix(NA,nrow=7,ncol=2)
       colnames(impute.results.1.Covariate2)=c("MICE","missForest")

rownames(impute.results.1.Covariate2)=c("MSE","Accuracy","MAE","RAE","RMSE","SSE","Bias")
impute.results.1.Covariate3=matrix(NA,nrow=7,ncol=2)
       colnames(impute.results.1.Covariate3)=c("MICE","missForest")

rownames(impute.results.1.Covariate3)=c("MSE","Accuracy","MAE","RAE","RMSE","SSE","Bias")
    #MSE:
    MSE.mice=matrix(0,ncol=3,nrow=1000)
    for (i in 1:1000)
        {

MSE.mice[i,1]=mse(final$Covariate1,cbind(missing.1[,1:4],complete(imputed_Data_mice,i))$Covariate1)
MSE.mice[i,2]=mse(as.numeric(final$Covariate2),as.numeric(complete(imputed_Data_mice,i)$Covariate2))
MSE.mice[i,3]=mse(as.numeric(final$Covariate3),as.numeric(complete(imputed_Data_mice,i)$Covariate3))
       }

MSE.mice.avg=c(mean(MSE.mice[,1]),mean(MSE.mice[,2]),mean(MSE.mice[,3]))
         impute.results.1.Covariate1[1,1]=mean(MSE.mice[,1])
         impute.results.1.Covariate2[1,1]=mean(MSE.mice[,2])
         impute.results.1.Covariate3[1,1]=mean(MSE.mice[,3])
    #Accuracy:
    accuracy.mice=matrix(0,ncol=3,nrow=1000)
    for (i in 1:1000)
       {

accuracy.mice[i,1]=accuracy(final$Covariate1,cbind(missing.1[,1:4],complete(imputed_Data_mice,i))$Covariate1)
accuracy.mice[i,2]=accuracy(as.numeric(final$Covariate2),as.numeric(complete(imputed_Data_mice,i)$Covariate2))
accuracy.mice[i,3]=accuracy(as.numeric(final$Covariate3),as.numeric(complete(imputed_Data_mice,i)$Covariate3))
       }

accuracy.mice.avg=c(mean(accuracy.mice[,1]),mean(accuracy.mice[,2]),mean(accuracy.mice[,3]))
         impute.results.1.Covariate1[2,1]=mean(accuracy.mice[,1])
         impute.results.1.Covariate2[2,1]=mean(accuracy.mice[,2])
         impute.results.1.Covariate3[2,1]=mean(accuracy.mice[,3])
    #Mean Absolute Error:
    MAE.mice=matrix(0,ncol=3,nrow=1000)
    for (i in 1:1000)
       {

MAE.mice[i,1]=mae(final$Covariate1,cbind(missing.1[,1:4],complete(imputed_Data_mice,i))$Covariate1)
MAE.mice[i,2]=mae(as.numeric(final$Covariate2),as.numeric(complete(imputed_Data_mice,i)$Covariate2))
MAE.mice[i,3]=mae(as.numeric(final$Covariate3),as.numeric(complete(imputed_Data_mice,i)$Covariate3))
       }

MAE.mice.avg=c(mean(MAE.mice[,1]),mean(MAE.mice[,2]),mean(MAE.mice[,3]))
         impute.results.1.Covariate1[3,1]=mean(MAE.mice[,1])
         impute.results.1.Covariate2[3,1]=mean(MAE.mice[,2])
         impute.results.1.Covariate3[3,1]=mean(MAE.mice[,3])
    #Relative Absolute Error:
    RAE.mice=matrix(0,ncol=3,nrow=1000)
    for (i in 1:1000)
       {
RAE.mice[i,1]=rae(final$Covariate1,cbind(missing.1[,1:4],complete(imputed_Data_mice,i))$Covariate1)
RAE.mice[i,2]=rae(as.numeric(final$Covariate2),as.numeric(complete(imputed_Data_mice,i)$Covariate2))
RAE.mice[i,3]=rae(as.numeric(final$Covariate3),as.numeric(complete(imputed_Data_mice,i)$Covariate3))
       }
RAE.mice.avg=c(mean(RAE.mice[,1]),mean(RAE.mice[,2]),mean(RAE.mice[,3]))
         impute.results.1.Covariate1[4,1]=mean(RAE.mice[,1])
         impute.results.1.Covariate2[4,1]=mean(RAE.mice[,2])
         impute.results.1.Covariate3[4,1]=mean(RAE.mice[,3])
    #RMSE:
    RMSE.mice=matrix(0,ncol=3,nrow=1000)
    for (i in 1:1000)
       {
RMSE.mice[i,1]=rmse(final$Covariate1,cbind(missing.1[,1:4],complete(imputed_Data_mice,i))$Covariate1)
RMSE.mice[i,2]=rmse(as.numeric(final$Covariate2),as.numeric(complete(imputed_Data_mice,i)$Covariate2))
RMSE.mice[i,3]=rmse(as.numeric(final$Covariate3),as.numeric(complete(imputed_Data_mice,i)$Covariate3))
       }
RMSE.mice.avg=c(mean(RMSE.mice[,1]),mean(RMSE.mice[,2]),mean(RMSE.mice[,3]))
         impute.results.1.Covariate1[5,1]=mean(RMSE.mice[,1])
         impute.results.1.Covariate2[5,1]=mean(RMSE.mice[,2])
         impute.results.1.Covariate3[5,1]=mean(RMSE.mice[,3])
    #SSE:
    SSE.mice=matrix(0,ncol=3,nrow=1000)
    for (i in 1:1000)
       {
```

```
SSE.mice[i,1]=sse(final$Covariate1,cbind(missing.1[,1:4],complete(imputed_Data_mice,i))$Covariate1)

SSE.mice[i,2]=sse(as.numeric(final$Covariate2),as.numeric(complete(imputed_Data_mice,i)$Covariate2))

SSE.mice[i,3]=sse(as.numeric(final$Covariate3),as.numeric(complete(imputed_Data_mice,i)$Covariate3))
       }

SSE.mice.avg=c(mean(SSE.mice[,1]),mean(SSE.mice[,2]),mean(SSE.mice[,3]))
         impute.results.1.Covariate1[6,1]=mean(SSE.mice[,1])
         impute.results.1.Covariate2[6,1]=mean(SSE.mice[,2])
         impute.results.1.Covariate3[6,1]=mean(SSE.mice[,3])
    #Bias
    Bias.mice=matrix(0,ncol=3,nrow=1000)
    for (i in 1:1000)
       {

Bias.mice[i,1]=bias(final$Covariate1,cbind(missing.1[,1:4],complete(imputed_Data_mice,i))$Covariate1)

Bias.mice[i,2]=bias(as.numeric(final$Covariate2),as.numeric(complete(imputed_Data_mice,i)$Covariate2))

Bias.mice[i,3]=bias(as.numeric(final$Covariate3),as.numeric(complete(imputed_Data_mice,i)$Covariate3))
       }

Bias.mice.avg=c(mean(Bias.mice[,1]),mean(Bias.mice[,2]),mean(Bias.mice[,3]))
         impute.results.1.Covariate1[7,1]=mean(Bias.mice[,1])
         impute.results.1.Covariate2[7,1]=mean(Bias.mice[,2])
         impute.results.1.Covariate3[7,1]=mean(Bias.mice[,3])
    #missing.15
imputed_Data_mice = mice(missing.15[,5:7], m=1000, maxit = 5, method =c('pmm','logreg','polr'), seed = 500)
    #Results tables:
impute.results.15.Covariate1=matrix(NA,nrow=7,ncol=2)
       colnames(impute.results.15.Covariate1)=c("MICE","missForest")

rownames(impute.results.15.Covariate1)=c("MSE","Accuracy","MAE","RAE", "RMSE","SSE","Bias")
impute.results.15.Covariate2=matrix(NA,nrow=7,ncol=2)
       colnames(impute.results.15.Covariate2)=c("MICE","missForest")
```

```
rownames(impute.results.15.Covariate2)=c("MSE","Accuracy","MAE","RAE
", "RMSE","SSE","Bias")
    impute.results.15.Covariate3=matrix(NA,nrow=7,ncol=2)
       colnames(impute.results.15.Covariate3)=c("MICE","missForest")

rownames(impute.results.15.Covariate3)=c("MSE","Accuracy","MAE","RAE
", "RMSE","SSE","Bias")
    #MSE:
    MSE.mice=matrix(0,ncol=3,nrow=1000)
       for (i in 1:1000)
       {

MSE.mice[i,1]=mse(final$Covariate1,cbind(missing.15[,1:4],complete(i
mputed_Data_mice,i))$Covariate1)
MSE.mice[i,2]=mse(as.numeric(final$Covariate2),as.numeric(complete(i
mputed_Data_mice,i)$Covariate2))
MSE.mice[i,3]=mse(as.numeric(final$Covariate3),as.numeric(complete(i
mputed_Data_mice,i)$Covariate3))
       }

MSE.mice.avg=c(mean(MSE.mice[,1]),mean(MSE.mice[,2]),mean(MSE.mice[,
3]))
          impute.results.15.Covariate1[1,1]=mean(MSE.mice[,1])
          impute.results.15.Covariate2[1,1]=mean(MSE.mice[,2])
          impute.results.15.Covariate3[1,1]=mean(MSE.mice[,3])
    #Accuracy:
    accuracy.mice=matrix(0,ncol=3,nrow=1000)
       for (i in 1:1000)
       {

accuracy.mice[i,1]=accuracy(final$Covariate1,cbind(missing.15[,1:4],
complete(imputed_Data_mice,i))$Covariate1)
accuracy.mice[i,2]=accuracy(as.numeric(final$Covariate2),as.numeric(
complete(imputed_Data_mice,i)$Covariate2))
accuracy.mice[i,3]=accuracy(as.numeric(final$Covariate3),as.numeric(

complete(imputed_Data_mice,i)$Covariate3))
       }

accuracy.mice.avg=c(mean(accuracy.mice[,1]),mean(accuracy.mice[,2]),
mean(accuracy.mice[,3]))
    impute.results.15.Covariate1[2,1]=mean(accuracy.mice[,1])
    impute.results.15.Covariate2[2,1]=mean(accuracy.mice[,2])
    impute.results.15.Covariate3[2,1]=mean(accuracy.mice[,3])
    #Mean Absolute Error:
    MAE.mice=matrix(0,ncol=3,nrow=1000)
       for (i in 1:000)
       {

MAE.mice[i,1]=mae(final$Covariate1,cbind(missing.15[,1:4],complete(imputed_Data_mice,i))$Covariate1)
MAE.mice[i,2]=mae(as.numeric(final$Covariate2),as.numeric(complete(imputed_Data_mice,i)$Covariate2))
MAE.mice[i,3]=mae(as.numeric(final$Covariate3),as.numeric(complete(imputed_Data_mice,i)$Covariate3))
       }

MAE.mice.avg=c(mean(MAE.mice[,1]),mean(MAE.mice[,2]),mean(MAE.mice[,3]))
         impute.results.15.Covariate1[3,1]=mean(MAE.mice[,1])
         impute.results.15.Covariate2[3,1]=mean(MAE.mice[,2])
         impute.results.15.Covariate3[3,1]=mean(MAE.mice[,3])
    #Relative Absolute Error:
    RAE.mice=matrix(0,ncol=3,nrow=1000)
       for (i in 1:1000)
       {

RAE.mice[i,1]=rae(final$Covariate1,cbind(missing.15[,1:4],complete(imputed_Data_mice,i))$Covariate1)
RAE.mice[i,2]=rae(as.numeric(final$Covariate2),as.numeric(complete(imputed_Data_mice,i)$Covariate2))
RAE.mice[i,3]=rae(as.numeric(final$Covariate3),as.numeric(complete(imputed_Data_mice,i)$Covariate3))
       }

RAE.mice.avg=c(mean(RAE.mice[,1]),mean(RAE.mice[,2]),mean(RAE.mice[,3]))
         impute.results.15.Covariate1[4,1]=mean(RAE.mice[,1])
         impute.results.15.Covariate2[4,1]=mean(RAE.mice[,2])
         impute.results.15.Covariate3[4,1]=mean(RAE.mice[,3])
    #RMSE:
    RMSE.mice=matrix(0,ncol=3,nrow=1000)
       for (i in 1:1000)
       {
```

```
RMSE.mice[i,1]=rmse(final$Covariate1,cbind(missing.15[,1:4],complete
(imputed_Data_mice,i))$Covariate1)

RMSE.mice[i,2]=rmse(as.numeric(final$Covariate2),as.numeric(complete
(imputed_Data_mice,i)$Covariate2))

RMSE.mice[i,3]=rmse(as.numeric(final$Covariate3),as.numeric(complete
(imputed_Data_mice,i)$Covariate3))

       }

RMSE.mice.avg=c(mean(RMSE.mice[,1]),mean(RMSE.mice[,2]),mean(RMSE.mi
ce[,3]))

         impute.results.15.Covariate1[5,1]=mean(RMSE.mice[,1])

         impute.results.15.Covariate2[5,1]=mean(RMSE.mice[,2])

         impute.results.15.Covariate3[5,1]=mean(RMSE.mice[,3])

    #SSE:

    SSE.mice=matrix(0,ncol=3,nrow=1000)

       for (i in 1:1000)

       {

SSE.mice[i,1]=sse(final$Covariate1,cbind(missing.15[,1:4],complete(i
mputed_Data_mice,i))$Covariate1)

SSE.mice[i,2]=sse(as.numeric(final$Covariate2),as.numeric(complete(i
mputed_Data_mice,i)$Covariate2))

SSE.mice[i,3]=sse(as.numeric(final$Covariate3),as.numeric(complete(i
mputed_Data_mice,i)$Covariate3))

       }

SSE.mice.avg=c(mean(SSE.mice[,1]),mean(SSE.mice[,2]),mean(SSE.mice[,
3]))

         impute.results.15.Covariate1[6,1]=mean(SSE.mice[,1])

         impute.results.15.Covariate2[6,1]=mean(SSE.mice[,2])

         impute.results.15.Covariate3[6,1]=mean(SSE.mice[,3])

    #Bias

    Bias.mice=matrix(0,ncol=3,nrow=1000)

       for (i in 1:1000)

       {

Bias.mice[i,1]=bias(final$Covariate1,cbind(missing.15[,1:4],complete
(imputed_Data_mice,i))$Covariate1)

Bias.mice[i,2]=bias(as.numeric(final$Covariate2),as.numeric(complete
(imputed_Data_mice,i)$Covariate2))

Bias.mice[i,3]=bias(as.numeric(final$Covariate3),as.numeric(complete
(imputed_Data_mice,i)$Covariate3))
```

```
       }

Bias.mice.avg=c(mean(Bias.mice[,1]),mean(Bias.mice[,2]),mean(Bias.mice[,3]))
         impute.results.15.Covariate1[7,1]=mean(Bias.mice[,1])
         impute.results.15.Covariate2[7,1]=mean(Bias.mice[,2])
         impute.results.15.Covariate3[7,1]=mean(Bias.mice[,3])
```

APPENDIX D

R code: missForest Package

```
#missForest
    installed.packages("missForest")
    library(missForest)
    MSE.missForest1=c(0,1000)
       MSE.missForest2=c(0,1000)
       MSE.missForest3=c(0,1000)
    Accuracy.missForest1=c(0,1000)
       Accuracy.missForest2=c(0,1000)
       Accuracy.missForest3=c(0,1000)
    MAE.missForest1=c(0,1000)
       MAE.missForest2=c(0,1000)
       MAE.missForest3=c(0,1000)
    RAE.missForest1=c(0,1000)
       RAE.missForest2=c(0,1000)
       RAE.missForest3=c(0,1000)
    RMSE.missForest1=c(0,1000)
       RMSE.missForest2=c(0,1000)
       RMSE.missForest3=c(0,1000)
    SSE.missForest1=c(0,1000)
       SSE.missForest2=c(0,1000)
       SSE.missForest3=c(0,1000)
    Bias.missForest1=c(0,1000)
       Bias.missForest2=c(0,1000)
       Bias.missForest3=c(0,1000)
#missing.05
for (i in 1:1000)
{
```

```
imputed_Data_missForest = missForest(missing.05)

#MSE:

MSE.missForest1[i]=mse(final$Covariate1,imputed_Data_missForest$ximp
$Covariate1)

MSE.missForest2[i]=mse(as.numeric(final$Covariate2),as.numeric(imput
ed_Data_missForest$ximp$Covariate2))

MSE.missForest3[i]=mse(as.numeric(final$Covariate3),as.numeric(imput
ed_Data_missForest$ximp$Covariate3))

#Accuracy:

Accuracy.missForest1[i]=accuracy(final$Covariate1,imputed_Data_missF
orest$ximp$Covariate1)

Accuracy.missForest2[i]=accuracy(as.numeric(final$Covariate2),as.num
eric(imputed_Data_missForest$ximp$Covariate2))

Accuracy.missForest3[i]=accuracy(as.numeric(final$Covariate3),as.num
eric(imputed_Data_missForest$ximp$Covariate3))

#Mean Absolute Error:

MAE.missForest1[i]=mae(final$Covariate1,cbind(missing.05[,1:4],
imputed_Data_missForest$ximp)$Covariate1)

MAE.missForest2[i]=mae(as.numeric(final$Covariate2),as.numeric(imput
ed_Data_missForest$ximp$Covariate2))

MAE.missForest3[i]=mae(as.numeric(final$Covariate3),as.numeric(imput
ed_Data_missForest$ximp$Covariate3))

#Relative Absolute Error:

RAE.missForest1[i]=rae(final$Covariate1,cbind(missing.05[,1:4],
imputed_Data_missForest$ximp)$Covariate1)

RAE.missForest2[i]=rae(as.numeric(final$Covariate2),as.numeric(imput
ed_Data_missForest$ximp$Covariate2))

RAE.missForest3[i]=rae(as.numeric(final$Covariate3),as.numeric(imput
ed_Data_missForest$ximp$Covariate3))

#RMSE:


RMSE.missForest1[i]=rmse(final$Covariate1,cbind(missing.05[,1:4],
imputed_Data_missForest$ximp)$Covariate1)

RMSE.missForest2[i]=rmse(as.numeric(final$Covariate2),as.numeric(imp
uted_Data_missForest$ximp$Covariate2))

RMSE.missForest3[i]=rmse(as.numeric(final$Covariate3),as.numeric(imp
uted_Data_missForest$ximp$Covariate3))

#SSE:
SSE.missForest1[i]=sse(final$Covariate1,cbind(missing.05[,1:4],
imputed_Data_missForest$ximp)$Covariate1)

SSE.missForest2[i]=sse(as.numeric(final$Covariate2),as.numeric(imput
ed_Data_missForest$ximp$Covariate2))

SSE.missForest3[i]=sse(as.numeric(final$Covariate3),as.numeric(imput
ed_Data_missForest$ximp$Covariate3))
```

```
#Bias
Bias.missForest1[i]=bias(final$Covariate1,cbind(missing.05[,1:4],
imputed_Data_missForest$ximp)$Covariate1)
Bias.missForest2[i]=bias(as.numeric(final$Covariate2),as.numeric(imp
uted_Data_missForest$ximp$Covariate2))
Bias.missForest3[i]=bias(as.numeric(final$Covariate3),as.numeric(imp
uted_Data_missForest$ximp$Covariate3))
}
             impute.results.05.Covariate1[1,2]=mean(MSE.missForest1)
             impute.results.05.Covariate2[1,2]=mean(MSE.missForest2)
             impute.results.05.Covariate3[1,2]=mean(MSE.missForest3)
impute.results.05.Covariate1[2,2]=mean(Accuracy.missForest1)
impute.results.05.Covariate2[2,2]=mean(Accuracy.missForest2)
impute.results.05.Covariate3[2,2]=mean(Accuracy.missForest3)
impute.results.05.Covariate1[3,2]=mean(MAE.missForest1)
impute.results.05.Covariate2[3,2]=mean(MAE.missForest2)
impute.results.05.Covariate3[3,2]=mean(MAE.missForest3)
impute.results.05.Covariate1[4,2]=mean(RAE.missForest1)
impute.results.05.Covariate2[4,2]=mean(RAE.missForest2)
impute.results.05.Covariate3[4,2]=mean(RAE.missForest3)
impute.results.05.Covariate1[5,2]=mean(RMSE.missForest1)
impute.results.05.Covariate2[5,2]=mean(RMSE.missForest2)
impute.results.05.Covariate3[5,2]=mean(RMSE.missForest3)
impute.results.05.Covariate1[6,2]=mean(SSE.missForest1)
impute.results.05.Covariate2[6,2]=mean(SSE.missForest2)
impute.results.05.Covariate3[6,2]=mean(SSE.missForest3)
impute.results.05.Covariate1[7,2]=mean(Bias.missForest1)
impute.results.05.Covariate2[7,2]=mean(Bias.missForest2)
impute.results.05.Covariate3[7,2]=mean(Bias.missForest3)
#missing.1:
     for (i in 1:1000)
     {
        imputed_Data_missForest = missForest(missing.1)
        #MSE:

MSE.missForest1[i]=mse(final$Covariate1,imputed_Data_missForest$ximp
$Covariate1)
MSE.missForest2[i]=mse(as.numeric(final$Covariate2),as.numeric(imput
ed_Data_missForest$ximp$Covariate2))
```

```
MSE.missForest3[i]=mse(as.numeric(final$Covariate3),as.numeric(imputed_Data_missForest$ximp$Covariate3))

       #Accuracy:

Accuracy.missForest1[i]=accuracy(final$Covariate1,imputed_Data_missForest$ximp$Covariate1)

Accuracy.missForest2[i]=accuracy(as.numeric(final$Covariate2),as.numeric(imputed_Data_missForest$ximp$Covariate2))

Accuracy.missForest3[i]=accuracy(as.numeric(final$Covariate3),as.numeric(imputed_Data_missForest$ximp$Covariate3))

       #Mean Absolute Error:

MAE.missForest1[i]=mae(final$Covariate1,cbind(missing.05[,1:4],imputed_Data_missForest$ximp)$Covariate1)

MAE.missForest2[i]=mae(as.numeric(final$Covariate2),as.numeric(imputed_Data_missForest$ximp$Covariate2))

MAE.missForest3[i]=mae(as.numeric(final$Covariate3,as.numeric(imputed_Data_missForest$ximp$Covariate3))

       #Relative Absolute Error:

RAE.missForest1[i]=rae(final$Covariate1,cbind(missing.05[,1:4],imputed_Data_missForest$ximp)$Covariate1)

RAE.missForest2[i]=rae(as.numeric(final$Covariate2),as.numeric(imputed_Data_missForest$ximp$Covariate2))

RAE.missForest3[i]=rae(as.numeric(final$Covariate3),as.numeric(imputed_Data_missForest$ximp$Covariate3))

       #RMSE:

RMSE.missForest1[i]=rmse(final$Covariate1,cbind(missing.05[,1:4],imputed_Data_missForest$ximp)$Covariate1)

RMSE.missForest2[i]=rmse(as.numeric(final$Covariate2),as.numeric(imputed_Data_missForest$ximp$Covariate2))

RMSE.missForest3[i]=rmse(as.numeric(final$Covariate3),as.numeric(imputed_Data_missForest$ximp$Covariate3))

       #SSE:

SSE.missForest1[i]=sse(final$Covariate1,cbind(missing.05[,1:4],imputed_Data_missForest$ximp)$Covariate1)

SSE.missForest2[i]=sse(as.numeric(final$Covariate2),as.numeric(imputed_Data_missForest$ximp$Covariate2))

SSE.missForest3[i]=sse(as.numeric(final$Covariate3),as.numeric(imputed_Data_missForest$ximp$Covariate3))

       #Bias

Bias.missForest1[i]=bias(final$Covariate1,cbind(missing.05[,1:4],imputed_Data_missForest$ximp)$Covariate1)

Bias.missForest2[i]=bias(as.numeric(final$Covariate2),as.numeric(imputed_Data_missForest$ximp$Covariate2))

Bias.missForest3[i]=bias(as.numeric(final$Covariate3),as.numeric(imputed_Data_missForest$ximp$Covariate3))
```

```
    }
    impute.results.1.Covariate1[1,2]=mean(MSE.missForest1)
       impute.results.1.Covariate2[1,2]=mean(MSE.missForest2)
       impute.results.1.Covariate3[1,2]=mean(MSE.missForest3)
    impute.results.1.Covariate1[2,2]=mean(Accuracy.missForest1)
       impute.results.1.Covariate2[2,2]=mean(Accuracy.missForest2)
       impute.results.1.Covariate3[2,2]=mean(Accuracy.missForest3)
    impute.results.1.Covariate1[3,2]=mean(MAE.missForest1)
       impute.results.1.Covariate2[3,2]=mean(MAE.missForest2)
       impute.results.1.Covariate3[3,2]=mean(MAE.missForest3)
    impute.results.1.Covariate1[4,2]=mean(RAE.missForest1)
       impute.results.1.Covariate2[4,2]=mean(RAE.missForest2)
       impute.results.1.Covariate3[4,2]=mean(RAE.missForest3)
    impute.results.1.Covariate1[5,2]=mean(RMSE.missForest1)
       impute.results.1.Covariate2[5,2]=mean(RMSE.missForest2)
       impute.results.1.Covariate3[5,2]=mean(RMSE.missForest3)
    impute.results.1.Covariate1[6,2]=mean(SSE.missForest1)
       impute.results.1.Covariate2[6,2]=mean(SSE.missForest2)
       impute.results.1.Covariate3[6,2]=mean(SSE.missForest3)
    impute.results.1.Covariate1[7,2]=mean(Bias.missForest1)
       impute.results.1.Covariate2[7,2]=mean(Bias.missForest2)
       impute.results.1.Covariate3[7,2]=mean(Bias.missForest3)
    #missing.15:
    for (i in 1:1000)
    {
          imputed_Data_missForest = missForest(missing.15)
       #MSE:
MSE.missForest1[i]=mse(final$Covariate1,imputed_Data_missForest$ximp$Covariate1)
MSE.missForest2[i]=mse(as.numeric(final$Covariate2),as.numeric(imputed_Data_missForest$ximp$Covariate2))
MSE.missForest3[i]=mse(as.numeric(final$Covariate3),as.numeric(imputed_Data_missForest$ximp$Covariate3))
       #Accuracy:
Accuracy.missForest1[i]=accuracy(final$Covariate1,imputed_Data_missForest$ximp$Covariate1)
Accuracy.missForest2[i]=accuracy(as.numeric(final$Covariate2),as.numeric(imputed_Data_missForest$ximp$Covariate2))
Accuracy.missForest3[i]=accuracy(as.numeric(final$Covariate3),as.numeric(imputed_Data_missForest$ximp$Covariate3))
```

```
       #Mean Absolute Error:
MAE.missForest1[i]=mae(final$Covariate1,cbind(missing.05[,1:4],imput
ed_Data_missForest$ximp)$Covariate1)
MAE.missForest2[i]=mae(as.numeric(final$Covariate2),as.numeric(imput
ed_Data_missForest$ximp$Covariate2))
MAE.missForest3[i]=mae(as.numeric(final$Covariate3),as.numeric(imput
ed_Data_missForest$ximp$Covariate3))
       #Relative Absolute Error:
RAE.missForest1[i]=rae(final$Covariate1,cbind(missing.05[,1:4],imput
ed_Data_missForest$ximp)$Covariate1)
RAE.missForest2[i]=rae(as.numeric(final$Covariate2),as.numeric(imput
ed_Data_missForest$ximp$Covariate2))
RAE.missForest3[i]=rae(as.numeric(final$Covariate3),as.numeric(imput
ed_Data_missForest$ximp$Covariate3))
       #RMSE:
RMSE.missForest1[i]=rmse(final$Covariate1,cbind(missing.05[,1:4],
imputed_Data_missForest$ximp)$Covariate1)
RMSE.missForest2[i]=rmse(as.numeric(final$Covariate2),as.numeric(imp
uted_Data_missForest$ximp$Covariate2))
RMSE.missForest3[i]=rmse(as.numeric(final$Covariate3),as.numeric(imp
uted_Data_missForest$ximp$Covariate3))
       #SSE:
SSE.missForest1[i]=sse(final$Covariate1,cbind(missing.05[,1:4],imput
ed_Data_missForest$ximp)$Covariate1)
SSE.missForest2[i]=sse(as.numeric(final$Covariate2),as.numeric(imput
ed_Data_missForest$ximp$Covariate2))
SSE.missForest3[i]=sse(as.numeric(final$Covariate3),as.numeric
(imputed_Data_missForest$ximp$Covariate3))
       #Bias
Bias.missForest1[i]=bias(final$Covariate1,cbind(missing.05[,1:4],imp
uted_Data_missForest$ximp)$Covariate1)
Bias.missForest2[i]=bias(as.numeric(final$Covariate2),as.numeric(imp
uted_Data_missForest$ximp$Covariate2)
Bias.missForest3[i]=bias(as.numeric(final$Covariate3),as.numeric(imp
uted_Data_missForest$ximp$Covariate3))
    }
    impute.results.15.Covariate1[1,2]=mean(MSE.missForest1)
       impute.results.15.Covariate2[1,2]=mean(MSE.missForest2)
       impute.results.15.Covariate3[1,2]=mean(MSE.missForest3)
    impute.results.15.Covariate1[2,2]=mean(Accuracy.missForest1)
       impute.results.15.Covariate2[2,2]=mean(Accuracy.missForest2)
       impute.results.15.Covariate3[2,2]=mean(Accuracy.missForest3)
    impute.results.15.Covariate1[3,2]=mean(MAE.missForest1)
```

```
       impute.results.15.Covariate2[3,2]=mean(MAE.missForest2)
       impute.results.15.Covariate3[3,2]=mean(MAE.missForest3)
    impute.results.15.Covariate1[4,2]=mean(RAE.missForest1)
       impute.results.15.Covariate2[4,2]=mean(RAE.missForest2)
       impute.results.15.Covariate3[4,2]=mean(RAE.missForest3)
    impute.results.15.Covariate1[5,2]=mean(RMSE.missForest1)
       impute.results.15.Covariate2[5,2]=mean(RMSE.missForest2)
       impute.results.15.Covariate3[5,2]=mean(RMSE.missForest3)
    impute.results.15.Covariate1[6,2]=mean(SSE.missForest1)
       impute.results.15.Covariate2[6,2]=mean(SSE.missForest2)
       impute.results.15.Covariate3[6,2]=mean(SSE.missForest3)
    impute.results.15.Covariate1[7,2]=mean(Bias.missForest1)
       impute.results.15.Covariate2[7,2]=mean(Bias.missForest2)
       impute.results.15.Covariate3[7,2]=mean(Bias.missForest3)
```

APPENDIX E

R code: multi-state model

```
#Load data into R:
RData=read.table(file="location",header=TRUE)
#RData=read.table(file="clipboard",header=TRUE)
       RData2=RData
       RData2$gender = as.factor(RData2$gender)
       RData2$r_stage = as.factor(RData2$r_stage)
       RData2$HER2 = as.factor(RData2$HER2)
       RData2$ER = as.factor(RData2$ER)
       RData2$PR = as.factor(RData2$PR)
       RData2$node = as.factor(RData2$node)
md.pattern(RData2)
dev.new()
k = dim(RData[,7:13])[2]
freq = numeric(k)
for(i in 1:k) freq = apply(RData[,7:13], 2,
function(x)mean(is.na(x)))
barplot(freq, col="black")
#imputation with missForest:
library(missForest)
imputed_Data2 = missForest(RData2[,-c(11,12)])
```

```
RData2.imputed=imputed_Data2$ximp

#Create a survTP object to use in TPmsm:

Rdata2.imputed.numeric=cbind(RData2.imputed[,1:4],as.numeric(RData2.imputed$gender),RData2.imputed$age,as.numeric(RData2.imputed$node),as.numeric (RData2.imputed$HER2),as.numeric(RData2.imputed$ER), as.numeric(RData2.imputed$PR),as.numeric(RData2.imputed$r_stage))

colnames(Rdata2.imputed.numeric)=colnames(RData2.imputed)

library(TPmsm)


breast_obj=with(Rdata2.imputed.numeric,survTP(time1,event1,Stime,event,gender,age,node,HER2,ER,PR,r_stage))

AJmodel=transAJ(breast_obj, s=0, t=1065, state.names=c("curative", "non-curative","death"), conf=TRUE,n.boot=1000, conf.level=0.95, method.boot="percentile")

PAJmodel=transPAJ(breast_obj, s=0, t=31065,state.names=c("curative", "non-curative","death"), conf=TRUE, n.boot=1000,conf.level=0.95, method.boot="percentile")

#change format of data to p3state.msm

library(p3state.msm)

breast_p3state=TPmsmOut(breast_TPmsm,package ="p3state.msm",names=c("time1","event1","Stime","event"))


colnames(breast_p3state)=c(colnames(breast_p3state)[1:5],"gender","age","node","HER2","ER","PR","r_stage")


obj1.p3state=p3state(breast_p3state,formula=~age+node+HER2+ER+PR+r_stage)

summary(obj1.p3state,model="CMM")

summary(obj1.p3state,model="CSMM")
```