Combining Multiple Imputation with cross-validation for calibration and assessment of Cox prognostic survival models

Ioannis Zavrakidis (s1552953)

Thesis advisor: Dr. Bart Mertens
Second advisor: Dr. Liesbeth de Wreede

MASTER THESIS
Defended on July 4, 2017
Specialization: Life and Behavioural Sciences

STATISTICAL SCIENCE
FOR THE LIFE AND BEHAVIOURAL SCIENCES
Acknowledgements

I would like to thank my supervisors Drs Liesbeth de Wreede and Bart Mertens for their endless support since the beginning of my thesis project. They were trying to help me overcome issues that were constantly arising concerning my thesis and more importantly making me think critically. I loved our cooperation and i am glad that it happened.

I wish to thank also my family and closest friends for their support and understanding during my studies, which gave me strength to continue and finally to reach the end of the road.

Finally, i want to express my gratitude to all members of the master truck Statistical Science for the Life and Behavioural sciences for creating such a comprehensive and well-structured Master’s program. I feel blessed to have been a student of such respected professors and have worked with so many great people. I will always have these years in my heart and my mind.
Contents

List of Tables 3

List of Figures 4

1 Introduction 5
  1.1 Aim of the Thesis ................................................. 5
  1.2 The Clinical study ............................................. 6
  1.3 Study population & Data description ......................... 7
  1.4 The Statistical Challenge ..................................... 7

2 Basic Statistical Methods 9
  2.1 Survival Analysis & Cox model ............................... 9
    2.1.1 Cox Regression ........................................... 12
    2.1.2 Proportional Hazards Assumption ....................... 12
    2.1.3 Measures for assessing the performance of a prediction model .............. 13
  2.2 Missing values ................................................... 16
    2.2.1 Multiple Imputation ...................................... 18
    2.2.2 Fully Conditional Specification ......................... 20
  2.3 Validation & Cross-validation ............................... 22

3 Statistical Methods for the analysis 23
  3.1 Preliminary Steps .............................................. 23
  3.2 Approach 1 .................................................... 24
  3.3 Approach 2 .................................................... 24
  3.4 Summary ....................................................... 27
  3.5 Important Notes ................................................ 27

4 Results 29
  4.1 Descriptive Statistics ....................................... 29
    4.1.1 Check of the Proportional Hazards Assumption ................. 34
  4.2 Missing data patterns ......................................... 36
    4.2.1 Lvdias-Distribution of imputed values ...................... 38
  4.3 Results from the 2 approaches ................................ 40
    4.3.1 Approach 2 vs Approach 1 plots .......................... 40
List of Tables

1.3.1 Description of the dataset .......................................................... 8
4.1.1 Descriptive statistics for the variables in the dataset ...................... 30
4.1.2 Scaled Schoenfeld Residuals for the original dataset ...................... 35
4.1.3 Scaled Schoenfeld Residuals for the censored dataset ...................... 36
4.2.1 Patterns of missing data ............................................................. 37
4.2.2 Statistics for the variables in the dataset, based on their missingness patterns ............................................................. 37
4.2.3 Distribution of imputed values-Lvdias .......................................... 38
4.3.1 C-index - Approach 1 ................................................................. 55
4.3.2 C-index - Approach 2 ................................................................. 55
4.3.3 Brier score - M=5 ................................................................. 56
4.3.4 Brier score - M=100 ................................................................. 56
4.3.5 Brier score - M=200 ................................................................. 56
4.4.1 Mean variances for the 2 approaches - Models without Lvdias .......... 58
4.4.2 Mean variances for the 2 approaches - Models with Lvdias ............... 58
4.4.3 Wilcoxon Signed-Ranks test for the Variances ........................... 59
4.5.1 Final Model for approach 2 ..................................................... 60
List of Figures

2.1.1 Illustration of right censorship ........................................... 11
2.2.1 Single Mean Imputation ...................................................... 17
2.2.2 Algorithm for (univariate) linear regression-based imputation .......... 19
2.2.3 MICE Algorithm for imputation ............................................ 20

3.2.1 Algorithm for approach 1 .................................................... 25
3.2.2 Approach 1 -schema ........................................................... 25
3.3.1 Algorithm for approach 2 .................................................... 26
3.3.2 Approach 2 -schema ........................................................... 27

4.1.1 Histograms for the continuous variables in the dataset................... 33
4.1.2 Kaplan Meier plot .............................................................. 34
4.2.1 Percentage of imputed values-Lvdias ....................................... 39
4.3.1 Approach 2 vs Approach 1(NoLvdias) - t=12 months ..................... 41
4.3.2 Approach 2 vs Approach 1(NoLvdias) - t=60 months ................... 42
4.3.3 Approach 2 vs Approach 1(NoLvdias) - t=84 months ................... 43
4.3.4 Approach 2 vs Approach 1 - t=12 months ................................ 44
4.3.5 Approach 2 vs Approach 1 - t=60 months ................................ 45
4.3.6 Approach 2 vs Approach 1 - t=84 months ................................ 46
4.3.7 Sum vs difference of survival probabilities - t=12 months ............... 48
4.3.8 Sum vs difference of survival probabilities - t=60 months ............... 49
4.3.9 Sum vs difference of survival probabilities - t=84 months ............... 50
4.3.10 Approach 2 vs Approach 1, split by status + MissingLvdias-t=12 months .... 52
4.3.11 Approach 2 vs Approach 1, split by status + MissingLvdias-t=60 months .... 53
4.3.12 Approach 2 vs Approach 1, split by status + MissingLvdias-t=84 months .... 54

A.0.1 Schoenfeld residuals plots ................................................. 70
A.0.2 Log-Minus-Log plots ......................................................... 74
Chapter 1

Introduction

1.1 Aim of the Thesis

When the ultimate goal of a study is to build a prediction model it is of great importance to validate this model for assessing its predictive performance, in order to see how it is expected to behave for future observations. In an ideal scenario, an independent dataset exists in order to validate the results on. However, this is rarely the case and the validation has to be done on the same dataset that the model derived from. Cross-Validation is one of the many techniques to do that properly and it is based on the splitting of the dataset in several folds, and each time a fold is being left out from the calibration of the model in order to act as the independent-test set. In Section 2.3 further details are introduced about cross-validation since we will use it in our analysis.

Moreover, a very common issue arising when it comes to analyzing data is that of missing values, and the approach the researcher will take to tackle it is vital. For that reason, several methods have been developed for dealing with this issue as better as possible. Currently, multiple imputation considered to be the most efficient and unbiased procedure and which we are going to use in this thesis. The idea behind it is to create several imputed datasets based on the original one by imputing different values for the missing observations each time. Section 2.2 elaborates more on the theoretical aspects of missing data as well as multiple imputation.

The statistical challenge of this project is to find the most appropriate way to combine multiple imputation with cross-validation. Although it might seems straightforward, there is a crucial detail that makes it non-trivial, and leaves ground for further investigation of this issue. In cross-validation, the goal is to test the model in a set of observations that was not used for its calibration. That is, the model has not seen the outcome of the observations in the validation set. On the other hand, multiple imputation makes use not only of the explanatory variables but also of the response variable. In other words, during the multiple imputation the procedure "sees" the outcomes of all observations in the dataset, and consequently, when it comes to the cross-validation part, the model indirectly "knows" the outcomes in the validation set. Therefore, by simply combining these 2 methods one after the other, an important assumption of validation is being violated, and that may lead to some optimism about model performance.

The aim of this thesis therefore is to compare 2 different approaches that attempt to overcome this issue, using a dataset from Leiden University Medical Center (LUMC), Leiden, the Netherlands. These methods are explained in detail in Chapter 3. Then, in Chapter 4 the results from the
analysis are presented and a discussion follows in Chapter 5. The rest of this chapter is about the description of the clinical study as well as of the data that is going to be used, and in Chapter 2 a brief introduction of the basic statistical concepts and methods that are necessary for this analysis is provided.

1.2 The Clinical study

Heart failure is one of the most common reasons for hospital admissions among those 65 years and older. But this condition is not limited to seniors. Heart failure is a condition that occurs when the heart cannot pump or fill with enough blood, which means that the heart must work harder to deliver blood to the body. Heart failure can be mild and cause minor symptoms, or it may be severe or even life-threatening. The most common symptoms of heart failure are shortness of breath, feeling tired, leg swelling, and other signs of fluid retention. It is estimated that there are around 26 million heart failure patients worldwide [ASA, American Heart Association], and accounts for 1-3 % of all hospital admissions in Europe and the United States. In the Netherlands, more than 120,000 people are diagnosed with heart failure [Leening and Siregar, 2014]. Factors that are associated with increased risk of this condition, are first of all the diseases that can damage the heart such as coronary heart disease, high blood pressure, diabetes etc. Moreover, unhealthy behaviors such as smoking tobacco, obesity, consuming food high in cholesterol and fats as well as zero or little physical activity, especially when combined with diseases like the aforementioned, can seriously elevate the probability of heart failure.

Currently there is no cure for heart failure. It is a chronic disease and it demands lifelong management. However, by treating it correctly and more importantly maintaining a healthy lifestyle, the symptoms and signs can be improved and possibly help making the heart stronger, letting eventually the patient to live longer. Nowadays, there are different types of treatments, but usually physicians prescribe a combination of medications. Besides that, sometimes it is necessary to undergo a surgery in order to fix the problem that was mainly responsible for the heart failure. Some examples are coronary bypass surgery & heart valve repair or replacement.

CRT (Cardiac Resynchronization Therapy) is another clinically proven treatment option for some individuals with heart failure, especially for those resistant to drugs. A CRT device, which is the size of a pocket watch is implanted with a local anaesthetic, just under the collarbone with flexible leads (coated wires) coming from it that are positioned in the heart. It sends small electrical impulses to both lower chambers of the heart to help them beat together in a more synchronized pattern. This may improve the heart’s ability to pump blood and oxygen to the body and hence the overall cardiac performance, reducing eventually cardiovascular and overall mortality. As with every treatment though, the benefits are not guaranteed and they depend on patient characteristics. Unfortunately, 40% of patients do not respond to CRT [ASA, American Heart Association] and therefore, several single-center studies have been carried out in the last decade to improve selection of CRT candidates. Multiple factors were found to influence the beneficial effect of CRT, both clinical and echocardiographic, as well as other parameters such as gender & age. Recommendation of CRT implantation is currently based on the severity of heart failure symptoms, left ventricular ejection fraction (LVEF) and QRS morphology and duration. Table 1.3.1 presents the basic variables of interest along with a short definition.

Despite the fact that several factors have been identified to have an effect on CRT, there is lack of individualized estimated prognosis. This procedure is costly and hence, it would be beneficial to be able to predict short-term and long-term survival probabilities for each patient. That is
the concern of a research study from the Department of Cardiology of Leiden University Medical Center, LUMC, to build a multi-parametric risk score for CRT, which will help in the shared decision-making between patient and physician. That analysis has already finished and the paper has been submitted [Höke et al., 2017].

1.3 Study population & Data description

For this thesis, the data that was used derived from Leiden University Medical Center (LUMC, Leiden, the Netherlands) and specifically from the Department of Cardiology. Patients included in the CRT registry between August 1999 and July 2013 were candidates for the analysis. First of all, patients with irremediable heart failure prior to CRT implantation or with recent myocardial infarction (< 3 months) were excluded. Further, only patients with QRS duration ≥ 120ms and New York Heart Association (NYHA) functional class II - ambulatory IV were considered as candidates. In total, \( N = 1053 \) patients were included in the final analysis. All of them underwent a thorough clinical evaluation and transthoracic 2-dimensional (2D) echocardiography prior to CRT implantation, and they regularly visited the outpatient clinic and/or the referral hospital for the long-term follow-up. Survival data were gathered from these records, where all-cause mortality was the end point.

In total, \( p = 14 \) variables were considered for the analysis. Table 1.3.1 presents these variables along with a small description, while in Table 4.1.1 in the results section, some basic descriptive statistics are shown. More information on how these measurements were obtained can be found in [Höke et al., 2017].

1.4 The Statistical Challenge

As it is apparent from Table 4.1.1, there is 1 variable, namely left ventricular diastolic function (Lvdias), which is observed only in 50% of the cases. There are also 4 other variables with missing values, but they count only for a small percentage of the dataset. In total, 50% of the patients have all their variables observed. That means we need to address this issue before moving on with the analysis. The simplest and probably most naive way to do so, is to remove from the analysis all the patients-observations that have at least one missing value. This will lead to simply removing half of the data and in throwing away so much information. Fortunately, methods have been developed to tackle this problem. The one that we are going to use here is the Multiple Imputation procedure, which is described in more details in Section 2.2. Finally, as stated in Section 1.1 it is crucial to consider how to combine it with cross-validation in order to minimize a potential optimism of the model.
### Variable Name

<table>
<thead>
<tr>
<th><strong>Variable Name</strong></th>
<th><strong>Abbreviation</strong></th>
<th><strong>Description</strong></th>
<th><strong>Possible Values</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>His</td>
<td>His</td>
<td>Atrioventricular junction ablation before CRT implantation, for patients with atrial fibrillation</td>
<td>True, False</td>
</tr>
<tr>
<td>Etiology of heart failure</td>
<td>Et</td>
<td>It is the reason of patient’s heart failure</td>
<td>Ischemic, Non-Ischemic</td>
</tr>
<tr>
<td>New York Heart Association</td>
<td>NYHA</td>
<td>Classification of patients’ heart failure according to the severity of their symptoms</td>
<td>I-IV (in this dataset no class I patients)</td>
</tr>
<tr>
<td>Diabetes mellitus</td>
<td>DM</td>
<td>If patient has diabetes</td>
<td>Diabetes, Non-Diabetes</td>
</tr>
<tr>
<td>Mitral regurgitation</td>
<td>Mr</td>
<td>Is leakage of blood backward through the Mitral valve each time the left ventricle contracts</td>
<td>0, 1</td>
</tr>
<tr>
<td>Left ventricular diastolic dysfunction</td>
<td>Lvdias</td>
<td>Diastolic dysfunction refers to when the diastole part of the pumping action of the heart is abnormal. The ventricles do not properly relax and become stiff meaning they cannot fill with blood properly.</td>
<td>Restrictive, Non-Restrictive</td>
</tr>
<tr>
<td>Left bundle branch block</td>
<td>Lbbb</td>
<td>Is a cardiac conduction abnormality seen on the electrocardiogram (ECG). In this condition, activation of the left ventricle of the heart is delayed, which causes the left ventricle to contract later than the right ventricle</td>
<td>0, 1</td>
</tr>
<tr>
<td>Atrial fibrillation</td>
<td>Af</td>
<td>Is an abnormal heart rhythm characterized by rapid and irregular beating. Often it starts as brief periods of abnormal beating which become longer and possibly constant over time</td>
<td>0, 1</td>
</tr>
<tr>
<td>Estimated glomerular filtration rate †</td>
<td>Egfr</td>
<td>Is the best test to measure the level of kidney function and determine the stage of kidney disease</td>
<td></td>
</tr>
<tr>
<td>Hemoglobin Levels †</td>
<td>Hb</td>
<td>Hemoglobin in the blood carries oxygen from the respiratory organs (lungs or gills) to the rest of the body (i.e. the tissues). There it releases the oxygen to permit aerobic respiration to provide energy to power the functions of the organism in the process called metabolism.</td>
<td></td>
</tr>
<tr>
<td>Left ventricular ejection fraction †</td>
<td>LVEF</td>
<td>An ejection fraction (EF) is the fraction of blood ejected from a ventricle of the heart with each heartbeat. Left ventricular ejection fraction is a measure of the efficiency of pumping into the systemic circulation.</td>
<td></td>
</tr>
<tr>
<td>QRS duration †</td>
<td>QRS</td>
<td>The QRS complex is a name for the combination of three of the graphical deflections seen on a typical electrocardiogram (EKG or ECG). It is usually the central and most visually obvious part of the tracing. It corresponds to the depolarization of the right and left ventricles of the human heart. In adults, it normally lasts 0.06 - 0.10s</td>
<td></td>
</tr>
</tbody>
</table>

†Continuous variable

Table 1.3.1: Description of the dataset

1.3.1
Chapter 2

Basic Statistical Methods

In this section some statistical techniques that are essential for this analysis are presented. Only the basic aspects are discussed here, and for more information readers are encouraged to look in some of the plenty books and articles that exist. Section 2.1 is about survival analysis in general as well as for Cox regression, Section 2.2 deals with missing data and the procedure of multiple imputation, and finally, Section 2.3 concerns cross-validation.

2.1 Survival analysis & Cox regression model

When the main interest of a study is the time to some event, there is a set of methods appropriate for analyzing such data, called survival analysis. Time here can be defined in several ways depending on the study. In general, we measure time from a well defined starting point until the event of interest occurs. Some simple examples are:

- Time since birth until death
- Time since onset of a disease until death
- Time until cardiovascular death after some treatment intervention
- Time since 1st day of unemployment until work resumption
- Time to birth of a child since marriage

Survival Analysis is predominantly used in medical studies where the death is in most cases the end point. That is the reason why the terms that are used intuitively have a negative connotation such as risk, hazard etc. However, the event of interest can be anything given that it is well defined. Other names also used for this topic are reliability theory, duration modeling, time-to-event analysis etc, depending on the field that it is being applied, i.e. engineering, economics, sociology.

Apart from the type of the response that is different in survival analysis, there is also one other issue that gave rise to this topic. Censoring is very common when we analyze this kind of data. Some subjects might never experience the event of interest during their presence in a study. In general, the only information we have about these subjects regarding their event-times is about the
time interval in which it did not occur, and we have no clue about their exact event-time and even if it occurs later. Mainly, the reasons why this may happen are:

- The subject never experiences the event of interest
- The subject is lost to follow-up during the study for several reasons
- The subject drops out of the study.

All the above are instances of right-censoring. There is also left-censoring where the limitations concerning the information about the time of the event are relative to the beginning of the study. That is, the subject has already experienced the event when the study started, but the exact time is unknown. For instance:

**Example left-censoring**

Some children have already learned to count before their age of 4, when a relevant study begins

Finally, interval censoring is another form of censoring, where the only information is about the time-interval where the event happened.

**Example interval censoring**

In a clinical study, patients are observed every 3 months. We know only that the disease of interest started between the 2 consecutive visits.

Regardless of the censoring scheme, the crucial issue is that it must be non-informative. Non-informative censoring is met when the distribution of survival times (T) provides no information about the distribution of censorship times (C), and vice versa [Kleinbaum and Klein, 2012].

Let $T$ denote a non-negative variable, that represents the survival time for a subject in the study. In general, 2 function are the main focus regarding $T$, the survival & hazard function, which are the key concepts in analyzing the distribution of event times. The survival function,

$$(2.1.0.1) \quad S(t) = P(T > t)$$

denotes the probability of remaining event-free until some time-point $t$, while the hazard function (or rate) denotes the instantaneous risk of experiencing the event, given that it has not happened by that time. It is defined as

$$(2.1.0.2) \quad h(t) = \lim_{\Delta t \to 0} \frac{P(t \leq T < t + \Delta t | T \geq t)}{\Delta t}$$

In case of continuous random variable $T$, we have

$$(2.1.0.3) \quad h(t) = \frac{f(t)}{S(t)} = -\frac{d \ln [S(t)]}{dt},$$

since

$$(2.1.0.4) \quad S(t) = \int_t^\infty f(x)dx,$$
An illustration of Uncensored observations

An illustration of right censored observations

Figure 2.1.1: Illustration of right censorship
where $f(x)$ is the probability density function.

By integrating the hazard function over the entire time period, we get the **Cumulative Hazard Function**, defined as

\begin{equation}
H(t) = \int_0^t h(t) \, dt = -\ln S(t)
\end{equation}

By knowing these functions, one can compute some other quantities of interest such as the **median survival**. Usually, in a study we are interested in relating the time to event of interest with a set of covariates. For instance, we want to compare 2 treatments based on their survival outcome, adjusting for some other parameters like age, gender etc. There are several models available for analyzing this kind of data. They can be **parametric**, where the distribution of $T$ is considered known or **non-parametric** like Kaplan-Meier, where we do not want to make an assumption about the distribution for $T$.

### 2.1.1 Cox Regression

A special and widely used model in the context of survival analysis is the **Cox proportional hazards regression model**. It is considered as a **semi-parametric** model, because it has less assumptions than a parametric but more than a non-parametric one. It allows testing for differences in survival times of two or more groups of interest, while allowing to adjust for other covariates. Cox regression is written in terms of the hazard model, connecting the predictors with the hazard function (function 2.1.1.1). The Cox proportional hazards regression model can be written as follows:

\begin{equation}
\begin{aligned}
\frac{h_a(t)}{h_b(t)} &= \frac{h_0(t) \exp(\beta_1 X_a)}{h_0(t) \exp(\beta_1 X_b)} = \frac{\exp(\beta_1 X_a)}{\exp(\beta_1 X_b)} = \exp(\beta_1(X_a - X_b)) = \exp(\beta_1(1 - 0)) = \exp(\hat{\beta}_1)
\end{aligned}
\end{equation}

which is independent of time, and independent of the baseline hazard.

### 2.1.2 Proportional Hazards Assumption

When the model to be used is the **Cox proportional hazards model**, as its name also indicates, the assumption of proportional hazards (PH) must be checked. This can be done in many ways and
a broad distinction is between graphical and numerical diagnostics. Starting with the former, a very useful tool for assessing the proportionality assumption for a categorical variable is the visual check of the survival curves for each level of it. A transformation of the survival plots however, which is based on the relationship between the survival and the hazard function, provides a better intuition and interpretation. If the PH assumption holds, \( S_1(t) = S_0(t)^{exp(\beta)} \), where \( S_0(t) \) is the survival function for the reference category and \( S_1(t) \) for the compared level respectively. Then the log-minus-log transformation of this function gives \( \log(-\log(S_1(t))) = \beta + \log(-\log(S_0(t))) \). Therefore, these two transformed survival functions will have a constant distance \( \beta \). Hence, when they are plotted against time, they must be parallel over the entire range of time under proportionality, while patterns of convergence, divergence or more importantly of crossing, indicate violation of the PH assumption. However, it is not hard to see that this graphical test of the PH assumption is useful only for categorical variables, and with not many levels. For numeric ones, it becomes impossible to plot the survival curves for each unique value, and the alternative of splitting it based on some criteria and then use it as a categorical is not advised.

The most common numerical diagnostic is the use of the Schoenfeld residuals. They are calculated for each variable and for each subject separately, and only for those who are not censored [Schoenfeld, 1982]. They are defined as: \( r_i = x_{ik} - \sum_{j \in R(t_i)} x_{jk} p_j \), for each observed event (\( \delta_i = 1 \)) and simply represent the difference between the observed covariate \( k \) for the individual that failed and the expected given the risk set at time \( t_i \) (a weighted-average of the covariate, weighted by each individual’s likelihood of failing at time \( t_i \) based on the fitted model \( p \)). Under PH assumption they are centered around zero and do not exhibit any non-random pattern, because we expect the difference between the variable values at failure times and a weighted average of them to not show any trends. An alternative proposed by [Grambsch & Therneau (1994)] is to use scaled Schoenfeld residuals that take into account the variance of the estimates. More specifically, they are defined as: \( r_i^w = M^{-1}(\beta) r_i \), where \( m \) is the total number of events and \( \hat{V} \) is the estimated variance-covariance matrix of \( \beta \). If the scaled Schoenfeld residuals exhibit some non-random pattern at each event-time, it is a clear indication of violation of the PH assumption. So, a plot of scaled Schoenfeld residuals against time can be informative, since we expect their slope with respect to time to be zero under PH.

## 2.1.3 Measures for assessing the performance of a prediction model

When the objective of a study is to build a prediction model, the final stage is usually the assessment of its predictive performance. There are many proposed methods to do that and in most cases depends on the type of the analysis. Most of them are applicable also for survival outcomes, with the most well known and extensively used being the **C-index and Brier score**. However, these two measures are telling different stories about the model’s performance and that has to be clear.

The **C-index**, as proposed by Harrell et al [Harrell et al., 1996] is a measure for the discriminative ability of the model. More specifically, **discrimination** shows how well a model can distinguish between subjects who experience the outcome and those who do not. It is the fraction of pairs of patients whose predicted survival probabilities are correctly ordered among all pairs that can actually be ordered. The very last point is due to the presence of censoring in the data, and says that not all pairs in the dataset can be used for the calculation of C-index. More precisely, a pair is considered usable if both individuals are not-censored or if the individual with the shortest time has the event. Hence, after getting all the usable pairs in the dataset, we take the proportion
of those for whom the model gave the correct ordering in terms of survival probability, based on their observed outcome. In other words, it is a rank order statistic for predictions against true outcomes [Steyerberg et al., 2010], and it is known to be insensitive in small changes in detecting small differences in predictions between models.

It is not hard to see that values of C-index are between 0 & 1, with 1 denoting a perfect discrimination and a value of 0.5 corresponding to a model with no predictive ability, that is the model is no better than a random guess. There is no gold standard for values that are considered good or bad, but as a rule of thumb values that exceed 0.7 are said to define a good model.

Brier score is a measure of the prediction error of the model. It is a quadratic scoring rule, where the squared differences between actual binary outcomes \( Y \) and predictions \( p \) are calculated. A common formulation is:

\[
Brier(\hat{S}, t_0) = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{S}_i(t_0|x))^2
\]

where \( \hat{S}(t_0|x) \) is the survival probability that was calculated from the model given the predictor \( x \), \( y \) the actual outcome of the event at instance \( t_0 \) (0 if it does not happen and 1 if it does) and \( N \) is the number of observations.

Example [Wikipedia] Suppose that one is forecasting the probability \( P \) that it will rain on a given day. Then the Brier score is calculated as follows:

- If the forecast is 100% \((P = 1)\) and it rains, then the Brier Score is 0, the best score achievable.
- If the forecast is 100% and it does not rain, then the Brier Score is 1, the worst score achievable.
- If the forecast is 70% \((P = 0.70)\) and it rains, then the Brier Score is \((0.70 - 1)^2 = 0.09\).
- If the forecast is 30% \((P = 0.30)\) and it rains, then the Brier Score is \((0.30 - 1)^2 = 0.49\).
- If the forecast is 50% \((P = 0.50)\), then the Brier score is \((0.50 - 1)^2 = (0.50 - 0)^2 = 0.25\), regardless of whether it rains.

In survival context though, again censoring produces difficulties in the calculation. For that reason, a weighted function is used for the derivation of Brier score. [Graf et al., 1999] suggested a robust weighting scheme based on the assumption that the censoring mechanism is independent of the covariates and it is known as Inverse Probability of Censoring Weighting (IPCW). The idea is that the censored observations are excluded and the observed ones are multiplied by some weight. Let us denote by \( \hat{G}(t|x) \) the estimate of \( P(T_{cens} > t|x) \). The weights then are \( \frac{1}{\hat{G}(t|x)} \) if an event occurs before time \( t \), and \( \frac{1}{\hat{G}(t|x)} \) if the subject is still at risk at time \( t \). [Gerds and Schumacher, 2006] [Schumacher et al., 1999].

In other words, in case of censoring the calculation of Brier score is done as follows:

\[
Brier(\hat{S}, t_0) = \frac{1}{N} \sum_{i=1}^{N} (y - \hat{S}(t_0|x))^2 w_i
\]
where

\[ w_i = \begin{cases} 
0, & \text{if } T_i < t_0 \text{ and } d_i = 0 \\
\frac{1}{G(t_i|x)}, & \text{if } T_i > t_0 \\
\frac{1}{G(T_i|x)}, & \text{if } T_i < t_0 \text{ and } d_i = 1
\end{cases} \]
2.2 Missing values

Dealing with missing values is the core problem in this thesis. Therefore, it is useful to present some basic concepts about it, and this section is concerned with this issue.

Missing values are very common in any kind of analysis and appropriate ways to handle them are needed. [Wood et al., 2004] reviewed 71 BMJ, JAMA, Lancet and NEJM papers, and reported that 89% had partly missing data. Obviously, researchers have to be careful with how to deal with this issue. The goal then is to compensate for the missing information in the data, by exploiting best what we already know. All statistical analyses are based on different kind of assumptions, but with missing data extra assumptions have to be made.

One very important criterion for succeeding in this procedure depends on whether we can identify the reasons why the missingness occurred. Hence, the missing mechanism plays a key role in the analysis. [RUBIN, 1976] defined three general classes of missing data mechanisms, namely:

- Missing completely at random (MCAR)
- Missing at random (MAR)
- Missing not at random (MNAR)

Define \( Y = (y_{ij}) \) a \((n \times K)\) rectangular dataset, where \( n \) is the number of subjects and \( K \) the number of variables, with no missing values. Further, define a matrix \( M = (m_{ij}) \), with \( m_{ij} = 0 \) for observed and \( m_{ij} = 1 \) for missing observations respectively. Hence, \( M \) denotes the missingness pattern.

MCAR simply means that the missingness pattern is completely independent of either the specific values that should have been obtained or the observed data and consequently of any inference we would like to make about it. If we denote \( Y_{obs} \) and \( Y_{mis} \) the observed and unobserved components of data,

\[
f(M|Y_{obs}, Y_{mis}, X) = f(M)
\]

An example is when in a survey a respondent missed a page and did not answer the questions there, or when in a clinical trial a participant moves in another city and get lost to follow-up. MAR is merely a conditional statement about the missingness pattern. It can be dependent on the unobserved data, but conditional on the observed data it is not. In other words, if subjects are stratified on the basis of similar values for the responses that have been observed, missingness is simply the result of a chance mechanism that does not depend on the values of the unseen responses,

\[
f(M|Y_{obs}, Y_{mis}, X) = f(M|Y_{obs}, X)
\]

As an example we can have: two measurements of the same variable are made at the same time. If they differ by more than a given amount a third is taken. This third measurement is missing for those that do not differ by the given amount. Finally, if neither MCAR nor MAR is the case, we say that the data are MNAR. That is, the missing data is related to the values that should have been obtained, even after conditioning on the observed data.

However, we cannot tell which of the 3 mechanisms is true just by the data at hand. In practice, MCAR is usually impossible, and hence it is common to start the analysis with an assumption of MAR. At the end, it is advised to perform always a sensitivity analysis, in order to check the validity of the results under MAR or MNAR.
There are different proposed methods to deal with missing data. The most easy one, is to perform a Complete Case analysis (CCA). It ignores any unit with missing values, and hence, every subject in the dataset with at least 1 missing value is excluded from the analysis. If the data is MCAR, then it is unbiased but apparently inefficient. Under MAR or MNAR is in general biased, however this is not always the case and even under MNAR it can be valid under some assumptions.

Another approach is to explicitly impute values for the unobserved cases. The simplest and in most cases naïve way to do this, is to use ad-hoc methods, which are just convenient ways to proceed with the analysis. Some very common examples include: mean imputation, last observation carried forward, missing category method etc. The idea is that they impute a single value for each missing based on some very general and abstract criteria. However, these methods do not produce valid inferences, because they do not consider the issues caused by missingness. For instance, in case of mean imputation the missing values are filled in with the mean value of the observed cases. Therefore, it is not hard to see that the variances and possible p-values will be biased and generally underestimated. Figure 2.2.1 shows the severity of bias that can be produced with a naïve mean imputation.
2.2.1 Multiple Imputation

By imputing data, the goal is to recreate the dataset and restore the 'lost' information. Obviously, it will never be possible to know the 'true' values of the unseen data. Therefore, we will always have extra uncertainty for the analysis and the estimates of interest produced by the imputation procedure. That was the reason for the rise of multiple imputation (MI), invented by Donald Rubin in the 1970s and evolved to a powerful and widely used method for analyzing incomplete data.

In multiple imputation, $M$ imputed datasets are built and after calculating the statistics/quantities of interest in each one of them, we average them based on some rules that we describe in the next paragraph. The reason of using $M$ imputed datasets is that we can get a variance estimate as well as a confidence interval for the statistics, which would be hard from just a single imputation. Also, multiple imputation is a simulation based approach and the Monte-Carlo error reduces as $M$ increases.

After having created these $M$ imputed datasets, we are ready to proceed with the analysis. Firstly, we analyze each dataset separately as we would have done if the dataset was complete and extract the estimate of interest, $\hat{\theta}_m$. Hence, we end up with $M$ estimates, $\hat{\theta}_1, \ldots, \hat{\theta}_M$, which we average. Moreover, these analyses give also an estimate of the variance of $\hat{\theta}_m$, which we also wish to average. [Rubin, 1987] developed some rules in order to pool these estimates and their variances taking into account the extra variability produced by the multiple imputation procedure.

The average estimate is simply the mean of the $M$ estimates:

$$\hat{\theta} = \frac{1}{M} \sum_{m=1}^{M} \hat{\theta}_m$$

The variance however, consists of two parts. The within imputation and the between imputation variance component.

$$\widetilde{Var} = \left(1 + \frac{1}{M}\right) \hat{\sigma}_b^2 + \hat{\sigma}_w^2$$

where,

$$\hat{\sigma}_w^2 = \frac{1}{M} \sum_{m=1}^{M} \hat{\sigma}_m^2$$

and

$$\hat{\sigma}_b^2 = \frac{1}{M-1} \sum_{m=1}^{M} (\hat{\theta}_m - \hat{\theta})^2$$

A very crucial issue as emerged from the theory of multiple imputation is the number of $M$ imputed datasets that someone should create. Despite the early beliefs that 3-5 datasets are sufficient, recent publications suggest that such small number of imputed datasets can be still affected by the Monte Carlo error [Royston, 2004] [White et al., 2011] [Van Buuren, 2012]. The computational power nowadays gives us the flexibility and the advantage of choosing a much bigger number, and it is common to perform a MI with at least $M = 100$ datasets.
Let \( n_{\text{obs}} \) be the number of fully observed individuals, and \( W \) be the design matrix, consisting of one column of 1’s and the rest of the \( n_{\text{obs}} \) \( X \)'s (where \( Y \) is observed). Then, the sampling distributions are:

\[
\hat{\sigma}^2 \sim \frac{\hat{\sigma}^2 \chi^2_{n_{\text{obs}}-2}}{n_{\text{obs}} - 2},
\]

\[
\begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{pmatrix} \sim N \left\{ \begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{pmatrix}, \sigma^2(W^TW)^{-1} \right\}
\]

Together with non-informative priors, this leads to expressions for posterior distributions of parameters.

1. Estimate \( \sigma^2, \beta_0 \) and \( \beta_1 \) using the complete case analysis, giving \( \hat{\sigma}, \hat{\beta}_0 \) and \( \hat{\beta}_1 \)

2. For \( m = 1, \ldots, M \):
   (a) Draw from posterior distribution of parameters:
   i. Draw a \( \sigma^{2(m)} \) from \( \hat{\sigma}^2(n_{\text{obs}} - 2)/\chi^2_{n_{\text{obs}}-2} \)
   ii. Draw a \( (\hat{\beta}_0^m, \hat{\beta}_1^m) \) from

   \[
   N \left\{ \begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{pmatrix}, \sigma^{2(m)}(W^TW)^{-1} \right\}
   \]

   (b) If \( Y \) is missing for subject \( i \), impute \( Y_i \) by

   \[
   Y_i^m = \beta_0^m + \beta_1^m X_i + \epsilon^m
   \]

   where \( \epsilon^m \sim N(0, \sigma^{2(m)}) \)

Figure 2.2.2: Algorithm for (univariate) linear regression-based imputation

2.2.2

There are many methods for imputing values with the most important being:

- Univariate (regression-based) imputation methods for a single variable
- Data augmentation imputation using a joint probability model for \( Y \)
- The method of fully conditional univariate regression specification

In this thesis we are going to use the last method, which is described in the next section, but firstly a very brief summary of the first method is provided in Figure 2.2.2(taken from Tim Morris). This method is for cases where only one variable is subject to missing data. Obviously, the imputation model will be different in each of the \( M \) datasets and consequently the imputed values of \( Y \)'s, since every time model parameters are draws from their posterior distribution.
Let $Y^m$ be the vector of missing values for variable $Y$, and $Y^o$ be the vector of observed values of $Y$. Moreover, $\phi_j$ are unknown parameters of the imputation model, which are not of interest since they are used only to model conditional relations for imputation procedure.

1. Specify an imputation model $P(Y^m_j|Y^o_j, Y_{-j})$ for variable $Y_j$ with $j=1,\ldots,p$.
2. For each $j$, fill in starting imputations $\hat{Y}^o_j$ by random draws from $Y^o_j$.
3. Repeat for $k = 1,\ldots,l$ for $l$ iterations.
4. Repeat for $j = 1,\ldots,p$.
5. Define $\hat{Y}^k_{-j} = (\hat{Y}^k_1, \ldots, \hat{Y}^k_{j-1}, \hat{Y}^k_{j+1}, \ldots, \hat{Y}^k_p)$ as the currently complete data except $Y_j$.
6. Draw $\hat{\phi}^k_j \sim P(\phi^k_j|Y^o_j, \hat{Y}^k_{-j})$.
7. Draw imputations $\hat{Y}^k_j \sim P(Y^m_j|Y^o_j, \hat{Y}^k_{-j}, \hat{\phi}^k_j)$.
8. End repeat $j$.
9. End repeat $k$.

Figure 2.2.3: MICE Algorithm for imputation

### 2.2.2 Fully Conditional Specification

It is very common to have datasets with more than one variable incomplete and especially when there are numerous predictors. The problem with applying the regression-based imputation for each one of them is apparent. When we are going to impute one variable using as predictors in the imputation model the rest, we will encounter the obvious problem that these predictors will also have missing values. [Van Buuren et al., 1999] [Raghunathan et al., 2001] developed an algorithm which simply cycles through the partly observed variables and apply the univariate regression imputation sequentially after having imputed those values that are necessary for the imputation model. It is called **Fully conditional specification (FCS)** or **Imputation by chained equations (MICE)** and Figure 2.2.3 describes how it works. (taken by Van Buuren).

The concept that makes FCS appealing is that it is able to specify a different conditional model type for each variable, i.e. **logistic** for binary variables, **Poisson** for count variables etc.

A very crucial issue when performing multiple imputation is the **imputation model**, which is not necessarily the same with the final model that the researcher intends to use for the analysis itself. In survival analysis though, as we also described in **Section 2.1** there is a different setting than a usual regression and it would be essential to take that into account also in the imputation model. [White and Royston, 2009] in their paper suggest to use the cumulative baseline hazard $H_o(t)$ in the imputation model along with the event indicator instead of the observed survival time. If $H_o(t)$ is not known, they suggest to approximate it by the Nelson-Aalen estimator of $H(t)$.

There are many practical issues when someone is about to use **multiple imputation** in practice. There are several textbooks available, such as [Molenberghs, 2015] and [Van Buuren, 2012], that...
explain thoroughly how this should be done correctly. In this section we just gave a very brief summary before describing the techniques that we are going to use in this thesis.
2.3 Validation & Cross-validation

In the last section of this chapter with the basic statistical methods that we use in this analysis, the basics of validation are presented. Especially we focus on cross-validation which is the method that we will use later.

When the goal of analysis is to build a prediction model then it is vital to assess also its ability to give valid results, or in other words its predictive performance. In general, there are 2 broad classes of validation, namely internal and external validation. Internal validation is when the model is tested in the same dataset that was used for its calibration. On the other hand, in external validation the model is tested in a very different sample, completely independent from the development data and originate from a different but plausibly related setting [Steyerberg, 2009].

External validation can be strict or more relax in terms of its setting. For instance, the independent sample that is going to be used for it might be from the same hospital registry but from a different time period (Temporal validation), or in its most extreme definition, it can be form a cohort in a different country, a different researcher and maybe a different time period (fully independent validation).

However, it is not always possible to have an independent dataset to perform an external validation. Therefore, different techniques have been developed in order to validate a model using the same sample used for its development. Internal validation can be performed in different ways, with the most commonly used be the apparent validation, split-sample validation, cross-validation and bootstrap. Apparent validation is simply when the full sample is used for training the model as well as for validating it. Obviously, it will give optimistic assessment of its performance. With split-sample validation, the dataset is randomly divided in 2 parts, where one is used for fitting and the other for validating the model respectively. The rules of splitting are not standard with the most typical splits be 50 : 50 or 2/3 : 1/3. Bootstrap is a technique where we treat the original dataset as being the population of interest and repeatedly draw samples from it (B times) with replacement. Then a model is trained in each of these bootstrap samples and then it is evaluated both in the same bootstrap sample (apparent validation) and in the original sample, and their difference shows the optimism. Usually B = 100 or B = 200 are adequate.

Finally, cross-validation can be seen as a different kind of split-sample technique. The dataset is split in several folds containing around the same amount of observations. Then, one fold is left out and the model is developed in the rest. Finally, it is applied to the left-out fold for its evaluation. This is done for all the folds and hence each observation is at the test set once. It is not hard to see that cross-validation is more powerful than the simple split-sample method, since it uses a bigger proportion of the dataset for calibrating the model. However, it is advised to apply this procedure several times, 50 or 100 iterations are common, in order to get more stable results. Again here, the rules of splitting are matter of taste and depend on the dataset at hand. 1, 5 & 10-fold cross-validation are the most typical settings.
Chapter 3

Statistical Methods for the analysis

In this chapter we describe the statistical methods that we are going to use for analyzing the dataset. As was mentioned in Chapter 1, we will try 2 different approaches which we would like to compare eventually. We will follow essentially the same idea as in the paper of Wood et al [Wood et al., 2015]. Section 3.2 is about the first and Section 3.3 about the second approach respectively. Section 3.1 describes the steps that we follow in both approaches, and finally, in Sections 3.4 and 3.5 we draw the attention to the conceptual differences between the 2 approaches as well as to some very important issues relevant to both analyses.

3.1 Preliminary Steps

Before diving into the core algorithms of the two approaches, there are some steps common to both of them, and we are going to present them here. Of course, the way they will be applied still depends on the specific approach.

First of all, we chose to apply a 25-fold cross-validation, which means that the dataset will be split in $k = 25$ folds, and each of them will be used as test set in turn while the rest 24 folds will be used as train set respectively. As mentioned in Section 2.2.2, [White and Royston, 2009] suggest to use the cumulative baseline hazard along with the event indicator in the imputation model, and we follow their proposal here. However, for calculating the Nelson-Aalen estimator we use only the train set each time. Further, we use all the available variables in the dataset in the imputation model.

The next step is practically in the direction of dealing with the problem we discussed in Chapter 1, concerning the combination of multiple imputation with cross-validation, which is the main statistical concern in this thesis. Since we have defined our 2 sets based on the $k^{th}$ fold, for $k = 1, 2, \ldots, 25$, we are going to delete the survival outcome in the associated test set ($k^{th}$ fold). By doing that, we omit entirely the response from the rest of the procedure for observations in the test set, and overcome the problem of potential bias. By saying survival outcome, we care here only for the event indicator, since we will use in the imputation model the Nelson-Aalen estimator instead of the survival time and consequently the latter does not concern us. This is the reason
for using only the train set in the calculation of the Nelson-Aalen estimator. Now, we are set to apply multiple imputation to our data and for that purpose we are going to use the MICE package in R [Stef Van Buuren, 2011]. The way that we will apply it though is different for each approach and will be described in the next sections.

Finally, we will build 2 models in both approaches, one with all predictors and one without the variable \textit{Lvdias} (left ventricular diastolic dysfunction), which is the "problematic" one here. It is known that clinically it has a significant effect on the survival outcome for patients with heart failure, but since it is in 50% of the cases missing, we would like to see if with multiple imputation we can compensate for that missing information and improve the predictive performance of the model. Possible different behavior in the 2 approaches is also of interest.

### 3.2 Approach 1

This approach is essentially based on the expectation of the predictions. At the end of this procedure, we will end up with \( M \) predictions for each subject which we will eventually average by taking their mean. Figure 3.2.1 shows the full algorithm for approach 1. Also, Figure 3.2.2 provides a schematic representation of it.

Since we have defined the partition of the data, as described in Section 3.1, we will apply multiple imputation. However, it is not actually "multiple" here, because we will impute only 1 dataset and we will do that using the entire data, that is both train & test set. Then, we set aside the test set and work only on the train set. We will fit a Cox model to it and as we said already, we will not perform any variable selection, meaning that the model will consist of all the 14 variables in the dataset. Afterwards, we will apply this model to the test set to get the predictions for subjects in it. The above steps have to be done for every fold of this partition of the data (25 times), such that every subject will be at the test set once.

The above procedure comprises one cycle of the algorithm and will give 1 prediction for each subject. Next, we will define a brand new partition of the data and follow the exact same steps to get the second set of predictions (2\textsuperscript{nd} cycle of the algorithm). In total, this will be done \( M \) times, resulting in \( M \) predictions for each subject. As was already mentioned, the idea of this approach is about working on the expectation of predictions, hence we will take the mean of the \( M \) derived predictions in order to get the final survival probability for each subject in the dataset. Here, we examine different values of \( M \), namely 5, 100 & 200

### 3.3 Approach 2

The second approach that we will follow is based on Rubin’s rules, as defined in Section 2.2.1. Figure 3.3.1 shows the full algorithm for this approach and Figure 3.3.2 the associated schematic representation.

First of all, we follow the preliminary steps as described in Section 3.1. Next, based on the defined partition, we use the \( k\textsuperscript{th} \) fold as the test set, for \( k = 1, 2, \ldots, 25 \). We are now going to apply a multiple imputation to the entire dataset -train & test set- \( M \) times instead of 1 time that we did in the first approach. That means we will have \( M \) imputed datasets. It should be clear that in this approach we work on all of these \( M \) datasets and the splitting in train and test sets is done in all of them simultaneously. Here we examine again the same values of \( M \) as in approach 1. Afterwards,
3.3. APPROACH 2

CHAPTER 3. STATISTICAL METHODS FOR THE ANALYSIS

Algorithm for approach 1
The steps that we are going to follow for the first approach:

1. Make a random partition of the data. Define k as the number of folds, in order to do a cross-validation
2. Define $k^{th}$ fold, as the testing and the rest as the training set respectively
3. Calculate the cumulative hazard with Nelson-Aalen estimator in the training set
4. Delete the outcome from the testing set. We care only about the event indicator, since survival time will not be used in the imputation model
5. Apply imputation ONCE in the whole dataset
6. Split the data according to the partition defined in step 2
7. Fit a Cox model to the training set. No variable selection here, use the pre-specified model
8. Apply the model to the testing set to get the predictions for those subjects
9. Do steps 2-8 $k$ times. That is, use every fold as a test set once
10. Do the above procedure(steps 1-9) $M$ times, for a multiple cross-validation
11. Average the $M$ predictions for each subject in the dataset

Figure 3.2.1: Algorithm for approach 1

(a) 1$^{st}$ fold as test set  
(b) 2$^{nd}$ fold as test set  
(c) $k^{th}$ fold as test set

Figure 3.2.2: Approach 1. This is the 1st cycle of the algorithm. The 5 steps mentioned have to be done for all the folds of the partition, in order to complete one cycle of the algorithm. Then after defining a brand new partition, another cycle is performed. In total $M$ cycles will be done.
Algorithm for approach 2
The steps that we are going to follow for the second approach:

1. Make a random partition of the data. Define $k$ as the number of folds, in order to do a cross-validation.
2. Define $k^{th}$ fold as the testing and the rest as the training set respectively.
3. Calculate the cumulative hazard with Nelson-Aalen in the training set.
4. Delete the outcome from the testing set. We care only about the event indicator, since survival time will not be used in the imputation model.
5. Apply Multiple Imputation to the entire dataset, with $M$ imputed datasets.
6. Split the data according to the partition defined in step 2, for all the $M$ datasets created in step 5.
7. Fit Cox model to the training set in each of the $M$ imputed datasets. No variable selection here, use the pre-specified model.
8. Pool the coefficients from the $M$ models according to Rubin’s rules, to get the final model.
9. Apply the final model to the associated testing set in each of the $M$ imputed datasets to get the predictions for those subjects, which yields in $M$ predictions which will be then averaged.
10. Do steps 2-9 $k$ times. That is, use every fold as a testing set once.

Figure 3.3.1: Algorithm for approach 2

we will fit a Cox model (pre-specified, no variable selection) to the associated training set in each of the $M$ imputed datasets.

Therefore, for the $k^{th}$ fold which acts as the test set, we will have $M$ fitted Cox models. By applying Rubin’s rules, we will pool their coefficients to get the ‘final’ model. Then we will apply this model to the associated test set in each of the $M$ imputed datasets in order to get the predictions for the subjects in it. This will produce $M$ predictions per subject and in order to get the final survival probability for each one, we will eventually average them.

Of course, the above procedure has to be done for all the $k$ folds of the partition, such that every subject is in the test set once. Hence, the outcome from this algorithm will be a vector with 1 prediction per subject.
3.4 Summary

The difference in the 2 approaches is clear. In the first one, we investigate the mean of the predictions while in the second the predictions of the mean. In other words, in the first case we build several models and calculate the predictions for each subject based on each one of them, ending up with several predictions which we then average. On the other hand, in the second approach we firstly average the several models based on Rubin's rules to acquire a final model, and then we compute the survival fractions. The crucial difference therefore, lies in **when** we pool and consequently **what** we pool.

Another difference between them is that in the first approach we define a brand new random partition of the data each time. That is, for running the algorithm $M$ times, we need $M$ different partitions, while in the second approach the partition is one and fixed.

Finally, it is not hard to see that the end-product for approach 1 will not be a single-final model, since it depends on several independent ones and we want to extract predictions from each one of them. On the contrary, a model for approach 2 can be derived. This will be achieved by applying the multiple imputation to the entire dataset without the cross-validation part, and then extracting the model based on the pooled coefficients from the $M$ imputed datasets.

3.5 Important Notes

At the end of both procedures, and after having obtained the relevant results that we are interested in, we have to compare them to see if the give similar results and if one of them is superior. To do that, we wish to obtain some measures. First of all, we will extract the C-index value for both approaches to have an indication of their discrimination ability, and moreover, we will calculate Brier score to assess their prediction error, as was described in Section 2.1.3.

An important note that we have to make again, is that we will not apply any variable selection to our models. That means, we will use all the predictors that are believed to have a significant impact on the survival outcome for patients with heart failure, as came up from previous analyses.
on the subject. However, as we will see not all of them are statistically significant, and of course we will not look for any kind of transformations or to correct for the violation of the proportional hazards assumptions or in general for the covariate functional form. Obviously, we are aware that this will possibly prevent us from working with the optimal model. Hence, it should be clear that we are merely interested in comparing the 2 approaches rather than getting a model for immediate use, which was the objective of another research [Höke et al., 2017] and probably of future studies.

In order to make predictions using a Cox model, one needs 2 quantities. The estimated coefficients from the model to get the linear predictor, but also the baseline survival ($S_0(t)$) extracted from the model. The way we will get the latter will be the same in both approaches, but we need one more step in the second one before applying it to get the predictions. This is because we have $M$ imputed datasets and for each one of them we have a different fitted model. Hence, we need to pool these $M$ baseline hazards as we will do for the coefficients. That is, using Rubin’s rules we will get the average baseline hazard for each training set. Meanwhile, for the first approach this is not necessary(and appropriate perhaps) since each subject’s prediction is derived from one imputed dataset & model and hence based on one baseline hazard.
Chapter 4

Results

In this section the results from the analysis are presented. Firstly, in Section 4.1 some descriptive statistics are shown along with the preliminary analysis on the data that is essential. Then, in the following sections the results from the 2 approaches that we described in Section 3 are provided.

4.1 Descriptive Statistics

Table 4.1.1 shows some basic descriptive statistics for the variables in the dataset. 14 predictors are going to be used for this analysis, 5 numeric and 9 categorical, where 3 and 2 respectively contain missing values. A visual depiction of the numeric variables is provided in Figure 4.1.1. For the categorical ones, Table 4.1.1 has all the necessary information.
Table 4.1.1: Descriptive statistics for the variables in the dataset

For the categorical variables, the table shows the levels, their frequencies and percentages, while for the continuous it shows the mean along with the 1st and 3rd quantiles as well as the standard deviation. The last column presents the number of missing values for each variable along with their percentage. For a full explanation of each variable, see Table 1.3.1

\[ N = 1,053 \]

<table>
<thead>
<tr>
<th>Categorical Variables</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td>Levels</td>
<td>Frequencies n(%)</td>
</tr>
<tr>
<td>Gender</td>
<td>Male</td>
<td>805 (76.45)</td>
</tr>
<tr>
<td></td>
<td>Female</td>
<td>248 (23.55)</td>
</tr>
<tr>
<td>His</td>
<td>True</td>
<td>42 (96.01)</td>
</tr>
<tr>
<td></td>
<td>False</td>
<td>1011 (3.99)</td>
</tr>
<tr>
<td>Et</td>
<td>Ischemic</td>
<td>587 (55.75)</td>
</tr>
<tr>
<td></td>
<td>Non-Ischemic</td>
<td>466 (44.25)</td>
</tr>
<tr>
<td>NYHA</td>
<td>Class II</td>
<td>250 (23.74)</td>
</tr>
<tr>
<td></td>
<td>— III</td>
<td>713 (67.71)</td>
</tr>
<tr>
<td></td>
<td>— IV</td>
<td>90 (8.55)</td>
</tr>
<tr>
<td>DM</td>
<td>Diabetes</td>
<td>221 (20.99)</td>
</tr>
<tr>
<td></td>
<td>Non-Diabetes</td>
<td>832 (79.01)</td>
</tr>
<tr>
<td>Mr</td>
<td>0</td>
<td>841 (79.86)</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>182 (17.28)</td>
</tr>
<tr>
<td>Lvdias</td>
<td>Restrictive</td>
<td>175 (16.61)</td>
</tr>
<tr>
<td></td>
<td>Non-Restrictive</td>
<td>354 (33.61)</td>
</tr>
<tr>
<td>Lbbblink</td>
<td>0</td>
<td>361 (34.28)</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>692 (65.72)</td>
</tr>
<tr>
<td>Af</td>
<td>True</td>
<td>177 (16.81)</td>
</tr>
<tr>
<td></td>
<td>False</td>
<td>876 (83.19)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Continuous Variables</th>
<th>Name, Unit of measure</th>
<th>Mean(1st − 3rd Quartile)</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Age, years</td>
<td>67.01 (61.2 - 74.4)</td>
<td>9.82</td>
</tr>
<tr>
<td></td>
<td>Egfr, ml/min</td>
<td>69.73 (47.1 - 85.3)</td>
<td>31.79</td>
</tr>
<tr>
<td></td>
<td>Hb, mmol/L</td>
<td>8.33 (7.6 - 9)</td>
<td>1.01</td>
</tr>
<tr>
<td></td>
<td>LVEF, %</td>
<td>25.9 (19.7 - 31.35)</td>
<td>8.16</td>
</tr>
<tr>
<td></td>
<td>QRS duration, ms</td>
<td>165.7 (146 - 181)</td>
<td>26.38</td>
</tr>
</tbody>
</table>
4.1. DESCRIPTIVE STATISTICS

CHAPTER 4. RESULTS

Histogram for variable Qrs

Mean = 165.7
Median = 164
SD = 26.38

Histogram for variable Age

Mean = 67.01
Median = 68.11
SD = 9.82

(a) Histogram for Qrs duration and Age
4.1. DESCRIPTIVE STATISTICS

CHAPTER 4. RESULTS

Histogram for variable Lvef

Mean = 25.9
Median = 25.86
SD = 8.16

Histogram for variable Hb

Mean = 8.33
Median = 8.4
SD = 1.01

(b) Histogram for \textit{Lvef} and \textit{Hb}
4.1. DESCRIPTIVE STATISTICS

CHAPTER 4. RESULTS

Mean = 69.73
Median = 65.41
SD = 31.79

(c) Histogram for Egfr

Figure 4.1.1: Histograms for the continuous variables in the dataset
First of all, a Kaplan Meier survival curve is presented in Figure 4.1.2 based on the full data. We observe that there are late events, with the last observation in the dataset being censored with a survival outcome of \( t = 167.3 \) months, and the last event observation at \( t = 156.1 \) months (roughly 13 years). Moreover, the median survival is \( t = 85.4 \) months.

Although the survival times for the subjects in the study range from 6 days to 167 months, the researchers are more interested in survival fractions for 1 & 5 years (12 and 60 months respectively). Moreover, there are not many events later in the study and as we will show later the assumption of proportional hazards is not met which is more due to the later times, and considering the fact that we are not going to investigate possible time-dependent hazard ratios, we decided to restrict our attention to the time interval [0, 84] months, which is also the 75% of the vector of time variable. Therefore, the very first thing to do, consider it as step 0 in algorithms 3.2.1 & 3.3.1, is to apply administrative censorship to our data at \( t = 84 \) months. Subjects with survival times longer than that will get a status indicator equal to Alive no matter what their initial status was, with time \( = 84 \) months.

### 4.1.1 Check of the Proportional Hazards Assumption

Since we are going to build several models for each approach derived from essentially different datasets, it would be cumbersome to assess the PH assumption within the algorithm. Moreover, as mentioned in Chapter 3, in this thesis we are going to use a pre-specified model, that is no variable selection will be performed here. Therefore, even if we were to detect a violation of the PH assumption, no action would be taken as explained in the previous section. However, it is of great importance to have an indication of whether it holds or not and possibly to what degree. For that reason, we decided to run a naive analysis with Cox model only to see where we stand. A multiple imputation with \( M = 20 \) imputed datasets is applied and Cox model is fitted to each one of them. Next, the PH assumption is checked. Table 4.1.2 shows the scaled Schoenfeld residuals tests for the variables of the model based on one of the multiple imputed datasets (the results from the other datasets are similar). For completeness, the associated plots of the scaled Schoenfeld residuals are presented in the Appendix in Figure A.0.1. Also, the log-minus-log plots for the 9 categorical variables are provided there in Figure A.0.2, where we observe a pattern of crossing curves in almost all the variables. This pattern seems to concern almost exclusively the very early times, which might be because the small number of events during that period. After the first 4-5 months the distance between the curves seems to stabilize and for the majority of the variables they seem to run in a parallel pattern.

In Table 4.1.2 we see that for almost all the variables there is no indication of violation of the proportional hazards assumption. However, this is not the case for variables Age and NYHA as
4.1. DESCRIPTIVE STATISTICS

<table>
<thead>
<tr>
<th></th>
<th>rho</th>
<th>chisq</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>0.12346</td>
<td>8.7115</td>
<td>0.00316</td>
</tr>
<tr>
<td>Gender2</td>
<td>-0.01836</td>
<td>0.1695</td>
<td>0.68053</td>
</tr>
<tr>
<td>His2</td>
<td>0.02177</td>
<td>0.2384</td>
<td>0.62534</td>
</tr>
<tr>
<td>Et2</td>
<td>0.05481</td>
<td>1.5682</td>
<td>0.21047</td>
</tr>
<tr>
<td>Nyha2</td>
<td>-0.13316</td>
<td>9.2768</td>
<td>0.00232</td>
</tr>
<tr>
<td>Nyha3</td>
<td>-0.10179</td>
<td>5.4182</td>
<td>0.01993</td>
</tr>
<tr>
<td>Dm2</td>
<td>0.04631</td>
<td>1.1093</td>
<td>0.29223</td>
</tr>
<tr>
<td>Egfr</td>
<td>0.04366</td>
<td>1.2449</td>
<td>0.26453</td>
</tr>
<tr>
<td>Lvialis2</td>
<td>-0.06617</td>
<td>2.1854</td>
<td>0.13932</td>
</tr>
<tr>
<td>Mr2</td>
<td>0.00515</td>
<td>0.0132</td>
<td>0.90845</td>
</tr>
<tr>
<td>Hb</td>
<td>0.04133</td>
<td>0.9572</td>
<td>0.32789</td>
</tr>
<tr>
<td>Lvef</td>
<td>0.00986</td>
<td>0.0483</td>
<td>0.82607</td>
</tr>
<tr>
<td>Qrs</td>
<td>0.07298</td>
<td>3.0470</td>
<td>0.08088</td>
</tr>
<tr>
<td>Lbb2</td>
<td>0.01736</td>
<td>0.1519</td>
<td>0.69670</td>
</tr>
<tr>
<td>Af2</td>
<td>-0.02357</td>
<td>0.3052</td>
<td>0.58063</td>
</tr>
<tr>
<td>GLOBAL</td>
<td>NA</td>
<td>32.2320</td>
<td>0.00599</td>
</tr>
</tbody>
</table>

Table 4.1.2: Scaled Schoenfeld Residuals for the original dataset

4.1.2

as for the global test of proportionality. As can be seen in Figure A.0.1 in the Appendix, the confidence intervals for all variables become bigger at the later times and consequently the uncertainty. Based on these results and the scientific interest of this project, we decided to limit the analysis in the interval [0, 84] months. Table 4.1.3 shows the numerical tests based on the scaled Schoenfeld residuals for the dataset with the administrative censorship. We see that only NYHA violates the proportionality now based on the 5%-level, and although the global test is still significant, the chi-square statistic is reduced.
4.2. Missing data patterns

A very crucial issue in this analysis is the missing data, especially for the variable \textit{Lvdias}. As described in Section 3, we are going to use \textit{multiple imputation}. Before however applying it, it is very important to check the missing patterns in our dataset. Hence, we think that it is of great interest to present here some statistics about it, to know what we are dealing with. First of all, Table 4.2.1 shows the missing patterns in the dataset, where 1 denotes a missing and 0 a non-missing case respectively. The first column gives the number of rows that the associated pattern occurs in the dataset, while the last column shows the number of variables that have at least 1 missing in the specific pattern. For instance, the first row of the table says that the dataset has 524 complete rows, while as can be seen in 5\textsuperscript{th} row, in 480 cases only \textit{Lvdias} is missing. Finally, the last row gives the number of missing values per variable. The second table, Table 4.2.2, provides some measures about the dependence of the variables in the dataset. The first column shows again the probability of observed values for each variable. The next 2 columns are about 2 coefficients that give us an impression of how the variables are connected in terms of missingness. \textbf{Influx coefficient} is simply the ratio of the number of variable pairs \((Y_j, Y_k)\) with \(Y_j\) missing and \(Y_k\) observed, divided by the total number of observed data cells. For a variable entirely missing, this coefficient is 1, and 0 for a complete one. On the other hand, \textbf{outflux coefficient} is defined in the opposite manner, by dividing the number of pairs \((Y_j, Y_k)\) with \(Y_j\) observed and \(Y_k\) missing, by the total number of incomplete cells. Apparently, outflux gives an indication of how useful this variable will be for imputing other variables in the dataset, while influx is an indicator of how easily this variable can be imputed. Analogously, outflux takes the value 1 for a completely observed variable and 0 otherwise. We see that all the variables are useful for the imputation procedure, except of course \textit{Lvdias} which does not seem very informative in general. As Van Buuren states in his

<table>
<thead>
<tr>
<th></th>
<th>rho</th>
<th>chisq</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>0.07175</td>
<td>2.44707</td>
<td>0.11774</td>
</tr>
<tr>
<td>Gender2</td>
<td>-0.04225</td>
<td>0.78832</td>
<td>0.37461</td>
</tr>
<tr>
<td>His2</td>
<td>-0.03353</td>
<td>0.49008</td>
<td>0.48389</td>
</tr>
<tr>
<td>Et2</td>
<td>0.05788</td>
<td>1.51821</td>
<td>0.21789</td>
</tr>
<tr>
<td>Nyha2</td>
<td>-0.13503</td>
<td>8.20680</td>
<td>0.00417</td>
</tr>
<tr>
<td>Nyha3</td>
<td>-0.12684</td>
<td>7.22901</td>
<td>0.00717</td>
</tr>
<tr>
<td>Dm2</td>
<td>0.07116</td>
<td>2.27258</td>
<td>0.13168</td>
</tr>
<tr>
<td>Egfr</td>
<td>0.02124</td>
<td>0.25046</td>
<td>0.61675</td>
</tr>
<tr>
<td>Lvdias2</td>
<td>-0.05026</td>
<td>1.11924</td>
<td>0.29008</td>
</tr>
<tr>
<td>Mr2</td>
<td>-0.00259</td>
<td>0.00297</td>
<td>0.95651</td>
</tr>
<tr>
<td>Hb</td>
<td>0.06768</td>
<td>2.20719</td>
<td>0.13737</td>
</tr>
<tr>
<td>Lvef</td>
<td>0.03471</td>
<td>0.53104</td>
<td>0.46617</td>
</tr>
<tr>
<td>Qrs</td>
<td>0.07721</td>
<td>2.93711</td>
<td>0.08657</td>
</tr>
<tr>
<td>Lbbb2</td>
<td>-0.05267</td>
<td>1.21279</td>
<td>0.27078</td>
</tr>
<tr>
<td>Af2</td>
<td>-0.02902</td>
<td>0.40039</td>
<td>0.62689</td>
</tr>
<tr>
<td>GLOBAL</td>
<td>NA</td>
<td>27.39875</td>
<td>0.02565</td>
</tr>
</tbody>
</table>

Table 4.1.3: Scaled Schoenfeld Residuals for the censored dataset
book [Van Buuren, 2012], a high outflux might turn out to be useless for the imputation procedure if it is unrelated to the incomplete variables, while the usefulness of a highly predictive variable is severely limited by a low outflux value. Therefore, we might have to consider if \textit{Lvdias} should be included in the imputation model.

\begin{table}[h]
\centering
\begin{tabular}{cccccccccccccccccccc}
\hline
\textbf{Age} & \textbf{Gender} & \textbf{AtrAct} & \textbf{His} & \textbf{Et} & \textbf{NYHA} & \textbf{Dm} & \textbf{Qrs} & \textbf{Lbbb} & \textbf{Af} & \textbf{Censor} & \textbf{Stime} & \textbf{Egfr} & \textbf{Hb} & \textbf{Lvef} & \textbf{Mr} & \textbf{Lvdias} \\
\hline
524 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\
2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\
480 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 2 \\
4 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 2 \\
21 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 2 \\
10 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\
2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\
7 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 7 & 20 & 30 & 524 & 583 \\
\hline
\end{tabular}
\caption{Patterns of missing data}
\end{table}

\begin{table}[h]
\centering
\begin{tabular}{ccc}
\hline
\textbf{Variable} & \textbf{pobs} & \textbf{influx} & \textbf{outflux} \\
\hline
Age & 1.00 & 0.00 & 1.00 \\
Gender & 1.00 & 0.00 & 1.00 \\
AtrAct & 1.00 & 0.00 & 1.00 \\
His & 1.00 & 0.00 & 1.00 \\
Et & 1.00 & 0.00 & 1.00 \\
NYHA & 1.00 & 0.00 & 1.00 \\
Dm & 1.00 & 0.00 & 1.00 \\
Egfr & 1.00 & 0.00 & 0.99 \\
Hb & 0.99 & 0.01 & 0.97 \\
Mr & 0.97 & 0.03 & 0.89 \\
Lvef & 0.98 & 0.02 & 0.92 \\
Qrs & 1.00 & 0.00 & 1.00 \\
Lvdias & \textbf{0.50} & \textbf{0.48} & \textbf{0.01} \\
Lbbb & 1.00 & 0.00 & 1.00 \\
Af & 1.00 & 0.00 & 1.00 \\
Censor & 1.00 & 0.00 & 1.00 \\
Stime & 1.00 & 0.00 & 1.00 \\
\hline
\end{tabular}
\caption{Statistics for the variables in the dataset, based on their missingness patterns}
\end{table}
4.2.1 Lvdias-Distribution of imputed values

Before starting with the core algorithm of this project, we would like to check how the multiple imputation procedure behaves in case of Lvdias (left ventricular diastolic dysfunction). That is, we wish to check the distribution of its imputed values to assess if they are random or there is a pattern.

In order to achieve that, we ran a multiple imputation with MICE 100 times (100 MI datasets), and save them. That gave us 100 imputed values for Lvdias, obviously only for the 524 subjects who have their values missing for it. Next, we calculated the frequency of the 2 possible values (Restrictive, Non-Restrictive) for each subject and finally order them based on the frequency of value Restrictive. Figure 4.2.1 provides the plot based on this ordered vector of percentages. As can be seen in Table 4.1.1 in the observed cases of Lvdias value "Restrictive" concerns 33% of the data and value "Non-Restrictive the rest 67%. Therefore, a naive random imputation would give 1/3 of the times the value "Restrictive" for all the subjects. However, we see in Figure 4.2.1 that this is not the case, since the percentages among the individuals are scattered in the whole range of probability values. This simply means that multiple imputation takes into consideration the specific characteristics of each subject and impute analogously. Obviously, the ideal here would be to have only 0 or 100% for each subject, which would mean that there is no uncertainty and the imputation would be able to successfully recover the true value for each one. However, this is not a realistic scenario since the true value is completely unknown and through multiple imputation we wish to draw possible data values from a distribution specifically modeled for each missing entry. Finally, Table 4.2.3 shows the five number summary for both values of the variable Lvdias, where we see it also numerically.

<table>
<thead>
<tr>
<th></th>
<th>Non-restrictive</th>
<th>Restrictive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>0.1700</td>
<td>0.0900</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>0.5400</td>
<td>0.2500</td>
</tr>
<tr>
<td>Median</td>
<td>0.6600</td>
<td>0.3400</td>
</tr>
<tr>
<td>Mean</td>
<td>0.6359</td>
<td>0.3641</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>0.7500</td>
<td>0.4600</td>
</tr>
<tr>
<td>Max</td>
<td>0.9100</td>
<td>0.8300</td>
</tr>
</tbody>
</table>

Table 4.2.3: Based on a multiple imputation with 100 imputed datasets, the frequency of the 2 possible values of variable Lvdias (Restrictive, Non-restrictive) was derived, among the 100 imputed values for each subject. This table provides the five number summary for both of them among the 524 subjects with missing value for Lvdias.
Figure 4.2.1: Percentage of imputed values-Lvdiases. x-axis is just a numbering of the 524 subjects with missing Lvdiases, in an increasing order of the percentage of the value "Restrictive"
4.3 Results from the 2 approaches

As was described in Chapter 3, we implemented the 2 approaches for different values of \( M \). Moreover, for each we built 2 models, one with and one without the variable \( Lvdias \). The aim of this thesis is to apply and eventually compare the 2 approaches, as well as to assess if \( M \) plays a key role in the performance of both of them. As was also made clear in Chapter 3, approach 1 does not produce a single model, and therefore we will present one here only for approach 2. Hence, since the comparison of the 2 algorithms will be done based on the predictions they give, we thought that the best way to present them is visually.

We will provide different sets of plots in each of the next subsections.

4.3.1 Approach 2 vs Approach 1 plots

These first plots are produced as follows: For each subject we have 2 predictions, 1 from each approach. The predictions from the first are sorted in an increasing order and we also sort the predictions from the second approach based on that ordering. Finally, on x-axis we plot the (ordered) predictions from the 1\(^{st} \) and on y-axis those from the 2\(^{nd} \) approach respectively. Obviously, if the 2 approaches agree perfectly, these points would be on a straight 45\(^{o} \) line, which is also plotted for easier inference.

The next 3 sets of plots shown in Figure 4.3.1 - Figure 4.3.3 are about the models without the variable \( Lvdias \). Specifically, the left plot is for the results produced from \( M = 5 \), the one in the middle from \( M = 100 \) and the far right from \( M = 200 \) MI datasets respectively. The survival fractions at \( t = 12, 60 \& 84 \) months are shown. In these models, the imputation is very light in a sense that only a very small percentage of cases have missing values. Nevertheless, even here the imputation in \( Lvdias \) plays also a key role since we use it in the imputation model and although it is not used in the final model it affects indirectly the rest.

We observe that the plots from analyses with different value of \( M \), are not dramatically different between each other. Maybe there is a bigger "explosion" for \( M = 5 \) around the 45\(^{o} \) line compared to the other two, but cannot be considered significant.
4.3. RESULTS FROM THE 2 APPROACHES

Figure 4.3.1: In the x-axis the ordered survival probabilities from the 1st approach are plotted against the survival probabilities from the 2nd approach in the y-axis, at $t = 12\text{months}$ for the different values of M. These survival fractions were produced from the model without the variable $Lvdias$. 
Figure 4.3.2: In the x-axis the ordered survival probabilities from the 1\textsuperscript{st} approach are plotted against the survival probabilities from the 2\textsuperscript{nd} approach in the y-axis, at \( t = 60 \text{months} \) for the different values of M. These survival fractions were produced from the model without the variable \textit{Lvdias}.
Figure 4.3.3: In the x-axis the ordered survival probabilities from the 1\textsuperscript{st} approach are plotted against the survival probabilities from the 2\textsuperscript{nd} approach in the y-axis, at $t = 84\text{months}$ for the different values of M. These survival fractions were produced from the model without the variable $Lvdias$. 
We now present the same plots for the full models, including *Lvdias*, in *Figure 4.3.4 - Figure 4.3.6*. Here it is crystal clear that by increasing the value of $M$, the 2 approaches become more alike and as a consequence there is less sparsity around the 45° line. Moreover, some extreme outliers in the case of $M = 5$ are not there for the $M = 100$ and $M = 200$. Here the variation in the predictions caused by the multiple imputation procedure is reduced significantly when we increase the value of imputations, and this becomes even more apparent for survival probabilities at later times.

![Figure 4.3.4](image)

*Figure 4.3.4*: In the x-axis the ordered survival probabilities from the 1st approach are plotted against the survival probabilities from the 2nd approach in the y-axis, at $t = 12$ months for the different values of $M$. The red points denote the subjects with missing value for the variable *Lvdias*. These survival fractions were produced from the full model.
Figure 4.3.5: In the x-axis the ordered survival probabilities from the 1st approach are plotted against the survival probabilities from the 2nd approach in the y-axis, at $t = 60$ months for the different values of $M$. The red points denote the subjects with missing value for the variable $Lvdias$. These survival fractions were produced from the full model.
Figure 4.3.6: In the x-axis the ordered survival probabilities from the 1\textsuperscript{st} approach are plotted against the survival probabilities from the 2\textsuperscript{nd} approach, in the y-axis, at $t = 84\text{months}$ for the different values of M. The red points denote the subjects with missing value for the variable $Lvdias$. These survival fractions were produced from the full model.
4.3.2 Sum vs difference of survival probabilities plots

In the previous plots (Figures 4.3.1 - 4.3.6), we observed that in both models, with and without the variable Lvdias, for small values of $M$ the 2 approaches give survival probabilities that differ more than the respective ones in case of higher values of $M$, especially in the former case. Of course it is hard to see how much different from those plots. Therefore, we explore further this issue by examining a second set of plots.

The goal of these plots is to see how much difference there is for the subjects in the dataset concerning their survival fractions extracted from the 2 approaches in the 3 time-points that we examine. For that reason, we present the plots in Figure 4.3.7 - Figure 4.3.9, where in the x-axis the sum of the 2 survival probabilities per subject from the 2 approaches is plotted against the associated difference in the y-axis. These plots are only for the full-model.

From these plots it is more clear that $M$ plays a key role in the 2 approaches. For $M = 5$ we have bigger differences amongst the 2 algorithms with more outliers. For each set of plots, that is for each time-point, the y-axis scales are the same for all 3 $M$'s, and we can see that the top one($M = 5$) is way more sparse than the others. Numerically, this can be seen also from the 5th and 95th percentile of the vector with the differences, shown in the legends of the plots.

Finally, we can observe that the biggest variation in the predictions concerns those in the middle. That is, there is more uncertainty in the estimation of the survival fraction at the middle of the probability range than at the low or high end. This is not unexpected of course, since it is always "easier" to predict for the extreme cases.
Figure 4.3.7: In the x-axis the sum of the survival fractions from the 2 approaches are plotted against their difference in the y-axis, at $t = 12$ months. The red dotted line denotes the 5th percentile of the difference while the green denotes the 95th respectively. The black solid line is for $y = 0$. 
Figure 4.3.8: In the x-axis the sum of the survival fractions from the 2 approaches are plotted against their difference in the y-axis, at \( t = 60 \) months. The red dotted line denotes the 5\(^{th}\) percentile of the difference while the green denotes the 95\(^{th}\) respectively. The black solid line is for \( y = 0 \)
Figure 4.3.9: In the x-axis the sum of the survival fractions from the 2 approaches are plotted against their difference in the y-axis, at $t = 84\text{ months}$. The red dotted line denotes the 5th percentile of the difference while the green denotes the 95th respectively. The black solid line is for $y = 0$. 
4.3.3 Approach 2 vs Approach 1, split by Status & Missing Lvdias plots

The last set of plots that we show here, is essentially the first set but now we split them based on the indicator status, that is if someone experiences the event or not, as well as based on the missing status of the variable Lvdias. For efficiency, we compare here only $M = 5$ and $M = 200$.

By examining these last plots, it should be clear that the variable Lvdias and the fact that it is missing in 50% of the cases causes most of the discrepancy between the 2 approaches. In all the 3 pairs of plots, we see in the second columns of the figures, that the majority of the points are on the 45° straight line, and if we examine also the numerical differences we will see that the observed differences are really small. We wouldn’t expect to get exactly the same survival probabilities anyway. Turning now to the first column of the plots, corresponding to the subjects with missing value for Lvdias, we see how much we reduce the variation around the 45° line as we increase the value of $M$. As can be seen, this is more apparent for later time points, where the survival predictions are more scattered in the whole range of probability values.

Finally, we see that the difference between the 2 approaches is not systematic in the sense that there is not a pattern concerning the probabilities they give. That is, we do not see one approach to give systematically lower or higher survival probabilities than the other, and the points are evenly divided up and down of the line. We can also see, that the model seems to discriminate between those with and those without the event, as we observe that the probabilities are lower for the former, although we can claim that this distinction could be even better.
Figure 4.3.10: In the x-axis the ordered survival probabilities from the 1st approach are plotted against the survival probabilities from the 2nd approach in the y-axis. The splitting was done according to the indicator status for the event at $t = 12$ months, and the status of the variable Lvdias, missing or not. Left plot concerns M=5 and right M=200 respectively.
4.3. RESULTS FROM THE 2 APPROACHES

CHAPTER 4. RESULTS

Figure 4.3.11: In the x-axis the ordered survival probabilities from the 1\textsuperscript{st} approach are plotted against the survival probabilities from the 2\textsuperscript{nd} approach in the y-axis. The splitting was done according to the indicator status for the event at $t = 60\text{months}$, and the status of the variable $Lvdias$, missing or not. Left plot concerns M=5 and right M=200 respectively.
Figure 4.3.12: In the x-axis the ordered survival probabilities from the $1^{st}$ approach are plotted against the survival probabilities from the $2^{nd}$ approach in the y-axis. The splitting was done according to the indicator status for the event at $t = 84$ months, and the status of the variable $Lvdias$, missing or not. Left plot concerns $M=5$ and right $M=200$ respectively.
4.3. RESULTS FROM THE 2 APPROACHES

C-index(%)  
Cases Survivors Censored M=5 M=100 M=200

<table>
<thead>
<tr>
<th>M=5</th>
<th>M=100</th>
<th>M=200</th>
</tr>
</thead>
<tbody>
<tr>
<td>74.56</td>
<td>75.72</td>
<td>76.08</td>
</tr>
<tr>
<td>72.19</td>
<td>73.18</td>
<td>73.23</td>
</tr>
<tr>
<td>72.05</td>
<td>73.18</td>
<td>73.05</td>
</tr>
</tbody>
</table>

Table 4.3.1: C-index values for Approach 1. For values M=5, 100 & 200, and time-points t=12, 60 & 84 months

<table>
<thead>
<tr>
<th>M=5</th>
<th>M=100</th>
<th>M=200</th>
</tr>
</thead>
<tbody>
<tr>
<td>75.66</td>
<td>76.33</td>
<td>76.20</td>
</tr>
<tr>
<td>72.17</td>
<td>73.69</td>
<td>73.39</td>
</tr>
<tr>
<td>72.03</td>
<td>72.81</td>
<td>73.25</td>
</tr>
</tbody>
</table>

Table 4.3.2: C-index values for Approach 2. For values M=5, 100 & 200, and time-points t=12, 60 & 84 months

4.3.4 Assessing the performance of the models

As we described in Section 2.1.3 we are going to use C-index and Brier score to assess our models’ discrimination power and their prediction error. Tables 4.3.1 and 4.3.2 present the C-index values for the first and second approach respectively, for all the 3 values of M that we examined (5, 100 & 200), as well as for the 3 time-points of interest, namely t = 12, 60 & 84 months.
As a general comment from these results, we can say that there are no big differences between the two approaches, but neither between different values of $M$. However, we can see here again, that the predictions derived from a procedure with $M = 5$ gave lower $C$-index values than from that with $M = 100$ in both approaches, as well as from that with $M = 200$.

Furthermore, we cannot claim that one of the two approaches is superior to the other, since the values are really close to each other and they seem to come closer and closer as we increase $M$. Therefore, there is no clear winner concerning the $C$-index value after this analysis at least.

We now move to the Brier score. Tables 4.3.3, 4.3.4 & 4.3.5 provide the results for the predictions derived from $M = 5$, $M = 100$ & $M = 200$ respectively, and for the time-points $t = 12$ 60 & 84 months.

<table>
<thead>
<tr>
<th></th>
<th>Approach 1</th>
<th>Approach2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>reference</td>
<td>Brier-NoLvdiasia</td>
</tr>
<tr>
<td>$t=12$</td>
<td>0.0734</td>
<td>0.0677</td>
</tr>
<tr>
<td>$t=60$</td>
<td>0.2257</td>
<td>0.1913</td>
</tr>
<tr>
<td>$t=83.99$</td>
<td>0.2499</td>
<td>0.2111</td>
</tr>
</tbody>
</table>

Table 4.3.3: Brier score for $M=5$. The first column correspond to the reference model(without covariates), the next 2 to the models( with & without $Lvdias$) from the 1st approach, and the last 2 respectively from the 2nd approach.

<table>
<thead>
<tr>
<th></th>
<th>Approach 1</th>
<th>Approach2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>reference</td>
<td>Brier-NoLvdiasia</td>
</tr>
<tr>
<td>$t=12$</td>
<td>0.0734</td>
<td>0.0675</td>
</tr>
<tr>
<td>$t=60$</td>
<td>0.2257</td>
<td>0.1912</td>
</tr>
<tr>
<td>$t=83.99$</td>
<td>0.2499</td>
<td>0.2111</td>
</tr>
</tbody>
</table>

Table 4.3.4: Brier score for $M=100$. The first column correspond to the reference model(without covariates), the next 2 to the models( with & without $Lvdias$) from the 1st approach, and the last 2 respectively from the 2nd approach.

<table>
<thead>
<tr>
<th></th>
<th>Approach 1</th>
<th>Approach2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>reference</td>
<td>Brier-NoLvdiasia</td>
</tr>
<tr>
<td>$t=12$</td>
<td>0.0734</td>
<td>0.0675</td>
</tr>
<tr>
<td>$t=60$</td>
<td>0.2257</td>
<td>0.1912</td>
</tr>
<tr>
<td>$t=83.99$</td>
<td>0.2499</td>
<td>0.2110</td>
</tr>
</tbody>
</table>

Table 4.3.5: Brier score for $M=200$. The first column correspond to the reference model(without covariates), the next 2 to the models( with & without $Lvdias$) from the 1st approach, and the last 2 respectively from the 2nd approach.

The most clear message that we can get from these tables is that the Brier score has only slight differences between the models and the different $M’s$. We have to move even to the 4th
decimal to find a difference. Moreover, we see that our models achieved a significant reduction of the \textit{Brier score} relative to the reference model. Trying now to unscramble these results, we can observe that the models which include also \textit{Lvdias} are slightly better than the more simple ones in all circumstances based on Brier score, which can give us an indication that the procedures we followed to impute its many missing values, achieved their purpose. Moreover, it is reasonable that as we move towards later time-points the models become more uncertain and hence, their prediction error becomes larger.

If we try to explain the Brier score results in association with the plots we examined earlier, we could say that within each value of $M$ the results are reasonable. For instance, for $M = 200$ we see that the difference in the prediction error of the 2 approaches for the model with Lvdias, is less than the associated one in the $M = 100$ and even less than in $M = 5$. This comes in agreement with the plots, where we saw that the 2 approaches give predictions which are very close between each other for $M = 100$ & 200 while for $M = 5$ those predictions are more different, and apparently this also reflects to the Brier scores.

\subsection*{4.4 Variance}

An important part of the usefulness of any kind of prediction is the assessment of its uncertainty. This can be done in various ways, but in our case here we are going to examine the variance that the two procedures give. How to estimate the variance can be done, again, in different ways, but we will keep things simple here.

For each subject we have $M$ predictions (survival probabilities) per approach, and we will simply calculate their variance in the same sense that we took their mean for the final prediction per subject. Therefore, we will get a vector with variances for each approach. Then, the idea is to run a \textit{paired t-test} to see if these variances differ significantly between the 2 approaches, or better if the mean of these differences are different from zero, and possibly the magnitude of this difference. However, there is huge violation of the normality assumption and the distribution is highly skewed, and so the t-test does not seem the best solution. Hence, we opt for the non-parametric equivalent, the \textit{Wilcoxon Signed-Ranks test}.

First of all, in Tables 4.4.1 and 4.4.2 we present the mean variances for each approach, for the models without and with Lvdias respectively. As can be seen there, the variances are really small, even to the sixth decimal. However, even now we can see that the first approach gives higher variances in all settings shown in the tables.

Wilcoxon Signed-Ranks test is performed for all the 9 different settings - 3 time-points & 3 values of $M$. The results are presented in Table 4.4.3.

One clear message for the these tables is that the first approach produces always higher variance than the second. Although the order of magnitude is really small, reaching even the 6th decimal, and consequently the respected differences of the variances between the two approaches, it still remains an interesting pattern. This is maybe reasonable and expected, since in the first approach we build several models from different partitions of the data whilst in the second we use the same imputation model each time to fill in the missing values as well as the same model to derive the the predictions for each subject. Wilcoxon Signed-Ranks tests (Table 4.4.3) showed that indeed the median of the differences between the two algorithms is not zero in all settings that we explored.
Table 4.4.1: Mean variances for the 2 approaches - Models without Lvdis

For each subject we calculate the variance of the M predictions per approach. Here, the mean of this vector with the variances per approach is provided. This is done for all the 3 values of M that we examine, as well as for the 3 time points of interest. These results are based on the model without Lvdis.

<table>
<thead>
<tr>
<th>Time-point</th>
<th>M=5</th>
<th>Approach 1</th>
<th>Mean(variance)</th>
<th>Approach 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>t=12 months</td>
<td>2.54e-05</td>
<td>6.68e-06</td>
<td></td>
<td></td>
</tr>
<tr>
<td>t=60 months</td>
<td>0.00019</td>
<td>6.96e-05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>t=84 months</td>
<td>0.00025</td>
<td>9.34e-05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M=100</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t=12 months</td>
<td>3.06e-05</td>
<td>1.04-05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>t=60 months</td>
<td>0.00022</td>
<td>0.00010</td>
<td></td>
<td></td>
</tr>
<tr>
<td>t=84 months</td>
<td>0.00027</td>
<td>0.00013</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M=200</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t=12 months</td>
<td>3.11e-05</td>
<td>8.93e-06</td>
<td></td>
<td></td>
</tr>
<tr>
<td>t=60 months</td>
<td>0.00022</td>
<td>8.79e-05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>t=84 months</td>
<td>0.00028</td>
<td>0.00011</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.4.2: Mean variances for the 2 approaches - Models with Lvdis

For each subject we calculate the variance of the M predictions per approach. Here, the mean of this vector with the variances per approach is provided. This is done for all the 3 values of M that we examine, as well as for the 3 time points of interest. These results are based on the full model (with Lvdis).

<table>
<thead>
<tr>
<th>Time-point</th>
<th>M=5</th>
<th>Approach 1</th>
<th>Mean(variance)</th>
<th>Approach 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>t=12 months</td>
<td>0.00015</td>
<td>0.00012</td>
<td></td>
<td></td>
</tr>
<tr>
<td>t=60 months</td>
<td>0.0012</td>
<td>0.00094</td>
<td></td>
<td></td>
</tr>
<tr>
<td>t=84 months</td>
<td>0.0015</td>
<td>0.0011</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M=100</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t=12 months</td>
<td>0.00019</td>
<td>0.00015</td>
<td></td>
<td></td>
</tr>
<tr>
<td>t=60 months</td>
<td>0.00145</td>
<td>0.00119</td>
<td></td>
<td></td>
</tr>
<tr>
<td>t=84 months</td>
<td>0.00178</td>
<td>0.00148</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M=200</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t=12 months</td>
<td>0.00019</td>
<td>0.00015</td>
<td></td>
<td></td>
</tr>
<tr>
<td>t=60 months</td>
<td>0.00146</td>
<td>0.00115</td>
<td></td>
<td></td>
</tr>
<tr>
<td>t=84 months</td>
<td>0.00180</td>
<td>0.00143</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
For each value of $M$ and each time-point of interest we perform a Wilcoxon Signed-Ranks test for the difference of the variances between the two approaches. Third column shows the results for the model without Lvdias predictor and fourth column the results for the full model respectively. The p-values for all these statistics are less than 2.2e-16.

<table>
<thead>
<tr>
<th>Time-point</th>
<th>Without Lvdias</th>
<th>V statistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>M=5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>t=12 months</td>
<td>539170</td>
<td>401490</td>
</tr>
<tr>
<td>t=60 months</td>
<td>537340</td>
<td>398680</td>
</tr>
<tr>
<td>t=84 months</td>
<td>536440</td>
<td>397990</td>
</tr>
<tr>
<td>M=100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>t=12 months</td>
<td>544770</td>
<td>448910,</td>
</tr>
<tr>
<td>t=60 months</td>
<td>543170</td>
<td>444110</td>
</tr>
<tr>
<td>t=84 months</td>
<td>542950</td>
<td>441770</td>
</tr>
<tr>
<td>M=200</td>
<td></td>
<td></td>
</tr>
<tr>
<td>t=12 months</td>
<td>548750</td>
<td>489650</td>
</tr>
<tr>
<td>t=60 months</td>
<td>547700</td>
<td>495990</td>
</tr>
<tr>
<td>t=84 months</td>
<td>549290</td>
<td>495250</td>
</tr>
</tbody>
</table>

### 4.5 Model for approach 2

As a final step in this analysis, we provide a model derived from approach 2 in Table 4.5.1. We see from Table 4.5.1 that the majority of the predictors are statistically significant. $Hs$ seems to have the less impact on the survival outcome. Moreover, $Age$, $Qrs$ and $Af$ are also non significant in this model, based on a 5% significance level. Regarding $Gender$ we see that men have more than 50% higher risk of dying at any time during the monitored period than women. As expected, patients in higher $NYHA$ class are in worse position and it seems that the risk increases rapidly for someone in class 3 or 4. Patients diagnosed with diabetes have a hazard rate 1.5 of that for those who do not suffer from the disease, while those with a Restrictive value of Lvdias have 40% greater risk of dying at any individual time. On the other hand now, $Egfr$, $Qrs$, $Lvef$ and $Hb$ are protective against death in heart failure patients, with higher values denoting better prognosis. The same is true also for $Hs$, where patients who underwent a ventricular junction ablation before CRT implantation are in lowest risk, and finally the presence of $Lbbb$ acts in a protective manner as well.
### Table 4.5.1: Coefficients and Hazard ratios for a model derived from approach 2. The column \textit{fmi} contains the fraction of missing information as defined by [Rubin, 1987], while the column \textit{lambda} is the proportion of the total variance that is attributable to the variable ($\lambda = (B + B/m)/T$) 

| Variable       | coef | exp(coef) | se   | t    | Pr(>|t|) | fmi | lambda |
|----------------|------|-----------|------|------|----------|-----|--------|
| Age            | 0.007| 1.007     | 0.006| 1.159| 0.246    | 0.019| 0.019  |
| GenderMan      | 0.435| 1.54      | 0.125| 3.481| 0.001    | 0.016| 0.016  |
| HisTrue        | -0.106| 0.90     | 0.245| -0.431| 0.666    | 0.028| 0.028  |
| EtIschemisch   | 0.287| 1.33      | 0.104| 2.769| 0.006    | 0.018| 0.018  |
| Nyha3          | 0.370| 1.44      | 0.138| 2.684| 0.007    | 0.013| 0.013  |
| Nyha4          | 0.827| 2.28      | 0.189| 4.387| 0.000    | 0.017| 0.017  |
| DmDiabetes     | 0.423| 1.53      | 0.109| 3.869| 0.000    | 0.014| 0.014  |
| Egfr           | -0.017| 0.98     | 0.002| -7.467| 0.000    | 0.024| 0.024  |
| LvdiasRestrictive| 0.335| 1.40    | 0.132| 2.540| 0.011    | 0.455| 0.454  |
| Mr1            | 0.262| 1.30      | 0.110| 2.377| 0.017    | 0.041| 0.041  |
| Hb             | -0.091| 0.91     | 0.049| -1.845| 0.065    | 0.034| 0.034  |
| Lvef           | -0.020| 0.98     | 0.006| -3.217| 0.001    | 0.067| 0.067  |
| Qrs            | -0.002| 0.99     | 0.002| -0.817| 0.414    | 0.024| 0.024  |
| Lbbb1          | -0.191| 0.82     | 0.098| -1.940| 0.052    | 0.023| 0.023  |
| AfTrue         | 0.133| 1.14      | 0.126| 1.057| 0.291    | 0.040| 0.040  |
Chapter 5

Discussion

5.1 Conclusion

The present thesis was concerned with the combined application of multiple imputation and cross-validation, in the process of building a Cox prognostic survival model. More precisely, two different approaches were investigated, and the objective was to compare them in order to assess if they are different and if one is superior to the other. The first method was based on the mean of the predictions while the second on the prediction of the mean. In other words, in the first we built several models from which we extracted the predictions for each subject in the dataset and finally, we took their mean. On the other hand, in the second approach we firstly pooled the several models using Rubin’ rules and then we got the predictions for the subjects based on that pooled-final model.

For this thesis, we used a dataset from the department of Cardiology of Leiden University Medical Center, LUMC, Leiden, the Netherlands. It concerns an ongoing research for heart failure patients underwent a CRT implantation. CRT(Cardiac Resynchronization Therapy) is a well-known treatment for those patients that are resistant to drugs and that project aims to build a multi-parametric risk score for individualized survival estimation. The relevant paper has already been submitted [Höke et al., 2017]. The main issue in this particular dataset and where the idea for this thesis stems from, is the abundance of missing values in one of the predictors, namely the left ventricular diastolic dysfunction (Lvdias). Clinically, it is considered as a very significant factor for the diagnosis as well as the prognosis of heart failure patients, and this has to be taken into account when building a prediction model for the survival estimation of those patients.

The results, as presented in Chapter 4, did not show a significant difference between the two methods. However, we observed that as we increase the number of imputations(M) the predictions are getting much closer to each other. This was observable in Figures 4.3.1 - 4.3.9 in Section 4.3 where for a small value of M, such as 5, there was more disagreement between the 2 approaches than for a higher value of M, such as 200. Finally, both of the procedures managed to compensate for the missingness in the variable Lvdias, by producing models that have greater predictive performance than the models without Lvdias as predictor, which is based on the Brier score, as presented in Tables 4.3.3 - 4.3.5. Finally, we ran a Wilcoxon signed-ranks test for the variances produced by the two approaches, and the results which presented in Table 4.4.3 showed that the first approach produced slightly higher variances for all the different values of M that we investigated. As a last step, we provided a model derived from approach 2 in Table 4.5.1, while as was made clear through
out the report the first approach does not have as an end-product a single model, and for that reason none was provided here.

From these results, we might conclude that the best approach could be justified also from the usability of each method in the daily practice, since they seem to have similar predictive performance.

5.2 Future Considerations

This thesis set the ground for further investigation on the topic. There are some issues that could be explored further and future studies should also consider them.

First of all, in building our models using the dataset from LUMC we used a pre-specified set of predictors that came up from previous analyses and considered to be significant for the outcome. Moreover we did not apply any kind of variable selection in either of the procedures. This had also an indirect effect in this case of Cox model, where the assumption of proportional hazards was not dealt with, even that the associated test showed that there is a violation. Instead, the only action we took here was the application of an administrative censorship at some time-point \( t = 84 \text{ months} \), which was justified from the objective of the clinical study as well as from the proportional hazards test. Moreover, we used as a maximum value \( M = 200 \), and in the future higher values could be checked, and the same holds for the number of folds for the cross-validation, where we used \( k = 25 \) and different values could be examined. Finally, we used an imputation model that included all the variables of interest plus the Nelson-Aalen estimator of the cumulative hazard instead of the survival time. Maybe a further exploration concerning the imputation model could give different results. For instance, as mentioned in Section 4.2 variable Lvdias (left ventricular diastolic dysfunction) might be better not to be used in the imputation model.

Another important issue that we did not investigate in detail, is that of the variance that each of the procedures gives. A possible alternative to what we did, would be to extract the variance of the baseline hazard and use it in some way in order to calculate the variances of the survival probabilities.

Lastly, it would be of great interest to see how these two procedures behave in a different dataset. One way to do that would be to run simulations for different scenarios and apply these two algorithms to check if they give similar results.
Bibliography


Appendices
Appendix A

Additional Material
APPENDIX A. ADDITIONAL MATERIAL
Figure A.0.1: Schoenfeld residuals plots for the variables in the model

**log-minus-log Plot for gender**

![log-minus-log Plot for gender](image)

- **Women**
- **Men**

Time, months
Figure A.0.2: log-minus-log plots for the categorical variables in the model
Appendix B

R code

# Code for approach1

rm(list = ls())

load("Rdata.Rda") # The dataset we will use

# Load some necessary packages

library(mice)
library(caret)
library(survival)
library(lattice)
library(timeROC)
library(dynpred)
library(pec)
library(gtools)

####

m = M # The number of M. That is, the number of repetitions/imputations we will run the algorithm

# Set up some matrices and lists to store the results from each iteration

all.coef <- vector("list", m)
coef1 <- data.frame(matrix(nrow = 14, ncol = 25))
all.coef.lvd <- vector("list", m)
coef1_lvd <- data.frame(matrix(nrow = 15, ncol = 25))

PrI_aft <- data.frame()
PrI_lvd <- data.frame()
PI_after <- data.frame(matrix(nrow = 1053, ncol = m))
PI_lvd <- data.frame(matrix(nrow = 1053, ncol = m))

final_preds_aft12 <- data.frame(matrix(nrow = 1053, ncol = m))
final_preds_aft60 <- data.frame(matrix(nrow = 1053, ncol = m))
final_preds_aft84 <- data.frame(matrix(nrow = 1053, ncol = m))
final_preds_lvd12 <- data.frame(matrix(nrow = 1053, ncol = m))
final_preds_lvd60 <- data.frame(matrix(nrow = 1053, ncol = m))
final_preds_lvd84 <- data.frame(matrix(nrow = 1053, ncol = m))

# Apply an administrative censorship at t=84 months
for (i in 1:dim(data)[1]) {
    if (data$Stime[i] > 84) {
        data$Stime[i] <- 84
        if (data$Censor[i] == 'Dead') {
            data$Censor[i] <- 'Alive'
        }
    }
}

n <- nrow(data)
summary(data) # just a check that everything was right

for (i in 1:m) {
    # The number of times we will perform cross-validation.
    # Make a random partition of the data in 25 folds
    folds <- split(sample(n, n, replace=FALSE), as.factor(1:25))

    # Again some matrices, which we want to be updated after each iteration
    # of a 25-fold cross-validation
    pred.r12_afterdf <- data.frame()
pred.r60_afterdf <- data.frame()
pred.r84_afterdf <- data.frame()
pred.r12_lvddf <- data.frame()
APPENDIX B. R CODE

```r
pred.r60.lvddf <- data.frame()
pred.r84.lvddf <- data.frame()

for( j in 1:25){  # the 25-fold cross-validation

my.data <- data  # Make a copy of the original data to keep the latter unaffected
id = 1:1053      # A helper vector for the procedure

# create a status indicator in a numeric type
my.data$Censor2 <- as.numeric(my.data$Censor)-1

# compute the Nelson-Aalen estimator for the cumulative hazard, using only the training set
haz1 <- nelsonaalen(my.data[-folds[[j]],], Stime, Censor2)

# ...and add it to the dataset
my.data$haz <- NA
my.data$haz[-folds[[j]]] <- haz1

# Remove the outcome in the left-out fold....
my.data$Stime[folds[[j]]] <- NA

# Do that for the whole outcome (time + status)
my.data$Censor[folds[[j]]] <- NA

# Also for the helper variable we made earlier
my.data$Censor2[folds[[j]]] <- NA

# Firstly, we run an imputation with zero iterations just to create the mids object.
# This is because we want afterwards to redefine the predictor matrix, by removing the # time variable as a predictor in the imputation models and use instead only # the cumulative hazard.

impl <- mice( my.data, m = 1, maxit = 0 )
pred <- impl$predictorMatrix
pred[, c("Stime") ] <- 0

# And finally, run an imputation with only m=1 MI dataset.
```

77
imp1 <- mice( my_data, m = 1, maxit = 50, pred = pred )

# Extract the imputed dataset
completed <- complete( imp1, 1 )

# We are going to calculate the prognostic indices manually. In order to do that,
# will run a cox model to the data but only for getting the X matrix from it,
# which then we will use to build the desired (final) models.

#... Model without Lvdias predictor...
tester_Aft <- coxph( Surv( Stime, Censor == "Dead" ) ~ Age + Gender + His + Et + Nyha + Dm + Egfr + Mr + Hb + Lvef + Qrs + Lbbb + Af,
data = completed, x = T, y = T, model = T )
baseda <- data.frame(cbind(tester_Aft$x, tester_Aft$y))

# We run the cox model only to the training set!
FinalMdl_aft <- coxph( Surv( time, status == 1 ) ~ Age + Gender2 + His2 + Et2 + Nyha2 + Nyha3 + Dm2 + Egfr + Mr2 + Hb + Lvef + Qrs + Lbbb2 + Af2,
data = baseda, subset = !id %in% folds[[j]], x = T, y = T, model = T )

... The same for the model with Lvdias variable

tester_lvd <- coxph( Surv( Stime, Censor == "Dead" ) ~ Age + Gender + His + Et + Nyha + Dm + Egfr + Mr + Hb + Lvef + Qrs + Lvdias + Lbbb + Af,
data = completed, x = T, y = T, model = T )
baseda_lvd <- data.frame(cbind(tester_lvd$x, tester_lvd$y))

# We run the cox model only to the training set!
FinalMdl_lvd <- coxph( Surv( time, status == 1 ) ~ Age + Gender2 + His2 + Et2 + Nyha2 + Nyha3 + Dm2 + Egfr + Mr2 + Hb + Lvef + Qrs + Lvdias2 + Lbbb2 + Af2,
data = baseda_lvd, subset = !id %in% folds[[j]], x = T, y = T, model = T )

# Save the coefficients for both models
cof1[,j] <- FinalMdl_aft$coefficients
cof1_lvd[,j] <- FinalMdl_lvd$coefficients
# Now, we use the original data to bring back the time & status variables that we # removed in the beginning (form the test set), # in order to use them for the validation of the models.

```r
baseda[folds[[j]], 'status'] <- as.numeric(data$Censor[folds[[j]]]) - 1
baseda[folds[[j]], 'time'] <- data$Stime[folds[[j]]]
```

# Next, we use the survfit function for getting the baseline hazard, for a # hypothetical subject with all covariates equal to zero # To do that, we create that subject and pass it as newdata argument in the function.

```r
check_af <- baseda[1,]
check_lvd <- baseda_lvd[1,]
check_af[1,] <- 0
check_lvd[1,] <- 0
```

```r
survfitcox <- survfit(FinalMdl_aft, newdata = check_af)
survfitcox_lvd <- survfit(FinalMdl_lvd, newdata = check_lvd)
```

# And then extract the cumulative hazard for the 2 models, along with the event–times

```r
Hazar <- cbind(survfitcox$cumhaz, survfitcox$time)
Hazar_lvd <- cbind(survfitcox_lvd$cumhaz, survfitcox_lvd$time)
```

# Further, we use the hazard to calculate the baseline survival, and extract the values for times t=12, 60 & 84 months.

```r
surv12_af <- min(exp(-Hazar[Hazar[,2] < 12,]][1]))
surv12_lvd <- min(exp(-Hazar_lvd[Hazar_lvd[,2] < 12,]][1]))
surv60_af <- min(exp(-Hazar[Hazar[,2] < 60,]][1]))
surv60_lvd <- min(exp(-Hazar_lvd[Hazar_lvd[,2] < 60,]][1]))
surv84_af <- min(exp(-Hazar[Hazar[,2] < 84,]][1]))
surv84_lvd <- min(exp(-Hazar_lvd[Hazar_lvd[,2] < 84,]][1]))
```

# Finally, in order to make predictions, we want the prognostic indices produced by the models, which will be done manually, # by multiplying the X matrix with the coefficients. However, we would like only the # predictions for the left–out fold (test set).
predid\_After \leftarrow \texttt{as.matrix}(\text{baseda}\{\text{folds}[[j]],!\text{names}(\text{baseda}) \in \text{c}('\text{time}', '\text{status}')\}) \ \%\% \ \texttt{coef1}[;,j]

predid\_lvd \leftarrow \texttt{as.matrix}(\text{baseda\_lvd}\{\text{folds}[[j]],!\text{names}(\text{baseda\_lvd}) \in \text{c}('\text{time}', '\text{status}')\}) \ \%\% \ \texttt{coef1\_lvd}[;,j]

\# ... save the PI's
PrI\_aft \leftarrow \texttt{rbind}(PrI\_aft, \ predid\_After)
PrI\_lvd \leftarrow \texttt{rbind}(PrI\_lvd, \ predid\_lvd)

\# Finally, calculate the actual survival predictions

\# person-specific cross-validated survival at 12 months
s12\_aft \leftarrow \texttt{surv12\_af}^\left(\texttt{exp}(\ predid\_After)\right)
s12\_lvd \leftarrow \texttt{surv12\_lvd}^\left(\texttt{exp}(\ predid\_lvd)\right)

\# person-specific cross-validated survival at 60 months
s60\_aft \leftarrow \texttt{surv60\_af}^\left(\texttt{exp}(\ predid\_After)\right)
s60\_lvd \leftarrow \texttt{surv60\_lvd}^\left(\texttt{exp}(\ predid\_lvd)\right)

\# person-specific cross-validated survival at 84 months
s84\_aft \leftarrow \texttt{surv84\_af}^\left(\texttt{exp}(\ predid\_After)\right)
s84\_lvd \leftarrow \texttt{surv84\_lvd}^\left(\texttt{exp}(\ predid\_lvd)\right)

\#... and save them!

pred\_r12\_afterdf \leftarrow \texttt{rbind}(\text{pred\_r12\_afterdf}, s12\_aft)
pred\_r60\_afterdf \leftarrow \texttt{rbind}(\text{pred\_r60\_afterdf}, s60\_aft)
pred\_r84\_afterdf \leftarrow \texttt{rbind}(\text{pred\_r84\_afterdf}, s84\_aft)

pred\_r12\_lvd\_df \leftarrow \texttt{rbind}(\text{pred\_r12\_lvd\_df}, s12\_lvd)
pred\_r60\_lvd\_df \leftarrow \texttt{rbind}(\text{pred\_r60\_lvd\_df}, s60\_lvd)
pred\_r84\_lvd\_df \leftarrow \texttt{rbind}(\text{pred\_r84\_lvd\_df}, s84\_lvd)

\texttt{print}(j)

\}

\# After have completed the 25-fold cross-validation, we want to save these predictions
\# before moving to the next iteration.

\# The partition we did in the beginning, shuffled the subjects in the dataset.
\# Therefore, we have to sort them again based in their original order.
# The next lines are for this purpose, regarding the dataframes with the predictions.

test12_aft <- data.frame(pred.r12_afterdf[ order(as.numeric(row.names(pred.r12_afterdf)))]

test60_aft <- data.frame(pred.r60_afterdf[ order(as.numeric(row.names(pred.r60_afterdf)))]

test84_aft <- data.frame(pred.r84_afterdf[ order(as.numeric(row.names(pred.r84_afterdf)))]

test12_lvd <- data.frame(pred.r12_lvddf[ order(as.numeric(row.names(pred.r12_lvddf)))]

test60_lvd <- data.frame(pred.r60_lvddf[ order(as.numeric(row.names(pred.r60_lvddf)))]

test84_lvd <- data.frame(pred.r84_lvddf[ order(as.numeric(row.names(pred.r84_lvddf)))]

#... and save them!
final_preds_aft12[,i] <- test12_aft
final_preds_aft60[,i] <- test60_aft
final_preds_aft84[,i] <- test84_aft
final_preds_lvd12[,i] <- test12_lvd
final_preds_lvd60[,i] <- test60_lvd
final_preds_lvd84[,i] <- test84_lvd

#... for the PI's
PI_after[,i] <- data.frame(PrI_after[ order(as.numeric(row.names(PrI_after)))]
PI_lvd[,i] <- data.frame(PrI_lvd[ order(as.numeric(row.names(PrI_lvd)))]

#... and the coefficients
all.coef[[i]] <- coef1
all.coef_lvd[[i]] <- coef1_lvd

print(i)

# Label these dataframes...
colnames(final_preds_aft12) <- paste("CV", 1:m)
colnames(final_preds_aft60) <- paste("CV", 1:m)
colnames(final_preds_aft84) <- paste("CV", 1:m)
colnames(PI_after) <- paste("CV", 1:m)
row.names(all.coef) <- colnames(baseda)[1:14]
colnames(final_preds_lvd12) <- paste("CV", 1:m)
APPENDIX B. R CODE

colnames(final_preds_lvd60) <- paste("CV", 1:m)
colnames(final_preds_lvd84) <- paste("CV", 1:m)
colnames(PI_lvd) <- paste("CV", 1:m)
row.names(all.coef_lvd) <- colnames(baseda_lvd)[1:15]

# Average the predictions for each desired time point...

#...For the model without Lvdias predictor...
MeanPR12_aft <- apply(final_preds_aft12, 1, mean)
MeanPR60_aft <- apply(final_preds_aft60, 1, mean)
MeanPR84_aft <- apply(final_preds_aft84, 1, mean)

#...and create a new dataframe with these averaged predictions per patient.
Mean_Preds_Aft <- data.frame(AftImp12_1st = MeanPR12_aft, AftImp60_1st = MeanPR60_aft, AftImp84_1st = MeanPR84_aft)

...and the same for the model with Lvdias

MeanPR12_lvd <- apply(final_preds_lvd12, 1, mean)
MeanPR60_lvd <- apply(final_preds_lvd60, 1, mean)
MeanPR84_lvd <- apply(final_preds_lvd84, 1, mean)

Mean_Preds_lvd <- data.frame(WLvdias12_1st = MeanPR12_lvd, WLvdias60_1st = MeanPR60_lvd, WLvdias84_1st = MeanPR84_lvd)

# Finally, save all the important results from the algorithm...

write.table(Mean_Preds_Aft, "Mean_Predictions_aft1_200MI"")
write.table(Mean_Preds_lvd, "Mean_Predictions_lvd1_2_200MI"")

write.table(final_preds_aft12, "Pred12_After_200MI"")
write.table(final_preds_aft60, "Pred60_After_200MI"")
write.table(final_preds_aft84, "Pred84_After_200MI"")
write.table(PI_after, "Prognostic_Index_After_200MI"")
saveRDS(all.coef, file = "Models_coefficients_After_200MI"")

write.table(final_preds_lvd12, "Pred12_Lvd_200MI"")
write.table(final_preds_lvd60, "Pred60_Lvd_200MI"")
write.table(final_preds_lvd84, "Pred84_Lvd_200MI"")
write.table(PI_lvd, "Prognostic_Index_Lvd_200MI"")
saveRDS(all.coef_lvd, file = "Models_coefficients_Lvd_200MI")

82
# Code for approach2

```r
rm(list = ls())
```

# Load the dataset
```r
load("Rdata.Rda") # The dataset we will use
```

# Load some necessary packages
```r
library(mice)
library(caret)
library(survival)
library(lattice)
library(timeROC)
library(dynpred)
library(pec)
library(gtools)
```

# Set up some matrices and lists to store the results from each iteration
```r
m <- 1 # A helper variable for running the algorithm
MI <- M # The number of imputations.

# I am gonna store here the coefficients for the model without Lvdias
coops <- data.frame(matrix(nrow = 14, ncol = MI))

# ... and for the model with Lvdias
coops_lvd <- data.frame(matrix(nrow = 15, ncol = MI))

# Here we will save the Prognostic Indices for the models with and without Lvdias
Pr1_aft <- data.frame()
Pr1_lvd <- data.frame()

# And here the predictions for the 3 different time-points, for the models with and without Lvdias respectively
checker_aft12 <- vector("list", MI)
checker_lvd12 <- vector("list", MI)

checker_aft60 <- vector("list", MI)
checker_lvd60 <- vector("list", MI)

checker_aft84 <- vector("list", MI)
checker_lvd84 <- vector("list", MI)

# Here we will save the baseline hazards and their associated variances for the models with and without Lvdias respectively
Haz_mean_after <- vector("list", MI)
Haz_mean_lvd <- vector("list", MI)

Haz_Var_aft <- vector('list', MI)
Haz_Var_lvd <- vector('list', MI)

# Apply administrative censorship at t=84 months
for(i in 1:dim(data)[1]) {
  if (data$Stime[i] > 84) {
    data$Stime[i] <- 84
    if (data$Censor[i] == 'Dead') {
      data$Censor[i] <- 'Alive'
    }
  }
}

n <- nrow(data)  # the number of observations
summary(data)  # a check that we did it correctly

# Make a random partition of the data in 25 folds
folds <- split(sample(n, n, replace=FALSE), as.factor(1:25))
id <- 1:1053  # a vector to help us in the process
for(j in 1:25){  # The algorithm itself (over the 25 folds)
  # We make a copy of the dataset, in order to keep the original unaffected
  my_data <- data

  # create a status indicator in a numeric type
  my_data$Censor2 <- as.numeric(my_data$Censor) - 1

  # compute the Nelson-Aalen estimator for the cumulative hazard, using only the
  # training set.
  haz1 <- nelsonaalen(my_data[-folds[[j]],], Stime, Censor2)

  # ... and add it to the dataset
  my_data$haz <- NA
  my_data$haz[-folds[[j]]] <- haz1

  my_data$Stime[ folds[[j]] ] <- NA  # Remove the outcome in the left-out part....
  my_data$Censor[ folds[[j]] ] <- NA  # Do that for the whole outcome (time + status )
  my_data$Censor2[ folds[[j]] ] <- NA  # Also for the helper variable we made earlier

  # Firstly, we run an imputation with zero iterations just to create the mids object.
  # This is because we want afterwards to redefine the predictor matrix, by removing the
  # time variable as a predictor in the imputation models and use instead only
  # the cumulative hazard.
  impl1 <- mice( my_data, m = 1, maxit = 0 )

  pred <- impl1$predictorMatrix
  pred[ , c("Stime") ] <- 0

  # And finally, run a Multiple imputation with m = M MI datasets.
  impl1 <- mice( my_data, m = MI, maxit = 50, pred = pred )

  # Now, we have M datasets, and we want to train a Cox model to each of them,
  # but only to the training part as defined in the beginning of the loop.

  # Model without Lvdias
  fit <- with(impl1, coxph( Surv( Stime, Censor == "Dead" ) ~ Age + Gender + His + Et + Nyha + Dm + Egfr + Mr + Hb + Lvef + Qrs + Lbbb + Af,
APPENDIX B. R CODE

x = T, y = T, model = T, subset = !id %in% folds[[j]])

# Model with Lvdias

fit_lvd <- with(imp1, coxph(Surv(Stime, Censor == "Dead") ~ Age + Gender + His + Et + Nyha + Dm + Egfr + Mr + Hb + Lvef + Qrs + Lvdias + Lbbb + Af, 
x = T, y = T, model = T, subset = !id %in% folds[[j]]))

# Then, using Rubin’s rule, we pool the coefficients from the MI datasets and save them.

# Model without Lvdias

Final_coefs <- pool(fit)
coops[, j] <- summary(Final_coefs[, 1]

# Model with Lvdias

Final_coefs_lvd <- pool(fit_lvd)
coops_lvd[, j] <- summary(Final_coefs_lvd[, 1]

# There is a problem here. We want to pool also the baseline hazards from these m=MI MI datasets. The mice package does not seem to be able to do that, at least not very straightforward. Therefore, below we run an extra loop over the MI datasets to do that.

# Also, we will take advantage of this loop to save all the MI datasets in the form of

liss_aft <- vector("list", MI)
liss_lvd <- vector("list", MI)

# Loop over the MI datasets
for (analys in 1:MI) {

completed <- complete(imp1, analys) # extract 1 each time

# We are going to calculate the prognostic indices manually. In order to do that, we will run a model to the data but only for getting the X matrix from it, which then we will use to build the desired (final) model.

# Model without Lvdias
base_model_Aft <- coxph(Surv(Stime, Censor=="Dead") ~ Age + Gender + His + Et + Nyha + Dm + Egfr + Mr + Hb + Lvef + Qrs + Lbbb + Af, model=TRUE, x=TRUE, y=TRUE,
data = completed)

basedata_aft <- data.frame(cbind(base_model_Aft$x, base_model_Aft$y))

# we use here as a subset only the training set.
Final_model_aft <- coxph(Surv(time, status==1) ~ Age + Gender + His + Et + Nyha +
Nyha3 + Dm2 + Egfr + Mr + Hb + Lvef + Qrs + Lbb2 + Af2, model=TRUE, x=TRUE, y=TRUE, data=basedata_aft, subset = !id %in%
folds[[j]])

liss_aft[[analys]] <- basedata_aft  # save the kth MI dataset

# Model with Lvdias
base_model_Lvd <- coxph(Surv(time, Censor==”Dead”) ~ Age + Gender + His + Et + Nyha +
Dm + Egfr + Mr + Hb + Lvef + Qrs + Lvdias + Lbb + Af, model=TRUE, x=TRUE, y=TRUE, data=completed)

basedata_lvd <- data.frame(cbind(base_model_Lvd$x, base_model_Lvd$y))

# Again only to the training set
Final_model_lvd <- coxph(Surv(time, status==1) ~ Age + Gender + His + Et + Nyha +
Nyha3 + Dm2 + Egfr + Mr + Hb + Lvef + Qrs + Lvdias + Lbb + Af, model=TRUE, x=TRUE, y=TRUE, data=basedata_lvd, subset =
!id %in% folds[[j]])

liss_lvd[[analys]] <- basedata_lvd  # save the kth MI dataset

# Next, we use the survfit function for getting the baseline hazard, for a
hypothetical subject with all covariates equal to zero
# To do that, we create that subject and pass it as newdata argument in the function.

check_af <- basedata_aft[1,]
check_lvd <- basedata_lvd[1,]
check_af[1,] <- 0
check_lvd[1,] <- 0

survfitcox_after <- survfit(Final_model_aft, newdata = check_af)
survfitcox_lvd <- survfit(Final_model_lvd, newdata = check_lvd)

# And then extract the cumulative hazard for the 2 models
Haz_tmp_after <- survfitcox_after$surv
Haz_tmp_lvd <- survfitcox_lvd$surv

# Finally, we save these hazards along with their variance.
if (analys == 1){
  Haz_mat_after <- Haz_tmp_after
  Haz_mat_lvd <- Haz_tmp_lvd

  var_Haz_gM_after <- survfitcox_after$std.err^2
  var_Haz_gM_lvd <- survfitcox_lvd$std.err^2

} else {
  Haz_mat_after <- cbind(Haz_mat_after, Haz_tmp_after)
  Haz_mat_lvd <- cbind(Haz_mat_lvd, Haz_tmp_lvd)

  var_Haz_gM_after <- cbind(var_Haz_gM_after, (survfitcox_after$std.err)^2)
  var_Haz_gM_lvd <- cbind(var_Haz_gM_lvd, (survfitcox_lvd$std.err)^2)
}

# Then, we average these hazards and the variances, along with the event-times.
Haz_mean_after[[j]] <- cbind(apply(Haz_mat_after, 1, mean), survfitcox_after$time)
Haz_mean_lvd[[j]] <- cbind(apply(Haz_mat_lvd, 1, mean), survfitcox_lvd$time)

varHaz_mean_after <- apply(var_Haz_gM_after, 1, mean)
varHaz_mean_lvd <- apply(var_Haz_gM_lvd, 1, mean)

# For the variance, using Rubin's rules step-by-step....
helpft <- function(x){
  out <- sum((x - mean(x))^2)
  out
}

haztmp_aft <- apply(Haz_mean_after[[j]], 1, helpft)
helptmp_aft <- cbind(varHaz_mean_after, haztmp_aft)

haztmp_lvd <- apply(Haz_mean_lvd[[j]], 1, helpft)
helptmp_lvd <- cbind(varHaz_mean_lvd, haztmp_lvd)

poolvarHazfunc <- function(x){
out <- x[1] + (1 + 1/MI) * (1/(MI-1)) * x[2]  # Rubin's rule formula
out
}

Haz_Var_aft[[j]] <- apply(helptmp_aft, 1, poolvarHazfunc)
Haz_Var_lvd[[j]] <- apply(helptmp_lvd, 1, poolvarHazfunc)

# Further, we use the mean hazard to calculate the baseline survival function, and extract the values for times t=12, 60 & 84 months.
surv12_af <- min(exp(-Haz_mean_after[[j]][Haz_mean_after[[j]][,2] < 12,]][,1])
surv12_lvd <- min(exp(-Haz_mean_lvd[[j]][Haz_mean_lvd[[j]][,2] < 12,]][,1])
surv60_af <- min(exp(-Haz_mean_after[[j]][Haz_mean_after[[j]][,2] < 60,]][,1])
surv60_lvd <- min(exp(-Haz_mean_lvd[[j]][Haz_mean_lvd[[j]][,2] < 60,]][,1])
surv84_af <- min(exp(-Haz_mean_after[[j]][Haz_mean_after[[j]][,2] < 84,]][,1])
surv84_lvd <- min(exp(-Haz_mean_lvd[[j]][Haz_mean_lvd[[j]][,2] < 84,]][,1])

# Finally, in order to make predictions, we want the prognostic indices produced by the model, which will be done manually, by multiplying the design matrix with the coefficients. However, we would like only the predictions for the left-out fold (test set). Obviously, this will be done in all MI datasets and this is the reason for the next loop.
predid_After <- data.frame(matrix(nrow = length(folds[[j]]), ncol = 2))
predid_lvd <- data.frame(matrix(nrow = length(folds[[j]]), ncol = 2))
for (i in 1:MI){
  predid_After[,i] <- as.matrix(liss_aft[i][folds[[j]],!names(basedata_aft)%in%c('time', 'status')]) %*% coefs[[j]]
predid_lvd[,i] <- as.matrix(liss_lvd[i][folds[[j]],!names(basedata_lvd)%in%c('time', 'status')]) %*% coefs_lvd[,j]
}
row.names(predid_After) <- folds[[j]]
row.names(predid_lvd) <- folds[[j]]

# Eventually, we calculate the actual predictions for t=12, 60 & 84 months by using the formula.
# person-specific cross-validated survival at t=12 months
APPENDIX B. R CODE

```r
s12_aft <- surv12_aft * (exp(as.matrix(predid_After[, i])))
s12_lvd <- surv12_lvd * (exp(as.matrix(predid_lvd[, i])))

# person-specific cross-validated survival at t=60months
s60_aft <- surv60_aft * (exp(as.matrix(predid_After[, i])))
s60_lvd <- surv60_lvd * (exp(as.matrix(predid_lvd[, i])))

# person-specific cross-validated survival at t=84months
s84_aft <- surv84_aft * (exp(as.matrix(predid_After[, i])))
s84_lvd <- surv84_lvd * (exp(as.matrix(predid_lvd[, i])))

# And save them......
checker_aft12[[i]] <- rbind(checker_aft12[[i]], s12_aft)
checker_lvd12[[i]] <- rbind(checker_lvd12[[i]], s12_lvd)
checker_aft60[[i]] <- rbind(checker_aft60[[i]], s60_aft)
checker_lvd60[[i]] <- rbind(checker_lvd60[[i]], s60_lvd)
checker_aft84[[i]] <- rbind(checker_aft84[[i]], s84_aft)
checker_lvd84[[i]] <- rbind(checker_lvd84[[i]], s84_lvd)

# Next, we take the mean of the predictions from the MI datasets.
predid_After12 <- as.matrix(apply(predid_After, 1, mean))
predid_lvd12 <- as.matrix(apply(predid_lvd, 1, mean))

# And save these Prognostic Indices......
PrI_aft <- rbind(PrI_aft, as.matrix(predid_After12))
PrI_lvd <- rbind(PrI_lvd, as.matrix(predid_lvd12))

print(j)

# The base algorithm is over. Now we will save the important results.
# First of all, we transform the lists in dataframes to make it easier.
prafted12 <- do.call(cbind.data.frame, checker_aft12)
prlvd12 <- do.call(cbind.data.frame, checker_lvd12)
```

90
APPENDIX B. R CODE

pr_aft_60 <- do.call(cbind.data.frame, checker_aft60)
prlvd_60 <- do.call(cbind.data.frame, checker_lvd60)

pr_aft_84 <- do.call(cbind.data.frame, checker_aft84)
prlvd_84 <- do.call(cbind.data.frame, checker_lvd84)

# As it might be obvious, the dataframes with the predictions, as well as the
# Prognostic indices, are not ordered. They are shuffled according to the splitting we
did in the beginning. Therefore, below we sort them based on their ID.

test12_aft <- data.frame(pr_aft_12[order(as.numeric(row.names(pr_aft_12)))]

test60_aft <- data.frame(pr_aft_60[order(as.numeric(row.names(pr_aft_60)))]

test84_aft <- data.frame(pr_aft_84[order(as.numeric(row.names(pr_aft_84)))]

test12_l <- data.frame(prlvd_12[order(as.numeric(row.names(prlvd_12)))]

test60_l <- data.frame(prlvd_60[order(as.numeric(row.names(prlvd_60)))]

test84_l <- data.frame(prlvd_84[order(as.numeric(row.names(prlvd_84)))]

PrIn_aft <- data.frame(PrI_aft[order(as.numeric(row.names(PrI_aft)))]

PrIn_lvd <- data.frame(PrI_lvd[order(as.numeric(row.names(PrI_lvd)))]

# Label them....
colnames(test12_aft) <- "SurvProb12_aft"
colnames(test60_aft) <- "SurvProb60_aft"
colnames(test84_aft) <- "SurvProb84_aft"
colnames(test12_l) <- "SurvProb12_lvd"
colnames(test60_l) <- "SurvProb60_lvd"
colnames(test84_l) <- "SurvProb84_lvd"
colnames(PrIn_aft) <- "PrognosticIndex_after"
colnames(PrIn_lvd) <- "PrognosticIndex_lvd"

#... and save them!
write.table(test12_aft, "Preds12_after_2nd_200MI")
write.table(test60_aft, "Preds60_after_2nd_200MI")
write.table(test84_aft, "Preds84_after_2nd_200MI")
write.table(test12_l, "Preds12_lvd_2nd_200MI")
write.table(test60_l, "Preds60_lvd_2nd_200MI")
write.table(test84_l, "Preds84_lvd_2nd_200MI")
write.table(PrIn_af, "PrognIndex_after_2nd_200MI")
write.table(PrIn_lvd, "PrognIndex_lvd_2nd_200MI")

# ...and also the dataframes with the coefficients
write.table(coefs, "Coefficients_after_2nd_200MI")
write.table(coefs_lvd, "Coefficients_lvd_2nd_200MI")

#...as well as the lists with the hazards and their variances!
saveRDS(Haz_mean_after, file = "Meanhazard_after_200MI")
saveRDS(Haz_mean_lvd, file = "Meanhazard_lvd_200MI")
saveRDS(Haz_Var_aft, file = "Varhazard_after_200MI")
saveRDS(Haz_Var_lvd, file = "Varhazard_lvd_200MI")