Predicting recurrence of Thrombosis: A comparison of different methods to build prediction models

Chahid Mohamed, A.

Predicting recurrence of thrombosis:

a comparison of different methods to build prediction models

Author: Abdelhak Chahid Mohamed

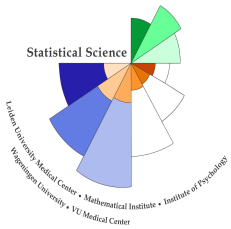
First supervisor: Prof. Dr S. le Cessie (Saskia)

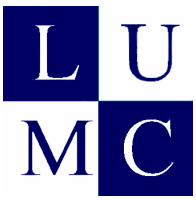
Second supervisor: Prof. Dr. J. Goeman (Jelle)

MASTER THESIS

Defended on April 16, 2019

Specialization: Statistical Science





STATISTICAL SCIENCE
FOR THE LIFE AND BEHAVIOURAL SCIENCES

Contents

1 Medical background **11**
- 1.1 Thrombus formation . . . 11
- 1.2 Venous thromboembolism (VTE) . . . 11
- 1.3 Risk factors . . . 12
- 1.4 Recurrent Thrombosis . . . 12
- 1.5 Anticoagulant guidelines . . . 12

2 The MEGA study on thrombosis **13**
- 2.1 Patients population . . . 13
- 2.2 Data description . . . 14
  - 2.2.1 Correlations between variables . . . 16

3 Survival analysis background **18**
- 3.1 Introduction . . . 18
- 3.2 Time to event data and censoring . . . 18
- 3.3 Basic survival analysis functions . . . 19
  - 3.3.1 Survival function . . . 19
  - 3.3.2 Hazard rate function . . . 20
- 3.4 Cox Proportional Hazards model . . . 21
- 3.5 The partial likelihood . . . 22
- 3.6 Cox Model assumptions . . . 24
- 3.7 Measure of discrimination in survival analysis . . . 24

4 Model selection methods **26**
- 4.1 Introduction . . . 26
- 4.2 Backward selection . . . 26
  - 4.2.1 Model selection using p-value procedure . . . 27
  - 4.2.2 Limitations . . . 27
- 4.3 Shrinkage method . . . 29
  - 4.3.1 Introduction . . . 29
  - 4.3.2 LASSO for linear regression model . . . 30
  - 4.3.3 Geometric interpretation for linear regression model . . . 31
  - 4.3.4 Lasso for Cox regression model . . . 32
  - 4.3.5 Lasso limitations . . . 36
- 4.4 Percentile lasso . . . 38
  - 4.4.1 Introduction . . . 38
  - 4.4.2 Percentile lasso . . . 38
  - 4.4.3 The percentile lasso algorithm . . . 39

4.4.4 Percentile lasso as an alternative to ordinary lasso . . . 40
4.5 Closed Testing as a variable selection method . . . 42
4.5.1 Introduction . . . 42
4.5.2 Closed testing procedure . . . 42
4.5.3 Confidence set for true and false discoveries . . . 43
4.5.4 Application . . . 44
4.5.5 Limitations and Shortcuts . . . 46
4.6 Model validation: Internal . . . 48
4.6.1 Split-sample . . . 48
4.6.2 K-fold Cross-Validation . . . 48
4.6.3 Bootstrapping: . . . 48

5 Results **51**
5.1 Developing the model . . . 51
5.2 Selecting candidate predictor variables by backward selection . . . 51
5.2.1 Model A . . . 52
5.2.2 Model C . . . 52
5.2.3 Predictors of recurrent thromboembolism models by backward selection . 53
5.2.4 Check of the Proportional Hazards Assumption . . . 53
5.2.5 Predictive value of the different models . . . 54
5.3 Selection of the risk factors by lasso method . . . 55
5.3.1 Model A . . . 55
5.3.2 Model C . . . 56
5.3.3 Predictive value of the different models . . . 56
5.3.4 Predictors of recurrent thromboembolism models by lasso . . . 57
5.3.5 Check of the Proportional Hazards Assumption . . . 58
5.4 Closed testing . . . 59
5.4.1 Selected models . . . 72
5.4.2 Predictors of recurrent thromboembolism models . . . 72
5.4.3 Predictive value of the different models . . . 73
5.4.4 Check of the Proportional Hazards Assumption . . . 74
5.5 Predictive performance of the models . . . 75
5.6 Models Summary . . . 75
5.6.1 Nomogram for risk of recurrent VT . . . 76

6 Discussion **78**
6.1 Choice of model and conclusion . . . 81

Bibliography **82**

7 Supplement **87**
7.1 Lasso instability . . . 87
7.1.1 Ordinary lasso and percentile lasso . . . 87
7.2 Extra notes on the closed testing procedure . . . 88
7.2.1 Exploratory vs. Confirmatory research . . . 88
7.2.2 Mild, flexible and post-hoc . . . 88
7.2.3 Hypothesis testing . . . 89
7.3 Schoenfeld residuals and numerical test for PH assumption . . . 89
7.3.1 Diagnostics for the Cox model: Schoenfeld residuals . . . 89
7.3.2 PH assumptions test: closed testing models . . . 92

7.3.3 Quintiles of the prognostic score closed testing 92
7.4 Cross-validation in linear regression 93
7.5 Results of the 1-SE rule for lasso 95
7.5.1 Model C: 1-SE rule 95
7.5.2 Model C: performance at 1-SE rule 95
7.5.3 lasso in conjunction with percentile lasso, backward selection and closed testing procedure coefficients estimates 96

8 R-code syntax **98**
8.1 Data preprocessing 98
8.2 Descriptive statistics 100
8.3 Backward elimination 104
8.4 Performing backward selection 107
8.5 Percentile lasso 110
8.6 Performing lasso analyses in conjunction with percentile lasso 118
8.7 Closed testing analyses 124
8.8 Nomograms for survival analysis 135
8.9 Coefficients plots 137
8.10 Sensitivity of lasso to data changes 140

Abstract

Prediction models are of major importance for many fields, including medicine for decision making.
An example is predicting if a patient with a venous thrombosis is at risk of experiencing a second thrombosis. A venous thromboembolism (VTE) arises when a blood clot is formed inside a blood vessel and blocks blood flow. VTE is a life-threatening disease and is considered as the most common vascular disease after myocardial infarction and stroke.

A considerable number of statistical methods have been proposed for variable selection to construct prediction models. The specific aims of this thesis were to provide a comprehensive overview, compare and assess the performance of three popular variable selection methods: Backward Elimination, LASSO in conjunction with Percentile Lasso and Closed Testing . Additionally, we attempted to identify the relative and absolute strengths and limitations of these variables selection methods.

The methods were used to build prediction models for recurrence of VTE, for patients with first VTE, using the MEGA study, a large follow-up study on 4956 patients with a first thrombosis. Two different prediction models were investigated: 1. A prediction model using clinical, genetic and laboratory factors (model A) and 2. A prediction model using only clinical and easy to obtain genetic factors (model C).

The results show that Backward selection has the advantage of being simple, available in most statistical packages and is widely used model selection method. On the other side, its performance is likely to depend on the choice for the stopping rule, and it results in regression coefficients β's that are usually inflated.
Lasso is considered as one of the new and well-acknowledged variables selection methods, with an important advantage over the traditional model selection methods that is mostly manifested in high-dimensional data or in presence of high multicollinearity among variables. However, lasso has the disadvantage of model instability due to fold assignment during the cross-validation. Similarly, as backward elimination, lasso tends to select randomly just one variable from a set of highly correlated variables.
Further, percentile lasso has the advantage of model stability selection but was quite computationally demanding.
Lastly, the closed testing method has the advantage of generating a collection of minimal models that can fit as good as the full model, further it quantifies the uncertainty of the selected models by providing the confidence set for each selected model and has the ability to work with any choice of a local test. For all that, the closed testing procedure in its standard form has the disadvantage of unfeasible computation.

Application of these different methods to the MEGA study showed that, for building model A , lasso (in conjunction with percentile lasso) has the best discriminative performance with 10 variables , whereas the backward elimination performance was slightly lower with 11 variables. In addition, the closed testing procedure could not be applied here due to computer time constrains because of the large number of candidate predictors. Model A by lasso requires three laboratory factors (factor VIII, D-dimer, and VWF) for which stopping the anticoagulant treatment is needed for a correct interpretation of the values.

The performance of the obtained models was slightly lower when considering a model with only clinical and genetic factors. Backward elimination selected 8 variables, lasso 12 variables, and with closed testing different models were selected ranging between 6 to 10 variables . With closed testing, three different models with 6 variables were selected. Model C_6 (Surgery, plaster cast, pregnant, hormone, location VT and gender) has the desired parsimonious character with a diminutive difference in terms of corrected for optimism C statistics, and moreover requires no laboratory measurements.

Our study results suggest that closed testing is indeed a useful method, that can be implemented as a variable selection method. In addition, these results indicate that models (shortlist) proposed by closed testing have slightly lower corrected C-statistic, but their added value is mostly its parsimonious character and clinical utility .

Keywords: Recurrence Venous Thrombosis, Anticoagulant, Variable Selection, lasso, Percentile lasso, Backward Selection, Closed Testing, Survival Analysis, Cross Validation, Confidence Set. Bootstrap Internal Validation.

Acknowledgements

After an intensive period of eight months, at Leiden University Medical Center (LUMC), that has been a period of an intensive learning process for me, not just in the scientific research field, but also on a personal level. Today is the day to write these words of thanks as the finishing touch on my thesis.

First of all, I would like to express my deep appreciation to my main supervisor Prof. Dr. Saskia Le Cessie, for giving me the opportunity to work with her. Her teaching style and enthusiasm for the medical statistics made a big impression on me. Her office was always open whenever I ran into difficulty or had a question about my thesis. She consistently allowed this thesis to be my own work, but also guide me, often with big doses of patience, toward the right path whenever she thought I needed it.

I would also like to thank my second supervisor Prof. Dr. J.J Goeman for his assistance and enthusiastic guidance. Without his dedicated involvement throughout the process, this thesis would have had totally another form. I would like to thank you very much for your advice and for broadening my "statistical-multiple-testing" horizon.

A special thanks to my friend George Kantidakis for his time and new insights. It was a wonderful "statistical" time at Leiden university! Also I would like to extend my gratitude to Ronja Hard for the time you took correcting my typos.

Last but not least, I must express my very profound gratitude to my father Hadj M'barek and my stepmothers (Ayada and Thraithmas). This accomplishment would not have been possible without them. Thank you for all your support through the years and unconditional love.

Finally, there are many brothers, sisters, and friends who saw me the last few months, working hard and sometimes beyond the "threshold". I'm certain that I did not provide you with the attention and time you deserve. Hopefully, there will be much more balance in the next challenge.

Thank you very much, everyone!
Abdelhak Chahid.

Utrecht, March 13, 2019.

Introduction

Prediction models play an important role in many research fields, including epidemiology for decision making, especially after the switch from the "subjective" decision to evidence-based medicine [48]. Often the research question involves selecting the true predictors from a set of candidate variables to build a prediction model. The main aim of model selection is to reduce the set of candidate variables to a small set that can replace the full set and account almost for the same variance as is accounted for by the full set of variables. A considerable number of methods have been proposed for variable selection. In this thesis three variables selection methods, each having their strengths and limitations are investigated to build a prediction model for the recurrent VT.

Backward selection was used to build a prediction model for the recurrent VT(Timp [54]). This method is considered as one of the most widely used model selection methods, and is easily available in most statistical packages. Additionally, in our thesis, we will investigate the performance of two other variable selection methods i.e. lasso in conjunction with percentile lasso and closed testing procedure.
Lasso is considered as one of the new and well-acknowledged methods to select variables, especially in high-dimensional data where the number of observations is smaller than or close to a number of candidate variables (n<<p). Lasso possesses an important property that other regularized methods (like a ridge and elastic net) do not have: it allows for automatic variable selection by shrinking some of the coefficients all the way to zero. By doing so, the lasso is performing variables selection. Furthermore, percentile lasso is considered here as a technique to stabilize lasso model selection.
In addition, we will discuss the closed testing procedure in the context of variable selection methods. Commonly, this method is used to control the family-wise error rate (FWER) when several statistical tests are performed simultaneously. To the extent of our knowledge, there are not enough studies done on its application as a variable selection method. Therefore, in this thesis we investigate the possibility of its application and compare its performance to the aforementioned methods.

Modeling the relationship between the recurrent VT and a set of candidate variables is a challenging problem. This requires selecting a subset of possible candidate variables that are associated with recurrent VT, and which will provide accurate predictions of future observations. Venous thromboembolism (VTE) arises when a blood clot is formed inside a blood vessel and blocks blood flow. The VTE is a serious, frequent, potentially lethal, and life-threatening chronic disease that requires immediate medical attention. In general, there are two types of VTE, Deep Venous Thrombosis (DVT) and Pulmonary Embolism (PE). It is widely known that patients who suffered from the first venous thrombosis, are at high risk of developing recurrent venous thromboembolism [54]. The occurrence of vein thrombosis is classified into two classes

i.e. Provoked and Unprovoked (idiopathic) VTE. The provoked VTE is defined as the one that is likely caused by some major clinical risk factors e.g., major surgery, hospitalization, immobility, trauma, pregnancy, whereas the unprovoked VTE is defined as the occurrence of VTE in a patient with no antecedent (within 3 months) major clinical risk factor for VTE.

Commonly, anticoagulant drugs are considered the best treatment to prevent the recurrence of VT. However, the anticoagulant treatment duration in the current guidelines is based on the previous VTE classification i.e. provoked versus unprovoked[27] . These guidelines suggest that all patients with a provoked VTE should cease anticoagulant treatment after 3 months, whereas patients with an unprovoked VTE are recommended to continue the treatment for at least 3 months [27].
The vast majority of published studies on the assessment of the risk of recurrent VT have focused on provoked versus unprovoked VTE categories. However, there are some problems with this approach:

- There is no unequivocal definition of unprovoked event [54]
- The definition of provoked VTE has been prolonged over years to contain more risk factors, by including, for instance, body mass index > 30 kg/m^2, prolonged travel, lower extremity paralysis or paresis, inflammatory bowel disease, congestive heart failure, and renal impairment [42].
- This approach does not take into consideration the difference between patients in the same group [54].

In this thesis, a dataset from the MEGA study [54] was used. In order to assess the association between some risk factors and recurrence of venous thrombosis 3750 patients aged 18-70 years with the first episode of venous thrombosis were followed [54]. Our dataset contains 38 candidate variables, divided into three clusters of factors: clinical, genetic and laboratory factors. Despite that predictors for the first venous thrombotic episode are well-established [40], this knowledge cannot always be directly used to predict the recurrent events [50].

Having the aforementioned limitations in mind, we aim to build a prediction model for all patients with first VT without a distinction among patients VTE classes i.e. provoked versus unprovoked VTE. The aim is to build two models: model A which uses clinical, genetic and laboratory factors as predictors and model C which uses clinical and genetic factors. Variable selection is performed using the three different statistical methods. In addition, the specific aims of this thesis were to provide a comprehensive overview, compare and assess the performance of these three popular variable selection methods to build a prediction model. Similarly, we attempted to identify the relative and absolute strengths and limitations of these variables selection methods. Moreover, we provide a few solutions to some limitations of the investigated methods.
To the best of our knowledge, despite the many proposed methods for variables selection, direct comparisons among the investigated methods, either theoretical or experimental, are rare. This thesis is attempting to provide such comparison, and more importantly, our thesis offers a comprehensive and practically relevant discussion on theoretical aspects of each method. By doing so, we hope to enrich the discussion about variables selection methods in the statistical community, and ultimately to aid the practitioner in making an evidence-based choice of model selection, that is more suitable to answer his/her research question.

The outline of this thesis is as follows. Chapter 1 provides a general introduction to the required medical backgrounds of venous thrombosis VT. In chapter 2, we give several views on the data through descriptive statistics. In chapter 3, a theoretical outline of the important survival analysis functions is presented. In chapter 4, a theoretical detailed overview and a discussion of the strengths and limitations of all three variable selection methods are provided, additionally, we offer a few practical examples to clarify the theory. In chapter 5, results from the application of the three model selection methods on MEGA dataset are summarized. In chapter 6, we give a detailed discussion of the obtained results, and we suggest some solutions. Moreover, we highlight directions for future research and finally, we formulate our conclusion.

Chapter 1

Medical background

1.1 Thrombus formation

Hemostasis (hemo- meaning blood, and stasis meaning stopping) is the natural process which induce bleeding to stop, by forming a clot to prevent blood loss after vascular damage. This process involves coagulation, i.e. changing the blood state from a liquid to a gelatinous state [32].

Thrombus formation is a complex process, it depends on a delicate interplay between bleeding and clotting, and involves various factors (von Willebrand factor and coagulation factors), fluid components (the platelets) and cells (endothelium) all working together in a balanced way to heal wounds (blood clotting) and prevent blood loss (bleeding). When an imbalance in the hemostasis system occurs, a blood clot may be formed in a venous system, causing a thrombus.

1.2 Venous thromboembolism (VTE)

Venous thromboembolism (VTE) ascribe the blood clot that develops in a vein and is considered as the third most common leading vascular disease after myocardial infarction and stroke [37]. The incidence of any category of VTE is estimated to be around 0.1%-0.2% per year, this rate is estimated to be higher for elderly persons, i.e. around 5 in 1000 persons per year [54].

There are two different types of VTE, Deep Venous Thrombosis (DVT) and Pulmonary Embolism (PE). The Deep Venous Thrombosis (DVT), happens when a blood clot (thrombus) develops in a deep vein usually in a leg, whereas Pulmonary Embolism happens when a DVT clot snap off and moves to the lungs and then blocks the blood flow, hence becoming a life-threatening embolus [41]. PE happened to be present in around 30-40% of patients with VTE, whereas it is observed that the death occurs in 6% of DVT cases and 12% of PE cases within 1 month of diagnosis [59]. DVT in his turn can be categorized into two levels: Proximal and Distal. The distal DVT occurs below the knee in the deep veins of the calf, where the majority of thrombi usually starts, whereas the proximal DVT occurs above the knee.

VTE is a serious and life-threatening condition that requires immediate medical attention [8]. Therefore it is important to pay attention to some signs and symptoms of VTE and seek medical attention if they occur. Some signs and symptoms of DVT include: Swelling and pain, red or

discolored skin and feeling of warmth in the affected leg, and for PE the common symptoms include: fast heart rate, rapid breathing, chest pain, feeling dizzy, or coughing up blood.

1.3 Risk factors

The venous thromboembolism VTE can be caused by anything that prevents your blood from circulating or clotting normally. The most common triggers are injury to a vein, surgery, active cancer, immobilization such as after surgery or sitting for long periods of time, such as when driving or flying, hospitalization, obesity, and aging. Additionally, it is known that pregnant women and women who use hormones like oral contraceptives or estrogen for menopause symptoms have a higher risk of developing VTE [50],[59].

1.4 Recurrent Thrombosis

Patients who suffered from the first episode of VTE are at major risk for recurrent venous thromboembolism (VTE) [54]. It was noted that the risk of recurrence of VTE, as well as the treatment (anticoagulant) duration, differs among two important categories of VTE: i.e. provoked and unprovoked VTE (idiopathic) [54] [26]. The provoked VTE is defined as the one that is likely caused by transient major risk factors (e.g., major surgery, hospitalization, immobility, trauma, pregnancy) or persistent risk factors: inheritable thrombophilia's, chronic heart failure, and cancer. The unprovoked VTE is defined as the occurrence of VTE in a patient with no antecedent (within 3 months) major clinical risk factor for VTE.

1.5 Anticoagulant guidelines

Current guidelines that are commonly followed by practitioners for administration of anticoagulant drug, recommend that patients with an unprovoked VTE to be treated with anticoagulation for at least 3 months. According to these guidelines, the decision to continue the treatment beyond 3 months should be evaluated for each patient based on the balance between the risk of recurrence if treatment is stopped and the risk of bleeding during the anticoagulation [27]. Furthermore, these guidelines suggest that all patients with a provoked VTE should cease anticoagulant treatment after 3 months. In general, there is an agreement for general patients with venous thromboembolism, that the risk of recurrent VTE can be alleviated by anticoagulant, with a large effect in preventing recurrent VTE in the first period following the VTE event, whereas there was not such benefit if the anticoagulant treatment was extended [12].

Despite that predictors for the first venous thrombotic episode are well-established [40], this knowledge cannot always be directly used to predict the recurrent events [50]. Take for instance age which is strongly associated with first events, or the presence of genetic thrombophilia, however these predictors are proven to be weakly associated with recurrent venous thrombosis [54],[50]. Therefore, one needs a model that can predict well the recurrence of VTE.

Chapter 2

The MEGA study on thrombosis

2.1 Patients population

The MEGA study (Multiple Environmental and Genetic Assessment of risk factor for venous thrombosis) is a large case-control study, aimed at estimating the incidence of recurrent venous thrombosis and at identifying those risk factors (combination) that are associated with the risk of recurrence of venous thrombosis [30]. Patients that suffered from a first episode of venous thrombosis (deep vein thrombosis and/or pulmonary embolism) were enrolled to MEGA study from six anticoagulation clinics in the Netherlands from March 1999 until September 2004. The partners of these patients, as well as individuals collected via random digit dialing, were invited to participate as control subjects (subjects without a history of venous thrombosis) [54].

All participants of the MEGA study were invited for an interview, furthermore, a blood sample or a buccal swab was collected for each participant at least three months after the discontinuation of anticoagulant therapy. If patients did not stop anticoagulant therapy, one year after the first thrombosis, blood samples were drawn from these patients [54]. Information on common laboratory (e.g. D-dimer, factor VIII...), as well as clinical factors (e.g. Age, BMI, gender...), have been measured, in addition, information on three genetic factors (factor V Leiden and blood type) were collected as well (table 2.1). This study was approved by the ethics medical committee of the Leiden University Medical Center (LUMC), and all participants provided written informed consent [54].

There were 4956 patients in the MEGA study, aged 18-70 years with the first episode of venous thrombosis, deep venous thrombosis (DVT) or pulmonary embolism (PE). Cases of the MEGA study were further followed for recurrent venous thrombosis (MEGA follow-up study). Out of the 4956 included patients, 1206 were excluded from the MEGA follow-up study for the following reasons: 225 patients did not consent, of 715 patients their follow-up ended before or at the moment of discontinuation of anticoagulant treatment. In addition, 266 patients were diagnosed with cancer within five years before VT or data were missing with regard for cancer diagnosis [54]. A flowchart for the number of included and excluded patients as well as the reasons are illustrated in figure 2.1. The vital status of all patients was obtained between 2007 and 2009 from the central Dutch population register [53].
Furthermore, between June 2008 and July 2009 a short answer form concerning recurrent VT were sent by mail to all survivors and consenting individuals, and supplemented by telephone interviews [39]. In addition, all patients were asked to complete a second questionnaire on the

presence of risk factors for VT after their first thrombosis [54].

2.2 Data description

One main question intended to be tackled with our dataset (MEGA follow-up study data) is the assessment of risk factors for recurrence of venous thrombosis. The dataset contains detailed information about 38 variables, some are continuous (e.g. BMI, Age . . .) and others are categorical (e.g. gender, surgery..). These 38 variables are classified into three different categories, namely: laboratory, clinical and genetic variables. The laboratory category consists of 21 variables, of which some of them are log transformed. The genetic category consists of 2 variables, and the clinical category contains 15 variables. Descriptive statistics of these variables are summarized in table 2.1.

The main question that this thesis intended to answer involves time until recurrent thrombosis. Therefore we need survival analysis methods, for this purpose we define the "Time to event" as the time between the instant of cessation of anticoagulant treatment, and the date of a recurrence or, in its absence, the date of returning the answers to the second questionnaire [53]. The "Event" is defined to be the "recurrence" of venous thrombosis. For a rigorous definition of recurrence, the interested reader is referred to section 2 from [54]

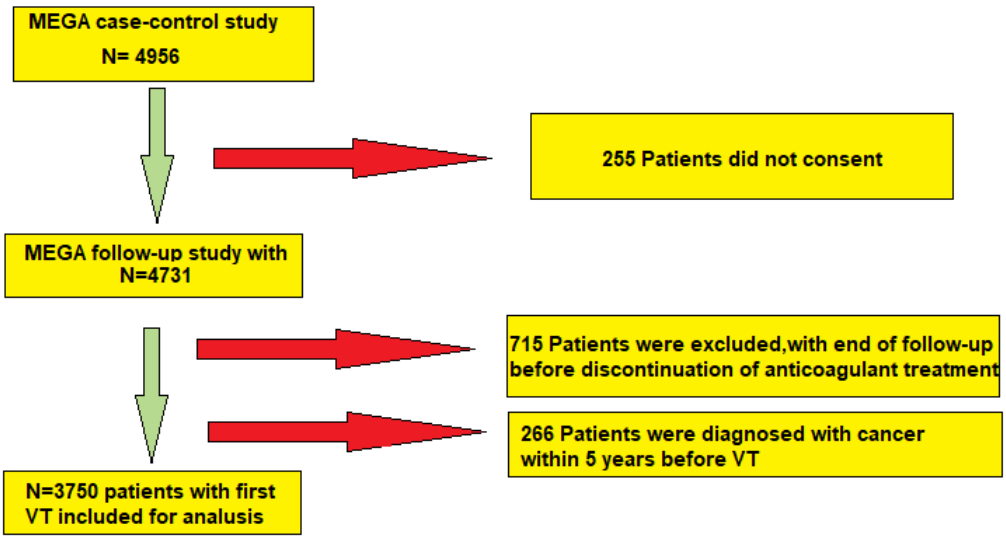


Figure 2.1: Flow diagram of the inclusion of the MEGA follow-up study. Included for analysis : N=3750 patients with first venous thrombosis [54]

Table 2.1: Baseline characteristics of the investigated cohort with n= 3750

Clinical factors	**Class**	**Numbers(%) or median(range)**	**Missing, N(%)**
Provoked additional factors	categorical	1893 (50%)	478(12.7%)
Cardiovascular disease	categorical	194 (5.1 %)	227 (6.1%)
BMI	continuous	26.2 (15.2 - 63.2)	304 (8%)
Age	continuous	48.4 (18 - 70)	0
Gender, male	categorical	1684 (45 %)	0
Pregnancy	categorical	160 (4.2 %)	16 (0.4%)
Surgery	categorical	566 (15 %)	13(0.3%)
Type VT:	categorical		
DVT		2231(59.4 %)	0
PE		1184 (31.5 %)	0
PE + DVT		335 (9 %)	0
Hormone use	categorical	1181 (31.5 %)	41 (1.1%)
Plaster cast	categorical	198 (5.2 %)	0
Hospitalization	categorical	582 (15.5 %)	0
Location VT:			
Proximal vs Distal	categorical	634 (17%)	880 (23.5%)
Cerebrovascular disease	categorical	82 (2.1%)	227 (6.1%)
Posthrombotic syndrome :	categorical		29.6%
mild		244(6.5%)	
severe		81(2.5%)	
Disease additional comorbiditie	categorical	503 (13.4%)	227(6.1%)
Genetic factors	**Class**	**Numbers(%)**	**Missing, N(%)**
factor V Leiden mutation	categorical	568 (15.1%)	314 (8.4%)
Blood group non-O vs O	categorical	2464 (65.7%)	329 (8.8%)
Laboratory factors	**Class**	**Median (range)**	**Missing, N(%)**
Von Willebrand factor*	continuous	5 (3.6 - 6.5)	1643(43.8%)
C-reactive protein (CRP)*	continuous	0.66 (-3.9 - 4.8)	1644 (43.8%)
Antithrombin	continuous	105 (56 - 158)	1643 (43.8%)
Fibronegen	continuous	3.4 (1.2 - 8.9)	1643 (43.8%)
Ddimer *	continuous	5.8 (3.8 - 10.4)	1834 (49%)
Factor II	continuous	112 (22 -173)	1829 (48.8%)
Factor V	continuous	0.93 (0.4 - 2.2)	1643 (43.8%)
Factor VII	continuous	112 (30 - 250)	1829 (48.8%)
Factor VIII*	continuous	5 (3.58 - 6.28)	1645 (44%)
Factor IX	continuous	107.5 (60.5 - 209.6)	1830 (48.8%)
Factor X	continuous	118 (10 - 201)	1829(48.8%)
Factor XI	continuous	104 (48 - 221)	1643(43.8%)
Protein C	continuous	115 (34 - 213)	1829 (48.8%)
TFPI	continuous	1.71 (0.42 - 4.17)	1649 (44%)
ETP	continuous	397.5 (0 - 1055.7)	1837 (49%)
APC ratio *	continuous	0.78 (-2.5 - 2.6)	1772 (47.3%)
Hemoglobin	continuous	8.7 (4.1 -11.6)	1669 (44.5%)
Protein S,free *	continuous	4.5 (2.8 -5.6)	1839 (49%)
White blood cell *	continuous	1.8 (0.35 - 4.07)	1669 (44.5%)
Monocyte percetage *	continuous	1.8 (-0.35 - 3.1)	1690 (45.1%)
Red cell Distribution width *	continuous	2.5 (2.3 - 3.3)	1670 (44.5%)

* A log-transformation was decided upon after a visual check of the distribution curve when a non-normal distribution was found [54].

2.2.1 Correlations between variables

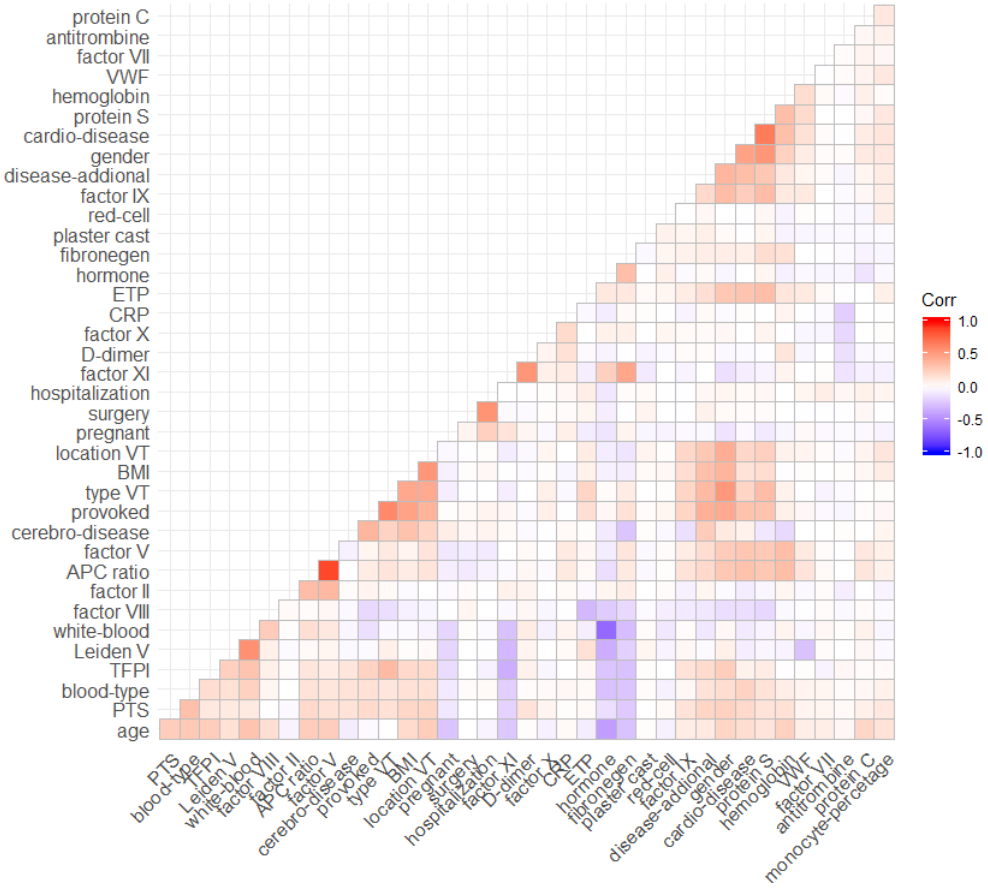


Figure 2.2: Heatmap of correlations between variables in the MEGA study

The heatmap correlation in figure 2.2 displays the correlation pattern among the 38 variables. When looking at the currently displayed correlation heatmap figure, high correlation values are observed in some clusters of factors. We have identified 5 clusters of variables that display a high correlation and have assigned names to them:

- Inertia factors: surgery, plaster cast and hospitalization (Hospital stay). A high correlation value (0.56) was observed between surgery and hospitalization.

- Coagulation factors and proteins : Protein C, Factor VII, Factor IX, Factor II, Factor X, Factor XI and C-reactive protein, These factors are highly correlated, a correlation values arranging between 0.2 (Protein C vs. C-reactive protein) and 0.6 (factor II vs. factor X) were observed in this cluster of variables.

- Clot formation factors : factor VIII, D-dimer, Von Willebrand factor (VWf) and Blood type. Correlation values arranging between 0.02 (factor VIII vs. Blood type) and 0.9 (factor VIII vs VWf) was observed.

- Location and type factors: Pulmonary embolism, Deep Vein Thrombosis and Proximal and Distal deep vein thrombosis (location). Correlation values arranging between 0.25 (PE vs location) and 0.83 (PE vs DVT) were observed.
- Gender factor: Gender, Hormone and Hemoglobin: a negative correlation value -0.65 between hormone and gender (male) and a positive correlation value of 0.56 between gender and hemoglobin.

Until 2002 , blood samples were acquired from the MEGA study population. Blood samples and measurements on laboratory markers were available for 2107 patients out of the total number of 3750 patients [53]. After June 2002 due to logistical reasons no blood samples were collected. This fact is major reason for the displayed missing pattern in figure 2.3.

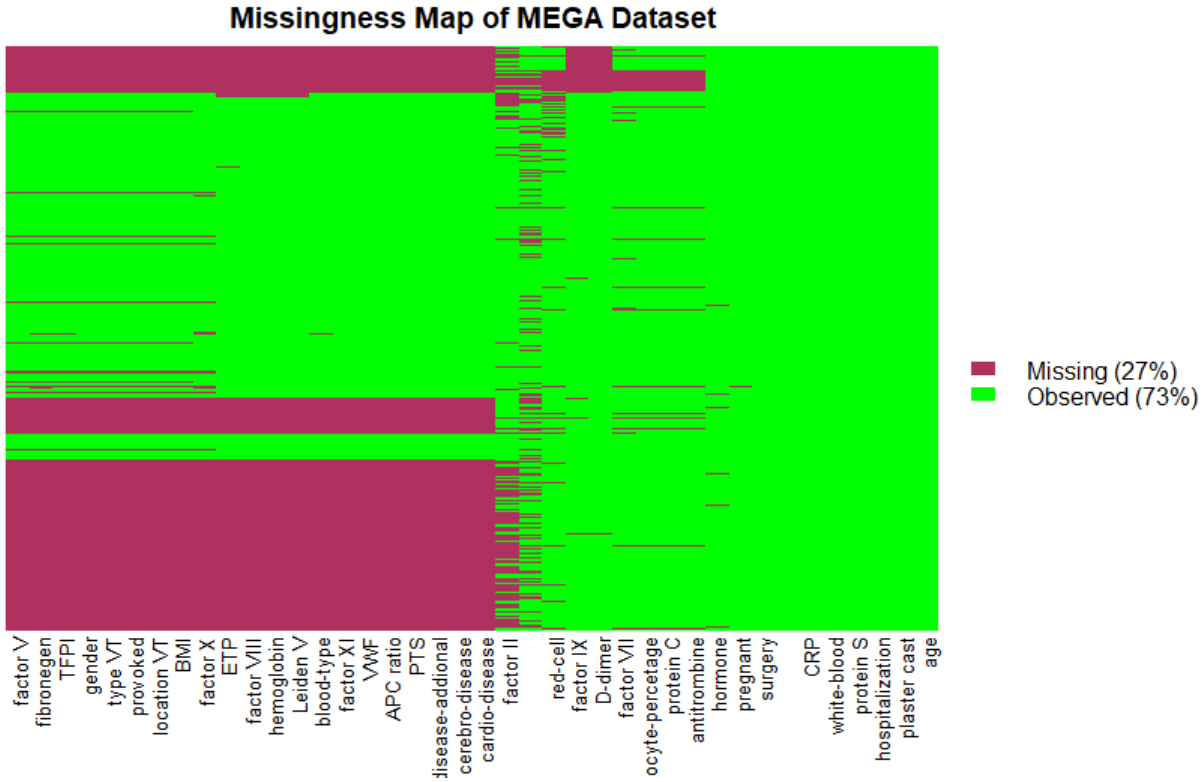


Figure 2.3: Missing values of the candidate predictors of recurrent thrombosis

Chapter 3

Survival analysis background

In this chapter, an overview of the survival analysis methods will be presented. The emphasis will be on the theoretical background, supported with some practical examples. We start by introducing the censoring types. Further, we will discuss the main different survival functions, the Cox proportion hazard (PH) model, partial likelihood and Cox model assumption. We close this chapter by the measure metric of discrimination in survival analysis.

3.1 Introduction

The term 'Survival analysis' originates from studies where the outcome of interest was death. Commonly in these type of studies, one is interested in the time between a certain start point (e.g. time of diagnosis) and the occurrence of the event. Now the scope of the survival analysis is broadened to contain different outcomes of interest, such as in our example recurrence of thrombosis.
Many epidemiological studies nowadays involve following patients over time, the follow-up time for the study may range from a few weeks to many years. The common endpoint of interest in those studies is time until an event occurs e.g. death, recurrence, relapse, time to develop heart disease, heart transplants and time until death etc. Besides the medical application, survival analysis has many filed of application (e.g sales, industry, manufacturing etc). In general, the survival analysis is defined as a set of statistical methods applied in order to analyze time to event data, to find out if there is a link between the covariates and survival.

3.2 Time to event data and censoring

In time to event research, not all individuals might experience the event in the study. There are many possible reasons. For instance, patients may drop out from the study, for a patient in this situation the survival time is considered to be at least as his last known observed time. Another possible reason is the occurrence of the event of interest outside the follow-up time, and the survival time for this patient is considered to be at least as long as the duration of the study period [57]. These two examples are known as right-censored observations. Censoring is an essential concern in survival analysis, illustrating a particular type of missing data.

In our study where the follow-up started in 1999 and last until 2010, table 3.1 shows data of the first 6 patients, where Time column represents the time since the inclusion in the study, and

Status column indicates the censorship i.e. censored (0) or event (1). For instance, patients 1 and 2 did not experience a recurrence of thrombosis during the follow-up period, whereas patient 3 had experienced a recurrence after 1.54 years.

Another type of censoring is known as left censoring, that is when the event of interest has already occurred before enrolment, which is not the case in our study, where any patient who did experience the recurrence of thrombosis before the study's start are not included. Finally, the last form of censoring is known as interval censoring. The interval censoring is encountered in many medical situations where a random variable of interest, in our case recurrence of thrombosis, is known only to lie within a time window and cannot be observed exactly. For instance, in an epidemiological study, assume patients are observed every 6 months. It might happen that the event of interest has occurred between the two consecutive visits. Figure 3.1 illustrates the three types of censoring.

ID	Time	Status
1	8.74	0
2	8.29	0
3	1.54	1
4	8.62	0
5	8.28	0
6	8.26	0

Table 3.1: The first subjects from the dataset.

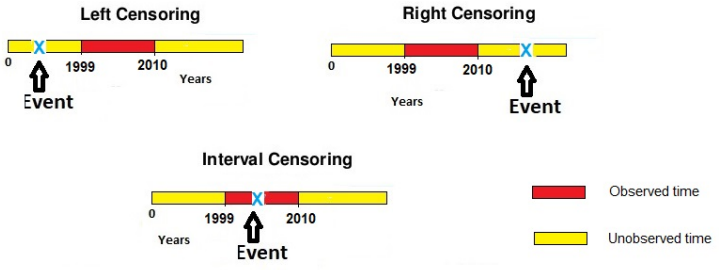


Figure 3.1: A graphical illustration of all three types of censoring.

3.3 Basic survival analysis functions

3.3.1 Survival function

In order to estimate important model parameters, survival analysis methods incorporate the information from both censored and uncensored observations. The dependent variable in survival analysis contains two arguments: time to event and the event status, this is an indicator whether the event under study has occurred or not. Two time dependent functions are the key concepts in analyzing the distribution of non-negative random variable T (time until an event) : Hazard and survival function. Let $f(t)$ be the probability density function (pdf) of T and $F(t)$ be the cumulative density function. The survival function $S(t)$ is the probability that a random individual will survive beyond time t.

$$S(t) := P(T > t) = 1 - F(t) \quad \text{equivalent} \quad F(t) = 1 - S(t); \quad \forall t \geq 0 \tag{3.1}$$

PDF and survival function are related as

$$f(t) = -\frac{d}{dt}S(t). \tag{3.2}$$

and

$$S(t) = \int_t^{\infty} f(u)du. \tag{3.3}$$

From (3.2) we can easy calculate $S(0)$ and $S(\infty)$, simply by using the property that PDFs integrate to one. $S(0) = \int_0^{\infty} f(u)du = 1$ and $S(\infty) = \lim_{t\to\infty} \int_t^{\infty} f(u)du = 0$ Note that this is a decreasing [1] function over time, as the time progresses there will be a smaller chance of surviving. Let $t_1 < t_2 < ... < t_D$ to be the distinct death times (time when events are observed) and let d_i be the number of individuals who experience the event of interest at time t_i. The Kaplan-Meier estimator (Kaplan and Meier, 1958) that is a.k.a product limit estimators is a non parametric method used to estimate the survival function $S(t)$. The estimate $\hat{S}(t)$ is given by:

$$\hat{S}(t) = \prod_{i:t_i \leq t} \left(1 - \frac{d_i}{Y_i}\right)$$

where Y_i is the number of individuals who are at risk at time t_i.

3.3.2 Hazard rate function

The hazard rate function $h(t)$ that is also known as the instantaneous risk of experiencing the event, is defined as the probability that an event will occur in the interval $[t, t + \Delta t]$, given that it has not occurred before. This is a positive function $h(t) \geq 0$ and might be an increasing, decreasing or constant function.

$$h(t) = \lim_{\Delta t\to 0} \frac{P(t \leq T < t + \Delta t | T \geq t)}{\Delta t} \tag{3.4}$$

The relation between hazard function h(t) and survival function S(t), can be derived from the following;

$$\begin{aligned} P(t \leq T < t + \Delta t | T \geq t) &= 1 - P(T \geq t + \Delta t | T \geq t) \\ &= 1 - \frac{P(T \geq t + \Delta t)}{P(T \geq t)} \\ &= 1 - \frac{S(t + \Delta t)}{S(t)} \end{aligned}$$

Hence:

$$h(t) = \lim_{\Delta t\to 0} \frac{1 - \frac{S(t+\Delta t)}{S(t)})}{\Delta t} = \lim_{\Delta t\to 0} \frac{S(t) - S(t + \Delta t)}{\Delta t.S(t)} = \frac{-1}{S(t)} \underbrace{\lim_{\Delta t\to 0} \frac{S(t) - S(t + \Delta t)}{\Delta t}}_{\text{derivative S'(t)}} = \frac{-S'(t)}{S(t)}$$

By using the logarithm derivatives rule, we recognize this last as: $h(t) = -\frac{d}{dt} \log(S(t))$. Note that by using (3.2) we may express the hazard rate as a function of the probability density function as well, hence $h(t) = \frac{f(t)}{S(t)}$. By introducing the cumulative hazard function as :

$$H(t) = \int_0^t h(u)du$$

[1] note that F is a monotoon increasing function over $[0, \infty[$

The relation between S(t) and H(t) can be written as:

$$S(t) = e^{-\int_0^t h(u)du} = e^{-H(t)} \quad \text{and equivalant by} \quad H(t) = -log(S(t)) \tag{3.5}$$

One way of estimating the cumulative hazard $H(t)$ by using the Product-Limit estimator is given as :

$$\hat{H}(t) = -log(\hat{S})$$

A better performance estimator $\hat{H}(t)$, when sample size is small, was given by Nelson-Aalen as:

$$\hat{H}(t) = \sum_{t_i<t} \frac{d_i}{Y_i}$$

3.4 Cox Proportional Hazards model

Suppose we wish to evaluate the impact of some predictor variables, called covariates in survival analysis, on the time to the reoccurrence of thrombosis. For example we may wish to investigate the impact of surgery and age on the time to the recurrence of thrombosis. This specific research question can be approached by a Cox Proportional-Hazards (PH) model. Cox PH model (David Cox, 1972,[7]) is a specific regression model that is used in many medical and engineering settings, to explore the association between one or more predictor variables and the time to event. In the Cox regression models, one can incorporate quantitative as well as categorical covariates to evaluate simultaneously the effect of several covariates on survival time. In our study the outcome of interest is the recurrence of thrombosis, hence the hazard rate will express the recurrence rate at a specific point in time.

Commonly, the data in survival analysis is expressed in the form of the triple indicators $(T_i, \delta_i, \boldsymbol{X}_i(t))$, for i-th individuals with $i = 1, ..., n$, where T_i indicate the time under study for the i-th patient, δ_i indicates the event indicator, with $\delta_i = 0$ right-censored event and $\delta_i = 1$ if reoccurrence has happened and $\boldsymbol{X}_i = (X_{i1}, .., Xip)^T$ the values of the covariates for i-th patient. The hazard function $h(t|\boldsymbol{X} = (X_{i1}, .., Xip))$ of the Cox PH model for individual i has the form:

$$h(t|\boldsymbol{X}) = h_0(t)e^{\beta^T \boldsymbol{X}} = h_0(t)e^{\sum_{j=1}^p \beta_j^T \boldsymbol{X}_j} \tag{3.6}$$

where $h_0(t|\boldsymbol{X})$ is the baseline hazard rate at time t. This is equal to hazard function when $X_{i1} = X_{i2} = .. = Xip = 0$. The baseline hazard rate is analogous to the intercept term in a multiple regression, and $\beta = (\beta_1, .., \beta_p)^T$ is a vector of regression coefficients. Once the conditional hazard rate is computed, the condition survival function $S(t|\boldsymbol{X})$ and cumulative hazard function $H(t|\boldsymbol{X})$ are found by the formula (3.5).

When the survival time distribution is parametric then using a maximum likelihood approach to estimate the survival time is appropriate, but often in real problems this is not the case, the survival time distribution is unknown. An interesting feature of Cox PH model (3.6) is that it is semi-parametric. This means that one can split the component of the model into two main parts: a parametric covariate effect on the hazard $e^{\sum_{j=1}^p \beta_j^T X_j}$, and a non-parametric part consisting of a baseline hazard $h_0(t)$, where no assumptions are made about its form.
When we consider two individuals with covariates values X and X', then the ratio of the hazards of these individuals can be written as:

$$\frac{h(t|X)}{h(t|X')} = \frac{h_0(t)e^{\beta^T X}}{h_0(t)e^{\beta^T X'}} = \frac{e^{\beta^T X}}{e^{\beta^T X'}} = e^{\beta^T (X - X')} \tag{3.7}$$

which show clearly how the baseline hazards are canceled out from this ratio, hence the hazard ratio for the two individuals is independent of lifetime t, constant and proportional. Actually, this is the reason why the Cox model is called a proportional hazards (PH) model. The ratio (3.7) is called the Hazard Ratio (HR) or Relative Risk (RR) of an individual with covariates X having the event as compared to an individual with covariates X'.
For instance, let X_1 indicates the gender effect, and suppose that all other covariates are constant then, $\frac{h(t|X)}{h(t|X')} = e^{\beta_1}$, represents the risk of having the *event* if the patient is a male r*elative* to the risk of having the *event* should the patient be a female.

$$X_1 = \begin{cases} X_1 = 1, & \text{individual is male} \\ X_1 = 0, & \text{individual is female} \end{cases}$$

Regarding the interpretation of the coefficients in a Cox model setting, one may choose between using the regression coefficients or the hazard ratio (HR). A positive regression coefficient for a covariate can be interpreted as the risk is higher for patients with higher values of the corresponding covariate, hence the prognosis is worse. This is equivalent to HR > 1. On the other hand, a negative regression coefficient suggests a better prognosis for patients with higher values of the corresponding covariate. This is equivalent to HR < 1. If the regression coefficient for a covariate is 0, which correspond to HR=1, this can be interpreted as the covariate having no effect on the outcome. In the Cox model, the HR is often used instead of regression coefficients for purpose of interpretation.

3.5 The partial likelihood

From the interpretation of the model, it is clair that β determines the effect of covariates X_j's. Hence β should be the focus of our inference. The contribution to the likelihood for an observed failure at time t is:

$$f_i(t) = h_i(t)S_i(t) = h_0(t)e^{\beta^T \boldsymbol{X}}[S_0(t)]^{e^{\beta^T \boldsymbol{X}}}$$

In the same way the contribution to the likelihood for a right censored observation at time t is given by:

$$S_i(t) = [S_0(t)]^{e^{\beta^T \boldsymbol{X}}}$$

As defined in the previous paragraphs, consider $\delta_i = 0$ if t_i is a censoring time and $\delta_i = 1$ if t_i is a failure time. Assume that patients to be independent of each other, and furthermore we assume that censoring is noninformative and in absence of ties [2] between the event times. Then the joint likelihood is given by:

[2] Two event are tied when they occur at the exact same recorded time.

$$\begin{aligned}
\prod_{i=1}^{n}[f_i(t_i)]^{\delta_i}[S_i(t_i)]^{1-\delta_i} &= \prod_{i=1}^{n}[h_i(t_i)S_i(t_i)]^{\delta_i}[S_i(t_i)]^{1-\delta_i} \\
&= \prod_{i=1}^{n}[h_i(t_i)]^{\delta_i}S_i(t_i) \\
&= \prod_{i=1}^{n}\Big[\frac{h_i(t_i)}{\sum_{j\in R(t_i)}h_j(t_i)}\Big]^{\delta_i}[\sum_{j\in R(t_i)}h_j(t_i)]^{\delta_i}S_i(t_i) \\
&= \underbrace{\prod_{i=1}^{n}\Big[\frac{e^{\beta^T\boldsymbol{X}}}{\sum_{j\in R(t_i)}e^{\beta^T\boldsymbol{X}}}\Big]^{\delta_i}}_{\text{Partial likelihood}}\prod_{i=1}^{n}[\sum_{j\in R(t_i)}h_0(t_i)e^{\beta^T\boldsymbol{X}}]^{\delta_i}S_i(t_i)
\end{aligned}$$

where $R(t_i)$ is the risk set at time t_i.
Estimation is difficult since $h_0(t)$ is an infinite dimensional nuisance parameter. Instead of the full likelihood, Cox (1972, JRSS B and 1975, Biometrika) proposed the partial likelihood. The partial likelihood can be introduced as:

$$L_p(\beta) = \prod_{i=1}^{n} L_i(\beta)^{\delta_i} \quad \text{with} \quad L_i(\beta) = \frac{h_i(T|X)}{\sum_{j\in R(t_i)}h_j(T|X)}$$

Hence:

$$L_p(\beta) = \prod_{i=1}^{n}\Big[\frac{e^{\beta^T\boldsymbol{X}}}{\sum_{j\in R(t_i)}e^{\beta^T\boldsymbol{X}}}\Big]^{\delta_i} \tag{3.8}$$

To give you an idea how the partial likelihood can be calculated, suppose we have a small data set (table 3.2), the following calculation will illustrate how partial likelihood is computed :

Table 3.2: a small data example for calculating the partial likelihood

Patient ID	t_i	δ_i	X_1
1	1	1	2
2	2	0	3
3	4	1	1
4	6	1	5

The partial likelihood is given by:

$$\begin{aligned}
L(\beta) &= \prod_{i=1}^{4}\Big[\frac{h_i(T|X)}{\sum_{j\in R(t_i)}h_j(T|X)}\Big]^{\delta_i} \\
&= \Big(\frac{h_1(1)}{h_1(1)+h_2(1)+h_3(1)+h_4(1)}\Big)\cdot\Big(\frac{h_3(4)}{h_3(4)+h_4(4)}\Big)\cdot\Big(\frac{h_4(6)}{h_4(6)}\Big) \\
&= \Big(\frac{e^{2\beta}}{e^{2\beta}+e^{3\beta}+e^{1\beta}+e^{5\beta}}\Big)\cdot\Big(\frac{e^{3\beta}}{e^{1\beta}+e^{5\beta}}\Big)
\end{aligned}$$

Taking the logarithm of function (3.8), the corresponding log-partial likelihood can be written as:

$$l(\beta) = logL(\beta) = \sum_{i}^{n} \delta_i(\beta^T \boldsymbol{X} - log(\sum_{j \in R(t_i)} e^{\beta^T \boldsymbol{X}})) \tag{3.9}$$

This function is maximized w.r.t β, by taking the derivative of this function, and set derivative equal to 0 . This yields the estimated coefficients of the Cox model.

3.6 Cox Model assumptions

Three important assumptions for the Cox regression model should be made, namely:

1. **Non-informative censoring:** Individuals who withdraw from the study should do so for motives unrelated to the study. An example of this would be a study where we compare the effect of two treatments (intervention vs. control) on survival time. If the treatment was effective, the patients in the intervention arm may be completely recovered and therefore they might feel no longer the need to follow-up. On the other hand, the ineffective treatment for the control group may lead to more incidents, and as a consequence the patients in this arm might become too sick to follow-up. In this situation, the censoring is informative leading to biased study results i.e. the true treatment effect is masked by informative censoring [43].

2. **The proportional hazards:** According to the PH model, for any two individuals with covariates values X and X', the hazards ratio (3.7) does not depend on time t. This implies that the ratio of the two hazards is a constant over time t. The proportional hazards assumption is often checked in two different ways, i.e. a graphical and/or numerical tests. For the former a Schoenfeld residuals plots are commonly used. One can also plot : $log(\hat{H}(t)) = log(-log(\hat{S}(t)))$ vs. $log(t)$ for two different subgroups (e.g. male vs. females). For this plot, if estimated survival curves are fairly separated (i.e. no cross between curves is observed) we can be confident that the PH assumption holds. For the later a numerical test called proportional hazards tests (Grambsch and Therneau (1994)) is performed by means of the function **cox.zph(.)** from the **(survival)** R package. In case the test results displayed a p-value smaller than 0.05, this may indicate that there are time-dependent covariates which one need to take care of and the PH assumption is violated.

3. **Linearity:** A linear relationship between the log hazard and each covariate is required, by taking the logarithm of (3.6):

$$log[h(t|\boldsymbol{X})] = log[h_0(t)] + \beta^T \boldsymbol{X}$$

Commonly the Martingale residuals plots are provided to check the linearity.

3.7 Measure of discrimination in survival analysis

The concordance c-index a.k.a C-statistics, is one of the most extensively used metric of model discrimination in the context of survival analysis. By model discrimination we mean the model ability to correctly classify subjects into one of two categories i.e. model is able to distinguish between individuals who will have the event from those who will not. In this regard, we say that a model has perfect discrimination in case it will assign each subject in the class to which it truly belongs, on the other side a model is considered to have a poor discrimination ability when it

assigns subject to the wrong class.

Since the aim of this thesis is the comparison of different methods to build prediction models, evaluating the performance of a predictive model is an essential step. For this purpose we have chosen to use the C- statistics (Harrell [22]), for its widespread use and being relatively simple to calculate and explain to a medical audience. The c-index is similar to the estimated area under Receiver Operating Characteristics (ROC) for a binary outcome. Similarly, its values range between 0 and 1. A c-index value of 0.5 or less indicates a random classification model (worse model), and a c-index value higher than 0.6 is generally considered to be clinically useful model.

The concept underlying the concordance c-index calculation is explained as follows : Consider all possible pairs (i, j), we denote $T_1, T_2...T_n$ to be the survival times of patients in our cohort, in addition we denote the predicted survival time by $\hat{t}_1, \hat{t}_2...\hat{t}_n$. Then a pair of patients is said to be concordant with the outcome if the model predicted survival time is larger for the patient who lived longer i.e. for a pair (i, j), $T_i > T_j$ and $\hat{t}_i > \hat{t}_j$ or $T_i < T_j$ and $\hat{t}_i < \hat{t}_j$. Furthermore, a pair (i, j) is said to be discordant, if $T_i < T_j$ and $\hat{t}_i > \hat{t}_j$ or $T_i > T_j$ and $\hat{t}_i < \hat{t}_j$. In the presence of right censored data, Harrell [22] proposed estimating the c-index as the mean of concordance C_{ij} over all pairs (i, j). We Restate Harrell's definition of the overall c-index for the survival analysis as follows:

$$C_{ij} = \begin{cases} 1, & \text{if } T_i > T_j \text{ and } \hat{t}_i > \hat{t}_j \\ \frac{1}{2}, & \text{if } \hat{t}_i = \hat{t}_j \\ 1, & \text{if } T_i < T_j \text{ and } \hat{t}_i < \hat{t}_j \\ 0, & \text{if discordant} \end{cases}$$

hence $c = \frac{1}{\#(M)} \sum_{(i,j) \in M} C_{ij}$, where M is the set of all usable pairs of subjects (i, j).

Chapter 4

Model selection methods

In this chapter, an overview of the statistical methods for variable selection will be presented. We start by Backward elimination in section 4.2, then we discuss shrinkage method in section 4.3 : lasso, followed by percentile lasso as an alternative in section 4.4 and as for the last one we discuss the Closed Testing procedure 4.5 as having been introduced by Goeman and Solari [20]. The emphasis is on theoretical background, technical details and the application in a survival setting, finally some limitations are discussed too.

4.1 Introduction

Model selection plays an important role in many research fields. Often the research question involves selecting the best predictors from a set of candidate variables. The main aim of model selection is to reduce the set of predictors to a small set that can replace the full set and account almost for the same variance as is accounted for by the full set of variables. The question is commonly phrased in terms of "which predictors out of the variables set do I really need?".

By applying the selection methods one 's aim is to separate between variables that have the true signal from those that are noise. This can be a challenging problem that a researcher may encounter, especially when there are many candidate variables subject to selection as well as having in mind the parsimony principle i.e. the desire to identify a model that can explain the phenomenon under investigation with a minimum number of predictor variables. Different variable selection methods have been proposed to yield the most appropriate model. e.g. Stepwise regression or automatic selection: Forward selection, Backward elimination, and Stepwise (Bidirectional) elimination. Regularized methods: ridge, lasso, and elastic net and Best subset regression etc. In the current chapter we will focus only on 3 methods i.e: backward selection, lasso in conjunction with percentile lasso and closed testing.

4.2 Backward selection

Backward selection a.k.a backward elimination is a variable selection method. This variable selection technique belongs to the automatic variable selection which is also known under the stepwise procedure terminology. Stepwise procedure techniques are widely used in medical research to build multivariate regression models. There are many reasons for the wide use of this

method: 1)- It is implemented in major statistical software (R, Stata, SAS, SPSS. etc.), 2)- Its computational mechanism is simple.

4.2.1 Model selection using p-value procedure

In backward selection, one starts by fitting a model with all candidate variables at the same time, testing the significance of each variable. Then sequentially one variable at a time with the largest p-value resulted for example by the likelihood ratio test (LRT) is dropped, so long as it is not significant at our chosen stopping criterion. After a variable is eliminated, we refit the model without the eliminated variable, the remaining variables with the largest p-value is considered next. We continue by successively re-fitting reduced models and applying the same rule, the procedure stops when there are no variables in the model that are statistically significant.

In order to perform backward elimination, we have to ensure that the number of observations (n) is higher than the number of candidates variables (p), because the partial likelihood estimates β's are not uniquely determined in case where $n << p$.
The stopping rule is the criterion of elimination at which backward elimination performs variables elimination, often this is denoted as P-to-stay. These criteria as reported [9][44][28] have clearly affect the size of the final selected model. In a study comparing many stopping rules in forward selection, Bendel & Afifi [1] found that the stopping rule has an important impact in withstanding noise variables and allowing true variables to enter the final model.

4.2.2 Limitations

Despite its widely extensive use in epidemiology as well as in other research fields for many years, blackward selection has it's own drawbacks . Harrell is one of the opponents of using the stepwise procedure techniques, he calls openly to reject it, because ***it violates every principle of statistical estimation and hypothesis testing*** [21]. Briefly, we give a summary according to Harrell [21] of the main pitfalls of stepwise procedure techniques:

1. The goodness of the fit measures will be biased and will be too high.
2. The likelihood ratio test statistics do not have the claimed χ^2 distribution.
3. The provided standard errors of regression coefficient estimates are biased low, as consequence the confidence intervals for effects are falsely narrow.
4. The regression coefficients β's are inflated (biased high in absolute value) away from the zero.
5. The provided p-values are too small.

In a Monte Carlo study, Derksen and Keselman [9] have investigated the effect of five parameters on the frequency of selecting true variables, from a large set of predictor variables that contains true (with $\beta_j \neq 0$) and noise predictors (with $\beta_j = 0$). The investigated parameters were :

- The effects of the correlation between predictor variables ($\rho_{X_j X_{j'}}$).
- The sample size n.
- the number of candidate predictor variables p.

- The level of significance for the inclusion and/or deletion of candidate variables.
- The type of subset selection algorithm.

Derksen and Keselman conclude that the number of true and noise variables that ended up in the selected models was very much influenced by :

1. The degree of collinearity among predictors $\rho_{X_j X_{j'}}$: they found that the increase of this parameter $\rho_{X_j X_{j'}}$ resulted in a decrease in the true predictors and an increase in the noise variables contained in the final subset.
2. The sample size n: Surprisingly a large sample size had a small positive effect on the number of true predictors in the final model.
3. The number of predictors p: this was mainly the important parameter that affect both the number of noise variables as well as the number of true variables that enter the final model i.e. when the number of candidate variables p increased, the amount of noise variables entered the final models increased too.
4. R^2 is always overestimated.

In addition, the stopping rule which is defined as the criterion of elimination at which backward selection performs variables elimination (often denoted as P-to-stay). These criteria as reported [9][44][28] have clearly effect on the size of the final selected model. In a study comparing many stopping rules in forward selection, Bendel & Afifi [1] found that the stopping rule has an important impact in withstanding noise variables and allowing true variables to enter the final model.

By investigating the effect of sample size n, the correlation $\rho_{X_j X_{j'}}$ and the number of variables p on the number of true and noise final model variables, Derksen and Keselman [9] have found that, even in the most favorable case i.e. where sample size of n= 90, and uncorrelated ($\rho_{X_j X_{j'}} = 0$) 12 candidate variables, 20 % of of the final model variables were noise. Further they showed also that 74 % of the selected variables were noise in the worst case scenario i.e. $n = 30, \rho_{X_j X_{j'}} = 0.8, p = 24$. In an independent study, Flack and Chang [13] have shown that the median percentage of noise variables in the final model ranged from 33% to 89%. This supports the finding by Derksen and Keselman [9], where they have shown that the average of true variables in the final model was less than the number of investigated true candidate variables

4.3 Shrinkage method

As we have mentioned in the previous section (4.2.2), one of the many drawbacks of variable selection by backward elimination is that it results in model coefficients that are inflated away from zero. In this section, we review some of the many alternatives to backward elimination, the so-called shrinkage methods with the main focus on the lasso.

In the current section, we will give an introduction to shrinkage (regularization) methods for linear regression. Subsequently, we will discuss the application of lasso in linear regression setting as well as for Cox PH model. Finally, we will touch upon some limitations of lasso and provide an alternative solution.

4.3.1 Introduction

By shrinkage methods we mean the shrinking process of the regression coefficients toward zero. The most known shrinkage methods are: ridge, lasso and elastic net, these are new and well-acknowledged methods in which we don't actually select variables explicitly but rather we fit a model containing all (p) candidate variables by using one of the mentioned methods that will shrink the coefficient estimates towards zero (ridge) or exactly zero (lasso) [15].

In linear regression, fitting the full model with many predictors without penalization will result in large noise and a scares signal, and the Ordinary Least Squares (OLS) estimates may not uniquely exist. This is the case for instance when the predictors are highly correlated (Multicollinearity), or the number of predictors exceeds the number of observations ($p >> n$, High-dimensional Data). The later is often the case in some medical data's (Omics data). In the case of severe multicollinearity, the design matrix X can be ill-conditioned, therefore $(X^tX)^{-1}$ might not exist. As we know, the least squares estimates depend upon $(X^tX)^{-1}$ ($\hat{\beta}_{ols} = (X^tX)^{-1}X^tY$), in this case we would have problems in computing $\hat{\beta}_{ols}$. The other consequence of multicollinearity is that the (OLS) estimate will produce coefficient estimates that have a high variance which will make the estimates very sensitive to small changes in the model. Consequently the estimated coefficient will be unstable and hard to interpret [2].

The shrinkage methods can be considered as an alternative to estimate the coefficients in case of multicollinearity, or when the number of predictors exceeds the number of observations ($p >> n$). As a consequence of shrinking the coefficients estimates toward zero some bias [1] is introduced. On the other hand the variance [2]of coefficients estimates might be significantly decreased. If the latter effect is small, the model will have large generalization power. When the variance in our model increase, the spread of the prediction will also increase leading to wrongly predicted values in the new data. The most important improvement of regularized methods regression over OLS is in the bias-variance trade-off. Although there are no explicit formulas for the bias and variance of the lasso estimate, in general we can say that: 1)The bias increases as λ (amount of shrinkage (4.1)) increases, 2) The variance decreases as λ increases [23].

[1] It simply means how far away is the estimated values from true values
[2] It is a measure of spread or variations in our predictions

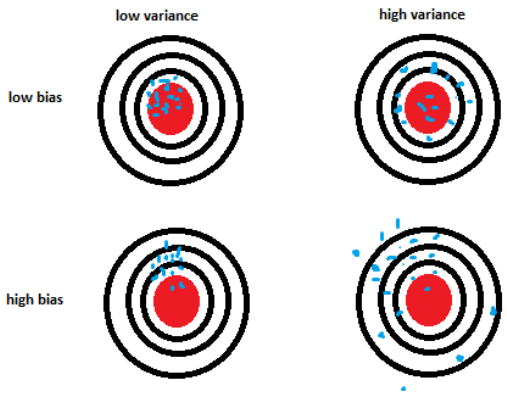


Figure 4.1: The Variance-Bias trade-off

4.3.2 LASSO for linear regression model

Lasso (Least Absolute Selection and Shrinkage Operator) works on the same principle as other regularized methods, in sense they all try to shrink the estimated coefficients toward zero. Lasso possesses an important property that other regularized methods do not have: it allows for automatic variable selection by shrinking some of the coefficients all the way to zero and consequently improvement of interpretability. This is why lasso is often used as a selection variable methods. By considering a standardized data,[3] the lasso estimate is defined by:

$$\hat{\beta}_{lasso} = \underset{\beta}{\operatorname{argmin}} \|Y - X\beta\|_2^2 \quad \text{subject to} \quad \sum_{j=1}^{p} |\beta_j| \leq c \tag{4.1}$$

We can also write the lasso equation in the equivalent Lagrangian form as:

$$\begin{aligned} \hat{\beta}_{lasso} &= \underset{\beta}{\operatorname{argmin}} \|Y - X\beta\|_2^2 + \lambda\|\beta\|_1 \\ &= \underset{\beta}{\operatorname{argmin}} \underbrace{\sum_{i=1}^{N}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2}_{\text{sum of squares}} + \underbrace{\lambda \sum_{j=1}^{p} |\beta_j|}_{\text{lasso penalty}} \end{aligned} \tag{4.2}$$

Where x_{ij} is the value of jth covariate for observation i and y_i is the response of the i-th. observation . The term $y_i - \sum_{j=1}^{p} x_{ij}\beta_j$ is actually the difference between the observed response y_i and the predicted response $\hat{y}_i = \sum_{j=1}^{p} x_{ij}\beta_j$. So, the solution β to the problem, is the β that minimizes the error, under the constraint $\sum_{j=1}^{p} |\beta_j| \leq c$. Note that the solution for β_0 is $\bar{y}$ and thereafter we fit a model without an intercept. Notice that decreasing c in (4.1) is the same as increasing the λ in (4.2). By making c very large (or $\lambda \approx 0$), there will be no constraint (penalization) at all and the value of estimate coefficients will be close to that of an OLS.

[3] Since lasso shrinks the coefficients associated with each variable, it is therefore, necessary to standardize the data such that all variables have a unite variance and the shrinkage value will affect all β's equally

For further illustration of the effect of λ on the number of variables that will end up in the final model, we have plotted the regularization path from our data (figure 4.2). This figure shows clearly the effect of the λ on the number of covariates that are included in the model. As we can see, a large value of λ forces all β's to be 0, hence fewer variables will end up in the model. Whereas a small value of λ , the coefficients start to take nonzero values, thus more covariates will be contained in the final model. In the next section will provide a geometric explanation of this effect.

In general there is no closed formula for calculating $\hat{\beta}_{lasso}(\lambda)$. The coefficient vector has to be determined through iterative processes for each λ. This is a numerically challenging problem, some resort to numerical optimization procedures. Tabshirani proposed quadratic programming to solve (4.1) in his original article [51], whereas Goeman [18] proposed a new algorithm that is based on a combination of gradient ascent optimization with the Newton–Raphson algorithm, and Efron et al. [11] used the LARS algorithm, which simultaneously solves (4.1) and (4.2) for all values of the tuning parameters c and λ .

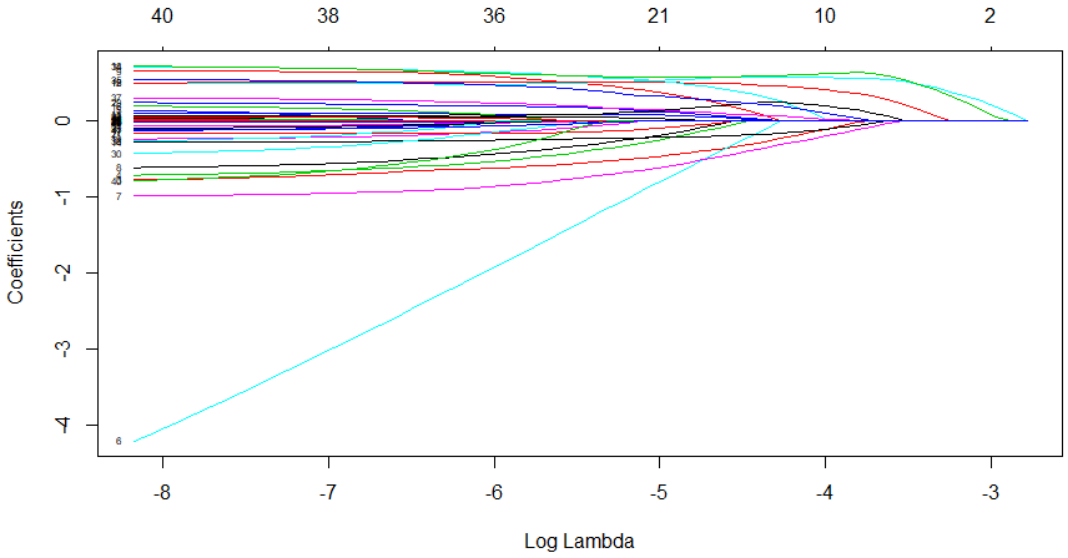


Figure 4.2: Regularization path for our dataset. For each λ , the $log(\lambda)$ versus individual coefficient values are displayed. Each curve traces the change of coefficient values for one variable. Starting from the right and moving to the left (large values of λ) no covariates are selected. Gradually, more covariates are included into the models (coefficients > 0).

4.3.3 Geometric interpretation for linear regression model

For simplicity, we will discuss here the geometric interpretation of lasso in two dimensions space (β_1,β_2). Note that this is only an intuition approach behind lasso method.

Let $f(\beta) = \|Y - X\beta\|_2^2$ be the loss function Residual Sum of Squares (first component of 4.2), its contour plot is shown in black in figure 4.3. There exists a minimum for this function.

Suppose this is in the middle of the black contours. Let us now add a new objective $g(\beta)$,where $g(\beta) = \lambda(|\beta_1|+|\beta_2|)$ (second component of 4.2) this is plotted as rhombus contour in red in figure 4.3. When we decrease λ (≈ 0), the contours of rhombus will expand, therefore the intersection of red rhombus with black f(β) contours comes closer to the center of the black circle, thus we get a non-penalized solution. i.e the $\beta_{lasso} = \beta_{ols}$ OLS estimates. And vice versa will happen to the contours when we increase λ. Now we have to find the minimum of the sum of this two objectives: $f(\beta) + g(\beta)$ but this is obtained when two contour plots meet each other. In the figure 4.3 is now clearly shown that when the first contour of f(β)(black contours) intersects the lasso constraint region (red) $|\beta_1| + |\beta_2| \leq c$. In the figure 4.3 this will result in $\beta_1 = 0$ and $\beta_2 \neq 0$ therefore the predictor X_1 is automatically eliminated. We have restricted our plot on (β_1,β_2) space, but the same argument also applies to the case when the number of predictors $p > 2$; the lasso constraint will have pointy edges (a diamond form), which increases the chances of eliminating variables. That is why lasso gives us sparse solution, making some of parameters exactly equal 0, in this context lasso does a kind of continuous subset selection.

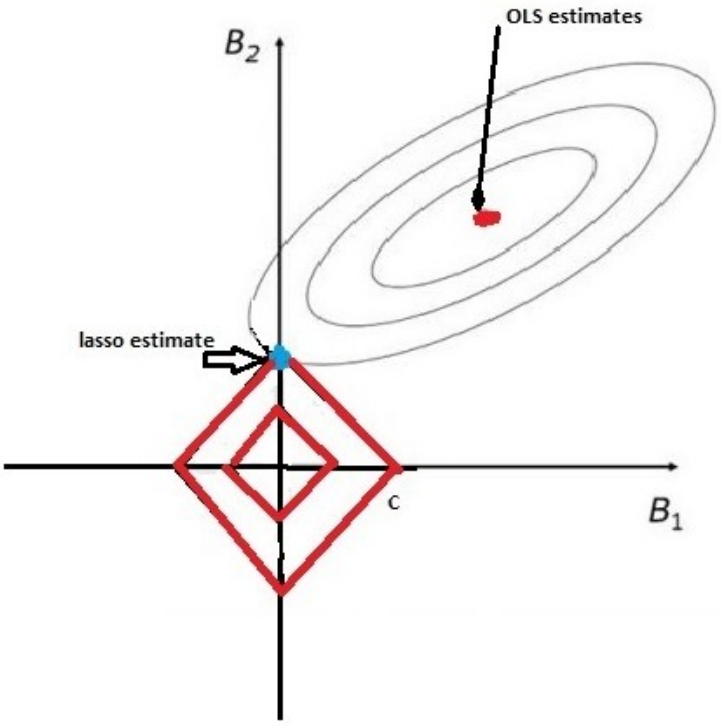


Figure 4.3: The black ellipses are the contours of the least squares error function, while the red contours are the constrain regions for lasso $\lambda(|\beta_1| + |\beta_2|) \leq c$ [23].

4.3.4 Lasso for Cox regression model

In the previous subsections, we discussed the original (i.e. linear regression models) setting for which lasso shrinkage method was intended to. In the current section we will further extend the application of lasso to Cox regression models. Recall that partial log likelihood for Cox regression

model that was introduced in chapter 3 and defined in (3.9) by:

$$l(\beta) = logL(\beta) = \sum_{i=1}^{n} \delta_i(\beta^T \boldsymbol{X} - log(\sum_{j \in R(t_i)} e^{\beta^T \boldsymbol{X}}))$$

Tibshirani [52] proposed to estimate β via the criterion:

$$\hat{\beta}_{lasso} = \text{argmin}\, l(\beta) \quad \text{subject to} \quad \sum_{j=1}^{p} |\beta_j| \leq c \tag{4.3}$$

This equation (4.3) can also be rewritten as:

$$\hat{\beta}_{lasso} = \underset{\beta}{\text{argmin}} \, (\underbrace{l(\beta)}_{\text{Partial log-likelihood}} + \underbrace{\lambda \sum_{j=1}^{p} |\beta_j|}_{\text{lasso penalty}}) \tag{4.4}$$

and the penalized log partial likelihood is given by:

$$l^{(\lambda)}(\beta) = l(\beta) + \lambda \sum_{j=1}^{p} |\beta_j| \tag{4.5}$$

We will discuss in the following subsection how the tuning parameter λ value is obtained.

Cross validation for Cox regression model

In comparison to linear regression, application of cross-validation to Cox model is a bit complex, this is due to the fact that the terms in log partial likelihood (3.9) are not independent, in sense that the term $log(\sum_{j \in R(t_i)} e^{\beta^T \boldsymbol{X}})$ in (3.9) depends on information about other observations, those that are still in the risk group, than the i-th observation itself.

The leave-one-out cross-validation (LOOCV) that was introduced by Verweij and van Houwelingen (1993) [56], takes into account that the components of the partial likelihoods are not independent as in linear or logistics regression model. This is one of many approaches that try to circumvent this problem. Unfortunately, LOOCV can be computationally demanding when the number of observation n and covariates p is very large. However, the same idea as introduced by Verweij and van Houwelingen (1993) can be used for k-fold cross-validation [34]. Similarly, the cross-validation penalized partial log likelihood (cvppl) can be defined as follow :

$$cvpl^{(\lambda)} = \sum_{k=1}^{K} \{ l(\hat{\beta}^{(\lambda)}_{(-k)}) - l_{(-k)}(\hat{\beta}^{(\lambda)}_{(-k)}) \} \tag{4.6}$$

where $l_{(-k)}$ is the log partial likelihood based on all observations except on those in the k-th fold, and $\hat{\beta}^{(\lambda)}_{(-k)}$ is the estimate of β that maximize the penalized log partial likelihood $l^{(\lambda)}_{(-k)}(\beta)$ when the k-th fold is left out (4.5).

Minimum λ in glmnet package

The tuning parameter λ in our equation (4.3) is an important key to determine the number of non-zero coefficients. But how is λ value chosen such that the predictive accuracy of our model is optimal? One way to achieve this by applying Cross Validation. The most common approach is K-fold cross validation. In the glmnet R package, the penalized partial log-likelihood deviance[4] is used as the loss function, instead of the log-likelihood function itself [47]. The idea is simple :

1. The training data T is partitioned into K separate sets of equal size: $T = (T_1, T_2, ..., T_K)$, commonly chosen K's are K = 5 and K = 10.
2. Fit the model to the training set T_k for a particular λ for each k = 1, 2, . . . , K, excluding the k-th fold , obtaining $\hat{\beta}^{(\lambda)}_{(-k)}$ estimate .
3. For each k fold compute the deviance :
$$Dev^{(\lambda)}_{(k)} = -2\Big[l(\hat{\beta}^{(\lambda)}_{(-k)}) - l_{(-k)}(\hat{\beta}^{(\lambda)}_{(-k)})\Big]$$
4. Compute the sum of deviance for a particular λ over all k-folds:
$$Dev^{(\lambda)} = \sum_{k=1}^{K} Dev^{(\lambda)}_{(k)}$$
5. Repeat 1 to 4 steps for a fine grid of values of λ's,
6. Find λ_{min} as the one that minimizes the $Dev^{(\lambda)}$.
$$\hat{\lambda}_{min} = \underset{\lambda \in (\lambda_1, \lambda_2, ... \lambda_m)}{\operatorname{argmin}} \; Dev^{(\lambda)}$$

when K = 1, This is called leave-one-out cross validation (LOOCV).

[4]The deviance is defined as dev= - 2 $l(\frac{M}{M_f})$ in words, this is : -2 times the log likelihood ratio of the model being evaluated compared to the full model (saturated model). This metric provides a measure of goodness-of-fit of the model of interest when compared to the full model

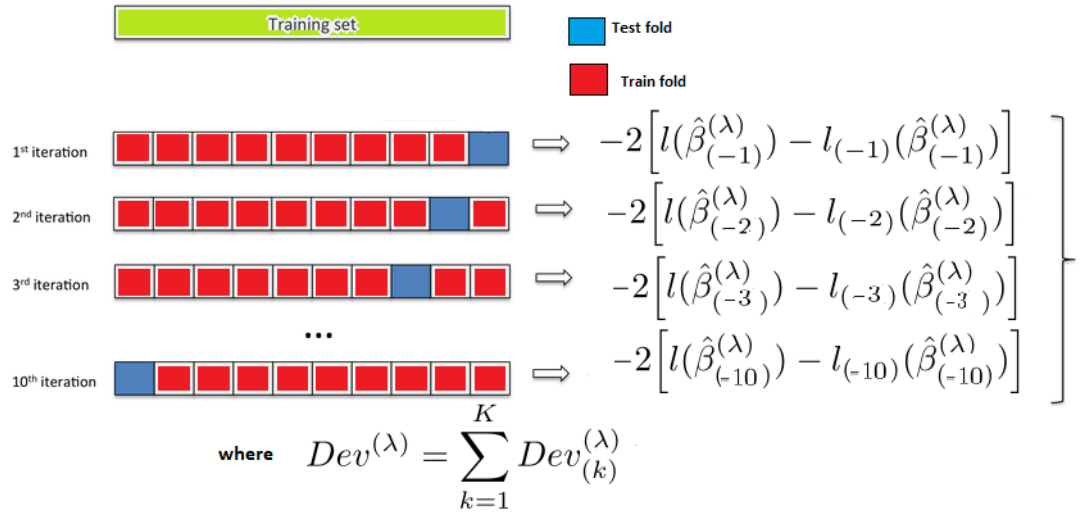


Figure 4.4: K fold cross validation method

Form the experimenters point of view, the choice λ_{min} is very conservative, i.e not eliminating sufficiently many predictors from the model. Another alternative choice was suggested by Tibshirani called one standard error rule.

One standard error rule

The one standard error rule is considered as an alternative rule for choosing the value of the tuning parameter λ. This can be described as follows:

1. Find the minimum deviance $Dev^{(\lambda)}$ and its corresponding (λ_{min})
2. Calculate also the standard error of the deviance as:

$$SE(Dev^{(\lambda_{min})}) = \frac{\sqrt{var(Dev^{(\lambda_{min})}_{(1)}, .., Dev^{(\lambda_{min})}_{(K)})}}{\sqrt{K}}$$

3. Find the largest λ such that the partial likelihood deviance curve is still within one standard error of $Dev(\hat{\lambda}_{min})$. We maintain:

$$\hat{\lambda}_{1SE} = \underset{Dev^{(\lambda)} \leq Dev^{(\lambda_{min})} + SE(Dev^{(\lambda_{min})})}{\text{argmax}} \lambda$$

Tibshirani has described this as : ***In words, we take the simplest (most regularized) model whose error (deviance) is within one standard error of the minimal error (deviance).***

The figure 4.5 is an illustration of our data for the values of (λ_{1SE}), λ_{min}, Partial Likelihood Deviance and the corresponding number of variables. As we can see in this figure there are two vertical lines, these lines are drawn at the values of λ_{min} left and λ_{1SE} right. The number of

non-zero coefficients is shown on the top of the figure 4.5. This means that if we would choose optimal tuning parameter λ_{min} we would get 13 non-zero coefficients for this example, instead of 6 predictors by using the one standard error rule (λ_{1SE}). The key point of the 1SE rule, is to detect the simplest model that can fit as well as the best model that is chosen by λ_{min} .

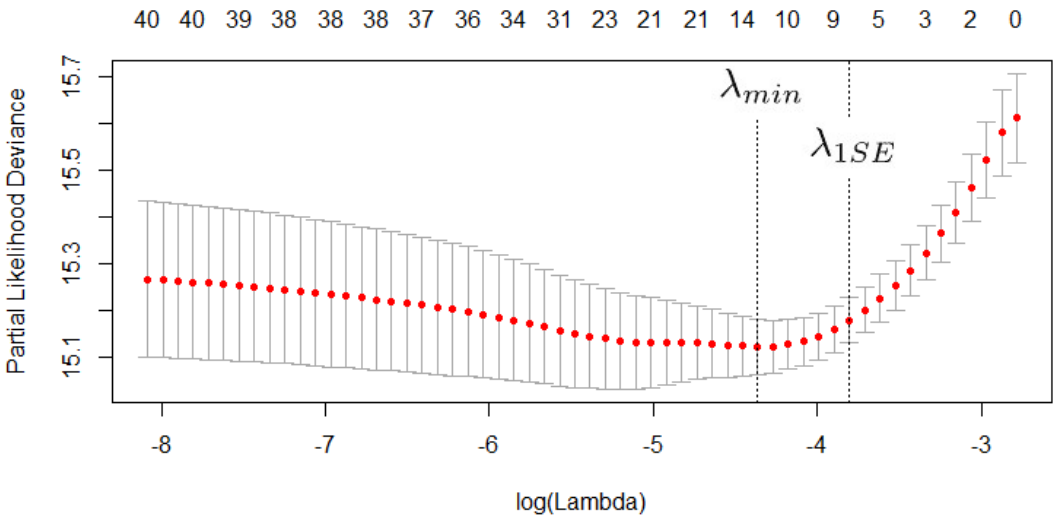


Figure 4.5: Tuning parameter: λ_{min} and λ_{1SE} rule

4.3.5 Lasso limitations

The stability of the selected model is required for two main reasons : reproducibility and generalization performance of an algorithm. Model instability was the main problem that have been risen during model selection by lasso in this thesis. Here by stability we main less variability in model selection, because one is interested in an algorithm that selects nearly the same variables set when one runs the algorithm again. As have been previously discussed, an optimal value for the tuning parameter $\hat{\lambda}$ is found by using cross-validation method. A downside of this approach is the fact that lasso can be very sensitive to the fold assignment used during cross-validation, this was extensively discussed by Bovelstad [3], Krstajic [29] and Roberts [45]. As a consequence of this extreme variability, the results from lasso analysis might not be reproducible and may lack interpretability too. In addition, it was pointed out by Zou and Hastie (2005) [61] that lasso tends to select randomly just one variable from a set of highly correlated variables. Furthermore, in case n << p (Efron et.[11]) pointed out that lasso can select not more than n predictors out of p candidate variables.

In order to illustrate the impact of cross-validation fold assignment on the optimal value of the tuning parameter $\hat{\lambda}$, and hence the number of variables of the model selected by the lasso, we have fitted lasso to our dataset 345 times. The results of this experiment are displayed in figure 4.6 where $\hat{\lambda}_{min}$ variates between 0.005 and 0.014, and on the meantime the corresponding size of the selected models by the lasso ranges from 11 to 23. Investigating the figure 4.6 we can clearly see that the variability among selected models using ordinary lasso is very large. This

is clearly showing the potential sensitivity of the lasso solution to the choice of $\hat{\lambda}$ as Roberts & Nowak, 2014 have demonstrated [45]. To overcome the lasso instability model selection, there were many approaches proposed to this problem [36],[45]. In the following section a method called the Percentile-lasso is introduced.

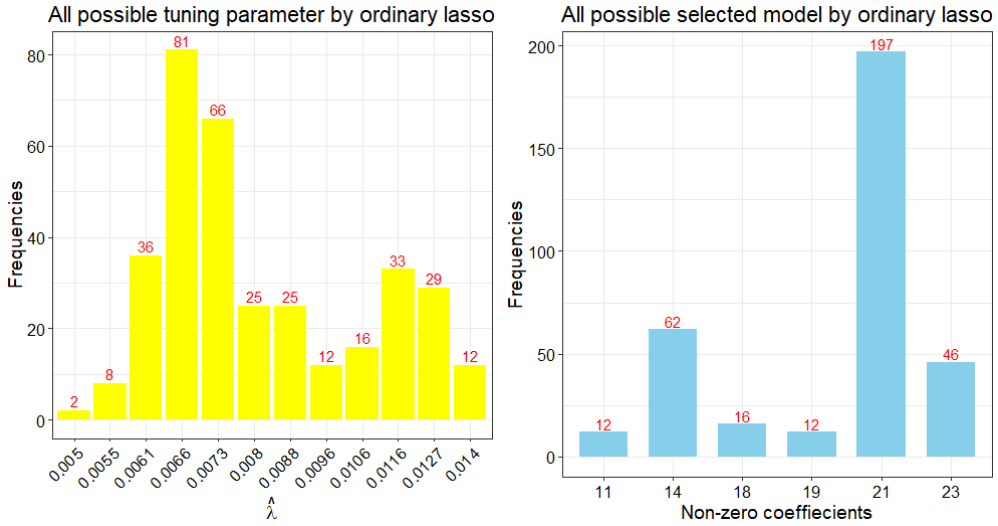


Figure 4.6: Analysis on our data set. The left side of the figure illustrates the frequency of values of optimal tuning parameter $\hat{\lambda}_{min}$ of non-zero coefficient estimates obtained from the lasso, and the right side the frequencies of selected models over 345 random fold assignments.

4.4 Percentile lasso

In the previous section, we have faced a common problem of non-stable model selection by the lasso. Since one important aim of this thesis is to build a model that is reproducible, we have tried to circumvent this problem through an alternative.
In the current section, we will start with an introduction to percentile lasso method as stabilization tool for ordinary lasso's model selection. Subsequently, we will discuss the application of percentile lasso algorithm. Finally, we will provide some arguments for using percentile lasso as an alternative method.

4.4.1 Introduction

The current section is based entirely on Roberts and Nowak article [45]. By fitting ordinary lasso on our dataset for 345 times, we have noticed that the proposed models by lasso are very different i.e. 6 different models were proposed by ordinary lasso, and the number of selected variables in a model range between 11 and 23 (figure 4.6). This observation raised the next question: which model containing between 11 and 23 variables should be chosen to serve as a basis for drawing conclusions ?

4.4.2 Percentile lasso

In a simulation study, Roberts and Nowak [45] showed that as the value of optimal $\hat{\lambda}$ increases, an improvement in the model selected by ordinary lasso was observed. By improvement, we mean that as $\hat{\lambda}$ increases the number of 'false positives' (false non-zero coefficient $\beta = 0$) decreases, whereas the number of 'true positives' (true non-zero coefficient $\beta \neq 0$) remains constant (figure 4.7). Note that the ordinary lasso will choose one of these specific $\hat{\lambda}$ solutions that could fall anywhere in $\hat{\lambda}$'s range. This phenomenon (figure 4.7) was the motivation to base the lasso solution on a specific percentile of a set of possible optimal $\hat{\lambda}$ values, instead of using a single value [45].

The percentile-lasso estimate the percentile θ of a set of optimal tuning parameters λ's that were generated from a cycle M of cross-validation $\Lambda(M) = \{\hat{\lambda}_1 .... \hat{\lambda}_m\}$. Roberts and Nowak have suggested that running the percentile-lasso with $\theta = 0.95$ will be appropriate in most circumstances. Nevertheless, a complete algorithm to estimate the percentile $\hat{\theta}$ is provided by the authors too.

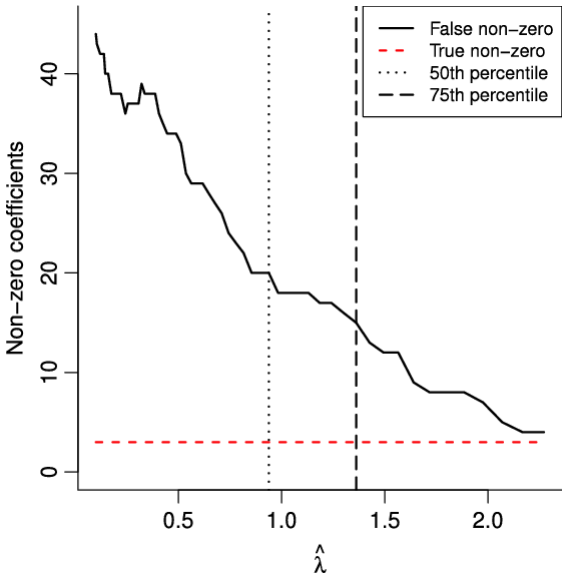


Figure 4.7: ***Number of true and false non-zero coefficient estimates obtained from the lasso fitted to the simulated data, corresponding to the values of*** $\hat{\lambda}$ ***obtained over 1000 random fold assignments. The vertical lines correspond to the 50th and 75th percentiles of the 1000 values of*** $\hat{\lambda}$ [45].

4.4.3 The percentile lasso algorithm

In this subsection we will present the percentile lasso algorithm as was described by Roberts and Nowak [45]. Note that the original algorithm was meant for linear regression settings, therefore we have adapted this algorithm to our thesis methodology. The algorithm is as follow:

1. Fit standard lasso M times using cross-validation for K folds (K=5 or K=10), and find the M optimal tuning parameters $\hat{\lambda}_{min}$ or $\hat{\lambda}_{1SE}$.

2. Denote the set $\Lambda(M) = \{\hat{\lambda}_1, \hat{\lambda}_2, ..\hat{\lambda}_M\}$ the M values of $\hat{\lambda}_{min}$ or $\hat{\lambda}_{1SE}$.

3. Let Θ denote a sequence percentile values θ, e.g. $\Theta = \{75\%, 80\%, 85\%, 90\%, 95\%\}$. Compute $\hat{\lambda}(\theta)$ the percentile of the set $\Lambda(M)$.

4. Re-estimate the parameters of the selected model (i.e from the lasso fitted with $\lambda = \hat{\lambda}(\theta)$) using the ordinary partial likelihood.

5. Compute the cross-validation error (partial likelihood deviance) of the re-estimated model.

6. For each θ in Θ repeat steps 3-5, and select $\hat{\theta}$ to be the value of $\hat{\theta} \in \Theta$ with the smallest cross validation error (deviance).

7. Compute $\hat{\lambda}(\theta)$ the $\hat{\theta}$ percentile of $\Lambda(M)$.

8. Fit the standard lasso with $\lambda = \hat{\lambda}(\theta)$ the percentile lasso solution.

4.4.4 Percentile lasso as an alternative to ordinary lasso

Roberts and Nowak [45] have suggested percentile lasso as an alternative method that can be used in conjunction with ordinary lasso to mitigate the model's variability caused by repeated cross validation. By restricting the optimal $\hat{\lambda}$'s solution on a percentile e.g. $\theta \geq 0.75$ of $\Lambda(M)$ values, the percentile lasso will automatically avoid choosing the smallest $\hat{\lambda}$, as was observed by ordinary lasso. Having this restriction in mind, Roberts and Nowak demonstrated that the percentile lasso can produce significant reductions in the model selection variability (instability) that were common with ordinary lasso (figure 4.6).
According to Roberts and Nowak the reduction in model instability can be attributed to two main factors : (1) the percentile-lasso tends to select values of $\hat{\lambda}$ that are larger than the ordinary lasso, and (2) the values of selected $\hat{\lambda}$ are consistent through fold assignments (i.e less variable) in comparison to the ordinary lasso. As a result, the percentile lasso is an effective alternative to the ordinary lasso.

The results of fitting ordinary lasso to our data set are displayed in figure 4.6. The selected models by ordinary lasso ranges from a model with 11 to a model containing 23 variables. In addition to ordinary lasso, we have fitted percentile lasso 345 times to our dataset i.e. we fit lasso 100 times and we estimated the percentile of $\Lambda(100)$ values of tuning parameter as was described in algorithm section (4.4.3), then we repeated this step 345 times, the results of this procedure are displayed in figure 4.8. We noticed that percentile lasso had selected only two models ranging from a model with 11 to a model containing only 13 variables (figure 4.8). This illustrates the fact that the percentile- lasso, compared to the ordinary lasso, is more likely to select parsimonious models without missing important variables, suggesting that the additional selected variables by the ordinary lasso could be noise. Our results are in agreement with the results from Roberts and Nowak simulation, although these can be seen as side effect benefit, because the foremost purpose of the percentile-lasso is to produce a stable model compared to the ordinary lasso.
Hence, the illustrated results in figure 4.8 suggest that percentile lasso is an effective tool for model stabilization: in 345 repetitions, the number of selected variables ranges between 11 and 13 variables, almost the same model is selected each time it is fitted. Because of this main benefit (model stability), one needs only a single fit of the percentile-lasso to produce interpretable results. This is in contrast to ordinary lasso (figure 4.6) where the results are highly sensitive to the fold assignment, therefore the model instability produced by ordinary lasso will make it hard to draw a meaningful conclusion.

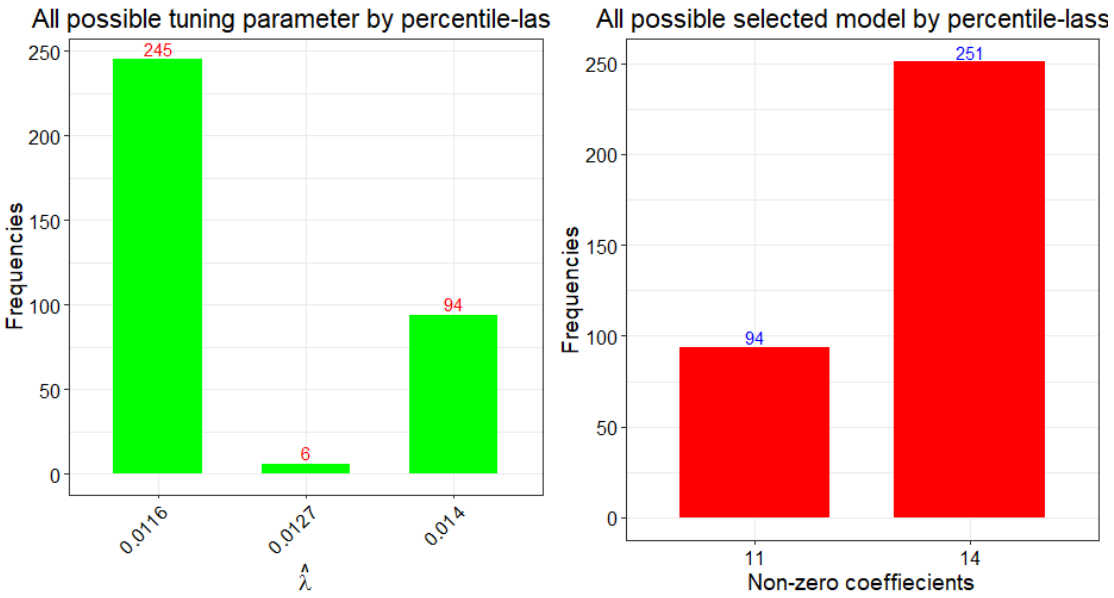


Figure 4.8: The left side of figure illustrate the frequency of values of optimal tuning parameter $\hat{\lambda}$ of non-zero coefficient estimates obtained from the percentile-lasso, and the right side are the frequencies of the selected models, over 345 repetition at $\hat{\lambda} = \hat{\lambda}(min(\hat{\theta}))$. With $min(\hat{\theta})$ we mean the percentile of $\Lambda(100)$ at minimal deviance for each single repetition of 345.

4.5 Closed Testing as a variable selection method

In the current section, we will introduce the closed testing method as was proposed by Goeman & Solari [14] for exploratory research. Variable selection can also be regarded as exploratory research unless some expert knowledge is used. Therefore we will discuss the application of this procedure here as variable selection method. In general, closed testing looks for a collection of sets with the smallest possible number of variables that still exhibit a significant effect on the outcome variable. In this context, closed testing can be seen as a variant of the hierarchical approach [35]. One important characteristic of this procedure is manifested in the possibility to choose freely a set of variables (selected hypothesis) , and if needed, apply one of the multiple hypothesis testing procedure to determine the confidence set for the contained number of false discoveries.

We start this section with an introduction to the closed testing procedure, followed by a brief discussion of this procedure. Subsequently, we provide a way to construct the confidence set for true and false discoveries, last but not least we touch upon some limitations. We close this section by applying the models to a subset from our data.

4.5.1 Introduction

The closed testing procedure was first proposed by Marcus et al. (Marcus, Peritz and Gabriel, 1976) [33]. This procedure is based on a set of intersection hypothesis that are ordered in a hierarchical way (figure 4.12) such that every hypothesis is a superset of a hypothesis one level above. With the global hypothesis (e.g. $H_0^L \cap H_0^G \cap H_0^S \cap H_0^B$) at the top of the tree, followed by non-elementary hypothesis (e.g. $H_0^L \cap H_0^G$) and the elementary hypothesis (e.g. H_0^L) at the bottom. This method was introduced in its original form as a simple and effective solution to control family wise error rate (FWER). Family-wise error rate (FWER) is defined as the probability of making at least one type I error when performing multiple hypotheses: P(rejecting at least one $H_0|H_0$). Under the assumption of independence, the chance of rejecting at least one of the k tests is defined as $1-(1-\alpha)^k$.

The closed testing methods are among the most powerful multiple inference methods [33],[58]. The present section aims to explain the way that closed testing procedure can be used to select candidate variables and to construct the confidence set for a number of false discoveries as well as the number of true discoveries. According to Goeman, this inferential procedure is in agreement with all the three most distinguishing features of exploratory research (7.2.1), namely mild, flexible and post hoc (7.2.2).

4.5.2 Closed testing procedure

In this subsection we will provide an overview of the closed testing procedure. First, let us introduce some notations and definitions. To use the closed testing procedure we consider a family of distinct elementary hypotheses of interest $H_0^1, ..., H_0^m$, out of which we want to select hypotheses to follow up. In our variable selection setting the notation H_0^1 means $H_0 : \beta_1 = 0$ and H_0^m means $H_0 : \beta_m = 0$. Consider now $H_I = \cap_i H_0^i$, with $I \subseteq \{1, 2, \ldots, m\}$ all possible intersection hypotheses. Note that an intersection hypothesis H_I of a collection of hypotheses is false if at least one hypothesis in the collection of hypotheses is false. Similarly, an intersection hypothesis H_I is true if and only if every hypothesis in the collection of hypotheses is true [17]. For instance, in a regression analysis if null hypothesis $H_0^{Location}$ states that the effect of location VT is zero

$\beta_{Location} = 0$, and the hypothesis H_0^{Gender} states that the effect of gender (Sex male) is zero $\beta_{Gender} = 0$, then the intersection hypothesis $H_0^{Location,Gender} = H_0^{location} \cap H_0^{Gender}$ states that the effects of both covariates are 0, i.e $\beta_{Gender} = \beta_{Location} = 0$.

In the closed testing procedure every single hypothesis H_0^i of this set H_I, is tested at level α using a particular local test. There are many local tests available that can be used for this purpose [4] [24][20]. Of note, the choice of local tests has an effect on the efficiency of the closed test procedure (shortcut) [10]. In our data the likelihood ratio test is used as local test. In general, the closed testing method proceeds as follows:

1. Test each elementary hypothesis H_0^i by a suitable local test.
2. Create the closure of the hypothesis set C i.e the set of all possible intersection hypothesis.
3. Perform a suitable local test for every member of the closure C, then reject an elementary hypothesis H_0^i if
   (a) it is rejected by its corresponding local test, and
   (b) every intersection hypothesis H_I that includes H_0^i is also rejected by its local test.

4.5.3 Confidence set for true and false discoveries

This subsection is entirely based on the article of Goeman & Solari [14]. Let us briefly introduce some important notations. Suppose there is a subset of a true hypotheses among a given m set of elementary hypotheses $H_1 \dots H_m$, and let $T \subseteq \{1, 2, \dots, m\}$ be the unknown indices of these true hypotheses. Denote R to be the rejection set i.e. a set of hypotheses that a researcher is interested to reject (selected hypotheses), and let Closure C to be the set of all possible intersection hypotheses. Furthermore we denote the set of all rejected hypotheses (discoveries) by the closed testing procedure on level α by $\mathcal{M} \subseteq C$. Let $\tau(R) = |T \cap R|$ be the number of false discoveries i.e hypotheses that are falsely rejected ($\beta = 0$), and $\phi(R) = \#R - \tau(R)$ to be the number of true discoveries i.e hypotheses that are correctly rejected ($\beta \neq 0$). For a given set R, these quantities ($\phi(R)$ and $\tau(R)$) are a function of model parameters that can be estimated and also a confidence interval can be constructed. In the following we will discuss how $(1-\alpha)\%$ boundary confidence sets are constructed for the number of false discoveries $\tau(R)$, as well as the number of true discoveries $\phi(R)$. A more detailed description and rigorous mathematical proof of the confidence set, we refer the interested readers to Goeman et al [20]. In this paper they showed that a $(1-\alpha)$ confidence set for the number of false discoveries $\tau(R)$ is given by:

$$\{0, ..., t_\alpha(R)\} \tag{4.7}$$

where $t_\alpha(R)$ is the size of the largest subset of R for which the corresponding intersection hypothesis is not rejected by the closed testing procedure: in abbreviated math
$t_\alpha(R) = max\{|I| : I \subseteq R, H_I \not\subset \mathcal{M}\}$. Because $\tau(R)$ only takes discrete value, we will talk in terms of confidence set rather than a confidence interval in this setting. On the other hand, the 100(1 - α)% confidence set for the true discoveries $\phi(R)$ for a given set R is given by [20]:

$$\{f_\alpha(R), ..., \#R\} \tag{4.8}$$

where $f_\alpha(R) = \#R - t_\alpha(R)$.

Let us clarify the construction of confidence set by an example from the chart figure 4.12. Consider for instance, the selected variables set (rejection set) $R = \{Surgery, BMI\}$. For this

specific R set, we get a value of $t_\alpha(R) = 1$. Therefore, when one rejects $H_0^{Surgery}$ and H_0^{BMI}, the $(1-\alpha)$- confidence set for the number of false discoveries as well as for the number of true discoveries is {0, 1} and {1,2} respectively, therefore for this selected set R, one can be confident of making at least one true discovery $\phi(R) = 1$ i.e with a model containing Surgery and BMI we are $(1-\alpha)$%- confident that at least one variable in the model is truly relevant for recurrence.

By investigating all possible confidence sets, the researcher has the possibility to select a set R that is more suitable to answer his research question. In doing this one is still keeping correct $(1-\alpha)$ coverage of the selected confidence set for the number of true or false discoveries [20]. Having said this, the user will have countless options in selecting a set R of variables and may review all options and their consequences in order to make his/her choice.

4.5.4 Application

To illustrate the mechanism of the closed testing procedure, let us examine the association between four candidate variables and the recurrence of thrombosis by using our dataset. Consider for instance the following four candidate variables to build a Cox regression model $h(t|\boldsymbol{X}) = h_0(t)e^{\sum_{j=1}^{4}\beta_j^T \boldsymbol{X}_j}$:

1. X_1 =Location: proximal vs distal DVT.
2. X_2 =Gender: male vs female.
3. X_3 =Surgery: within 3 months before VT.
4. X_4 =BMI: body mass index.

Figure 4.12 illustrates the application of the closed testing procedure to subset of covariates from the MEGA study dataset. This chart displays all 15 $(2^4 - 1)$ possible intersection hypotheses H_I. In the current discussed closed testing procedure, no method of multiplicity control was applied in our study. In the following illustrative chart (figure 4.12), the elementary hypothesis $H_0^{Location} : \beta_{Location} = 0$ refers to location null hypothesis, $H_0^{Gender} : \beta_{Gender} = 0$ refers to gender null hypothesis, $H_0^{Surgery} : \beta_{Surgery} = 0$ refers to surgery null hypothesis and finally the elementary hypothesis $H_0^{BMI} : \beta_{BMI} = 0$ refers to BMI null hypothesis. Furthermore a non-elementary hypothesis like $H_0^{Location} \cap H_0^{Gender}$ refers to location and gender together. i.e: $H_0 : \beta_{Location} = \beta_{Gender} = 0$, in words we say that the regression coefficient of location as well as the regression coefficient of gender is 0. Hence by rejecting the null hypothesis $H_0^{Location} \cap H_0^{Gender}$ we mean that either gender or location regression coefficient is not zero.

In order to test each hypothesis of the closure C i.e. testing the null model against the saturated model, the likelihood ratio test for the Cox model is applied as a local test. The associated p-values for each elementary as well as for non- elementary are displayed in figure 4.12. In addition the rejected as non rejected hypothesis by the closed testing procedure are marked in red and blue respectively. In this example, three out of 4 elementary hypotheses corresponding to each of the 4 examined candidate covariates were rejected by closed testing, in addition all non-elementary hypothesis were rejected at the fixed significance level $\alpha = 0.05$ too.

Applying the closed testing procedure to our dataset has resulted in 14 rejections out of 15 hypotheses, among which there are 3 elementary hypotheses: Location, gender, and surgery (figure 4.12). The ***cherry R*** package includes the ***closed(.)*** function that enables us to perform the closed testing procedure in ***R***. The displayed results (figure 4.9) were generated by using

the object created by the closed () function in R. This result provides us with a lower confidence bound on the number of false discoveries $\tau(R)$ as well as the upper confidence bound on the number of true discoveries $\phi(R)$ among the collection of all four tested hypotheses. We conclude that there are likely at least 3 true discoveries among the four selected hypotheses.

```
ct.sub <- closed(test, hypotheses)
> ct.sub
Closed testing result on 4 elementary hypotheses.
At confidence level 0.95: False hypotheses >= 3; True hypotheses <= 1.
```

Figure 4.9: Closed object results

Sometimes one is interested in a small subset of hypotheses R, in this case one can investigate the number of true and false discoveries among the corresponding subset by using ***pick(.)*** function from the cherry ***R*** package. For instance if we select variables location and gender i.e. R = {Gender, Location}, the displayed output figure 4.10 suggest that these variables are both true discoveries ($\beta_{Gender} \neq 0$ and $\beta_{location} \neq 0$) i.e. these variables should remain in the model. Whereas when we pick surgery and BMI covariates, we note that there is only evidence for one true discovery among surgery and BMI covariates ($\beta_{Surgery} \neq 0$ or $\beta_{BMI} \neq 0$) (figure 4.11).

```
pick(ct.sub,c("locatie1varII","sex_J"))
2 hypotheses selected. At confidence level 0.95:
False null-hypotheses >= 2; True null-hypotheses  <= 0.
```

Figure 4.10: pick object results for the selected variables Location and Gender together

```
> pick(ct.sub,c("oper3mnd", "bmi"))
2 hypotheses selected. At confidence level 0.95:
False null-hypotheses >= 1; True null-hypotheses  <= 1.
```

Figure 4.11: Pick object results for the selected variables surgery and BMI together

Furthermore, ***Defining set*** and ***Shortlist set*** are two important concepts that will be introduced here which can give more insights into the structure of the results of closed testing. The defining set is defined as ***a collection of sets of hypotheses with the property that for each set in the collection we can be confident that it contains at least one true discovery*** i.e $\beta \neq 0$ [17]. In the cherry package, this was given by ***defining(.)*** function. In our case, the defining collection is the following singleton sets:

- {Location }
- {Gender }
- {Surgery }

As each of these sets corresponds to a singleton rejected hypotheses, we can conclude with 95% confidence that these covariates are truly rejected hypotheses (true discoveries). This is exactly what the graphical illustration is depicting in figure 4.12. We see that all the three elementary hypothesis {location, gender and surgery} were rejected, and any set that contains these elementary hypothesis were also rejected. Remember that an intersection hypothesis H_I of a collection of hypotheses is false if at least one hypothesis in the collection of hypotheses is false. Furthermore, when we select for example a set of hypothesis R = { Location, Gender, Surgery, BMI }, we will be confident that we have selected at least 3 truly relevant covariates (true discoveries).

In addition, the second concept is introduced here. The **shortlist** is defined as ***a collection of sets of hypotheses with the property that at least one of the sets in the collection contains only true discoveries*** i.e ($\beta \neq 0$)[17]. Moreover, these sets construct the smallest models that fit as good as the full model, in other words the shortlist will retains a collection of the smallest set of variables that still display a significant effect on the outcome variable [35]. The shortlist is given by function ***shortlist(.)*** from Cherry R package. In the current discussed example, the shortlist resulted in a single set containing: { location, Gender, Surgery } .

4.5.5 Limitations and Shortcuts

In general, the size of the graphical representation(figure 4.12) for m elementary hypotheses is 2^m–1. In other words, the closed testing procedure has the disadvantage of performing an $O(2^m)$ intersection hypotheses. Though, even for a moderately large m hypothesis, say m around 20-30, the standard form of closed testing will result in unfeasible computation, let alone if a large number of hypotheses is to be investigated say $m > 1000$ (e.g. in GWAS).

To reduce the overall complexity of $O(2^m)$, methods for avoiding such large calculation of some hypothesis tests were investigated, these are known as shortcuts. Shortcuts methods can be useful to conduct closed testing without evaluating all $2^m - 1$ hypotheses. Some shortcuts methods have the ability to reduce the closed testing complexity from $O(2^m)$ to $O(m^2)$. This is beyond the scope of this thesis, nevertheless I refer the interested reader to consult [20], [16] and chapter 2 from [34].

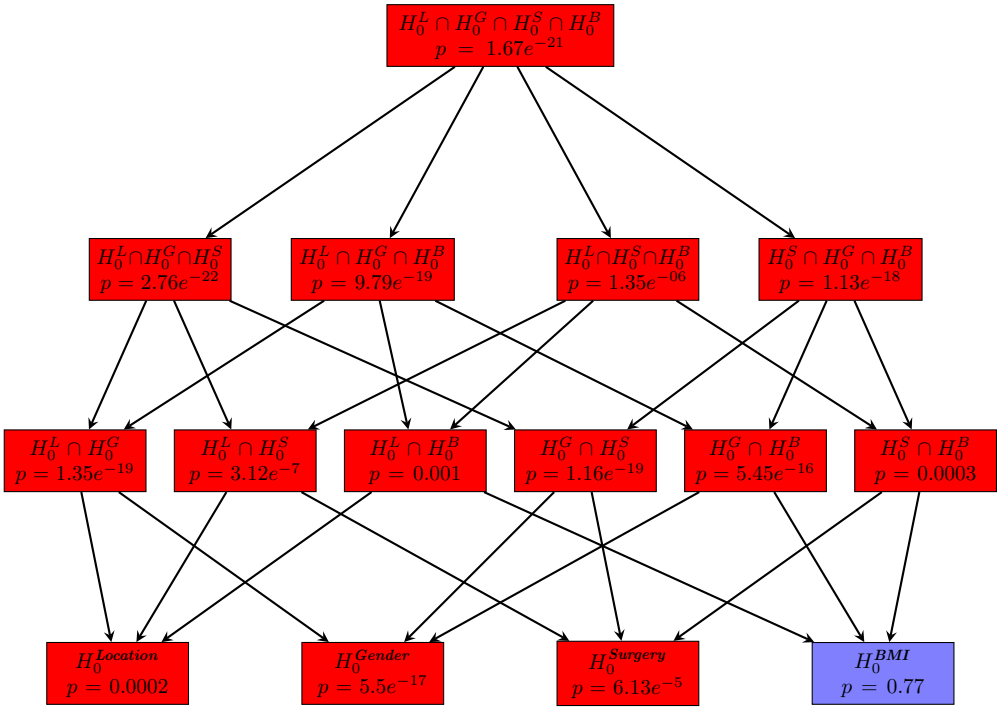


Figure 4.12: A graphical illustration for all intersection hypotheses for a set of m=4 hypotheses (elementary and non-elementary) by closed testing procedure for m= 4 hypotheses, $H_1^{location}$, H_2^{Gender}, $H_3^{Surgery}$ and H_4^{BMI} . Rejected hypotheses are colored in red. In the graph $H_1^L \cap H_2^G$ is the same as : $H_1^{Location} \cap H_2^{Gender}$.

4.6 Model validation: Internal

The goal of a predictive model is to provide a reliably predicted outcome for a new subject. Model validation is an essential step to evaluate the reliability of models before they can be put in clinical practice. In general, there are two different validation classes. Internal and external validation. We talk about Internal Validation when the performance of the model is assessed on the same data set as was developed. Whereas in an External Validation, the model performance is assessed on data from a different population. In this study, we have only access to the MEGA study data set, and no other data set was available to validate our model, therefore internal validation will be the main focus of this section.
In this section, we will provide a brief overview of three commonly used internal validation techniques: Split Sample (a.k.a Training - Testing), Cross-validation, and Bootstrap.

4.6.1 Split-sample

A straightforward and popular old approach. Commonly the data is randomly split into a training (2/3) and testing part (1/3): the former part is assigned to develop the model and the latter to measure its performance. There are many known drawbacks of this approach e.g. model instability resulted by developing the model using just a part of the data. By chance, the model could show a good or poor performance, and moreover, the spilt-sample approach requires a large sample size in order to be reliable [48].

4.6.2 K-fold Cross-Validation

Cross-validation is an extension of split-sample validation technique. In k-th fold cross-validation (e.g. k=5 or k=10), the data is divided into k equalized subsets. Each time, the model is fitted to k-1 subsets that form a training set together and tested in the k-th fold. In this way, all patients have served once to test the model. The error estimation is averaged over all k testing sets in order to get a total cross-validation error of our model. However, the whole cross-validation procedure may need more computational time, one needs more repetition of the whole technique in order to obtain truly stable results [48].

4.6.3 Bootstrapping:

For the bootstrapping techniques, one will generate M samples with replacement from the original data set, of the same size as the original data set, i.e. the entire dataset is used for model development. Often, 100–200 bootstrap samples may be sufficient to obtain stable estimates. In our case, 200 bootstrap samples of dataset A including 1241 patients were drawn with replacement for model A, and 200 bootstrap samples of dataset C including 1881 patients were drawn with replacement for model C. We fit the model in each bootstrap sample, and evaluate its performance in the bootstrap sample (estimate of bootstrap performance c^{boot}) and in the original dataset (estimate of original performance c^{org})). The technique consists of the following steps:

1. Develop the model in the original dataset, and calculate the c^{app}.
2. Generate a sample of size n from our dataset with replacement (bootstrap sample).
3. Fit the developed model in the bootstrap sample, and calculate the c^{boot}.

4. Apply the fitted model from the bootstrap sample to the original data, and calculate c^{org}.

5. Repeat 2-4 steps B times (e.g 100-200 times).

6. Compute the optimism as :

$$Opt = \frac{1}{B}\sum_{b=1}^{B}(c_b^{boot} - c_b^{org})]$$

7. Compute the corrected C statistic as:

$$\text{C- corrected} = c^{app} - Opt$$

Commonly split-sample internal validation performs worse (underestimate performance and high variability) than bootstrapping [49]. Whereas the Cross-validation is considered not as precise as the bootstrap: in many cases, cross-validation has to be repeated many times to achieve adequate precision [49]. Steyerberg and Harrell [49] strongly recommend the bootstrap resampling approach for internal validation, as it results in a stable and nearly unbiased estimate of performance.
Since our data set size is not that large, we will abstain from use split-sample approach, instead, we will perform bootstrapping internal validation in our analysis.

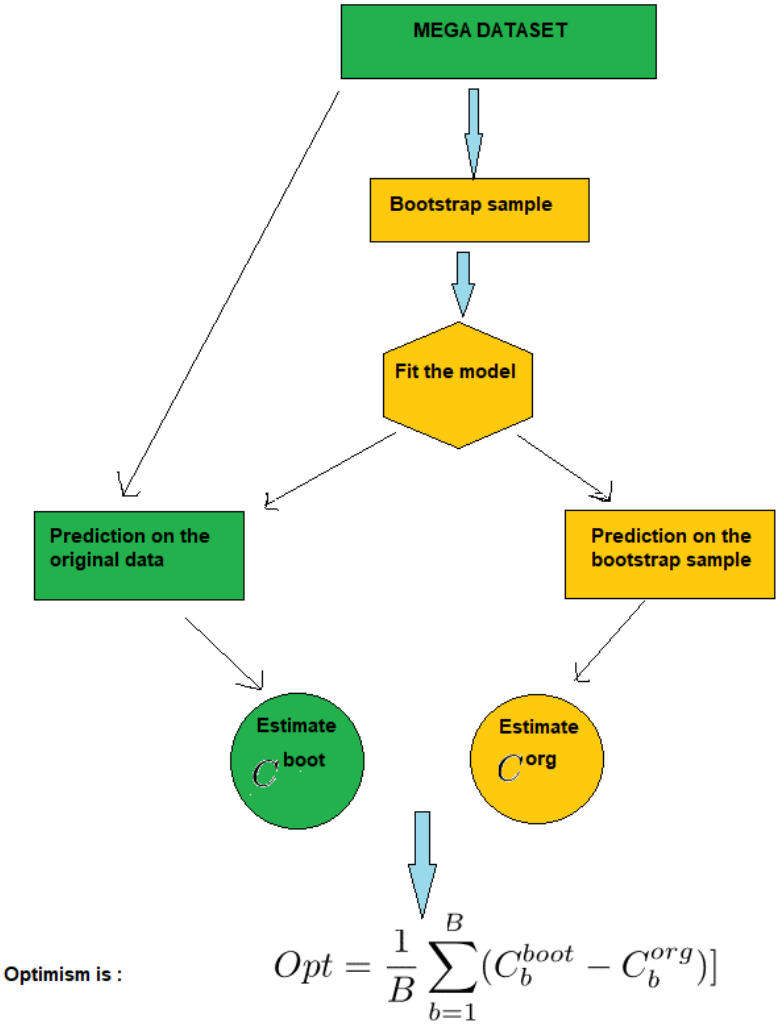


Figure 4.13: A schema to illustrate the bootstrap internal validation

Chapter 5

Results

In this section, we will provide an overview of the results from the previously discussed statistical methods applied to our original dataset. Firstly, in Section 5.1 we introduce our investigated models. In section 5.2 selecting candidate variables using Backward elimination results are provided. Then, in the following sections 5.3 and 5.4 the results from the selecting candidate variables using lasso as well as Closed Testing are described respectively. Finally, in section 5.5 we calculated the degree of discrimination of each model, by means of Harrell's C statistics and the corrected C-index for optimism.

5.1 Developing the model

Two models were developed on our data, by means of Cox regression where we perform model selection using backward selection, lasso in conjunction with percentile lasso and closed testing. In the current thesis, we studied two models, a full model that contains all candidate variables .i.e. clinical, laboratory and genetic factors (model A), and a model containing clinical and genetic factors only (model C). The identification of candidate variables for inclusion in the study was based on : **1) consistent inclusion in the previous prediction models, 2) reported of the association with recurrence of VTE in literature or 3) expert opinion** (Jasmijn Timp-2018).

Our dataset, data with missing values, contains records on 3750 patients with first VT, either provoked or unprovoked, among which 86.48% of the patients were censored and 13.52% had recurrence of thrombosis. The median follow-up time was 5.72 years with an IQR (3.19-7.43), the most type of first VT were DVT events n= 2231 (59.4%). The mean age of participants was 48.4 years and 45% were men. All analyses were performed using only the patients with complete data. For the development of model A, n= 1241(33%) patients were involved, and for model C, n= 1881 (50%) patients were involved. The baseline characteristics are summarized in table 2.1.

5.2 Selecting candidate predictor variables by backward selection

In this section, we provide an overview of the results of backward selection methods. The analyses were performed in the R-software environment version 3.4.2. In order to perform backward

selection on our dataset, we used the **selectCox()** function from the **pec** package . The technical details for this method were explained in Section (4.2).

5.2.1 Model A

In this section we will describe the results for Cox regression models obtained by backward selection method using all candidate variables with a removal criterion P-value = 0.1. Among 38 variables that the full model contains, only 11 variables were selected. The hazard ratios (HR) and the 95% confidence intervals estimated from the Cox proportional hazard model are summarized in table 5.1.

We have observed a strong positive effect i.e. patients with one of these factors will have a lower risk of recurrence of thrombosis, with the most important predictors being the hormone use with a HR=0.44 and 95%CI(0.231, 0.834) and surgery with HR= 0.45 and 95%CI (0.227, 0.893), followed by Type of the first VT (Pulmonary Embolism) and Fibrinogen (table 5.1). This result can be interpreted as a patient who uses hormone, holding other factors constant, will have 56% less risk of recurrence of thrombosis.

In addition, a slightly negative effect was observed for factor X variable with a HR=1.01 and a 95%CI of (1.00, 1.02) and factor XI with a HR=1.007 and a 95%CI of (0.999, 1.016). This can be interpreted as a patient with one of these factors, keeping the other covariates at a constant value, will have a slightly higher risk of recurrence of thrombosis.

Lastly, a strong negative effect indicating a strong relationship between the patients with one of these factors and increase risk of recurrence of thrombosis. The most important predictors for this category are factor VIII with a HR= 2.52 and a 95%CI of (1.64, 4.35), Gender (male) with a HR=1.99 and a 95%CI of (1.22, 3.25), Type of the first VT (PE+DVT) with a HR of 1.74 and a 95%CI of (1.091, 2.803), followed by APC ratio and D-dimer. These results can be interpreted for instance as: a patient with a higher level of factor VIII, holding the other covariates constant, will have a higher risk of recurrence of thrombosis. Or being a male, a patient will have 2 times higher risk of recurrence of thrombosis in comparison with a female patient, holding the other covariates constant.

5.2.2 Model C

We repeated the backward selection procedure, but now only using clinical and genetic candidate variables (model C). The hazard ratios (HR) and the 95% confidence intervals estimated from the Cox model for model C are summarized in the right-side of table 5.1. Among 17 candidate variables, backward selection for model C resulted in 8 selected variables.

A strong positive effect predictors, patients with one of these factors have a lower risk of recurrence of thrombosis, were observed for the pregnancy predictor with a HR= 0.13 with 95%CI of (0.021, 0.9), followed by surgery with a HR= 0.42 with 95%CI of (0.26, 0.7), plaster cast, hormone use and Type of the first VT (PE: Pulmonary Embolism).

In addition a strong negative effect was observed indicating a strong negative relationship between a patient with one of these factors and the risk of recurrence of thrombosis i.e. higher risk of recurrence of the thrombosis. With gender (male) being the most important predictor

for this category with a HR= 1.81 and a 95%CI of (1.26, 2.60) and location of VT (distal DVT) with a HR= 1.53 and a 95%CI of (1.10, 2.13).

5.2.3 Predictors of recurrent thromboembolism models by backward selection

The selected variables by the different models are presented in table 5.1. The following predictors were found to be common among all selected models: surgery, hormone use, gender and type of the first VT (PE and PE & DVT). In addition, the following laboratory factors were additionally predictive for recurrence event for model A: Fibrinogen, protein C, factor X, factor XI, APC ratio, factor VIII and D-dimer, whereas none of the genetic factors were significant.

On the other hand, the following clinical predictors were additionally predictive for recurrence for model C: plaster cast, pregnant and location VT. Furthermore, one genetic factor predictor was also predictive for a recurrent event i.e. factor V Leiden.

Table 5.1: The remaining variables after backward selection, and their corresponding regression coefficients for models A and C

	Model A		**Model C**	
Clinical factors	Hazard Ratio (HR)	(95% CI for HR)	Hazard Ratio (HR)	(95% CI for HR)
Surgery	0.4504	(0.227, 0.893)	0.4266	(0.260, 0.700)
Hormone use	0.4391	(0.231, 0.834)	0.4749	(0.298, 0.756)
Gender (male)	1.9971	(1.228, 3.249)	1.8153	(1.267, 2.602)
Type of 1st VT(DVT,PE,PE+DVT):				
PE	0.7034	(0.465, 1.065)	0.8910	(0.653, 1.216)
PE+DVT	1.7490	(1.091, 2.803)	1.4704	(0.978, 2.211)
Plaster cast	-	-	0.4637	(0.191, 1.128)
Pregnant	-	-	0.1371	(0.021, 0.901)
Location of DVT (Prox vs Dist DVT):				
Distal DVT	-	-	1.5380	(1.108 2.136)
Genetic factors	Hazard Ratio (HR)	(95% CI for HR)	Hazard Ratio (HR)	(95% CI for HR)
Factor V Leiden	-	-	1.4636	(1.074, 1.994)
Laboratory factors	Hazard Ratio (HR)	(95% CI for HR)	Hazard Ratio (HR)	(95% CI for HR)
Fibrinogen	0.7548	(0.573, 0.995)	-	-
Protein C	0.9905	(0.981, 1.000)	-	-
Factor x	1.0099	(1.000, 1.020)	-	-
Factor xi	1.0075	(0.999, 1.016)	-	-
APC ratio *	1.3064	(1.050, 1.626)	-	-
Factor viii*	2.5236	(1.464, 4.351)	-	-
D-dimer *	1.2850	(1.000, 1.652)	-	-

* log transformed factors.

5.2.4 Check of the Proportional Hazards Assumption

Before using any Cox predictive model we need to have an indication of whether the proportional hazards (PH) assumption holds or not, and possibly to what degree. There are many ways to check the (PH) assumption. Here we have chosen to check the validity of (PH) assumption by using a statistical test (Grambsch and Therneau (1994)) implemented in **cox.zph(.)** function from the **survival** R package. Further we provide just one time an illustration of graphical diagnostic based on the scaled Schoenfeld residuals for model A. The Schoenfeld residuals plots for this model are presented in the Supplement figure 7.3. A systematic departure from a horizontal line is an indication that proportional hazards assumption are violated, i.e. if that line is fairly

flat and straight, then PH assumption is supported. We see no pattern with time in figure 7.3, hence the assumption of proportional hazards appears to be supported for all covariates of our models.

Our conclusion from the diagnosis of the Schoenfeld residuals figure 7.3 is also supported by the statistical test result for the proportional hazards assumption table 5.2 and table 5.3. The output from the displayed test results in tables 5.2 and 5.3 are non-significant for model A as well as for model C, indicating no violation evidence for the (PH) assumption.

Table 5.2: PH assumption numerical test for model A

	rho	chisq	p
Surgery	-0.02	0.05	0.82
Hormone	-0.04	0.20	0.66
Gender	-0.09	1.06	0.30
TypeVT2(PE)	-0.05	0.31	0.58
TypeVT3(PE+DVT)	0.16	3.64	0.06
Fibrinogen	-0.15	2.16	0.14
Protein C	-0.05	0.38	0.54
Factor X	-0.13	2.69	0.10
Factor XI	0.14	2.67	0.10
APC ratio	0.05	0.32	0.57
Factor VIII	0.16	4.12	0.04
D-dimer	-0.11	1.84	0.17
GLOBAL		18.08	0.11

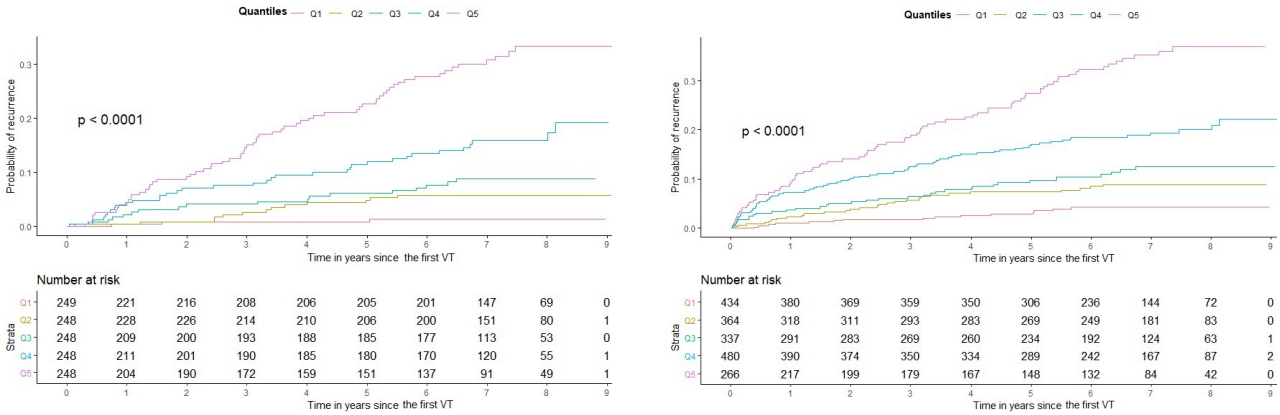
Table 5.3: PH assumption test for model C

	rho	chisq	p
Surgery	0.05	0.53	0.47
Plaster-cast	-0.02	0.07	0.80
Pregnant	-0.06	0.74	0.39
Hormone	-0.03	0.23	0.63
Location VT	-0.07	1.30	0.25
Gender	-0.04	0.32	0.57
TypeVT2 (PE)	-0.06	0.93	0.33
TypeVT3 (PE+DVT)	0.11	3.09	0.08
Leiden V	0.03	0.20	0.66
GLOBAL		6.95	0.64

5.2.5 Predictive value of the different models

Figure 5.1 illustrates the Kaplan-Meier curves for quintiles of the prognostic score for the backward selected models A and C. In order to examine the model ability of discrimination between risk of recurrence among patients, inverse Kaplan Meier plots for the observed risk of recurrence in quintiles of the prognostic scores were generated (figure 5.1). The prognostic score for each patient was calculated by the linear predictor : $\beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p$, of the model and then devided into 5 risk groups of patients, using the 20, 40, 60 and 80% quintiles of the linear predictor estimate.

Increasing quintiles of the prognostic score in figure (5.1) corresponded to an increased observed risk of recurrence. Patients in risk group 1 (figure 5.1 a and b) displayed a low recurrence risk, whereas patients in the 5th risk group showed a high risk. In general, we see a good discrimination between the 5 risk groups, which is also supported by a small log-rank p-value ($p< 0.0001$). More importantly, we observe an increase in model ability to distinguish between 5 risk groups when moving form a model encompassing 8 variables (model C) to a model with 11 variables (model A). Furthermore, it is also noticeable that risk groups 1 and 2 are barely distinguishable for the first 2.5 years after the of the first VT for model A.



(a) model A (b) model C

Figure 5.1: Probability of recurrence stratified by quintiles of prognostic scores of the backward selected models

5.3 Selection of the risk factors by lasso method

In this section, we provide an overview of the results of a regularized method (lasso). The analyses for lasso were performed in the R-software environment version 3.4.2. To perform lasso model selection on our dataset, **glmnet** package was used for this purpose. This package has possibility to apply lasso method for a Cox regression model. In order to overwhelm the model instability problem by lasso, we have implemented percentile lasso in conjunction with the ordinary lasso. The technical details for this approach are explained in section (Percentile lasso 4.4).

We estimated the percentile $\hat{\theta}$ by following the steps of percentile lasso algorithm in (4.4.3), where we use M=100. The percentile is estimated to be $\hat{\theta} = 95\%$. i.e. we extract 95% percentile $\hat{\lambda}$ value from $\Lambda(100)$ values of the optimal tuning parameters. This percentile correspond with a value of $\hat{\lambda}_A = 0.01398578$ for model A. By repeating the same steps for model C as for model A, we got a $\hat{\lambda}_C = 0.004330508$. Lastly, we then have fit the ordinary lasso at this optimal values.

5.3.1 Model A

By using the estimated $\hat{\theta} = 95\%$ percentile of $\Lambda(100)$ values of the optimal tuning parameters. The optimal lambda was found to be $\hat{\lambda}_A = 0.01398578$. This corresponds to a model with 10 predictors among 38 candidate variables that represent the full model (model A). The estimated hazard ratios (HR) from the Cox proportional hazard model are summarized in the left-side of table 5.4.
The strong predictors are factor VIII with HR =1.82 followed by gender (male) HR=1.79 and type of the first VT (PE+DVT) with HR=1.58. The remaining predictors have more modest effects with hazard ratios smaller than 1.3, these are posthrombotic syndrome 2 with HR = 1.29, factor V Leiden with HR = 1.28, VWF with HR= 1.21 and D-dimer with HR= 1.06.
Finally, we noticed a category of predictors that have a mild protective effect, these are: hormone use with HR = 0.73, surgery with HR= 0.79, type of the first VT (PE) with HR=0.85 and pregnant with HR= 0.99.

5.3.2 Model C

By repeating the same steps for model C i.e. a model that contains clinical and genetic factors, we found $\hat{\theta} = 95\%$ which corresponds with an optimal $\hat{\lambda}_C = 0.004330508$. This resulted in a model with 12 predictors among 18 candidate variables that represent the model C. The estimated hazard ratios (HR) of the Cox proportional hazard model are summarized on the right-side of table 5.4.

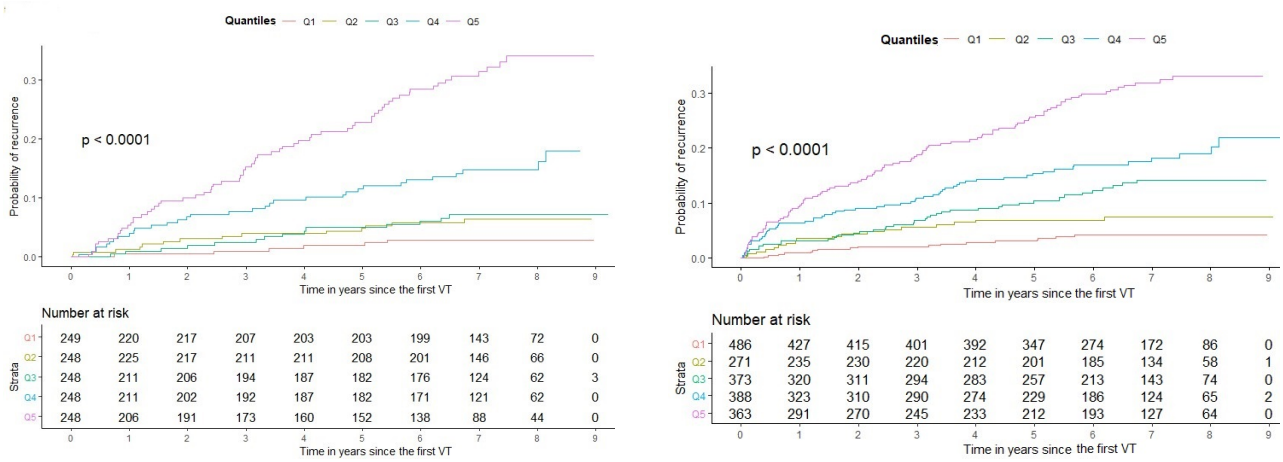
We noticed that some predictors have a strong negative effect such as: gender (male) with HR=1.83, followed by location of VT (Distal) with a HR= 1.47, posthrombotic syndrome 2 (severe) with HR = 1.42, type of the first VT (PE+DVT) with a HR =1.40 and factor V Leiden with HR = 1.36. Only one predictor, i.e. blood type with HR = 1.16, showed a modest effects with hazard ratios $1 <$ HR < 1.3.

Finally, we observed two categories of predictors, the first have a strong protective effect, these are: pregnant with HR = 0.33, surgery with HR= 0.51, hormone use with HR= 0.53 and plaster cast with HR= 0.62, and the second category have a mild protective effect, these are cardiovascular disease with HR= 0.76, posthrombotic syndrome 1 (mild) with HR=0.84, type of the first VT (PE) with a HR =0.93 and hospitalization with HR= 0.96.

5.3.3 Predictive value of the different models

Figure 5.2 illustrates the Kaplan-Meier curves for quintiles of the prognostic score for the selected models by lasso method. In order to examine the model ability of discrimination between risk of recurrence among patients, inverse Kaplan Meier plots for the observed risk of recurrence in quintiles of the prognostic scores were generated (figure 5.2). Increasing quintiles of the prognostic score in figure 5.2 corresponded with an increased observed risk of recurrence.

In general, we see a good discrimination between the 5 risk groups, which is also supported by a small log-rank p-value ($p < 0.0001$). More important, we observe an increase in model ability to distinguish between 5 risk groups when moving form a model encompassing 10 variables (model A) to a model with 12 variables (model C). Furthermore, It is also noticeable that risk groups 2 and 3 are barely distinguishable for model A, whereas for model C this was just the case for the first 2.5 years after the first VT .



(a) model A (b) model C

Figure 5.2: Probability of recurrence stratified by quintiles of prognostic scores of the lasso selected models

5.3.4 Predictors of recurrent thromboembolism models by lasso

Table 5.4 represents a summary of selected variables by mean of the lasso. The following predictors were found to be common among the models: surgery, hormone use, gender and type of the first VT (PE and PE & DVT), pregnant, PTS 2 (severe) and factor V Leiden.

Furthermore, the following laboratory factors were additionally predictive for recurrence event for model A: factor VIII, D-dimer, and VWF. On the other hand, the following clinical predictors were additionally predictive for recurrence for model C: plaster cast, location VT, cardiovascular disease and hospitalization. Furthermore, one extra genetic factors was also predictive for a recurrent event i.e. blood-type.

Table 5.4: Overview of the selected variables by lasso with their corresponding regression coefficients across models A and C

	Model A		**Model C**	
Clinical factors	hazard ratios (HR)	95% CI	hazard ratios (HR)	95% CI
Surgery	0.7919	-	0.5147	-
Hormone use	0.7376	-	0.5375	-
Sex (male)	1.7980	-	1.8359	-
Type of 1st VT(DVT,PE,PE+DVT):				
PE	0.8540	-	0.9324	-
PE+DVT	1.5829	-	1.4040	-
Plaster cast	-	-	0.6206	-
Pregnant	0.9981	-	0.3291	-
Location of DVT (Prox vs Dist DVT):				
Distal DVT	-	-	1.4784	-
Cardiovascular disease	-	-	0.7653	-
Posthrombotic syndrome 1	-	-	0.8443	-
Posthrombotic syndrome 2	1.2988	-	1.4210	-
Immobilization	-	-	0.9622	-
Genetic factors	hazard ratios (HR)	95% CI	hazard ratios (HR)	95% CI
Factor V Leiden	1.2820	-	1.3634	-
Blood type	-	-	1.1614	-
Laboratory factors	hazard ratios (HR)	95% CI	hazard ratios (HR)	95% CI
Factor VIII*	1.8200	-	-	-
D-dimer *	1.0635	-	-	-
VWF *	1.2159	-	-	-

* log transformed factors.

5.3.5 Check of the Proportional Hazards Assumption

In order to check the validity of PH assumption, we have performed a statistical test using the cox.zph() function from R package. The cox.zph() function from R is not applicable for a glment nor for a coxnet objects, and in order to circumvent this problem, we extracted the selected covariate by lasso from model A as well as model C, subsequently we built a multivariate Cox regression model for each one (unpenalized models). The results of such a statistical test are displayed in the tables 5.5 and 5.6. Neither the covariates nor the global test is statistically significant ($p > 0.05$) for both models. This is an indication of no violation evidence for the (PH) assumption. We conclude that the PH assumptions are not violated for these models.

	rho	chisq	p
Surgery	-0.00	0.00	0.98
Pregnant	-0.04	0.00	1.00
Hormone	-0.03	0.13	0.71
Gender	-0.10	1.26	0.26
TypeVT2	-0.07	0.70	0.40
TypeVT3	0.17	3.72	0.05
PTS_J1	-0.11	1.55	0.21
PTS_J2	-0.02	0.03	0.85
Leiden V	0.05	0.38	0.54
factor VIII	0.07	0.39	0.53
VWF	0.05	0.24	0.62
D-dimer	-0.15	3.53	0.06
GLOBAL		13.02	0.37

Table 5.5: PH assumption test, model A

	rho	chisq	p
Surgery	0.06	0.75	0.39
Plaster cast	-0.01	0.05	0.82
Hospitalization	-0.03	0.23	0.63
Pregnant	-0.05	0.73	0.39
Hormone	-0.03	0.23	0.63
Cardio-disease	-0.04	0.34	0.56
Location VT	-0.07	1.24	0.27
Gender	-0.03	0.27	0.60
TypeVT2	-0.06	0.90	0.34
TypeVT3	0.11	3.12	0.08
PTS_J1	-0.02	0.12	0.73
PTS_J2	0.02	0.12	0.73
Blood-type	-0.04	0.46	0.50
Leiden V	0.02	0.15	0.70
GLOBAL		8.22	0.88

Table 5.6: PH assumption test, model C

5.4 Closed testing

In the current section, we will present the results of the closed testing procedure as was discussed in more detail in section 4.5. Since model A has more than 30 variables, the closed testing procedure will be computationally intensive. Therefore we will restrict the application of closed testing procedure in this thesis only to model C . Further, two important concepts from the closed testing section are reintroduced here. Remember that **the defining set** is defined as ***a collection of sets of hypotheses with the property that for each set in the collection we can be confident that it contains at least one true discovery*** i.e $\beta \neq 0$ [17]. Furthermore, these sets of variables are the smaller sets for which the same statement holds [20].
In addition, remember that **The shortlist** is defined as ***a collection of sets of hypotheses with the property that at least one of the sets in the collection contains only true discoveries*** i.e. $\beta \neq 0$. Further, with shortlist we aim to identify a collection with the smallest possible number of variables that exhibit a significant effect on the outcome variable. In other words, we aim to identify the smallest models that can fit as good as the full model.

The results of defining set are summarized in table 5.7. As can be seen, there is no singleton hypothesis. This implies that none of the variables are indispensable. On the other hand, large number is observed of set of the hypothesis (25) containing 2 variables, 12 sets containing 3 variables, 2 sets containing 4 variables and 4 sets containing 5 variables. This could be explained by the fact that the amount of evidence for a single variable is not sufficient in our data, due to multicollinearity among variables or more individual predictors are close to significant and thus collectively form overall significant model.

Furthermore, a heat-map correlation among these covariates is displayed in figure 5.3. By inspecting the correlation among these covariates, a higher correlation coefficient is observed between hormone and gender (-0.63), since the hormone is only used by women. Further, a high positive correlation was observed between hospitalization and surgery (0.54). This might be explained by having a surgery could lead to hospital stay. An intermediate correlation (-0.26) is observed between location VT and type of the first VT, as location VT (distal vs proximal) indicates the physical position of VT, that might correspond with type of first VT i.e. deep vein

thrombosis and pulmonary embolism.
Lastly, a moderate correlation (0.23) is noted between pregnancy and hospitalization, which can be attributed to the fact that pregnancy might lead to frequent hospital stays. For the remaining covariates, the observed correlation coefficients were very low between variables.

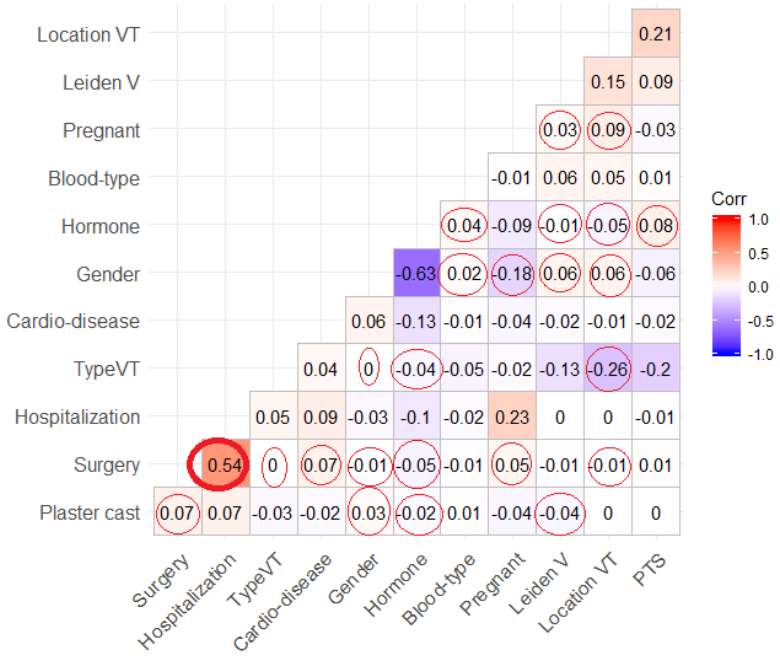


Figure 5.3: Correlation matrix for the defining set variables

Table 5.7: Defining rejection set results

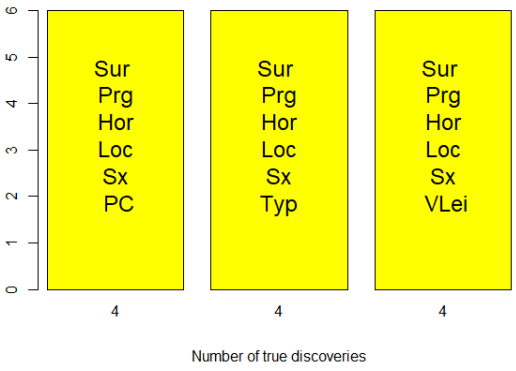
Defining sets	Number of variables
Surgery, Plaster cast	2
Surgery, Hospitalization	2
Surgery, Pregnant	2
Surgery, Hormone	2
Plaster cast, hormone	2
Pregnant, Hormone	2
Surgery, Cradio disease	2
Surgery, Location	2
Pregnant, Location	2
Hormone, Location	2
Surgery, Gender	2
Plaster cast, Gender	2
Pregnant, Gender	2
Hormone, Gender	2
Location, Gender	2
Surgery, Type VT	2
Hormone, Type VT	2
Location,Type VT	2
Gender, Type VT	2
Hormone, Post thromboti syndrome	2
Gender , blood type	2
Surgery, Leiden V	2
Pregnant,Leiden V	2
Hormone, Leiden V	2
Gender, Leiden V	2
Plaster cast, Pregnant, cardio disease	3
Plaster cast,Pregnant, Type VT	3
Pregnant, Cardio disease, Type VT	3
Plaster cast,Pregnant, PTS	3
Gender,Cardio disease, PTS	3
Pregnant,Type VT, PTS	3
Plaster cast, Pregnant,Blood type	3
Hormone,Cardio disease, Blood type	3
Pregnant,Type VT,blood type	3
Surgery, PTS,blood group	3
Plaster cast,Location, Leiden V	3
Location,PTS,Leiden V	3
Pregnant, Cardio disease,PTS, blood type	4
Cardio disease, location,blood type, Leiden V	4
Plaster cast,Cardio disease, Location,PTS,blood type	5
Plaster cast,cardio disease, Type VT, PTS, Leiden V	5
Plaster cast, cardio disease,Type VT, Blood type, Leiden V	5
Plaster cast,Type VT, PTS,blood type, Leiden V	5

The closed testing procedure in this thesis was applied by means of **close()** function from the cherry package in R [17], with a P-value = 0.1. Among 17 variables contained in model C

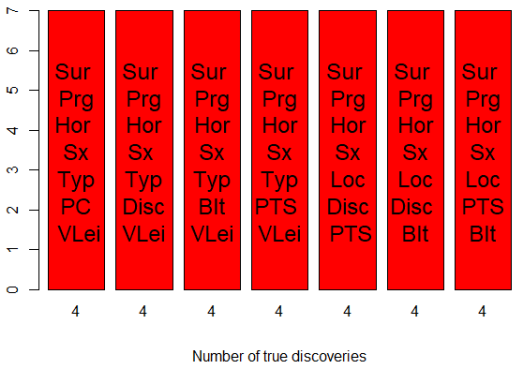
(i.e model with clinical and genetic factors), the shortlist has resulted in a collection of 23 sets of variables i.e. 23 possible minimal models that can fit as good as the full model. This collection contains : 3 models with 6 variables , 7 models with 7 variables, 5 models with 8 variables, 6 models with 9 variables , and as for last 2 models with 10 variables. This collection of models was displayed through figure 5.4 to 5.6.

The models from the shortlist collection are sorted by the order of number of variables in such way each histogram will represent one category of this collection. In order to display such histograms, we have abbreviated our covariates names as follows : Sur= surgery, Prg=pregnant, Hor=hormone, Loc= location first VT, Sx =gender, PC= plaster cast, Typ =type of first VT, VLeid = factor V Leiden, Disc = cardiovascular disease, Blt= blood-type, PTS= post-thrombotic syndrome, Hosp = hospitalization.

For instance, figure 5.4 (a) displays an intersection of 6 hypotheses i.e a model with 6 variables, the first bar from the left to the right contains the following variables: surgery, pregnant, hormone, Location VT, gender and plaster cast variables. Note that each bar of the histogram represents an intersection hypotheses of 6 variables (i.e each bar forms a model with 6 variables) and that each one contains likely at least 4 variables that were declared to be true discoveries (true variables $\beta \neq 0$) and at most 2 variables are false discoveries (noise $\beta = 0$).



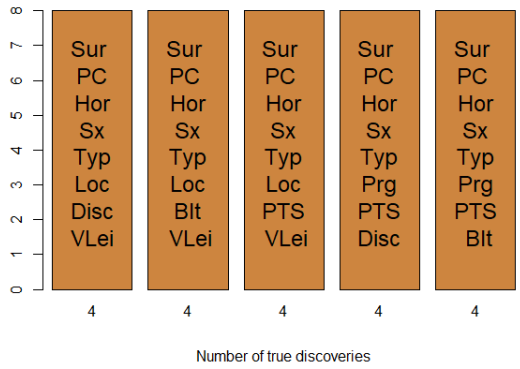
(a) All possible models with 6 variables resuted by shortlist



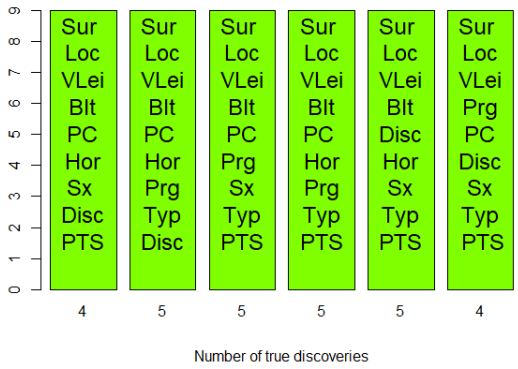
(b) All possible models with 7 variables resuted by shortlist

Figure 5.4: Shortlist results: all possible models with 6 and 7 variables and their number of true discoveries (i.e true variables $\beta \neq 0$)

In figure 5.4 (b), a set of bars are displayed, each one contains 7 variables that together form a model that fit as good as the full model. Furthermore, each model contains likely at least 4 true covariates and at most 3 false discoveries (noise). Further, in figure 5.5 (a) each bar contains a model with 8 variables, with at least 4 true discoveries (true variables i.e $\beta \neq 0$) and 4 false discoveries (noise). In addition, figure 5.5(b) is an illustration of all sets of models with 9 variables contained in shortlist collection, each models contains at least 4 or 5 true discoveries and at most 4 or 5 false discoveries.



(a) All possible models with 8 variables resuted by shortlist



(b) All possible models with 9 variables resuted by shortlist

Figure 5.5: Shortlist results: all possible models with 8 and 9 variables and their number of true discoveries (i.e true variables $\beta \neq 0$)

The last figure 5.6, each bar constructs a model of 10 variables, in which at least 5 are true discoveries and at most 5 are false discoveries.

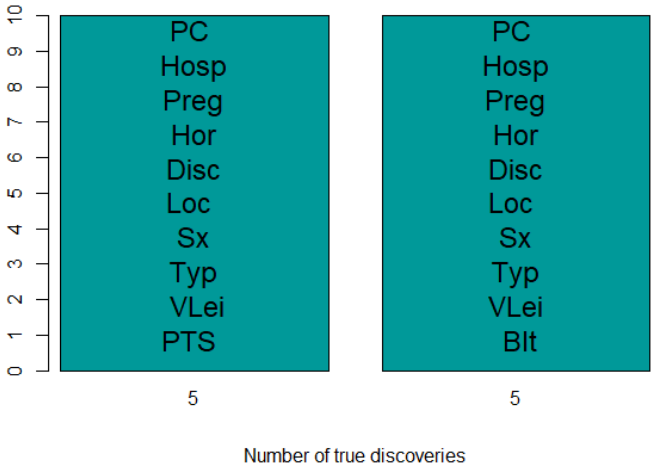


Figure 5.6: Shortlist results: all possible models with 10 variables and their number of true discoveries (i.e true variables $\beta \neq 0$)

Now let us choose one set of variables i.e a model displayed in the shortlist results, and compare it with other shortlist sets figure 5.4 to 5.6. Note that any shortlist set is the smallest models that can fit as good as the full model. The researcher may consider one set of variables from the shortlist as his main model, and further compare it with the remaining sets. In our example, we have chosen the left one model of figure 5.4 as our main model with the variables

: surgery, plaster cast, pregnant, hormone use, location of the first VT and gender, this was denoted as $modelC_6$. This model was displayed in green color in the center of all figures through 5.7 to 5.11.

For instance, if a researcher had chosen this model to be used as a prediction model for the recurrence of thrombosis, he can be 90% confident that his model is containing at least 4 true discoveries (TD) and at most 2 false discoveries (FD). The main distinction of the closed testing procedure from the other discussed methods in this thesis is manifested in the fact that this procedure provides the researcher with a broad view to all possible alternative models that can perform as good as the full model. In addition, this method will give the researcher also the possibility to change or substitute some covariates.

Figures 5.7 to 5.11 display all possible model alternatives to the main model ($modelC_6$). For instance, in figure 5.7 the alternative 1 model is a model that is roughly the same as the $modelC_6$, with the only difference is that $modelC_6$ contains plaster cast as a covariate whereas in the alternative 1 model the plaster cast covariate is substituted by Type of first VT. Further, the alternative 2 model is a model that contains the same variables as the $modelC_6$, with the only difference is that $modelC_6$ contains plaster cast as a covariate whereas in the alternative 2 model the plaster cast covariate is substituted by factor V Leiden. This can further explained in the context of missing variable information. Suppose for in stance now the plaster cast covariate, patient plaster cast information for the last 3 months before VT, is often missing in practice. In this case, the researcher can choose to substitute the plaster cast covariate by using one of the next covariates: Type of first VT or factor V Leiden. In this case, an alternative model (alternative 1 or 2) will have the same characteristics as $modelC_6$ i.e. these alternatives are models that contain at least 4 true discoveries (true covariates) and at most 2 false discoveries (noise).

In addition, the researcher will have the ability to replace the plaster cast variable with a joint variables. So for example, he can choose to substitute plaster cast by a joint variables PTS and cardiovascular disease figure 5.7 (alternative 3). Another alternative is to replace plaster cast covariate with joint variables blood-type and PTS (alternative 4) or with blood type and cardiovascular disease covariates (alternative 5). When one of these last alternative models (alternative 3, 4 or 5 figure 5.7) is chosen, the researcher can be 90% confident that his model will contain at least 4 true discoveries (TD) and 3 at most false discoveries (FD). To clarify more what is said, consider for instance the alternative 3, this model contains the same variables as $modelC_6$, the only difference is that plaster cast in the main model ($modelC_6$) is now replaced in alternative 3 by the joint variables PTS and Cardiovascular disease.

Let us right now proceed to figure 5.8. This figure is a summary of all possible alternative models for location VT and location VT and plaster cast together. To explain this further, consider the model $modelC_6$ to be the main chosen model by the researcher. Suppose for instance that patient information for location VT (Proximal vs. distal DVT) is missing. In this case, the researcher can choose to replace the location VT covariate by a joint variable (alternative 4) i.e. factor V Leiden and Type of the first VT (i.e. PE or PE&DVT). In this case, the model alternative 4, is a model containing at least 4 true discoveries and at most 3 false discoveries. Furthermore, this model (alternative 4) is a model containing the same variables as $modelC_6$, with the only difference is that $modelC_6$ contains location VT as a covariate, whereas in the alternative 4 the location VT covariate is substituted by factor V Leiden and Type the first VT covariates together.

We remain our focus in figure 5.8 and let us move to investigate the triple substitution. i.e. evaluating the possibility to replace the location VT covariate with the triple covariates: Type VT, PTS and cardiovascular disease (alternative 5) or by using blood-type, PTS and type of the first VT covariates (alternative 6). In this case, the researcher can be 90% confident to have an alternative model (5 or 6) that contains at least 4 true discoveries and at most 4 false discoveries. Note also that model alternative 5 is a model containing the same variables as the main model $modelC_6$, with the only difference is that $modelC_6$ contains location VT as a covariate whereas in the alternative 5 this covariate is replaced by a triple covariates type of the first VT, PTS, and cardiovascular disease.

Similarly, alternative 6 is a model containing the same variables as $modelC_6$, with the only difference is that $modelC_6$ contains location VT as a covariate, whereas in the alternative 6 the location VT covariate is substituted by a triple covariates blood-type, PTS and type of the first VT. Hence we move from a model with 6 covariates (4 TD and 2 FD) toward a model containing 8 variables (4 TD and 4 FD). One extra observation is noted for these alternatives i.e. they do only differ in one covariate. In alternative 5 the researcher will have to use cardiovascular disease covariate while this covariate is substituted by blood-type in the alternative 6.

Last but not least, let us consider the replacement of the joint covariates i.e. plaster cast and location VT figure 5.8. In case the researcher is missing information on both covariates i.e. plaster cast and location VT he may consider three alternative models 1, 2 or 3 . Suppose he/she had chosen to use model alternative 1, therefore he/she can be 90% confident to have an alternative model that contains at least 4 true discoveries and at most 3 false discoveries. For the model alternative 1, the joint covariates plaster cast and location VT are substituted together by the triple covariates Type VT, factor Leiden V and Cardiovascular disease. In addition, alternative 2, is a model where the joint covariates plaster cast and location VT are replaced together by the triple covariates type of the first VT, factor V Leiden and blood-type.
Similarly, for the model alternative 3, the joint covariates plaster cast and location VT are substituted together by the triple type of the first VT, factor V Leiden, and PTS. Note also in these three alternatives (1,2 and 3) covariates type of the first VT, factor V Leiden are common among all three alternatives.

Analogously, figures 5.9 through 5.11, display all possible model alternatives that a researcher might consider to substitute the single covariates pregnant (within 3 months before VT), gender (male), hormone use (at the time of VT, including: hormone replacement therapy and hormonal contraceptive) and Surgery (within 3 months before VT) as well as for the joint covariates pregnant & plaster cast together (joint replacement). If we considered the last figure 5.11, that displays two alternative models in case of the surgery information is missing. The investigator will have two possible model alternatives to consider. Each model alternative has at least 5 true discoveries and at most 5 false discoveries. In each model alternative, the surgery covariate is replaced by 5 different covariates. Thus for alternative 1, besides that this contains 4 main covariates of the model $modelC_6$: plaster cast, pregnant, hormone use, location VT, and gender. This model will replace the surgery by type of the first VT, factor V Leiden, cardiovascular disease, PTS, and hospitalization (immobility in bed in hospital, within 3 months before VT). Hence moving from a model with 6 covariates ($modelC_6$ with 4 TD and 2 FD) to a model with 10 covariates (alternative 1 with 5 TD and 5 FD). Furthermore, we have noticed that these alternatives (1 and 2) have 4 covariates in common, these were colored in the same color: Type VT, factor Leiden V, Cardiovascular disease and Hospitalization. On the contrary, these alternatives (1 and 2) differ only in one covariate PTS (alternative 1) vs. blood type (alternative

2).

Figure 5.7: A graphical illustration to illustrate different minimal models. In the center, the model we started with ($modelC_6$ main model), and the colored nodes are the alternative models. In the leaves of the colored nodes, plaster cast (in green) from the main model is replaced by the colored variables in the alternative model. With TD we mean true discoveries (true covariates) and FD stand for false discoveries (noise).

Figure 5.8: A graphical illustration to illustrate different minimal models. In the center, the model we started with ($modelC_6$ main model), and the colored nodes are the alternative models. In the leaves of the colored nodes, location VT or location & plaster cast (in green) from the main model is replaced by the colored variables in the alternative model. With TD we mean true discoveries (true covariates) and FD stand for false discoveries.

Figure 5.9: A graphical illustration to illustrate different minimal models. In the center, the model we started with $modelC_6$ (main model), and the colored nodes are the alternative models. In the leaves of the colored nodes, pregnant or pregnant & plaster cast (in green) from the main model is replaced by the colored variables in the alternative model. With TD we mean true discoveries (true covariates) and FD stand for false discoveries.

Figure 5.10: A graphical illustration to illustrate different minimal models. In the center, the model we started with ($modelC_6$, main model), and the colored nodes are the alternative models. In the leaves of the colored nodes, hormone use or gender (in green) from the main model is replaced by the colored variables in the alternative model. With TD we mean true discoveries (true covariates) and FD stand for false discoveries.

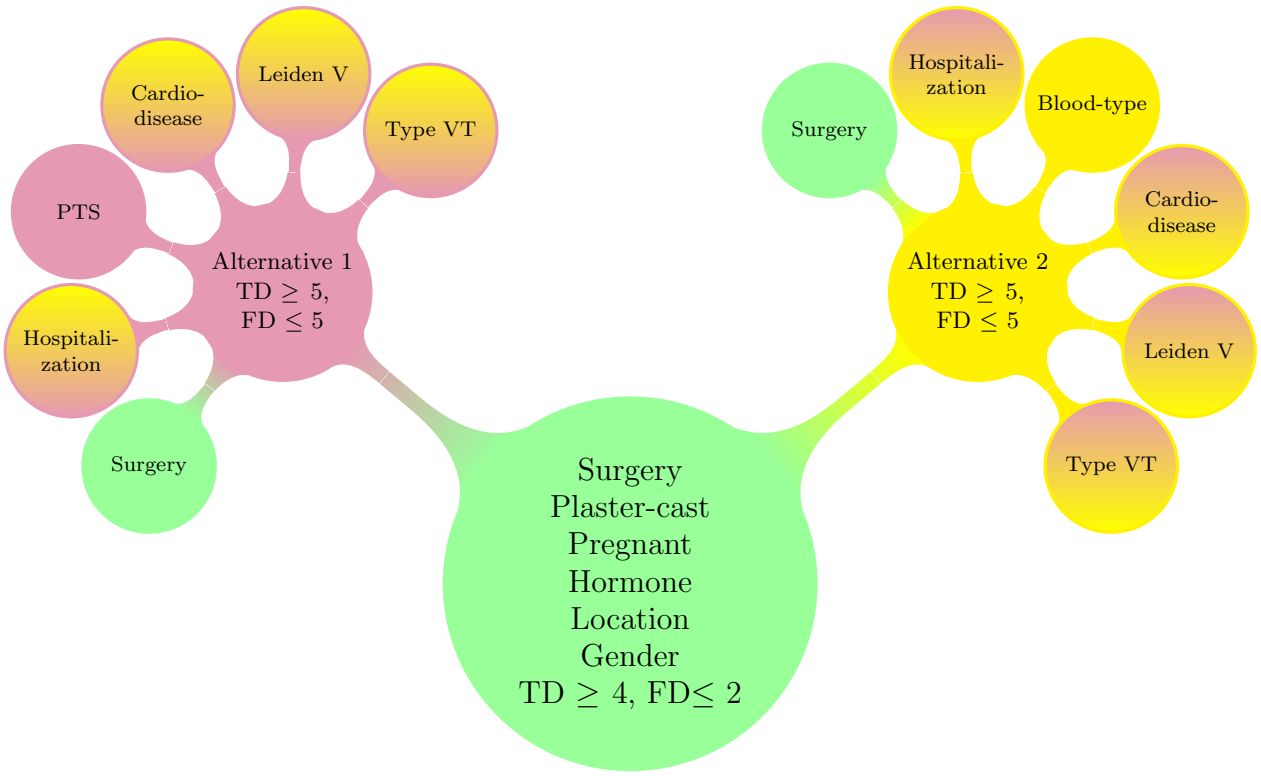


Figure 5.11: A graphical illustration to illustrate different minimal models. In the center, the model we started with ($modelC_6$, main model), and the colored nodes are the alternative models. In the leaves of the colored nodes, surgery (in green) from the main model is replaced by the colored variables in the alternative model. With TD we mean true discoveries (true covariates) and FD stand for false discoveries.

5.4.1 Selected models

In this section, we will describe the results of the Cox regression models obtained by the closed testing procedure. In table 5.8 we provide an overview of some of the Cox regression models from the shortlist. From each of the shortlist models with the same number of variables, we selected one model, therefore we do not expect a significant difference in terms of discrimination power among models of the same size. Of note, that all models in the shortlist do not differ significantly from the full model. The choice of the presented models in table 5.8 was based on a balance between the number of missing values and the optimal ease of use of the model in clinical practice (i.e. clinical variables only). Based on these criteria, we will present here only one model from each set category of the shortlist collection (figure 5.4 to 5.6).

5.4.2 Predictors of recurrent thromboembolism models

In the following subsections, I will describe in more details the results of two models (Model C_6 and Model C_9) displayed in the table 5.8.

Model C_6:

A strong positive effect is observed for some predictors i.e. patients with one of these factors will have a lower risk of recurrence of thrombosis. The most important predictors with a strong positive effect are pregnancy with a HR=0.12 and 95%CI(0.01, 0.87), surgery with HR= 0.41 and 95%CI (0.25, 0.67), followed by plaster cast HR= 0.43 and 95%CI (0.18, 1.06), and hormone use HR= 0.49 and 95%CI (0.31, 0.79). In addition, a strong negative effect was observed among the following predictors: gender with an HR=1.92 and a 95%CI of (1.35, 2.75) followed by location VT (Distal DVT) with an HR=1.82 and a 95%CI of (1.37,2.42).

Model C_9:

A strong positive effect is observed for the same predictors as model C_6. i.e. pregnancy, surgery and plaster cast. Additionally, model C_9, has some unique predictors with strong positive effect: cardiovascular disease with a HR= 0.67 and a 95%CI of (0.35,1.26), PTS 1 HR= 0.7 and a 95%CI of (0.41,1.18), and a slightly positive effect was observed for type of the first VT (PE) with HR= 0.9 and a 95%CI of (0.65,1.23). Furthermore, a strong negative effect is observed among the following unique predictors for model C_9: PTS2 with a HR=1.5 and a 95%CI of (0.78,2.9), type of the first VT (PE & DVT) with a HR=1.45 with a 95%CI of (0.96,2.18), factor V Leiden with a HR=1.4 and 95%CI of (1.02,1.9).

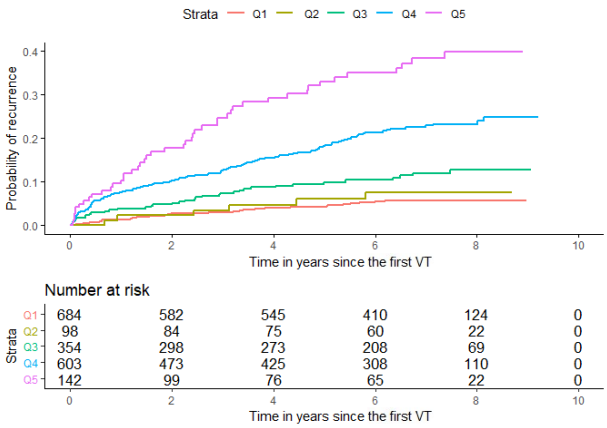
Table 5.8: Variables and their corresponding HR across closed testing selected models

Variables	model C_6	model C_7	model C_8	model C_9	model C_{10}
Clinical factors	HR & (95% CI)	HR & (95% CI)	HR & (95% CI)	HR & (95% CI)	HR & (95% CI)
Surgery	0.41 (0.25, 0.67)	0.41 (0.25, 0.76)	0.43 (0.26, 0.71)	0.45 (0.27, 0.73)	-
Hormone use	0.49 (0.31,0.79)	0.48 (0.30,0.77)	0.46 (0.29,0.74)	-	0.46 (0.29, 0.73)
Gender (male)	1.92 (1.35,2.75)	1.9 (1.32,2.71)	1.87 (1.30,2.68)	2.8 (2.11,3.7)	1.78 (1.24, 2.56)
Type of 1st VT					
PE	-	-	0.7 (0.53,0.94)	0.90 (0.65, 1.23)	0.88 (0.64, 1.21)
PE+DVT	-	-	1.44 (0.96,2.17)	1.45 (0.96, 2.18)	1.50 (1.003, 2.27)
Plaster cast	0.43 (0.18,1.06)	-	0.43 (0.17,1.05)	0.47 (0.19,1.16)	0.45 (0.18, 1.11)
Pregnant	0.12 (0.01,0.87)	0.12 (0.01,0.88)	0.13 (0.01,1,006)	0.17 (0.02,1.27)	0.13(0.01, 0.98)
Location of DVT:					
Distal DVT	1.82 (1.37,2.42)	1.8(1.34,2.41)	-	1.53 (1.10,2.13)	1.53 (1.10, 2.14)
Cardio-disease	-	0.6 (0.31,1.13)	0.62 (0.33,1.18)	0.67 (0.35,1.26)	0.60 (0.32, 1.14)
PTS					
PTS1	-	0.74 (0.44,1.24)	0.70 (0.41,1.19)	0.7 (0.41, 1.18)	0.71 (0.42,1.20)
PTS2	-	1.77 (0.92,3.4)	0.62 (0.96,3.53)	1.5 (0.78, 2.9)	1.48 (0.77, 2.85)
Hospitalization	-	-	-	-	0.64 (0.42, 0.99)
Genetic factors	HR & (95% CI)	HR & (95% CI)	HR & (95% CI)	HR & (95% CI)	HR & (95% CI)
Factor V Leiden	-	-	-	1.4 (1.02,1.90)	1.44 (1.06, 1.97)
Blood type	-	-	-	-	-

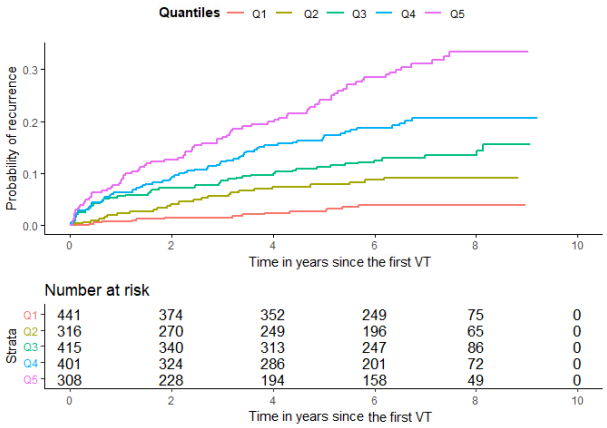
5.4.3 Predictive value of the different models

Figure 5.12 illustrates the Kaplan-Meier curves for quintiles of the prognostic score for closed testing models: model C_6 and model C_9. The remaining models (model C_7, model C_8 and model C_{10}) are given in the supplementary 7.3.3. In order to examine the model ability of discrimination between risk of recurrence among patients, inverse Kaplan Meier plots for the observed risk of recurrence in quintiles of the prognostic scores were generated (figure 5.12). The prognostic score for each patient was calculated by discretizing the linear predictor : $\beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p$, of the model into 5 risk groups of patients, using the 20, 40, 60 and 80% quintiles of the linear predictors estimate.

Increasing quintiles of the prognostic score in figure 5.12 corresponded in an increased observed risk of recurrence. Patients in risk group 1 (figure 5.12 a and b) display a low recurrence risk, whereas patients in the 5th risk group show a high risk. In general, we see a good discrimination between the 5 risk groups, which is also supported by a small log-rank p-value ($p< 0.0001$). Importantly, we observe an increase in model ability to distinguish between 5 risk groups when moving form a model encompassing 6 variables to a model with 9 variables. Furthermore, it is also noticeable that risk groups 1 and 2 are barely distinguishable for the first 4 years after the first VT for model C_6.



(a) model C_6



(b) model C_9

Figure 5.12: Probability of recurrence stratified by quintiles of prognostic scores of the closed testing models

5.4.4 Check of the Proportional Hazards Assumption

In this paragraph we perform a test to check whether the PH assumption holds or not and possibly to what degree. The results of PH assumption test are displayed in table 5.9 and 5.10. Covariates are not statistically significant (p>0.05). Additionally, the global test is not statistically significant for the model ($modelC_6$) nor for model ($modelC_9$). This is an indication of no violation evidence for the (PH) assumption.
In supplementary, we have included the PH assumption test results for the remaining models (7, 8 and 10), also no indication of violation of PH assumptions was observed. We conclude that the PH assumptions are not violated for these models.

	rho	chisq	p
Surgery	0.04	0.33	0.57
Plaster-cast	-0.02	0.07	0.79
Pregnant	-0.06	0.89	0.34
Hormone	-0.02	0.12	0.73
Location VT	-0.03	0.16	0.69
Gender VT	-0.02	0.14	0.71
GLOBAL		1.58	0.95

Table 5.9: PH assumption numerical test for model C_6

	rho	chisq	p
Surgery	0.05	0.68	0.41
Plaster-cast	-0.02	0.07	0.79
Pregnant	-0.05	0.69	0.41
Hormone	-0.01	0.03	0.86
Cardio-disease	-0.03	0.26	0.61
Location VT	-0.07	1.23	0.27
TypeVT2(PE)	-0.06	0.83	0.36
TypeVT3(PE+DVT)	0.11	3.00	0.08
Blood-type	-0.04	0.48	0.49
Leiden V	0.03	0.16	0.69
GLOBAL		7.35	0.69

Table 5.10: PH assumption test for model C_9

5.5 Predictive performance of the models

In order to evaluate model performance, we used an internal validation procedure based on the bootstrap method. The corrected for optimism Harrell C statistic was calculated using 200 bootstrap samples. Each sample was drawn with replacement from the original dataset including 1241 (model A) or 1881 patients (model C), then the model was refitted in the bootstrap sample and original sample. The detailed description of bootstrap internal validation is given in section 4.6.

Table 5.11 displays a summary of the models performance in terms of C statistic. The analysis results indicate that model A selected by lasso had the highest predictive performance with a corrected Harrell C-statistics of 0.703. The discriminative performance was somewhat lower in the model A selected by backward selection with a corrected C-statistics of 0.697. On the other hand, models selected from original model C, closed testing procedure had the highest predictive performance with a corrected Harrell C-statistic values ranging between 0.689 (model C_6) and 0.683 (model C_{10}). The discriminative performance was barely lower in the model selected by lasso with a corrected C-statistic of 0.688 (model with 12 variables), followed by model selected by backward elimination with a corrected C-statistic of 0.682 (model with 8 variables).

Table 5.11: Predictive performance of the selected models by backward selection, lasso and closed testing

	Model A		**Model C**	
Methods	Harrell C and 95%CI	Corrected C **	Harrell C and 95%CI	Corrected C**
Backward selection	0.748 (0.730,0.812)	0.697	0.701 (0.680, 0.741)	0.682
LASSO	0.737	0.703	0.708	0.688
Closed testing :				
Model C_6	*	*	0.692 (0.665, 0.722)	0.689
Model C_7	*	*	0.695 (0.668,0.733)	0.689
Model C_8	*	*	0.698 (0.671,0.734)	0.687
Model C_9	*	*	0.693 (0.665,0.728)	0.684
Model C_{10}	*	*	0.697 (0.697,0.733)	0.684

* Closed testing was applied only to model C
** Harrell C statistic corrected for optimism

5.6 Models Summary

Table 5.12 provides a summary of the selected variables by the backward selection, lasso and closed testing methods. The following predictors were found to be common among all methods through all selected models (model A and model C): surgery, hormone use, gender and type the first VT (PE and PE & DVT). Furthermore, the following predictors were commonly predictive for recurrence event for model C among all methods: pregnant, plaster cast, location VT and factor V Leiden. Additionally, the following predictors were found to be common for model A selected by backward selection and lasso: D-dimer and factor VIII. Lastly, the predictors cardiovascular disease, PTS 2, hospitalization and blood type were additionally present in model C selected by lasso and closed testing.

Table 5.12: A summary of the selected candidate variables across the investigated methods. For closed testing methods all the three suggested models with 6 variables are included in the table under the names C_{6A}, C_{6B} and C_{6C}. Furthermore, by C_{alt} we mean that the variable is selected in at least one of the possible models from the shortlist collection.

		Selected variables by backward selection		**Selected variables by lasso**		**Selected variables by closed testing**			
Clinical factors	Class	model A	model C	model A	model C	C_{6A}	C_{6B}	C_{6C}	C_{alt}
Gender	Categorical	✓	✓	✓	✓	✓	✓	✓	☑
Pregnant	Categorical	-	✓	✓	✓	✓	✓	✓	☑
Cardiovascular disease	Categorical	-	-		✓	-	-	-	☑
Type of the first VT:									
PE	Categorical	✓	✓	✓	✓	-	✓	-	☑
PE & DVT	Categorical	✓	✓	✓	✓	-	✓	-	☑
Hormone use	Categorical	✓	✓	✓	✓	✓	✓	✓	☑
Surgery	Categorical	✓	✓	✓	✓	✓	✓	✓	☑
Plaster cast	Categorical	-	✓	-	✓	✓	-	-	☑
PTS 1 (mild)	Categorical	-	-	✓	-	-	-	-	☑
PTS 2 (severe)	Categorical	-	-	✓	✓	-	-	-	☑
Location VT(proximal vs distal)	Categorical		✓		✓	✓	✓	✓	☑
Hospitalization	Categorical	-	-	-	✓	-	-	-	☑
Genetic factors	Class								
factor V Leiden	Categorical	-	✓	✓	✓	-	-	✓	☑
Blood type	Categorical	-	-	-	✓	-	-	-	☑
Laboratory factors	Class								
D-dimer *	Continuous	✓	-	✓	-				
Factor II	Continuous								
Factor V	Continuous								
Factor VII	Continuous								
Factor VIII *	Continuous	✓	-	✓	-				
Factor IX	Continuous								
Factor X	Continuous	✓	-						
Factor XI	Continuous	✓	-						
Von Willebrand factor* (VWF)	Continuous	-	-	✓	-				
Protein C	Continuous	✓	-						
Fibrinogen	Continuous	✓	-						
APC ratio *	Continuous	✓	-						

* A log-transformation was decided upon after a visual check of the distribution curve and a non-normal distribution was found.

5.6.1 Nomogram for risk of recurrent VT

In this section, we will show how model C_6 can be used to develop a nomogram (figure 5.13), that can be used to compute risk scores and expected probability of recurrent VT from the individual's values for the following predictors: pregnant, surgery, hormone use, location of VT, plaster cast and gender. The nomogram's points can be calculated as follow [38], [60]: In order to determine how many points an individual will obtain for the predictors pregnant, surgery, hormone use, the location of VT, plaster cast and gender, we should draw a straight line upward to the points line. Then we sum the points obtained for each predictor which are located on the total point axis. Subsequently, drawing a straight line downward, the individual's cumulative recurrence rate after 2 and 5 years is found.

Note that we can easily find other cumulative recurrence rates too, a complete R code is provided in the supplement (8). To make this more clear we provide one example, suppose a patient is a female (0 points), without surgery (42 points), without plaster cast (39 points), not pregnant (100 points), without hormone use (33 points) and no location VT (0 points) (i.e. proximal DVT) will have a total score of 214 points that correspond to a probability of recurrent VT 0.96 and 0.90 for 2 years and 5 years respectively.

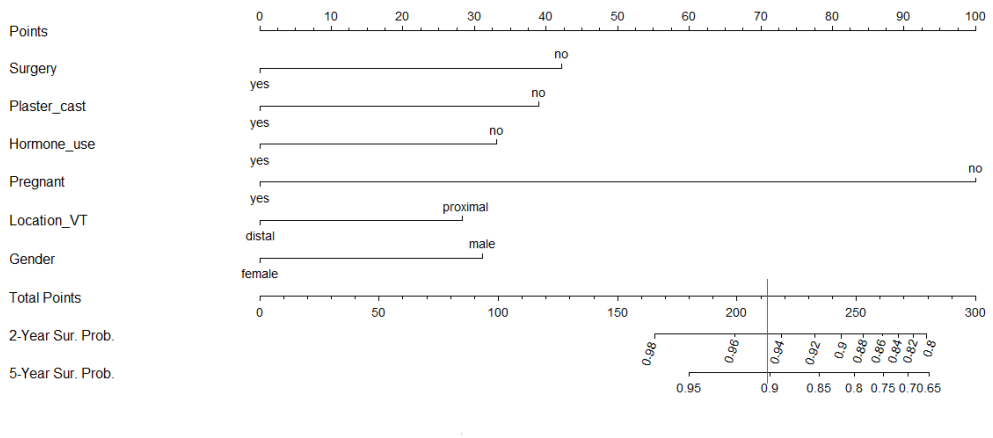


Figure 5.13: Nomogram predicting 2 years and 5 years probability of recurrent VT , for an individual with a first VT using values of pregnant, surgery, hormone use, location of VT, plaster cast and gender.

Chapter 6

Discussion

In this thesis we aimed at building a model to predict the recurrence of thrombosis by comparing three statistical methods for variable selection: backward selection, lasso in conjunction with percentile lasso and closed testing procedure as was developed by Goeman and Solari [20]. We examined the association between 38 candidate variables model A (n=1222) as well as 17 variables model C (n=1898) with recurrent VT (outcome variable). Different sets of predictors were selected by three different statistical methods. The obtained results from our study, show that there is strong evidence of an association between recurrent VT and some candidate variables investigated in this study.

Four covariates were found to be common among all of the three investigated statistical methods: surgery, hormone use, gender and type of the first VT (PE and PE & DVT). This important finding is in agreement with the three currently most used prediction models for recurrent VT i.e. the DASH score model, Vienna model, and Rodger model. Moreover, these four predictors were also common among other investigated models [50], [6],[31],[25]. Additionally, each method had a set of own specific covariates selected to be associated with the recurrent VT table (5.12).

Backward elimination method is considered as one of the most widely used model selection methods. In our study, this method was investigated with a stopping rule of p = 0.10, that has led us to select a model with 11 predictors (model A) and 8 predictors (model C). The selected models by backward selection had almost a comparable performance in terms of corrected C-statistics with lasso and closed testing. However, the size of the backward selected model A was larger than the model A selected by lasso, suggesting that some included covariates by backward selection might be irrelevant variables of the recurrent VT.

The performance of backward elimination depend probably also on the choice of removal criterion. In this thesis, we only considered the P=0.1 as the stopping rule. However, a small p-value and other stopping rules such as AIC or BIC should be considered too, that could result in a more parsimonious model with better predictive performance. Additionally, we did not consider backward elimination followed by post- selection shrinkage in the current study. This approach was proven to give a better results than the lasso in a study by van Houwelingen et.al [55].

In addition to backward elimination, penalized regression method lasso was considered in this thesis. Through cross-validating the strength of the penalty value (λ_{min} and λ_{1se}), lasso provides the researcher with a flexible model selection tool to control the model size. The important ad-

vantage of lasso over the traditional model selection methods is mostly manifested in cases when the number of observations is smaller than or close to a number of candidate variables (n<<p), or in presence of high multicollinearity among variables.

In this thesis, we faced one of the most common problems of the lasso method: model instability due to fold assignment during the cross validation. Further, we noted in our study that ordinary lasso was very sensitive to the way the variables were coded, so a small change in the variables code in R (i.e. type of the first VT as factor instead dividing type of the first VT into 2 variables), lasso selected a very different models (Supplement 7.1.1).
Further, in this thesis, we did not provide the 95% CI of the estimation and the corresponding standard errors for lasso. This is considered as a theoretical challenge for lasso and its extensions, which is still a topic of heated debate among statisticians. We therefore suggest this as a topic for further study. One last limitation of importance here is that lasso, as backward elimination, tends to select randomly just one variable from a set of highly correlated variables.

A powerful approach, percentile lasso, to handle this problem was discussed in this thesis. Percentile lasso, in a nutshell, estimates the percentile θ of a set M (i.e. M=100) of optimal tuning parameter $\hat{\lambda}$'s that were generated by ordinary lasso. Then the percentile lasso solution $\hat{\lambda}(\hat{\theta})$ is considered further as lasso tuning parameter value.
One small experiment was conducted in this thesis to verify if percentile lasso can help to achieve model stability. Undoubtedly, the provided percentile lasso solution $\hat{\lambda}_{min}(\hat{\theta})$ across the 345 repetitions resulted in a more stable model, i.e. only two models were selected by percentile lasso, whereas the selected models by ordinary lasso were more variable. This can be justified by the fact that the percentile lasso tends to select large values of $\hat{\lambda}_{min}$ than ordinary lasso.

Lasso in conjunction with percentile lasso resulted in a model of 10 predictors selected from 38 variables model A (n=1241 observations), outperforming backward selection in terms of model parsimony and discrimination power (table 5.11). For model C, fitting lasso with the percentile λ resulted in a model with 12 variables selected from 17 candidate variables (n=1881 observations). On the other hand, the discriminative power of the model resulted by lasso was barely different from the other selected models by closed testing and backward selection table 5.11.
Additionally, in terms of model parsimony, lasso had the worst performance than the other investigated methods, suggesting that lasso at percentile λ_{min} might encompass more noise variables that can be found to have a spurious association with the recurrent VT. Moreover, this might also be attributed to the fact that a large number of candidate variables had some explanatory power, or to the fact that the cross-validation deviance prefers larger models when the number of observation is much larger than the number of variables. Therefore, we propose a further simulation study to investigate these differences.

In addition, allowing for more shrinkage by using 1-SE rule could eliminate the suspected noise variables. With the 1-SE penalty, lasso selected a model with 6 variables (model C) outperforming backward selection in terms of model parsimony (Supplement 7.5.1), whereas the discriminative power of the model was just 1% below the other selected models by backward selection and closed testing (table 7.2). If model parsimony is desirable, using a stronger penalty, like the 1-SE rule, is advisable. However, we are aware that the 1-SE penalty can introduce more bias in the regression coefficients estimates and corresponding standard errors. Moreover, the model performance could be improved without losing model parsimony, if we could correct for the extra shrinkage involved by the 1-SE rule.
Last but not least, we noted that analyzing our data with percentile lasso was quite computa-

tionally demanding, as we had to run M times K folds cross-validation for the lasso, obtaining, therefore, M optimal tuning parameters which can be used to estimate the percentile θ.

Lastly, we investigated the closed testing procedure as was proposed by Goeman and Solari. Besides, it is considered to be a powerful procedure for controlling FWER. The closed testing as was implemented in this thesis, has the ability to provide the researcher with more insights in the model selection process in contrast to lasso and backward selection. This was manifested clearly in its ability to produce a collection of minimal models that can fit as good as the full model. Hence the researcher will have several minimal models from which he/she can choose a model that can be more suitable to answer his/her research question.
In addition, the Closed testing quantifies the uncertainty of the selected models by providing the confidence set for each selected model i.e. the researcher will be able to display the $(1 - \alpha)$ boundary for the number of true and false finding for each selected model. Furthermore, closed testing is a flexible testing procedure i.e. it works with any choice of a local test. On the grounds of these advantages of closed testing over the lasso and backward selection methods, we would like to propose this new method for further application to build predictive models.

In our thesis, applying the closed testing to model C, resulted in a collection of multiple minimal models with 6 to 10 variables, which we called Shortlist. Remember that the shortlist is defined as a collection of models with the property that each model encompass the smallest possible number of variables that exhibit a significant effect on the outcome variable as the full model. Moreover, the shortlist models from closed testing outperforms the other methods, lasso and backward selection, in terms of model parsimony and model performance measured by corrected Harrell C statistics. This result might be explained by the fact that the closed testing tries at least to select the important variables, therefore generating all possible minimal models that still have a significant association with the recurrent VT.

There are some theoretical and computational challenges that should be addressed for further study. The closed testing procedure in its standard form has the disadvantage of performing an $O(2^m)$ intersection hypotheses. Therefore, even for moderately large number of candidates variables m, say m around 20-30, the standard form of closed testing will result in unfeasible computation.
Avoiding testing all hypothesis in the closure C, by using more efficient local test may reduce the computational time of this procedure, mainly when some hypothesis in the hierarchical structure appears to be non-significant. One additional point that should be mentioned here is the power of this procedure. To the best of our knowledge [19] the power property of this procedure depends essentially on the local test which makes it very sensitive to the implemented local tests, therefore the researcher should find a balance between the pros (computational) and cons (power) of the implemented local test.
Last note, when comparing lasso, backward and closed testing coefficients estimate, we noted that closed testing has the largest estimate (figure 7.7). Therefore we suggest further study to investigate the effect of shrinkage of the estimate coefficients under closed testing.

In this study, missing values was an important issue. The great part of missing values was a consequence of the change in the measuring mechanism i.e. after 2002 no blood samples were gathered, instead, DNA buccal swabs were collected. Our analysis was based only on the observed data, the so-called Complete Case. Important to note by this approach is that the sample size is reduced, which may lead to imprecise estimates. In some situations using complete cases may yield biased results. In our study we think that this is not so much the case. An alternative

approach is to use Multiple Imputation methods, that have the ability to preserve sample size, and often performs better than complete case approach.

Many approaches have been proposed for variable selection with multiply imputed data. There is plenty of research done for the conventional (e.g. backward elimination) selection methods with multiply imputed data. However, how to apply lasso variable selection methods to the data with missing values is an important yet unresolved problem. To the best of our knowledge, several approaches to combine the lasso method with multiple imputation techniques have been proposed [5],[46]. Not using multiple imputation methods was one limitation of this study. For this reason, we propose a comparison between backward selection, lasso and closed testing with multiple imputation as topic for a further study.

6.1 Choice of model and conclusion

In this thesis, we investigated three methods in order to build a model for the prediction of recurrent VT. We are getting to the stage where we can make a choice among the proposed models.
Our choice of the presented models was based on a balance between the number of missing values, maximal discriminative performance and the optimal ease of use of the model in clinical practice. Model A, developed by lasso shows the best discriminative performance with 10 variables, this performance was somehow low for model C selected under closed testing and backward selection. On the other hand, model A requires three laboratory factors (factor VIII, D-dimer and VWF) for which stopping the anticoagulant treatment is needed for a correct interpretation of the values. In addition, the laboratory factors had the highest percentage of missing values. At the other end of the spectrum is model C_6 selected by closed testing, has the desired parsimonious character with a diminutive difference in terms of corrected C statistics, and moreover requires no laboratory measurements. Importantly, any model C chosen by closed testing can be used as an alternative to model C_6.
Further, note that some alternative models for C contains genetic factors i.e. blood type and factor V Leiden, but these are not that hard to measure, even from a buccal swab. Therefore, we would propose to choose model C_6 selected under closed testing method, which has slightly lower corrected C-statistic but its added value over model A is mostly that its parsimony character and clinical utility.

One last essential suggestion before models are introduced into clinical practice and one that is of the utmost importance is that their predictive performance be externally validated i.e. that the model performance should be evaluated based on data that were not used to develop the model. This completely was a clear limitation of our study. Therefore, in order to help assess the generalizability of our model, we suggest the performance of external validation.

Bibliography

[1] Robert B Bendel and Abdelmonem A Afifi. Comparison of stopping rules in forward "stepwise" regression. *Journal of the American Statistical association*, 72(357):46–53, 1977.

[2] The Minitab Blog. What are the effects of multicollinearity and when can i ignore them? http://blog.minitab.com/blog/adventures-in-statistics-2/what-are-the-effects-of-multicollinearity-and-when-can-i-ignore-them. Accessed: 29-03-2019.

[3] Hege M Bøvelstad, Ståle Nygård, Hege L Størvold, Magne Aldrin, Ørnulf Borgan, Arnoldo Frigessi, and Ole Christian Lingjærde. Predicting survival from microarray data—a comparative study. *Bioinformatics*, 23(16):2080–2087, 2007.

[4] Werner Brannath and Frank Bretz. Shortcuts for locally consonant closed test procedures. *Journal of the American Statistical Association*, 105(490):660–669, 2010.

[5] Qixuan Chen and Sijian Wang. Variable selection for multiply-imputed data with application to dioxin exposure study. *Statistics in medicine*, 32(21):3646–3659, 2013.

[6] Sverre C Christiansen, Suzanne C Cannegieter, Ted Koster, Jan P Vandenbroucke, and Frits R Rosendaal. Thrombophilia, clinical factors, and recurrent venous thrombotic events. *Jama*, 293(19):2352–2361, 2005.

[7] David R Cox. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202, 1972.

[8] WORLD THROMBOSIS DAY. Know the facts: Know thrombosis. http://www.worldthrombosisday.org/issue/thrombosis/. Accessed: 29-03-2019.

[9] Shelley Derksen and Harvey J Keselman. Backward, forward and stepwise automated subset selection algorithms: Frequency of obtaining authentic and noise variables. *British Journal of Mathematical and Statistical Psychology*, 45(2):265–282, 1992.

[10] Dr S. Kristiansen M. Pasquier Dr L. Banken, Dr H.U. Burger. The Closed Test Procedure. http://www.sascommunity.org/seugi/SEUGI1993/The%20Closed%20Test%20Procedure.pdf. Accessed: 2018-10-01.

[11] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

[12] Jennifer Fahrni, Marc Husmann, Silvia B Gretener, and Hong H Keo. Assessing the risk of recurrent venous thromboembolism–a practical approach. *Vascular health and risk management*, 11:451, 2015.

[13] Virginia F Flack and Potter C Chang. Frequency of selecting noise variables in subset regression analysis: a simulation study. *The American Statistician*, 41(1):84–86, 1987.

[14] Andrew Gelman. Exploratory data analysis for complex models. *Journal of Computational and Graphical Statistics*, 13(4):755–779, 2004.

[15] Gerardnico. Statistics - (shrinkage—regularization) of regression coefficients. https://gerardnico.com/data_mining/shrinkage. Accessed: 29-03-2019.

[16] Jelle Goeman, Rosa Meijer, Thijmen Krebs, and Aldo Solari. Simultaneous control of all false discovery proportions in large-scale multiple hypothesis testing. *arXiv preprint arXiv:1611.06739*, 2016.

[17] Jelle Goeman, Aldo Solari, and Rosa Meijer. Using the cherry r package. 2018.

[18] Jelle J Goeman. L1 penalized estimation in the cox proportional hazards model. *Biometrical journal*, 52(1):70–84, 2010.

[19] Jelle J Goeman and Aldo Solari. Rejoinder to" multiple testing for exploratory research". *arXiv preprint arXiv:1208.3297*, 2012.

[20] Jelle J Goeman, Aldo Solari, et al. Multiple testing for exploratory research. *Statistical Science*, 26(4):584–597, 2011.

[21] Frank E Harrell. Ordinal logistic regression. In *Regression modeling strategies*, pages 311–325. Springer, 2015.

[22] Frank E Harrell, Kerry L Lee, and Daniel B Mark. Multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. *Statistics in medicine*, 15(4):361–387, 1996.

[23] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[24] Nicholas A Heard and Patrick Rubin-Delanchy. Choosing between methods of combining-values. *Biometrika*, 105(1):239–246, 2018.

[25] John A Heit, David N Mohr, Marc D Silverstein, Tanya M Petterson, W Michael O'fallon, and L Joseph Melton. Predictors of recurrence after deep vein thrombosis and pulmonary embolism: a population-based cohort study. *Archives of internal medicine*, 160(6):761–768, 2000.

[26] C Kearon, W Ageno, SC Cannegieter, B Cosmi, G-J Geersing, PA Kyrle, Subcommittees on Control of Anticoagulation, Predictive, and Diagnostic Variables in Thrombotic Disease. Categorization of patients as having provoked or unprovoked venous thromboembolism: guidance from the ssc of isth. *Journal of Thrombosis and Haemostasis*, 14(7):1480–1483, 2016.

[27] Clive Kearon, Elie A Akl, Joseph Ornelas, Allen Blaivas, David Jimenez, Henri Bounameaux, Menno Huisman, Christopher S King, Timothy A Morris, Namita Sood, et al. Antithrombotic therapy for vte disease: Chest guideline and expert panel report. *Chest*, 149(2):315–352, 2016.

[28] William J Kennedy and Theodore A Bancroft. Model building for prediction in regression based upon repeated significance tests. *The Annals of Mathematical Statistics*, 42(4):1273–1284, 1971.

[29] Damjan Krstajic, Ljubomir J Buturovic, David E Leahy, and Simon Thomas. Cross-validation pitfalls when selecting and assessing regression and classification models. *Journal of cheminformatics*, 6(1):10, 2014.

[30] Saskia Kuipers, Suzanne C Cannegieter, Carine JM Doggen, and Frits R Rosendaal. Effect of elevated levels of coagulation factors on the risk of venous thrombosis in long-distance travelers. *Blood*, 113(9):2064–2069, 2009.

[31] Paul Alexander Kyrle, Frits R Rosendaal, and Sabine Eichinger. Risk assessment for recurrent venous thrombosis. *The Lancet*, 376(9757):2032–2039, 2010.

[32] LUMEN. Hemostasis. https://courses.lumenlearning.com/boundless-ap/chapter/hemostasis/. Accessed: 29-03-2019.

[33] Ruth Marcus, Peritz Eric, and K Ruben Gabriel. On closed testing procedures with special reference to ordered analysis of variance. *Biometrika*, 63(3):655–660, 1976.

[34] Rosa Janna Meijer et al. *Efficient multiple testing for large structured problems*. Department of Medical Statistics & Bioinformatica, Faculty of Medicine, Leiden University Medical Center (LUMC), Leiden University, 2015.

[35] Nicolai Meinshausen. Hierarchical testing of variable importance. *Biometrika*, 95(2):265–278, 2008.

[36] Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

[37] Fatemeh Moheimani and Denise E Jackson. Venous thromboembolism: classification, risk factors, diagnosis, and management. *ISRN hematology*, 2011, 2011.

[38] AI Franco Moreno, MJ García Navarro, J Ortiz Sánchez, RM Martín Díaz, E Madroñal Cerezo, Cristina Lucía de Ancos Aracil, N Cabello Clotet, I Perales Fraile, S Gimeno García, C Montero Hernández, et al. A risk score for prediction of recurrence in patients with unprovoked venous thromboembolism (damoves). *European journal of internal medicine*, 29:59–64, 2016.

[39] Banne Nemeth, JF Timp, A van Hylckama Vlieg, Frits R Rosendaal, and Suzanne C Cannegieter. High risk of recurrent venous thrombosis in patients with lower-leg cast immobilization. *Journal of Thrombosis and Haemostasis*, 16(11):2218–2222, 2018.

[40] NHS. Causes deep vein thrombosis. https://www.nhs.uk/conditions/deep-vein-thrombosis-dvt/causes. Accessed: 09-10-2018.

[41] Office of the Surgeon General (US et al. The surgeon general's call to action to prevent deep vein thrombosis and pulmonary embolism. 2008.

[42] Martin H Prins, Anthonie WA Lensing, Paolo Prandoni, Philip S Wells, Peter Verhamme, Jan Beyer-Westendorf, Rupert Bauersachs, Henri Bounameaux, Timothy A Brighton, Alexander T Cohen, et al. Risk of recurrent venous thromboembolism according to baseline risk factor profiles. *Blood advances*, 2(7):788–796, 2018.

[43] Priya Ranganathan, CS Pramesh, et al. Censoring in survival analysis: potential for bias. *Perspect Clin Res*, 3(1):40, 2012.

[44] John O Rawlings, Sastry G Pantula, and David A Dickey. *Applied regression analysis: a research tool*. Springer Science & Business Media, 2001.

[45] Steven Roberts and Gen Nowak. Stabilizing the lasso against cross-validation variability. *Computational Statistics & Data Analysis*, 70:198–211, 2014.

[46] Nick Sabbe, Olivier Thas, and J-P Ottoy. Emlasso: logistic lasso with missing data. *Statistics in medicine*, 32(18):3143–3157, 2013.

[47] Martin Sill, Thomas Hielscher, Natalia Becker, Manuela Zucknick, et al. c060: Extended inference with lasso and elastic-net regularized cox and generalized linear models. *Journal of Statistical Software*, 62(5):1–22, 2014.

[48] Ewout W Steyerberg. *Clinical prediction models: a practical approach to development, validation, and updating*. Springer Science & Business Media, 2008.

[49] Ewout W Steyerberg, Frank E Harrell Jr, Gerard JJM Borsboom, MJC Eijkemans, Yvonne Vergouwe, and J Dik F Habbema. Internal validation of predictive models: efficiency of some procedures for logistic regression analysis. *Journal of clinical epidemiology*, 54(8):774–781, 2001.

[50] Michael B Streiff. Predicting the risk of recurrent venous thromboembolism (vte). *Journal of thrombosis and thrombolysis*, 39(3):353–366, 2015.

[51] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[52] Robert Tibshirani. The lasso method for variable selection in the cox model. *Statistics in medicine*, 16(4):385–395, 1997.

[53] Jasmijn F Timp, Willem M Lijfering, Linda E Flinterman, Astrid van Hylckama Vlieg, Saskia le Cessie, Frits R Rosendaal, and Suzanne C Cannegieter. Predictive value of factor viii levels for recurrent venous thrombosis: results from the mega follow-up study. *Journal of thrombosis and haemostasis*, 13(10):1823–1832, 2015.

[54] Jasmijn Fleur Timp et al. *Risk factors and predictors for recurrent venous thrombosis: building blocks for a prognostic model*. PhD thesis, 2016.

[55] Hans C van Houwelingen and Willi Sauerbrei. Cross-validation, shrinkage and variable selection in linear regression revisited. *Open Journal of Statistics*, 3(2):79, 2013.

[56] Pierre JM Verweij and Hans C Van Houwelingen. Cross-validation in survival analysis. *Statistics in medicine*, 12(24):2305–2314, 1993.

[57] Bhanukiran Vinzamuri and Chandan K Reddy. Cox regression with correlation based regularization for electronic health records. In *2013 IEEE 13th International Conference on Data Mining*, pages 757–766. IEEE, 2013.

[58] Peter H Westfall and Stanley S Young. Resampling-based multiple testing: Examples and methods for p-value adjustment (wiley series in probability and statistics). 1993.

[59] Richard H White. The epidemiology of venous thromboembolism. *Circulation*, 107(23 suppl 1):I–4, 2003.

[60] Zhongheng Zhang, Giuliana Cortese, Christophe Combescure, Roger Marshall, Minjung Lee, Hyun Ja Lim, Bernhard Haller, et al. Overview of model validation for survival regression model with competing risks using melanoma study data. *Annals of translational medicine*, 6(16), 2018.

[61] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320, 2005.

Chapter 7

Supplement

7.1 Lasso instability

7.1.1 Ordinary lasso and percentile lasso

Here is an illustration (figures :7.1 and 7.2) depicting the sensitivity of the ordinary lasso to the changes we made in the R code of our analysis. Of note, percentile lasso is still selecting two models 7.2, whereas ordinary lasso 7.1 has a λ_{min} varying between 0.0046 and 0.0153, and on the meantime, the corresponding size of the selected model by ordinary lasso ranges from 9 to 24. This figure depicts clearly the high variability among selected models using the ordinary lasso .

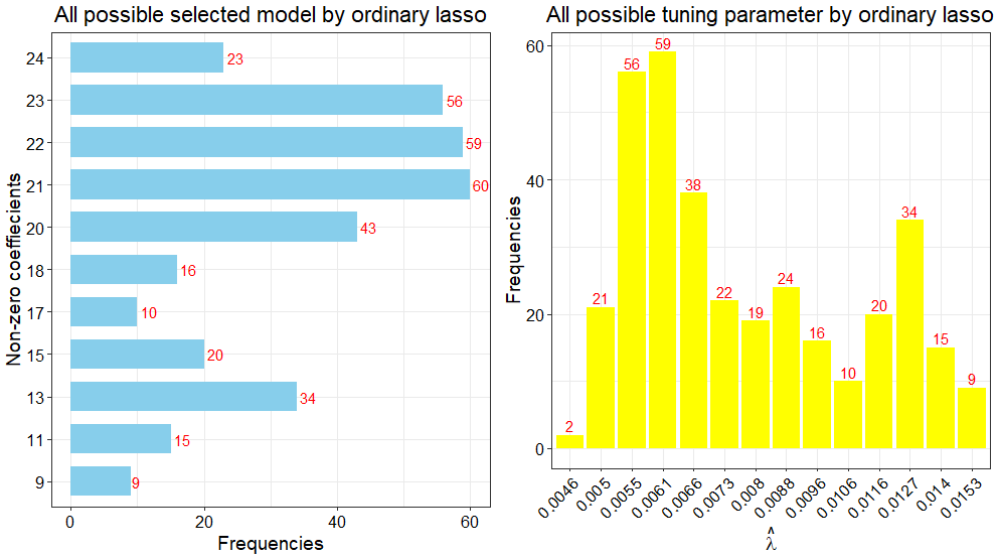


Figure 7.1: Analysis on our data set. The left side of figure illustrate the frequency of selected models, and the right side are values of optimal tuning parameter $\hat{\lambda}_{min}$ of non-zero coefficient estimates obtained from the lasso, over 345 random fold assignments

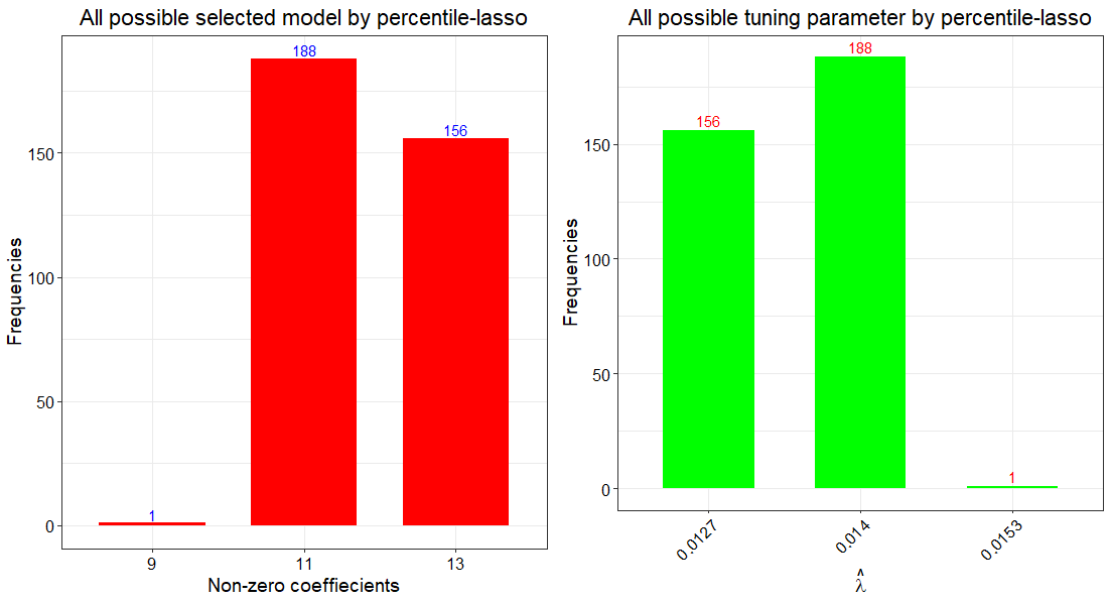


Figure 7.2: The left side of figure illustrate the frequency of selected models, and the right side are values of optimal tuning parameter $\hat{\lambda}$ of non-zero coefficient estimates obtained from the percentile-lasso, over 345 repetition at $\hat{\lambda} = \hat{\lambda}(min(\hat{\theta}))$. With $min(\hat{\theta})$ we mean the percentile of $\Lambda(100)$ at minimal deviance for each single repetition of 345.

7.2 Extra notes on the closed testing procedure

7.2.1 Exploratory vs. Confirmatory research

Statistical data analysis often falls into two main phases: exploratory and confirmatory. The exploratory data analysis (EDA) is the first part of your data analysis process, As the name suggests, one is exploring: looking for clues, the EDA can help you in figuring out what to make of the data, generating and formulating a hypothesis. This first phase of analysis can be compared to detective work in the sense of one is gathering evidence. This first essential step is commonly followed by Confirmatory Data Analysis. Confirmatory Data Analysis (CDA) is the phase where you evaluate your evidence "quantifies the extent to which these discrepancies [deviations from a model] could be expected to occur by chance" (Gelman; 2004) [14]. In CDA the traditional statistical tools of inference, significance, and confidence can be used to evaluating evidence. Exploratory analysis and confirmatory analysis "can—and should—proceed side by side" (Tukey; 1977).

7.2.2 Mild, flexible and post-hoc

By a mild inferential procedure we mean that one can expect occurrence of a number of false discoveries among the selected hypotheses and these can be removed in the validation phase. Furthermore an inferential procedure is flexible if it does not dictate to the user the rejections, but it provides the researcher with a complete freedom to choose which hypotheses to select or not to select. This freedom of "picking" and "choosing" is not possible in the confirmatory

research setting, here the collection of hypotheses has been done prior to the experiment. Lastly, a post hoc inferential procedure allows the user to review the consequences of any choice of rejected hypotheses made after seeing the data [20].

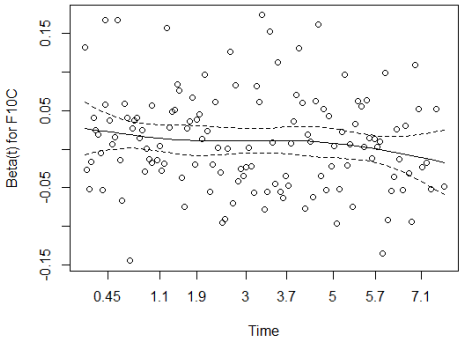
7.2.3 Hypothesis testing

Hypothesis testing is a statistical method that is performed in making statistical decisions by using some experimental data. The aim in hypothesis testing is to evaluate whether or not some pre-specified hypothesis, called the null-hypothesis, can be rejected based on evidence that is present in the data. A hypothesis test evaluates two mutually exclusive statements (null hypothesis vs. alternative hypothesis) in order to determine which statement is most likely supported by data. That is, if one is true, the other must be false. Even though the alternative hypothesis does not have to be explicitly specified.
Null and alternative hypothesis are two types of statistical hypotheses. The null hypothesis, denoted by H_0, is the commonly accepted fact, this assumes that the observation is due to a chance factor. So, with respect to our data under study, we can formulate the null as:$H_0 : \beta_j = 0$ $\forall$ j in $(1, .., p)$ this is the same as saying: there are no variables associated with second thrombosis. The alternative hypothesis, denoted by H_1 or H_a, states the opposite and is usually the hypothesis you are trying to prove. For $H_1 : \beta_j \neq 0$ $\exists$ j in $(1, .., p)$ in words this mean there is at least one variable associated with second thrombosis.
When we say that a finding is statistically significant this is equivalent to say a null hypothesis is rejected, this tells us that there is at least one variable X from the variables set $X_1 \ldots . X_p$ that is in fact associated with the outcome Y. In order to conclude whether a null-hypothesis can be rejected, a statistical test is needed.
If our statistical test is significance (suppose $p \leq 0.05$), we reject the null hypothesis and accept the alternative hypothesis. Alternatively if statistical test is not significance (suppose $p > 0.05$), we fail to reject the null hypothesis and cannot accept the alternative hypothesis. One should note that you cannot accept the null hypothesis, but only find evidence against it. We make here a distinction between "fail to reject" and "acceptance", because "fail to reject" implies that the data are not sufficiently supporting the alternative hypothesis over the null hypothesis. whereas "acceptance" implies that the null hypothesis is true. Note that we are not trying to prove that the null hypothesis is true, because the null hypothesis is assumed to be a true statement until the contrary is proven.

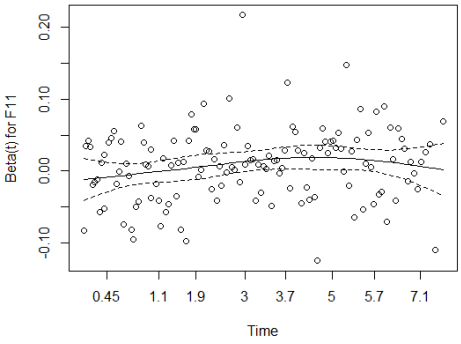
7.3 Schoenfeld residuals and numerical test for PH assumption

7.3.1 Diagnostics for the Cox model: Schoenfeld residuals

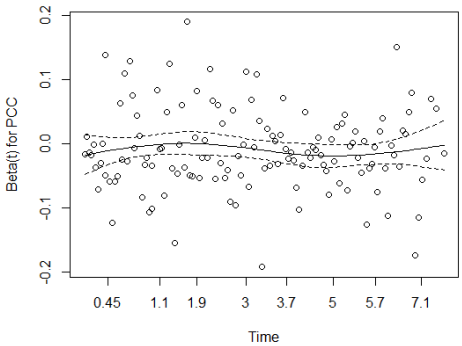
In the figures we see two dashed lines forming a band range of a +/- 2-standard-error around the middle solid line which is a smoothing spline fit to the plot. A systematic departure from a horizontal line are indication that proportional hazards assumption are violated, i.e if that line is fairly flat and straight, then PH assumption is supported.. The assumption of proportional hazards appears to be supported for all covariates of our model, which is in agreement with the test presented in (Results 5, figure 5.2 and figure 5.3) .



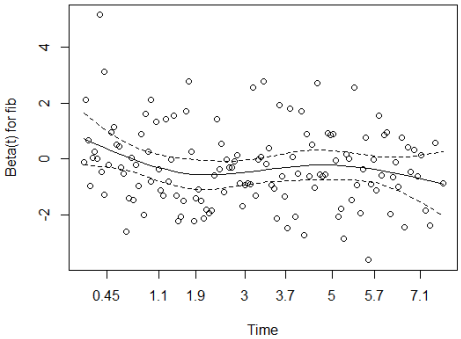
(a) factor 10



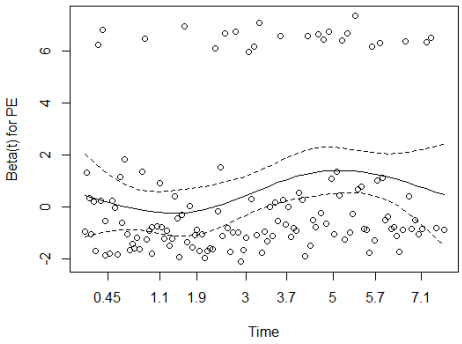
(b) factor 11



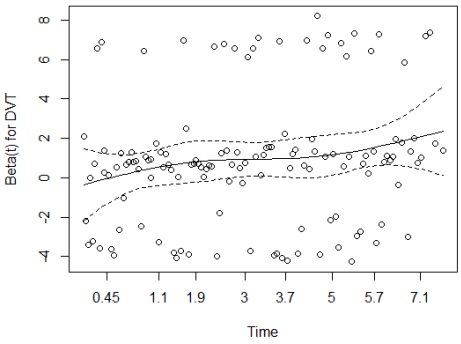
(c) Protein C



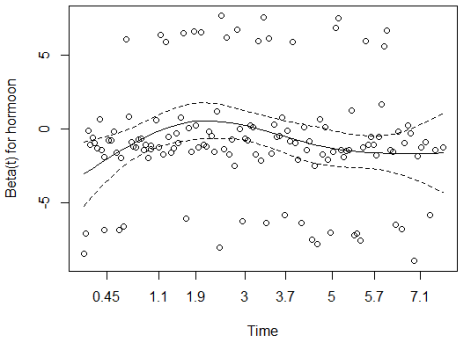
(d) Fibrinogen



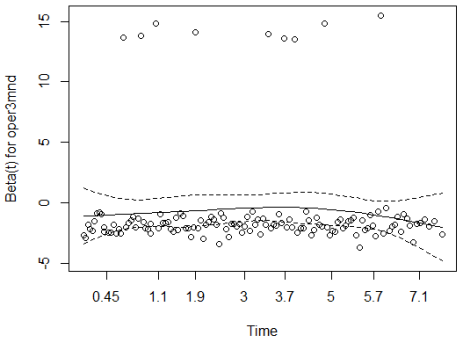
(e) PE



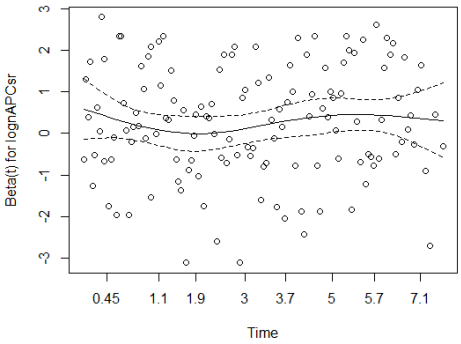
(f) PE+DVT



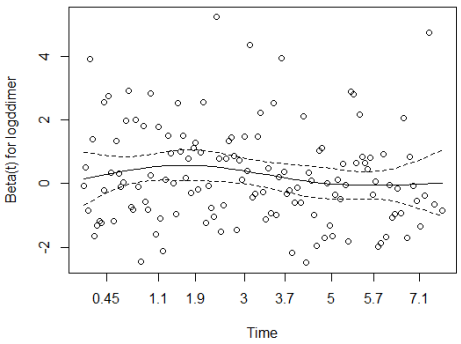
(a) Hormone



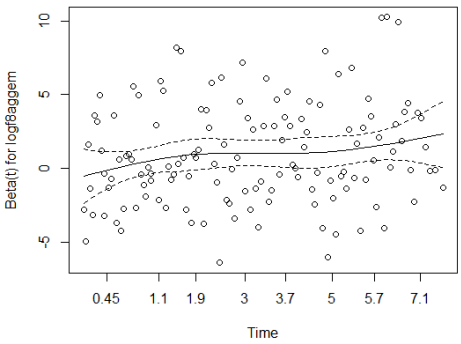
(b) Surgery



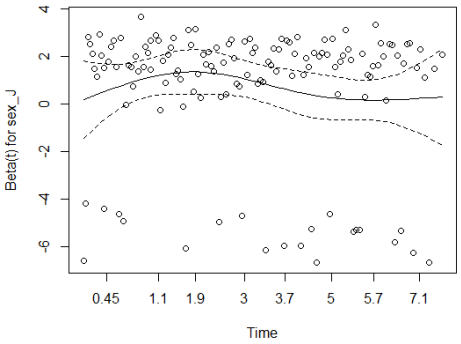
(c) APC ratio



(d) Ddimer



(e) Factor 8



(f) Sex

Figure 7.3: the scaled Schoenfeld residuals for model A by backward selection

7.3.2 PH assumptions test: closed testing models

In this subsection we present a numerical test as an indication of whether the proportional hazards (PH) assumption holds or not, and possibly to what degree. Here we have chosen to check the validity of (PH) assumption by using a statistical test (Grambsch and Therneau (1994)) implemented in **cox.zph(.)** function from the **survival** R package. The output from the displayed test results in figure 7.4 are non-significant for model 7, model 8 and model 10, indicating no violation evidence for the (PH) assumption.

	rho	chisq	p
Surgery	-0.01465	0.05120	0.8210
Hospital	-0.00227	0.00126	0.9717
Pregnant	-0.05853	0.83061	0.3621
Hormone	-0.03469	0.28486	0.5935
Cardio-dis	-0.03417	0.27683	0.5988
Location VT	-0.07331	1.31293	0.2519
Gender	-0.03910	0.36940	0.5433
PE	-0.06244	0.95024	0.3297
PE & DVT	0.11482	3.20054	0.0736
PTS 1	-0.02012	0.09607	0.7566
PTS 2	0.02025	0.09930	0.7527
Leiden V	0.02494	0.15134	0.6973
GLOBAL	NA	7.41403	0.8291

(a) model 10

	rho	chisq	p
Surgery	0.0454	0.4968	0.4809
PC	-0.0168	0.0676	0.7948
Pregnant	-0.0610	0.8914	0.3451
Hormone	-0.0273	0.1744	0.6762
Cardio-dis	-0.0400	0.3844	0.5352
Gender	-0.0359	0.3099	0.5777
PE	-0.0341	0.2779	0.5981
DVT	0.1101	2.9125	0.0879
PTS 1	-0.0294	0.2062	0.6498
PTS 2	0.0112	0.0297	0.8632
GLOBAL	NA	6.1345	0.8038

(b) model 8

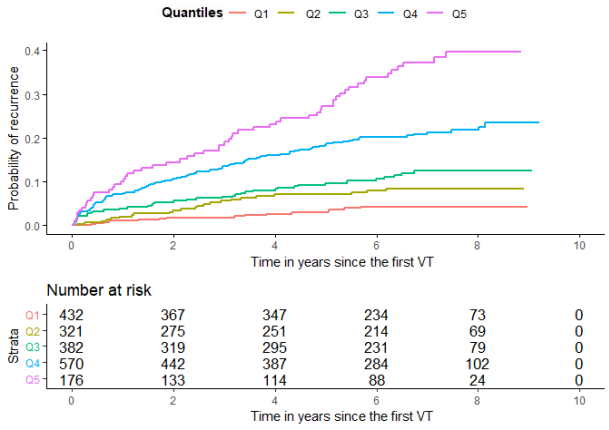
	rho	chisq	p
Surgery	0.0465	0.5220	0.4700
PC	-0.0162	0.0631	0.8016
Pregnant	-0.0610	0.8939	0.3444
Hormone	-0.0279	0.1869	0.6655
Gender	-0.0384	0.3544	0.5516
PE	-0.0282	0.1874	0.6651
DVT	0.1126	3.0694	0.0798
Leiden V	0.0214	0.1104	0.7397
GLOBAL	NA	5.7311	0.6773

(c) model 7

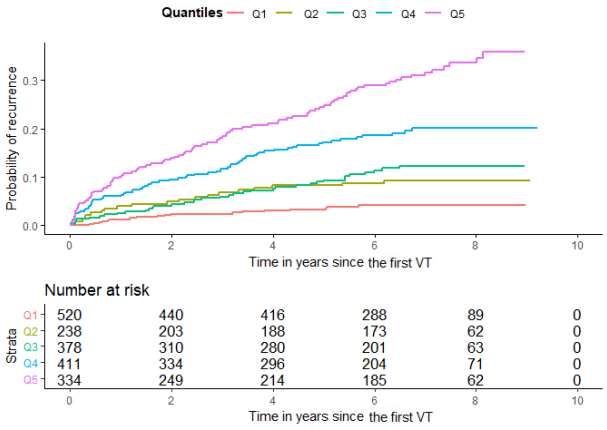
Figure 7.4: Test results for PH assumption for models selected by closed testing

7.3.3 Quintiles of the prognostic score closed testing

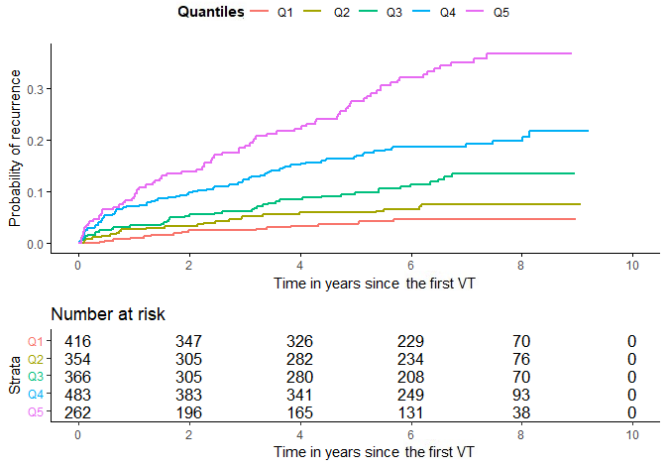
Figure 7.5 illustrates the Kaplan-Meier curves for quintiles of the prognostic score for the selected models by closed testing method. In order to examine the model ability of discrimination between risk of recurrence among patients, inverse Kaplan Meier plots for the observed risk of recurrence in quintiles of the prognostic scores were generated for each model.



(a) model C_7



(b) model C_8



(c) model C_{10}

Figure 7.5: Probability of recurrence stratified by quintiles of prognostic scores of the selected models by closed testing procedure

7.4 Cross-validation in linear regression

The tuning parameter λ in our equation (4.3) is an important key to determine the number of non-zero coefficients, thus an estimator $\hat{f}_\lambda$ that has a large predictive power. But how is the λ chosen such that the predictive accuracy of our estimator $\hat{f}_\lambda$ is optimal? One way to achieve this aim is by applying Cross Validation. The most common approach is K-fold cross validation, the basic idea is simple:

1. The training data T is partitioned into K separate sets of equal size: $T = (T_1, T_2, ..., T_K)$, commonly chosen K's are K = 5 and K = 10
2. Fit the model $\hat{f}^{\lambda}_{-k}$ to the training set for each k = 1, 2, . . . , K, excluding the kth-fold T_k.

3. Compute the fitted values for the observations in T_k, based on the training data that excluded this fold.

4. for each k fold compute the cross-validation (CV) error:

$$CV^{(\lambda)}_{error_k} = \frac{1}{|T_k|} \sum_{(x,y)\in T_k} (y - \hat{f}(x)^{(\lambda)}_k)$$

5. Compute the overall model cross-validation error:

$$CV^{(\lambda)}_{error} = \frac{1}{|K|} \sum_{k=1}^{K} (CV^{(\lambda)}_{error_k})$$

6. Find λ_{min} as the one with minimum $CV^{(\lambda)}_{error}$.

$$\hat{\lambda}_{min} = \underset{\lambda\in(\lambda_1,\lambda_2,....\lambda_m)}{\operatorname{argmin}} CV^{(\lambda)}_{error}$$

7. Apply $\hat{f}^{(\lambda_{min})}(x)$ to the test set to assess test error

when K = 1, This is called leave-one-out cross validation (LOOCV).
form the experimenters point of view this choice (λ_{min}) is very conservative, i.e not eliminating sufficiently many predictors from the model. One other alternative choice was suggested by Tibshirani.

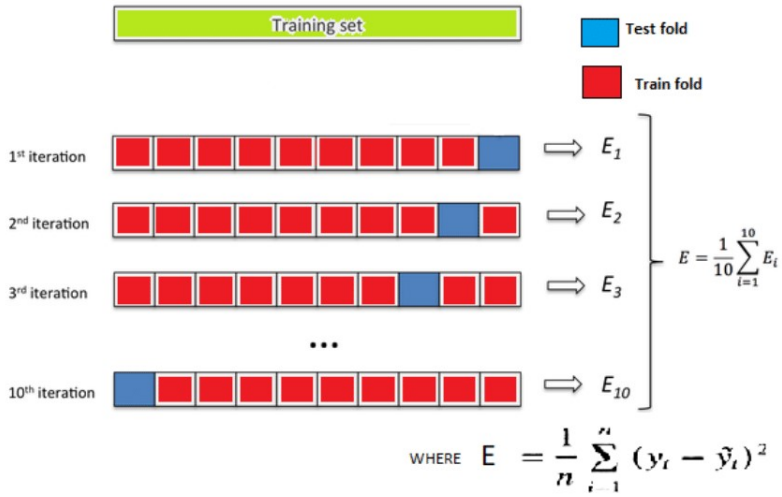


Figure 7.6: K fold cross validation method for linear model

7.5 Results of the 1-SE rule for lasso

7.5.1 Model C: 1-SE rule

Table 7.1: Variables and their corresponding HR for model C under lasso method

Variables	Hazard Ratios (HR)
Surgery	0.88
Hormone use	0.97
Location VT (Distal DVT vs Proximal)	1.16
Gender (male)	1.98
Type VT (PE & DVT)	1.01
Factor V Leiden	1.05

7.5.2 Model C: performance at 1-SE rule

Table 7.2: Predictive performance of the selected model C across lasso in conjunction with percentile lasso, backward selection and closed testing

Methods	Harrell C	Corrected C	Number of variables
Lasso λ_{min}	0.708	0.688	12
Lasso λ_{1-SE}	0.683	0.672	6
Backward selection	0.701 (0.680, 0.741)	0.682	8
Closed testing :			
Model C_6	0.692 (0.665, 0.722)	0.689	6
Model C_7	0.695 (0.668,0.733)	0.689	7
Model C_8	0.698 (0.671,0.734)	0.687	8
Model C_9	0.693 (0.665,0.728)	0.684	9
Model C_{10}	0.697 (0.697,0.733)	0.684	10

7.5.3 lasso in conjunction with percentile lasso, backward selection and closed testing procedure coefficients estimates

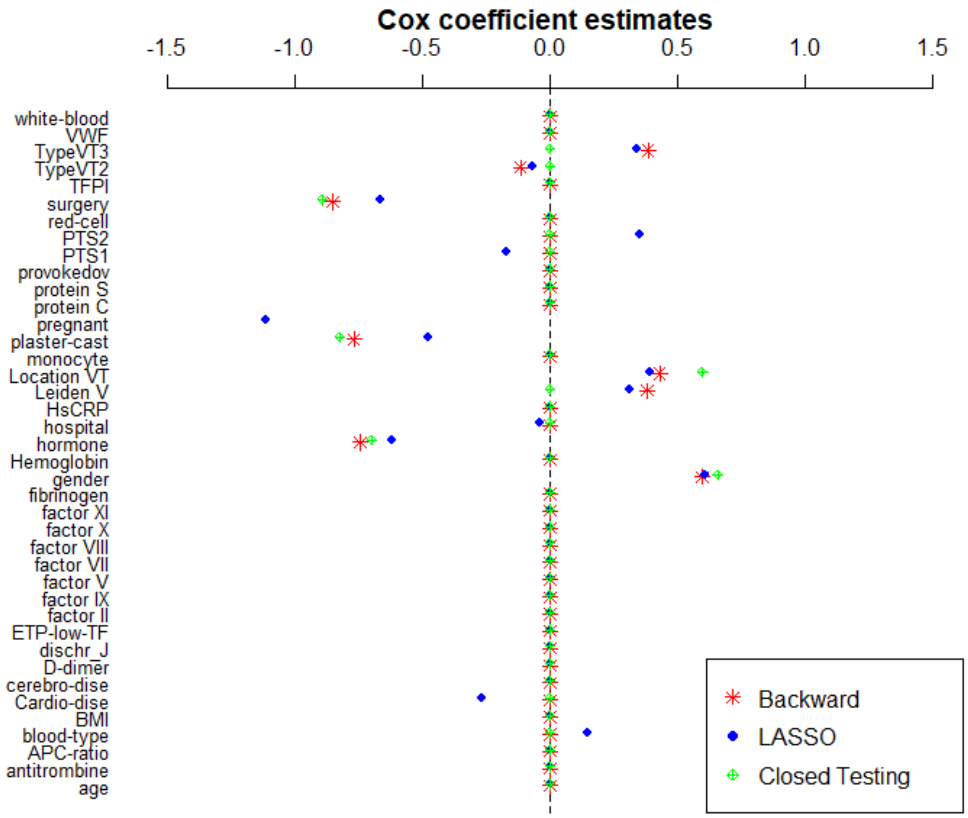


Figure 7.7: Regression coefficients estimates for model C, under lasso, backward selection and closed testing.

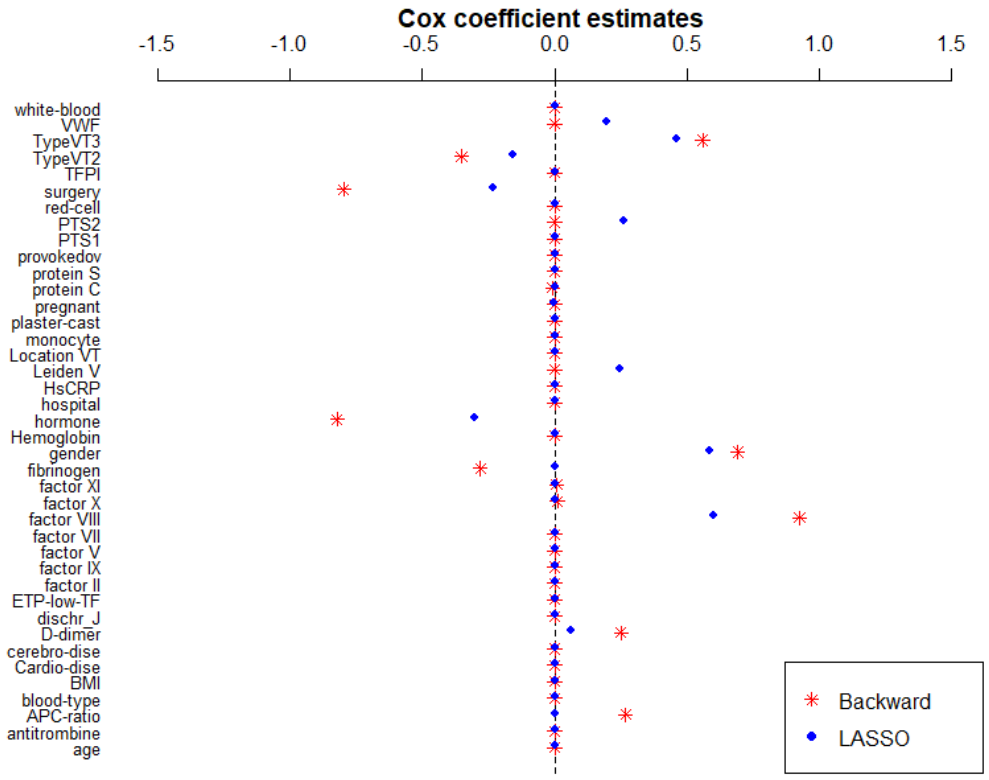


Figure 7.8: Regression coefficients estimates for model A. A comparison between lasso and backward selection

Chapter 8

R-code syntax

8.1 Data preprocessing

```
#install.packages("haven")

#---------------------------#
# MEGA Data preprocessing #
#---------------------------#

library(haven)

dat <- read_dta("...data source...")

# remove the next variables variables: because they were replaced by new variables
    :
# dischr_J -> dischr_S
# provokedov -> provokedov_S
# locatie1var -> locatie1var_S
# PTS1var -> PTS_J
# id <- is not needed


dat <- subset(dat, select = -c(dischr_J,provokedov,locatie1var,PTS1var,id))

# Some data variables were changed to factors;

dat$PTS_J <-  as.factor(dat$PTS_J)

dat$TypeVT <- as.factor(dat$TypeVT)
# note that TypeVT has 3 categorie i.e: DVT=1,PE=2 and DVT+PE=3;

dat$locatie1varII <- ifelse(dat$locatie1var_S==2,1,0)
# this will result in Proximal=0 vs Distal=1

# we remove SNPscore, because it has so many missing variables: 70%
dat <- subset(dat , select = -c(SNPscore,locatie1var_S))

# note that some variables are already coded as 0/1, thus
# there's no need to code them as factor


# Peredictors:

```

```
clinical.f <- c("lft", "oper3mnd","gips3mnd","bedzk3mnd","zwan3mnd", "hormoon",
"discardio","discerebro","bmi","locatie1varII","sex_J","provokedov_S",
"TypeVT","dischr_S","PTS_J")

genetics.f <- c("blgroup","fvleiden_J")

laboratory.f <- c("fib","AT3","PCC","F7C","f9aggem" ,"f2c","F10C","F11","lognAPCsr
    ",
"factorV","TFPI","Hgb","nETPLT","logf8aggem" ,"logvwf","logpsfree",
"logddimer","logWBC","logmoperc","logrdw","logHsCRP")


A.var.org <- c(clinical.f,genetics.f,laboratory.f)# model A variables;
C.var.org <- c(clinical.f,genetics.f)# model C variables;



# Survival analysis time and indecator,

# recidieftime -> time to event
# recidiefunprov -> event

dat$time <- dat$recidieftime
dat$status <- dat$recidiefunprov

# remove the redundant variables from the data,
dat <- subset(dat , select =-c(recidieftime,recidiefunprov))
```

8.2 Descriptive statistics

```
1
#-------------------------#
3 # missing values plot  #
#-------------------------#
5

7 # change columnames to english:

9 j <- c( "age","surgery","plaster cast","hospitalization","pregnant","hormone","
    cardio-disease","cerebro-disease",
"BMI","location VT","gender","provoked","type VT","disease-addional","PTS","blood-
    type","Leiden V","fibronegen",
11 "antitrombine","protein C","factor VII","factor IX","factor II", "factor X","
    factor XI","APC ratio",
"factor V", "TFPI","hemoglobin","ETP","factor VIII", "VWF","protein S","D-dimer","
    white-blood",
13 "monocyte-percetage","red-cell","CRP")
data_frame.j <-  dat[,-c(41,42)]# remove time and statuts columns
15 colnames(data_frame.j)<- j

17
library(Amelia)
19 missmap(data_frame.j,col = c("maroon", "green"),y.labels = NULL,y.at = NULL,
main = "Missingness Map of MEGA Dataset")
21

23 #-----------------#
# Correlations   #
25 #-----------------#

27 library("ggcorrplot")
library("Hmisc")
29
# change columnames to english:
31
j <- c( "age","surgery","plaster cast","hospitalization","pregnant","hormone","
    cardio-disease","cerebro-disease",
33 "BMI","location VT","gender","provoked","type VT","disease-addional","PTS","blood-
    type","Leiden V","fibronegen",
"antitrombine","protein C","factor VII","factor IX","factor II", "factor X","
    factor XI","APC ratio",
35 "factor V", "TFPI","hemoglobin","ETP","factor VIII", "VWF","protein S","D-dimer","
    white-blood",
"monocyte-percetage","red-cell","CRP")
37 data_frame <- na.omit(dat[,-c(39,40)])#
colnames(data_frame)<- j
39 corr <- as.data.frame(rcorr(as.matrix(data_frame),type = "pearson")$r)
ggcorrplot(corr, hc.order = TRUE, type = "lower" )
41


43
#-----------------------------------#
45 # Functions to produce baseline  #
#  characteristics                  #
47 #-----------------------------------#

49 # This function will calculate: the median, min and max of the
# required variables;
51
```

```
fun.1 <- function(x){
num.na <-  sum(is.na(x))
percent <- round(num.na /nrow(dt),3)
p <- list(median= median(x,na.rm=TRUE),
minimum=min(x,na.rm=TRUE),
maximum=max(x,na.rm=TRUE),
NA_numb = num.na,
NA_per = percent
)
return(p)

}

# This function will calculate: the sum and the percentage
# of variables of interest,

sp  <- function(x){
sm <- sum(x,na.rm=TRUE)
pr <- sum(x,na.rm=TRUE)/nrow(dt)
num.na <-  sum(is.na(x))
percent <- round(num.na /nrow(dt),3)

return(list(sum = sm, percentage = pr,
num_NA= num.na , NA_per = percent ) )
}

attach(dat)



# These are some characteristics of the chosen predictors.

#-------------------------#
# 1- clinical factors     #
#-------------------------#

#lft: Age
fun.1(lft)

# bmi: BMI
fun.1(bmi)

# sex_J : gender(male)
sp(sex_J)

# zwan3mnd : pregnant
sp(zwan3mnd)

# discardio: cardiovascular disease
sp(discardio)

# provokedov_S: provoked additional factors
sp(provokedov_S)

# typeVT: Type of the first event,

#DVT: Deep Vein Thrombosis:
sp(TypeVT==1)

# PE: polmunary embolism
sp(TypeVT==2)

```

```
# DVT + PE
sp(TypeVT==3)

# hormoon: hormone use
sp(hormoon)

# oper3mnd : surgery
sp(oper3mnd)

# gips3mnd: Plaster cast
sp(gips3mnd)

# bedzk3mnd : Hospitatlization
sp(bedzk3mnd)

#PTS_J: Posthrombotic syndrome, mild or severe.
#PTS_J: Posthrombotic syndrome, mild or severe.

sp(PTS_J==1)# milde
sp(PTS_J==2)# severe


# locatie1var :  location VT
sp(locatie1varII)

# discerebro: Cerebrovascular disease
sp(discerebro)

# dischr_S: disease additional comorbidities
sp(dischr_S)

#--------------------#
# Genitic factors   #
#--------------------#

#fvleiden_J : factor V Leiden
sp(fvleiden_J)

# blgroup: blood group
sp(blgroup)


#-------------------------#
#  Laboratory factors    #
#-------------------------#

#logddimer
fun.1(logddimer)

#f2c -> Factor II/prothrombin
fun.1(f2c)

#factorV
fun.1(factorV)

#F7C : factor VII
fun.1(F7C)

#logf8aggem: factor VIII
fun.1(logf8aggem)

#f9aggem: factor IX
```

```
fun.1(f9aggem)

#F10C: factor X
fun.1(F10C)

#F11: factor XI
fun.1(F11)

#logvwf :VWF
fun.1(logvwf)

#PCC -> Protein C
fun.1(PCC)

#fib -> Fibrinogeen
fun.1(fib)

#AT3: Antithrombine
fun.1(AT3)

#TFPI: TFPI
fun.1(TFPI)

#logHsCRP :CRP
fun.1(logHsCRP )

#nETPLT:
fun.1(nETPLT)

# lognAPCsr -> APC ratio
fun.1(lognAPCsr)

# Hgb -> Hemoglobin
fun.1(Hgb)

# logpsfree -> protein S
fun.1(logpsfree)

#logWBC ->White boold cell
fun.1(logWBC)

# logmoperc -> monocyte percetage
fun.1(logmoperc)

#logrdw -> red cell Distribution width
fun.1(logrdw)
```

8.3 Backward elimination

```
1
# These are the required R packages to perform our backward elimination analyses.
3
Packages <- c("MASS", "pec" ,"survcomp","survAUC","Hmisc") # load multiple
    packages
5 lapply(Packages, require, character.only = TRUE)

7
#-----------------------------------------------------#
9 # Functions to perform backward selection analysis  #
#-----------------------------------------------------#
11

13 # This function will first select variables in the Cox regression model
# using fastbw() function from the rms package, subsequently will return
15 # a fitted Cox regression model with the selected variables.
# backward selection in our analysis is used with p =0.1 value.
17 # Of note,this function is 99% similar to the function selectCox() from rms
    package;

19 selCox_my <- function (formula, data, rule = "p")
{
21 fit <- rms::cph(formula, data, surv = TRUE, x=TRUE, y=TRUE)
bwfit <- rms::fastbw(fit, rule = rule,type="individual",sls=0.1)
23 if (length(bwfit$names.kept) == 0) {
newform <- update(formula, ".~1")
25 newfit <- prodlim::prodlim(newform, data = data)
}
27 else {
newform <- update(formula, paste(".~", paste(bwfit$names.kept,
29 collapse = "+")))
newfit <- rms::cph(newform, data, surv = TRUE ,x=TRUE, y=TRUE)
31 }
out <- list(fit = newfit, In = bwfit$names.kept)
33 out$call <- match.call()
class(out) <- "selectCox"
35 out
}
37

39 # This function will use our selected model by selCox_my() function;
# to make internal validation, and produce the Harrell C statistics;
41


43
Boot.H <- function(B,X,df){
45
# Arguments:
47 # X: covariates,
# B: number of bootsrap,
49 # df: data frame
# return: backward model,C_corrected,C_apparent and 95% C ci.
51
dt <- as.data.frame(df)
53 Xs <- paste(X, collapse=" + ")
form <- as.formula(paste("Surv(time, status) ~",
55 paste(Xs, collapse="+")))

57 # Fit the model in the original data;
```

```
back.df <- selCox_my(form ,rule = "p",data=dt)
fit <- back.df$fit

# Calculate the apparent Harrel C statistics;
ttt <- quantile(dt$time)
k <- predictSurvProb(fit,newdata=dt,times=ttt)
harrelC1 <- rcorr.cens(k[,3],with(dt,Surv(time, status)))
C_indx_app <-  harrelC1[1]


# Empty  matrix to store bootstrap results;
M <- matrix(nrow = B, ncol = 3,
dimnames = list(paste('Sample',1:B),
c("C_orig","C_boot","Optimism")))

n = nrow(dt)
set.seed(701)

for(i in 1:B){

# Draw a random sample from our data:
obs.boot <- sample(x = 1:n, size = n, replace = T)
data.boot <- dt[obs.boot, ]

# Fit the model on bootstrap sample:
back.cx <- selCox_my(form ,rule = "p",data=data.boot)
a <- back.cx$fit

# Apply model to original data:
ttt <- quantile(dt$time)
k <- predictSurvProb(a,newdata=dt,times=ttt)
harrelC1 <- rcorr.cens(k[,3],with(dt,Surv(time, status)))
M[i, 1] <- harrelC1[1]

# Apply model to bootstrap data
tt <- quantile(data.boot$time)
k1 <- predictSurvProb(a,newdata=data.boot,times=tt)
harrelC2 <- rcorr.cens(k1[,3],with(data.boot,Surv(time, status)))
M[i, 2] <- harrelC2[1]

# Optimism:
M[i,3] <- M[i, 2]- M[i, 1]

}
C_indx <-  C_indx_app - mean(M[,3])
K <- data.frame(M)

# Confidence interval for C statistics by Percentile Method;
c.boo <-  sort(K$C_boot)
up <- quantile(c.boo,.975)
down <- quantile(c.boo,.025)

return(list(backward_model=fit,C_index_corrected=C_indx,
C_app= C_indx_app,ci_95_Cindx=c(down,up)))
}



# This function will claculate the 95% CI for the  HR's from the cox
# regression model;
```

```
CI_coef <- function(obj_bw){

# Arguments;
# obj.bw: backward selection object from Boot.H function,
# Returns: 95% CI of the HR's,

m <- obj_bw$backward_model
Beta_sd <- sqrt(diag(m$var))
Beta <- round(m$coefficients,4)

up <-down <- numeric(length(Beta))
for(i in 1: length(Beta)) {
M <-  exp(Beta[i] + c(-1,1) * 1.96 * Beta_sd[i])
down [i] <-  round(M[1],3)
up[i] <- round(M[2],3)
}
HR<- exp(Beta)
M <- data.frame(HR,down,up )
return(M)
}
```

8.4 Performing backward selection

```
#-----------------------------------#
# Performing backward selection #
#-----------------------------------#

# Model A analysis;

dat.a <- na.omit(dat)# Remove missing values;
A.bw <- Boot.H(200,A.var.org,dat.a)
CI_coef(A.bw) # HR's and their 95% CI


# Cox PH assumptions; model A
test.ph.A <- cox.zph(A.bw$backward_model)
plot(test.ph.A)# schoenefeld residuals
ph.A <- data.frame(test.ph.A$table)

# R to latex
print(xtable(ph.A, type = "latex", tabular.environment="longtable"), file = "PH_
    bwa.tex")


# prognostic scores:


# Model A;

cox_back <- A.bw$backward_model
obj.pred <- predict(cox_back,type = "lp")


# The prognistic scores diveded into quantiles ;
groups <- factor(cut(obj.pred,c(-Inf,
quantile(obj.pred, 0.20),
quantile(obj.pred, 0.40),
quantile(obj.pred, 0.60),
quantile(obj.pred, 0.80),
Inf)),
labels =c("Q1", " Q2", "Q3", "Q4","Q5"))

fit <- survfit(Surv(time, status)~ groups  ,data =dat.a )

library("survminer")
ggsurvplot(
fit ,
size = 0.5 , # change line size
palette = " jco ", #
#conf.int = TRUE , # Add confidence interval
risk.table = TRUE , # Add risk table
censor = FALSE ,
pval = TRUE ,
#risk.table.col = " strata ", # Risk table color by groups
xlab = " Time in years since 1 VT  ",
ylab = " Probability of recurrence ",
break.time.by = 1,
legend.labs = c("Q1", " Q2", "Q3", "Q4","Q5"),
risk.table.height = 0.35 ,
surv.median.line = "hv",
fun = "event",
ggtheme = theme_classic() # Change ggplot2 theme
```

```
)


#-------------#
# Model C     #
#-------------#


C.bw <-    Boot.H(200,C.var.org  ,dat.c)
CI_coef(C.bw) # 95%CI



## Data summary for backward;

df.B <- data.frame(rbind(A.bw$C_app,C.bw$C_app),
rbind(A.bw$C_index_corrected,C.bw$C_index_corrected))

colnames(df.B) <-  c("C_indx app","C_indx corrected")
rownames(df.B)<- c("Full model A"," Model C")


# follow-up time,
mySurvival <- with(dat , Surv(time, status))
summary(mySurvival)



# Cox PH assumptions; model C
test.ph.C <- cox.zph(C.bw$backward_model)
ph.C <- data.frame(test.ph.C$table)
print(xtable(ph.C, type = "latex", tabular.environment="longtable"), file = "PH_
    bwc.tex")


rm(list = c('groups','fit', 'obj.pred','cox_back')) # remove the data

cox_back <- C.bw$backward_model
obj.pred <- predict(cox_back,type = "lp")


# The prognistic scores diveded into quantiles ;
groups <- factor(cut(obj.pred,c(-Inf,
quantile(obj.pred, 0.20),
quantile(obj.pred, 0.40),
quantile(obj.pred, 0.60),
quantile(obj.pred, 0.80),
Inf)),
labels =c("Q1", " Q2", "Q3", "Q4","Q5"))

fit <- survfit(Surv(time, status)~ groups  ,data =dat.c )

ggsurvplot(
fit ,
size = 0.5 , # change line size
palette = " jco ", #
#conf.int = TRUE , # Add confidence interval
risk.table = TRUE , # Add risk table
censor = FALSE ,
pval = TRUE ,
#risk.table.col = " strata ", # Risk table color by groups
```

```
xlab = " Time in years since 1 VT  ",
ylab = " Probability of recurrence ",
break.time.by = 1,
legend.labs = c("Q1", " Q2", "Q3", "Q4","Q5"),
risk.table.height = 0.35 ,
surv.median.line = "hv",
fun = "event",
ggtheme = theme_classic() # Change ggplot2 theme
)
```

8.5 Percentile lasso

```
1
# These are the required R packages to perfom our analyses.
Packages <- c("MASS", "pec" ,"survcomp","survAUC","glmnet") # load multiple
    packages
lapply(Packages, require, character.only = TRUE)

# Function to calculate the C- statistics

C_stat <- function(df) {
# Arguments:
# df: the data frame,
# Return: Harrell C statistics,

time <- df$time
status <- df$status
x <- df$lp
n <- length(time)
r <- order(time, -status)
time <- time[r]
status <- status[r]
x <- x[r]

a <- which(status == 1)
b <- 0
cr <- 0

for (i in a) {
for (j in ((i + 1):n)) {
if (time[j] > time[i]) {
b <- b + 1

if (x[j] < x[i])
cr <- cr + 1 # if j has smaller PI than i we add 1
if (x[j] == x[i])
cr <- cr + 0.5 # if the pairs have same PI we add 0.5
}
}
}
return(round(cr/b, digits = 4))
}



#-----------------------------------------------------------#
# The next function: Per.CV.Err() and Per.Lasso()           #
#  were developed by S. Roberts  and G. Nowak.              #
# Nevertheless, we have introduced some changes to them.    #
#-----------------------------------------------------------#

#-----------------------------------------------------------#
#  Per.CV.Err()  function calculates the unpenalized CV     #
# prediction error for a sequence of lambda values.         #
# These values are specified by a cv.glmmnet object and     #
# a corresponding vector of indices for the lambdas.  Uses  #
#  cross-validation via the glmnet package to estimate      #
#  the prediction error. This version of the function is to  #
# be used with Per.Lasso.                                   #
#-----------------------------------------------------------#
```

```
59
61 Per.CV.Err <- function(x,y,K,cv.glmn,lam.idcs,fam,alpha,cv.rep) {
   # Arguments:
63 #   x - Matrix of variables that are penalized.
   #   y - The response vector.
65 #   K - The number of folds.
   #   cv.glmn - The original cv.glmmnet object on the complete data.
67 #   lam.idcs - The indices of the lambdas (that were used in cv.glmn) for
   #              which to calculate the unpenalized CV error.
69 #   fam - The family used in the original cv.glmnet object.
   #   alpha - The alpha value in the glmnet function for the elastic net.
71 #   cv.rep - The number of repetitions for calculating the CV error.
   #
73 # Returns:
   #   A vector of average CV errors the same length as lam.idcs.
75

77 ## The betas.
   betas <- cv.glmn$glmnet.fit$beta
79
   ## The lambdas.
81 lams <- cv.glmn$lambda

83 ## Number of observations.
   n <- length(y)
85
   ## The matrix of CV errors.
87 res.cvms <- matrix(NA,cv.rep,length(lam.idcs))

89 for (j in 1:cv.rep) {

91 ## New set of fold IDs.
   cur.fld.ids <- sample(rep(1:K,length.out=n))
93
   ## Going through each lambda index.
95 for (i in 1:length(lam.idcs)) {

97 cur.ind <- lam.idcs[i]; cur.beta <- betas[,cur.ind]

99 # If all betas are zero.
   # this is the step where we Re-estimate the parameters
101 # of the selected model (i.e. from the lasso fitted with lambda= lambda(theta))
   # using ordinary ( no penalty), based on two cases;
103

105 if (all(cur.beta==0)) {

107 ## Refit cv.glmnet to complete data.
   cur.cv.glmn <- cv.glmnet(x,y,lambda=lams,foldid= cur.fld.ids,
109 alpha=alpha,family=fam)

111 ## Setting CV error to the corresponding error from
   ## cv.glmnet fitted above.
113 res.cvms[j,i] <- cur.cv.glmn$cvm[cur.ind]

115 } else {
   ## Subsetting x for only non-zero betas.
117 x.nz <- x[,cur.beta!=0,drop=F]

119 if(dim(x.nz)[2]>1) {
   # because, in some cases, 1-SE rule will choose only one variables,
```

```
121 # this will make a problem for cv.glment, because x will not be seen as a matric
        of 2 column.

123
    # this is the step where we Re-estimate the parameters
125 # of the selected model (i.e. from the lasso fitted with lambda = lambda(theta))
    # using ordinary least squares  ( no penality):
127

129 ## Running cv.glmnet without penalty to get CV error.
    npen.lam <- c(0.01,0) # me: no penalty
131 npen.cv <- cv.glmnet(x.nz,y,foldid=cur.fld.ids,alpha=alpha,
    lambda=npen.lam,family=fam)# me: re-estimated model
133 ## CV error.
    #  in this step :
135 # Compute the cross-validation error of the re-estimated model
    res.cvms[j,i] <- npen.cv$cvm[npen.lam==0]
137
    } else{
139 res.cvms[j,i] <- Inf
    }
141 }

143 }

145 }

147 return(colMeans(res.cvms))

149 }

151


153 #-----------------------------------------------------------------#
    # Per.Lasso: This function implements the Percentile Lasso.  #
155 #  It's based on repeatedly fitting the lasso on a different #
    # assignment of cross-validation folds.                      #
157 #-----------------------------------------------------------------#

159
    Per.Lasso <- function(x,y,K=10,alpha=1,M=100,per,fam="cox",cv.rep) {
161 # Arguments:
    #   x - Matrix of variables that are penalized.
163 #   y - Response vector.
    #   K - The number of folds.
165 #   alpha - The alpha value in the glmnet function for the elastic net.
    #   M - The number of times to repeat fold assignments, thus optimal lambdas
167 #   per - The probabilities corresponding to the percentiles of lambda.
    #   fam - The family used in the cv.glmnet function.
169 #   cv.rep - The number of repetitions for calculating the CV error.this
    #            repitition are needed for the new model to compute the CV Deviance
171 #
    # Returns (a list consisting of):
173 #   glmn.fit - The original glmnet model fitted to get the sequence of
    #              lambdas.
175 #   res.sum - A summary of results that states the average unpenalized CV
    #             prediction error and the number of non-zero variables for
177 #             each type of optimal lambda (e.g., minimum or a given
    #             percentile).
179 #   betas - A matrix of beta estimates where each column consists of the
    #           estimates for a particular type of optimal lambda.
181
```

```
183 ##############################################
    ## Initial parameters and run of cv.glmnet. ##
185 ##############################################

187 ## Some parameters.
    p <- ncol(as.matrix(x)); n <- length(y)
189
    ## Setting up the folds.
191 fld.ids <- sample(rep(1:K,length.out=n))

193 ## Running cv.glmnet.
    cv.glmn <- cv.glmnet(x,y,foldid=fld.ids,alpha=alpha,
195 family=fam)
    lams <- cv.glmn$lambda
197 lam.min <- cv.glmn$lambda.min; lam.1se <- cv.glmn$lambda.1se
    mod.glmn <- cv.glmn$glmnet.fit
199 betas.glmn <- mod.glmn$beta

201 ## Result indices and names.
    res.idcs <- c(which(lams==lam.min),which(lams==lam.1se))
203 res.names <- c("min","1se")

205 ##########################################
    ## Running glmnet over repeated folds. ##
207 ##########################################
    #  step 2 and step3 :
209 #Let Lambda(M) = (lambda_1, . . . , lambda_M) denote the M values of lambda_m (
        optimal lambdas).

211 ## Minimum and 1-se lambda for each fold.
    fld.min.lams <- fld.1se.lams <- rep(NA,M)
213
    for (i in 1:M) {
215
    ## Re-choosing folds.
217 new.fld.ids <- sample(fld.ids)

219 ## Running cv.glmnet.
    fld.cv.glmn <- cv.glmnet(x,y,lambda=lams,foldid=new.fld.ids,
221 alpha=alpha,family=fam)

223 ## Minimum and 1se lambdas and errors.
    fld.min.lams[i] <-
225 lams[which(fld.cv.glmn$lambda==fld.cv.glmn$lambda.min)]
    fld.1se.lams[i] <-
227 lams[which(fld.cv.glmn$lambda==fld.cv.glmn$lambda.1se)]

229 }

231 ##########################################################################
    ## Percentiles of the minimum and 1se lambdas from the repeated ##
233 ## folds and percentile with smallest least squares CV error.   ##
    ##########################################################################
235
    ## Choosing appropriate percentiles of minimum and 1se lambdas and
237 ## their corresponding indices.
    #  step 4: Compute lambda(theta), the theta-percentile of Lambda(M).
239
    per.min.lams <- quantile(fld.min.lams,probs=per,type=1)
241 per.min.idcs <- match(per.min.lams,lams)
    per.1se.lams <- quantile(fld.1se.lams,probs=per,type=1)
```

```
per.1se.idcs <- match(per.1se.lams,lams)

## CV errors for the percentiles.
cvms.min <- Per.CV.Err(x,y,K,cv.glmn,per.min.idcs,fam,alpha,cv.rep)
cvms.1se <- Per.CV.Err(x,y,K,cv.glmn,per.1se.idcs,fam,alpha,cv.rep)


## Indices of percentile (lambda) with smallest least squares CV error.
cvm.min.idc <- which(lams==per.min.lams[which.min(cvms.min)])
cvm.1se.idc <- which(lams==per.1se.lams[which.min(cvms.1se)])

## Tacking on to result indices and names.
res.idcs <- c(res.idcs,per.min.idcs,per.1se.idcs,cvm.min.idc,
cvm.1se.idc)
res.names <- c(res.names,paste("min",per),paste("1se",per),"min-cvm",
"1se-cvm")

###############
## Results. ##
###############

## Model sizes and lambdas chosen by each optimal value of lambda.
res.cvms <- Per.CV.Err(x,y,K,cv.glmn,res.idcs,fam,alpha,cv.rep)
res.sum <- data.frame("opt.lams"=lams[res.idcs],"cvm"=res.cvms,
"n.var"=mod.glmn$df[res.idcs],
row.names=res.names)

## Matrix of betas corresponding to the optimal lambdas.
res.betas <- as.matrix(betas.glmn[,res.idcs])
colnames(res.betas) <- res.names

## Returning results.
return(list("glmnet.fit"=mod.glmn,"res.sum"=res.sum,"betas"=res.betas))

}


# Having found the optimal lambda, by the previous functions,
# now  lasso_app() function is applied to found  the apparent C statistics
# and other valuable objects.


lasso_app <- function(X,df,lambda){
# Arguments:
# X: covariates,
# df: data frame,
# lambda: the optimal lambda,
# Returns:
# model,apparnet C index,predicted values and model_fit


Xs <- paste(X, collapse=" + ")
form <- as.formula(paste("Surv(time, status) ~",
paste(Xs, collapse="+")))

# Fit the model in the original data,

x.org <- model.matrix(form, df)
y.org <- Surv(df$time,df$status)

fit <- suppressWarnings(glmnet(x.org, y.org, family="cox", alpha=1,lambda = lambda
    ,
```

```
intercept=FALSE))

# calculation of C index:
pr <- predict(fit, newx = x.org, type = "link")
df.la <- data.frame(time= df$time,status= df$status,
lp= as.numeric(pr)) # needed for C index calculation
C.id <- C_stat(df.la) # this is a function
C_indx_app <- round(C.id,3)# this is the apparent c_index
coef.min = coef(fit, s = lambda)
k <- coef.min[which(coef.min != 0),]
k <- as.matrix(k)
colnames(k) <- "coefficients"

return(list(model= k,C_ind_apparent = C_indx_app ,predicted = pr, fit.model = fit)
    )

}


# The lasso.boot function : this function will apply bootsrap to find
# the corrected C index.


lasso.boot <- function(B,X,df,C_indx_app,lambda) {
# Arguments:
# B: bootsrap number: Harrell et al suggest100-200 times;
# X: covariates;
# df: data frame;
# C_index_app: C index apparent;
# l.min.b : minimal lambda
# return: corrected C index;


Xs <- paste(X, collapse=" + ")
form <- as.formula(paste("Surv(time, status) ~",
paste(Xs, collapse="+")))
# empty Rx2 matrix for bootstrap results
x.org <- model.matrix(form, df)
y.org <- Surv(df$time,df$status)

M <- matrix(0,nrow = B, ncol = 3,
dimnames = list(paste('Sample',1:B),
c("C_orig","C_boot","Optimism")))

n = nrow(df)
set.seed(701)

for(i in 1:B){

# draw a random sample
obs.boot <- sample(x = 1:n, size = n, replace = T)
data.boot <- df[obs.boot, ]

# fit the model on bootstrap sample

x.boot <- model.matrix(form, data.boot)
y.boot <- Surv(data.boot$time,data.boot$status)

fit.boot <- suppressWarnings(glmnet(x.boot, y.boot, family="cox", alpha=1,
lambda = lambda,intercept=FALSE))

# apply model to original data
```

```
365
pr.or <- predict(fit.boot, newx =  x.org, type = "link")
367 df.la.or <- data.frame(time= df$time,status= df$status,
lp= as.numeric(pr.or))# needed for C index calculation
369
M[i, 1] <- C_stat(df.la.or) # C-index_originel sample
371
# apply model to bootstrap data
373 pr.boot <- predict(fit.boot, newx = x.boot,type = "link")
df.la.bo <- data.frame(time= data.boot$time,status= data.boot$status,
375 lp= as.numeric(pr.boot))# needed for C index calculation
M[i, 2] <- C_stat(df.la.bo) # C-index_bootstrap sample
377
# Optimism:
379 M[i,3] <- M[i, 2]- M[i, 1]

381 }
C_indx <-  round(C_indx_app - mean(M[,3]),3)# corrected C-index
383
# K <- head(M,10) # if you want to display everytime the boot.sample statistics
385 # return(list(K,C_index_corrected=C_indx))
return(C_indx)
387 }
```

```
2 #-----------------------------#
# lasso prognistics scores  #
4 #-----------------------------#

6 # model A;

8 rm(list = c('groups','fit', 'obj.pred','cox_back')) # remove the data

10 obj.pred <- app_mod$predicted


12
# The prognistic scores diveded into quantiles ;
14
groups <- factor(cut(obj.pred,c(-Inf,
16 quantile(obj.pred, 0.20),
quantile(obj.pred, 0.40),
18 quantile(obj.pred, 0.60),
quantile(obj.pred, 0.80),
20 Inf)),
labels =c("Q1", " Q2", "Q3", "Q4","Q5"))
22
fit <- survfit(Surv(time, status)~ groups  ,data =dat.a )
24
ggsurvplot(
26 fit ,
size = 0.5 , # change line size
28 palette = " jco ", #
#conf.int = TRUE , # Add confidence interval
30 risk.table = TRUE , # Add risk table
censor = FALSE ,
32 pval = TRUE ,
#risk.table.col = " strata ", # Risk table color by groups
34 xlab = " Time in years since 1 VT  ",
```

```
ylab = " Probability of recurrence ",
break.time.by = 1,
legend.labs = c("Q1", " Q2", "Q3", "Q4","Q5"),
risk.table.height = 0.35 ,
surv.median.line = "hv",
fun = "event",
ggtheme = theme_classic() # Change ggplot2 theme
)

# model C;

rm(list = c('groups','fit', 'obj.pred','cox_back')) # remove the data

obj.pred <- app_mod.c$predicted

# The prognistic scores diveded into quantiles ;
groups <- factor(cut(obj.pred,c(-Inf,
quantile(obj.pred, 0.20),
quantile(obj.pred, 0.40),
quantile(obj.pred, 0.60),
quantile(obj.pred, 0.80),
Inf)),
labels =c("Q1", " Q2", "Q3", "Q4","Q5"))

fit <- survfit(Surv(time, status)~ groups  ,data =dat.c )

ggsurvplot(
fit ,
size = 0.5 , # change line size
palette = " jco ", #
#conf.int = TRUE , # Add confidence interval
risk.table = TRUE , # Add risk table
censor = FALSE ,
pval = TRUE ,
#risk.table.col = " strata ", # Risk table color by groups
xlab = " Time in years since 1 VT  ",
ylab = " Probability of recurrence ",
break.time.by = 1,
legend.labs = c("Q1", " Q2", "Q3", "Q4","Q5"),
risk.table.height = 0.35 ,
surv.median.line = "hv",
fun = "event",
ggtheme = theme_classic() # Change ggplot2 theme
)
```

8.6 Performing lasso analyses in conjunction with percentile lasso

```
1
2
3 #--------------------------------#
  # lasso and percentile lasso     #
5 #--------------------------------#
6
7 ################
  # Model A      #
9 ################
10
11 # Remove missing values;
   dat.a <- na.omit(dat)
13
   X <- A.var.org # variables
15 df <- dat.a
16
17 # objects for percentile lasso;
   Xs <- paste(X, collapse=" + ")
19 form <- as.formula(paste("Surv(time, status) ~",
   paste(Xs, collapse="+")))
21
   x <- model.matrix(form, df)
23 y <- Surv(df$time,df$status)
24
25 library(ggplot2)
26
27 A <- list()
   for(i in 1:345){
29 A[[length(A)+1]] <- Per.Lasso(x,y,K=10,per=c(0.75,0.8,0.85,0.9,0.95),cv.rep = 10)$
       res.sum
30
31 }
32
33 h <- matrix(0,length(A),4)
   for(i in 1:length(A)){
35 k <- A[[i]]
   h[i,1] <- k[1,3] # variables by lasso
37 h[i,2] <- k[7,3] # variables by percentile
   h[i,3] <- round(k[1,1],4) # extract the lambdas_min
39 h[i,4] <- round(k[7,1],4) # extract the lambdas at cvm
   colnames(h) <- c("stand","per","lam.min","lam.cvm")
41
42 }
43
44 H <- data.frame(h)
45
46 # Lambdas
47 lam.min <- table(H$lam.min)
   stand.l.min <- data.frame(lam.min)
49
50 #--------#
51 # plot   #
   #--------#
53
54 library(ggplot2)
55
   ggplot(stand.l.min, aes(x = Var1, y = Freq)  ) +
```

```
geom_bar(stat = "identity",fill = "yellow") +
geom_text(aes(label = Freq),vjust = -0.3,color ="red") + ###
labs( y = "Frequencies",x=expression(hat(lambda)),
title = "All possible tuning parameter by ordinary lasso")+
theme_bw()+
theme(plot.title = element_text(hjust = 0.5,size = rel(1.5)),
axis.text = element_text(size = rel(1.1),color = "black"),
axis.title.y = element_text(size = rel(1.3) ),
axis.title.x = element_text(size = rel(1.3) ),
axis.text.x = element_text(angle = 45, hjust = 1))


# non-zero coeffiecients

V <- table(H$stand)
V.df <- data.frame(V)

ggplot(V.df , aes(x = Var1, y = Freq), hjust = -0.2) +
geom_bar(stat = "identity",fill = "skyblue",width = 0.7) +
geom_text(aes(label = Freq), hjust = -0.2,color ="red") + ###
coord_flip()+
labs(x = "Non-zero coeffiecients", y = "Frequencies",
title = "All possible selected model by ordinary lasso")+
theme_bw()+
theme(plot.title = element_text(hjust = 0.5,size = rel(1.5)),
axis.text = element_text(size = rel(1.1),color = "black"),
axis.title.y = element_text(size = rel(1.3) ),
axis.title.x = element_text(size = rel(1.3) ))



#-------------------------#
# plot percentile lasso   #
#-------------------------#

# Lambdas:
lam.p <- table(H$lam.cvm)
lam.per <- data.frame(lam.p)

#plots:
ggplot(lam.per, aes(x = Var1, y = Freq)  ) +
geom_bar(stat = "identity",fill = "green",width = 0.6) +
geom_text(aes(label = Freq),vjust = -0.3,color ="red") + ###
labs( y = "Frequencies",x=expression(hat(lambda)),
title = "All possible tuning parameter by percentile-lasso")+
theme_bw()+
theme(plot.title = element_text(hjust = 0.5,size = rel(1.5)),
axis.text = element_text(size = rel(1.1),color = "black"),
axis.title.y = element_text(size = rel(1.3) ),
axis.title.x = element_text(size = rel(1.3) ),
axis.text.x = element_text(angle = 45, hjust = 1))


# non-zero coeffiecients plots
P <- table(H$per)
per.l <- data.frame(P)


ggplot(per.l, aes(x = Var1, y = Freq) ) +
geom_bar(stat = "identity",fill = "red",width = 0.7) +
geom_text(aes(label = Freq), vjust = -0.2,color ="blue") + ###
labs(x = "Non-zero coeffiecients", y = "Frequencies",
```

```
119 title = "All possible selected model by percentile-lasso")+
    theme_bw()+
121 theme(plot.title = element_text(hjust = 0.5,size = rel(1.5)),
    axis.text = element_text(size = rel(1.1),color = "black"),
123 axis.title.y = element_text(size = rel(1.3) ),
    axis.title.x = element_text(size = rel(1.3) ))
125


127 # lambdas: 0.013985781 -->  11 variables
    # lambdas: 0.011611241 --> 14 variables
129
    # Model A: results
131
    per.lam <- 0.01398578  # optimal percentile lambda;
133 app_mod <- lasso_app(A.var.org,dat.a,per.lam)
    C_indx.A <- lasso.boot(200,A.var.org,dat.a, app_mod$C_ind_apparent ,per.lam)
135


137 # at the 1 SE rule for A
    pr.1se.a <- 0.03545905 # only 3 variables were chosen
139 app_mod.1se.a <- lasso_app(A.var.org,dat.a,pr.1se.a)
    C_indx.1se.a <- lasso.boot(200,A.var.org,dat.a, app_mod.1se.a$C_ind_apparent ,pr.1
        se.a)
141



143
    # Transform coeffiecients to HR;
145 HR.A <- exp(app_mod$model)


147
    # Testing the PH assumptions for lasso model A:
149
    coef.min <- coef(app_mod$fit.model , s = "lambda.min")
151 k <- coef.min[which(coef.min != 0),]
    L <- names(k)
153 L[5] <- "TypeVT"
    L[7] <- "PTS_J"
155 M <- L[-6]# remove the TypeVT3

157 Xs <- paste(M, collapse=" + ")
    form <- as.formula(paste("Surv(time, status) ~",
159 paste(Xs, collapse="+")))

161 cox.obj.a <- coxph(form, data =  dat.a, x=TRUE)
    D <- cox.zph(cox.obj.a)
163
    f <- data.frame(D$table)# change the name to english for thesis
165 rownames(f)<- c("Surgery","Pregnant","Hormone","Gender","TypeVT2","TypeVT3","PTS_
        J1","PTS_J2","Leiden V",
    "factor VIII","VWF","D-dimer","GLOBAL")
167 f <- round(f,4)

169 # R outpt to latex:

171 library(xtable)
    print(xtable(f, type = "latex", tabular.environment="longtable"), file = "PH.tex")
173



175
    #-------------#
177 # Model C     #
    #-------------#
```

```
179
# Remove the laboratory variables;
181 dat.C <- subset(dat, select = -c(fib, AT3, PCC, F7C, f9aggem, f2c, F10C, F11, lognAPCsr,
factorV, TFPI, Hgb, nETPLT, logf8aggem, logvwf, logpsfree,
183 logddimer, logWBC, logmoperc, logrdw, logHsCRP))

185 dat.c <- na.omit(dat.C)# remove messing values;

187
X <- C.var.org
189 df <- dat.c

191 Xs <- paste(X, collapse=" + ")
form <- as.formula(paste("Surv(time, status) ~",
193 paste(Xs, collapse="+")))

195 x <- model.matrix(form, df)
y <- Surv(df$time, df$status)
197

199 ## Which percentile to choose:

201 # we will repeat (345times) per.lasso to check for its stability; thus to choose
    one model.
# note that percentile lasso will often provide you two models,
203 # models arranging between 11 and 14 variables. choose one model.
# we note that there is almost no difference in model performence (apparent_c-indx
    ),
205 # therfore we will choose model with 11 variables.--> see the next code;

207 C <- list()
for(i in 1:345){
209 C[[length(C)+1]] <- Per.Lasso(x,y,K=10,per=c(0.75,0.8,0.85,0.9,0.95),cv.rep = 10)$
    res.sum

211 }

213 # lambdas: 0.004330508 --> 14 variables(only)

215 # Model C: results

217 per.lam.c <-  0.004330508 # optimal percentile lambda;
app_mod.c <- lasso_app(C.var.org, dat.c, per.lam.c)
219 C_indx.c <- lasso.boot(200,C.var.org, dat.c, app_mod.c$C_ind_apparent ,per.lam.c)

221 # at the 1 SE-rule for C

223 pr.1se <- 0.025363873
app_mod.1se <- lasso_app(C.var.org, dat.c, pr.1se)
225 C_indx.1se <- lasso.boot(200,C.var.org, dat.c, app_mod.1se$C_ind_apparent ,pr.1se)

227
# Transform coeffiecients to HR;
229
HR.c <- exp(app_mod.c$model)
231
exp(app_mod.1se$model)
233
# Testing the PH assumptions for lasso model C:
235
coef.min <- coef(app_mod.c$fit.model , s = "lambda.min")
237 k <- coef.min[which(coef.min != 0),]
```

```
L <- names(k)
L[9] <- "TypeVT"
L[11] <- "PTS_J"
M <- L[-c(10,12)]# remove the TypeVT3 and PTS_J2

Xs <- paste(M, collapse=" + ")
form <- as.formula(paste("Surv(time, status) ~",
paste(Xs, collapse="+")))

cox.obj.c <- coxph(form, data =  dat.c, x=TRUE)

D <- cox.zph(cox.obj.c)
f <- data.frame(D$table)

# change the names.
rownames(f)<- c("Surgery","Plaster cast","Hospitalization","Pregnant","Hormone","
    Cardio-disease","Location VT",
"Gender","TypeVT2","TypeVT3", "PTS_J1","PTS_J2","Blood-type","Leiden V","GLOBAL")

f <- round(f,4)
print(xtable(f, type = "latex", tabular.environment="longtable"), file = "PH_c.tex
    ")



#-----------------------------#
#  figures lasso chapter      #
#-----------------------------#

# Figures for lasso : lambda and partial likelihood

library(glmnet)


# Model A:

# Remove missing values;
dat.a <- na.omit(dat)#
X <- A.var.org
df <- dat.a

# objects for percentile lasso;
Xs <- paste(X, collapse=" + ")
form <- as.formula(paste("Surv(time, status) ~",
paste(Xs, collapse="+")))

x <- model.matrix(form, df)
y <- Surv(df$time,df$status)


## Running cv.glmnet.
cvfit <-  cv.glmnet(x, y, family = "cox")
plot(cvfit)

coef.min = coef(cvfit, s = "lambda.1se")
active.min = which(coef.min != 0)
index.min = coef.min[active.min]

# Plot the path:
fit = glmnet(x, y, family = "cox")
plot(fit, xvar = "lambda", label = TRUE)
```

8.7 Closed testing analyses

```
#-----------------------------#
# Closed testing analysis    #
#-----------------------------#

# load multiple packages
Packages <- c("MASS", "pec" ,"survcomp","survAUC","Hmisc","cherry")
lapply(Packages, require, character.only = TRUE)


## Model C: Closed testing is applied only to model,c

# fit the full model
fullfit <-  coxph(Surv(time,status)~ ., data = dat.c )

# The test function:
# This function is used as our local test for in closed testing procedure

mytest <- function(H_I) {
# Arguments :
# H_I : the intersection hypothesis of interest,
# Return: p_value:   are the regression coefficient 0 or not?

# fit the full model
others <- setdiff(C.var.org, H_I)
a <- 1
others <- get(ifelse(length(others)==0,"a", "others"))
Xs <- paste(others, collapse=" + ")
form <- as.formula(paste("Surv(time, status) ~",
paste(Xs, collapse="+")))
cx <- coxph(form , data = dat.c,x=TRUE)
anov <- anova(cx, fullfit ,test='LRT')
pvalue <-anov$P[2]
return(pvalue)
}

# perfoming closed testing:

ct <- closed(mytest,C.var.org,alpha = 0.1)
def <- defining(ct)
a <- shortlist(ct)

# check the number of TD (True Discovery) and FD (False Discovery)
pick(ct,a[[22]])

# to find out the the number of all false hypotheses;

ct1 <- numeric(length(a))
d <- numeric(length(a))
for(i in 1:length(a)){
m <- shortlist(ct)[[i]]
d[i] <- length(shortlist(ct)[[i]])
ct1[i] <- pick(ct,m )
#print(ct1)
}

# create a data frame containing the number hypothesis
# and their number of false hypotheses.

CT <- data.frame(Hyp=d,fals = ct1)
```

```
60
62
#-------------------------------#
64 # Barplots of shortlists      #
#-------------------------------#
66
# We create a barplot for each shortlist model.
68 # these are the variables names of the chosen models of 6 ,7,8,9 and 10 variables:

70 nm6 <- c("Sur \n Prg \n Hor \n Loc \n Sx \n PC","Sur \n Prg \n Hor \n Loc \n Sx \n
      Typ ",
"Sur \n Prg \n Hor \n Loc \n Sx \n VLei")
72
nm7 <- c("Sur \n Prg \n Hor  \n Sx \n Typ \n PC \n VLei"  ,"Sur \n Prg \n Hor \n
     Sx  \n Typ \n Disc \n VLei ",
74 "Sur \n Prg \n Hor \n Sx \n Typ \n Blt \n VLei "    ,"Sur \n Prg \n Hor \n Sx \n
     Typ \n PTS  \n VLei ",
"Sur \n Prg \n Hor \n Sx \n Loc \n Disc \n PTS", "Sur \n Prg \n Hor \n Sx \n Loc \
     n Disc  \n Blt ",
76 "Sur \n Prg \n Hor \n Sx \n Loc \n PTS \n Blt ")

78

80 nm8  <- c("Sur \n PC \n Hor \n Sx \n Typ \n Loc \n Disc \n VLei ","Sur \n PC  \n
     Hor \n Sx \n Typ \n Loc \n Blt \n VLei",
"Sur \n PC  \n Hor \n Sx \n Typ \n Loc \n PTS \n VLei","Sur \n PC \n Hor \n Sx \n
     Typ \n Prg \n PTS \n Disc ",
82 "Sur \n PC  \n Hor \n Sx \n Typ \n Prg \n PTS \n Blt")

84 nm9 <- c("Sur \n Loc \n VLei \n Blt \n PC \n Hor \n Sx  \n Disc \n PTS ", "Sur \n
     Loc \n VLei \n Blt \n PC  \n Hor \n Prg \n Typ \n Disc ",
"Sur \n Loc \n VLei \n Blt \n PC \n Prg  \n Sx \n Typ \n PTS ","Sur \n Loc \n VLei
      \n Blt \n PC  \n Hor \n Prg \n Typ \n PTS ",
86 "Sur \n Loc \n VLei \n Blt \n Disc \n Hor \n Sx \n Typ \n PTS ","Sur \n Loc \n
     VLei \n Prg \n PC \n Disc \n Sx \n Typ \n PTS")

88

90 nm10 <- c("PC \n Hosp \n Preg \n Hor \n Disc \n Loc  \n Sx \n Typ \n VLei\n PTS  "
     ,
"PC \n Hosp \n Preg \n Hor \n Disc \n Loc  \n Sx \n Typ \n VLei \n Blt")
92

94 m6 <- which(CT$Hyp==6)# hypo of 6 variables
m7 <- which(CT$Hyp==7)# hypo of 7 variables
96 m8 <- which(CT$Hyp==8)# hypo of 8 variables
m9 <- which(CT$Hyp==9)# hypo of 9 variables
98 m10 <- which(CT$Hyp==10)# hypo of 10 variables

100

102 # these are the chosen models.
# m6 : 8
104 # m7 : 20
# m8 : 7
106 # m9 : 5
# m10: 1
108
# Barplot of model with 6 variables.
110 bp6 <- barplot(CT$Hyp[m6],
#main=" Shortlist models of 6 variables",
```

```
112 axes=FALSE, col="yellow",
    xlab="Number of true discoveries")
114 text(bp6, 1.5 ,nm6, cex=1.5,pos=3 )
    axis(2,seq(0,10,1), line=-0.5, cex.axis=1 )
116 axis(1, at=bp6, labels=CT$fals[m6], tick=FALSE, line=-0.5, cex.axis=1)

118

    # Barplot of model with 7 variables.
120 bp7 <- barplot(CT$Hyp[m7],
    #main=" Shortlist models of 7 variables",
122 axes=FALSE, col="red1",
    xlab="Number of true discoveries")
124 text(bp7, 1,nm7, cex=1.5,pos=3 )
    axis(2,seq(0,10,1), line=-0.5, cex.axis=1 )
126 axis(1, at=bp7, labels=CT$fals[m7], tick=FALSE, line=-0.5, cex.axis=1)

128

    # Barplot of model with 8 variables.
130 bp8 <- barplot(CT$Hyp[m8],
    #main=" Shortlist models of 8 variables",
132 axes=FALSE, col="tan3",
    xlab="Number of true discoveries")
134 text(bp8, 1,nm8, cex=1.5,pos=3 )
    axis(2,seq(0,10,1), line=-0.5, cex.axis=1 )
136 axis(1, at=bp8, labels=CT$fals[m8], tick=FALSE, line=-0.5, cex.axis=1)

138 # Barplot of model with 9 variables.
    bp9 <- barplot(CT$Hyp[m9],
140 #main=" Shortlist models of 9 variables",
    axes=FALSE, col="chartreuse",
142 xlab="Number of true discoveries")
    text(bp9, 1,nm9, cex=1.5,pos=3)
144 axis(2,seq(0,10,1), line=-0.5, cex.axis=1 )
    axis(1, at=bp9, labels=CT$fals[m9], tick=FALSE, line=-0.5, cex.axis=1)
146

148 # Barplot of model with 10 variables.
    bp10 <- barplot(CT$Hyp[m10], #main=" Shortlist models of 10 variables",
150 axes=FALSE, col="#009999",
    xlab="Number of true discoveries")
152 text(bp10, 0.25,nm10, cex=1.5,pos=3 )
    axis(2,seq(0,11,1), line=-0.5, cex.axis=1 )
154 axis(1, at=bp10, labels=CT$fals[m10], tick=FALSE, line=-0.5, cex.axis=1)

156

158 #-----------------------------------#
    # correlations for definingsets  #
160 #-----------------------------------#

162 library(ggcorrplot)
    library(Hmisc)
164 vr <- c("oper3mnd","gips3mnd","bedzk3mnd","zwan3mnd","hormoon","discardio","
        locatie1varII","sex_J","TypeVT","PTS_J",
    "blgroup","fvleiden_J","time","status")
166

    data_frame <- dat.c[,vr]
168 # change columnames to english:
    k <- c("Surgery","Plaster cast","Hospitalization","Pregnant","Hormone","Cardio-
        disease","Location VT","Gender","TypeVT","PTS",
170 "Blood-type","Leiden V","time","status")
    colnames(data_frame)<- k
```

```
corr <- as.data.frame(rcorr(as.matrix(data_frame),type = "pearson")$r)
ggcorrplot(corr, hc.order = TRUE, type = "lower",lab = TRUE)


#-------------------------#
# Application: subset   #
#-------------------------#


#The full model.
hypotheses <- c( "locatie1varII","sex_J","oper3mnd", "bmi" )

Xs <- paste(hypotheses, collapse=" + ")
form <- as.formula(paste("Surv(time, status) ~",
paste(Xs, collapse="+")))

# Full model for the  subset.
fullfit_sub <-  coxph(form, data = dat.c )


#The test function;

test <- function(H_I) {
# Arguments :
# H_I : the intersection hypothesis of interest,
# Return: p_value:   are the regression coefficient 0 or not?
# fit the full model
others <- setdiff(hypotheses, H_I)
a <- 1
others <- get(ifelse(length(others)==0,"a", "others"))
Xs <- paste(others, collapse=" + ")
form <- as.formula(paste("Surv(time, status) ~",
paste(Xs, collapse="+")))
cx <- coxph(form , data = dat.c,x=TRUE)
anov <- anova(cx, fullfit_sub ,test='LRT')
pvalue <-anov$P[2]
return(pvalue)
}



ct.sub <- closed(test, hypotheses)

# check the number of true and false discoveries.
pick(ct.sub,c("locatie1varII","sex_J"))
pick(ct.sub,c("oper3mnd", "bmi"))



# The defining and the shortlist for the example;
defining(ct.sub)
shortlist(ct.sub)

#---------------------------------#
# Bootsrap for closed testing   #
#---------------------------------#

library("riskRegression")

# This function to perform bootsrap internal validation for closed testing.

Boot.H.CT <- function(B,X) {
```

```
# Arguments:
# X: covariates,
# B: number of bootsrap,
# return: backward model,C_corrected,C_apparent and 95% C ci.

Xs <- paste(X, collapse=" + ")
form <- as.formula(paste("Surv(time, status) ~",
paste(Xs, collapse="+")))

# Fit the model in the original data: the apparten c_index
cox1 <- coxph(form, data =  dat.c, x=TRUE)
v <- summary(cox1)

# This is the apparent c_index
v <- summary(cox1)
C_indx_app <-v$concordance


# Empty  matrix to store the bootstrap results,
M <- matrix(nrow = B, ncol = 3,
dimnames = list(paste('Sample',1:B),
c("C_orig","C_boot","Optimism")))

n = nrow(dat.c)
set.seed(1)
for(i in 1:B){

# draw a random sample
obs.boot <- sample(x = 1:n, size = n, replace = T)
data.boot <- dat.c[obs.boot, ]

# fit the model on bootstrap sample
cox.bo <- coxph(form, data=data.boot, x=TRUE)



# apply model to original data
ttt <- quantile(dat.c$time)
k <- predictSurvProb(cox.bo,newdata=dat.c,times=ttt)
harrelC1 <- rcorr.cens(k[,3],with(dat.c,Surv(time, status)))
M[i, 1] <- harrelC1[1]

# apply model to bootstrap data
tt <- quantile(data.boot$time)
k1 <- predictSurvProb(cox.bo,newdata=data.boot,times=tt)
harrelC2 <- rcorr.cens(k1[,3],with(data.boot,Surv(time, status)))
M[i, 2] <- harrelC2[1]

# optimism
M[i,3] <- M[i, 2]- M[i, 1]

}
C_indx <-  C_indx_app - mean(M[,3])
K <- data.frame(M)

# confidence interval for C statistics: by Percentile Method
c.boo <-  sort(K$C_boot)
up <- quantile(c.boo,.975)
down <- quantile(c.boo,.025)

return(list(model=cox1,C_index_corrected=C_indx,
C_app= C_indx_app,ci_95_Cindx=c(down,up)))
```

```
}



# Cox model summary for all shortlist chosen models;

# model 6:
X6 <- paste(a[[8]], collapse=" + ")
form6 <- as.formula(paste("Surv(time, status) ~",
paste(X6, collapse="+")))

cox6 <- coxph(form6, data =  dat.c, x=TRUE)
summary(cox6)

# model 7:
X7 <- paste(a[[20]], collapse=" + ")
form7 <- as.formula(paste("Surv(time, status) ~",
paste(X7, collapse="+")))

cox7 <- coxph(form7, data =  dat.c, x=TRUE)
summary(cox7)


# model 8:
X8 <- paste(a[[7]], collapse=" + ")
form8 <- as.formula(paste("Surv(time, status) ~",
paste(X8, collapse="+")))

cox8 <- coxph(form8, data =  dat.c, x=TRUE)
summary(cox8)

# model 9:
X9 <- paste(a[[5]], collapse=" + ")
form9 <- as.formula(paste("Surv(time, status) ~",
paste(X9, collapse="+")))

cox9 <- coxph(form9, data =  dat.c, x=TRUE)
summary(cox9)

# model 10:
X10 <- paste(a[[1]], collapse=" + ")
form10 <- as.formula(paste("Surv(time, status) ~",
paste(X10, collapse="+")))

cox10 <- coxph(form10, data =  dat.c, x=TRUE)
summary(cox10)

# This function will find the 95% CI for the
# HR from the cox regression model.

CI_coef <- function(cx){
# Arguments;
# cx: cox model object;
# Returns: 95% CI of the coeffiecients;
Beta_sd <- sqrt(diag(cx$var))
Beta <- round(cx$coefficients,4)

up <-down <- numeric(length(Beta))
for(i in 1: length(Beta)) {
M <-  exp(Beta[i] + c(-1,1) * 1.96 * Beta_sd[i])
down [i] <-  round(M[1],3)
```

```
up[i] <- round(M[2],3)
}
HR<- round(exp(Beta),3)
M <- data.frame(HR,down,up )
return(M)
}

CI_coef(cox6)
CI_coef(cox7)
CI_coef(cox8)
CI_coef(cox9)
CI_coef(cox10)


# perfomance and predictores for  the chosen models:
# the c index

Boot.H.CT(200,a[[1]])
Boot.H.CT(200,a[[5]])
Boot.H.CT(200,a[[7]])
Boot.H.CT(200,a[[20]])
Boot.H.CT(200,a[[8]])


# Function to check the Assumption cox :


PH <-  function(i){
# Argument:
# i : model
# return:

Xs <- paste(a[[i]], collapse=" + ")
form <- as.formula(paste("Surv(time, status) ~",
paste(Xs, collapse="+")))

cox1 <- coxph(form, data =  dat.c, x=TRUE)
return(cox.zph(cox1))

}

# Check the PH assumtion for the chosen models

# model 6:
ph6 <- data.frame(PH(8)$table)
print(xtable(ph6, type = "latex", tabular.environment="longtable"), file = "PH_6.
    tex")

# model 9:
ph9 <- data.frame(PH(9)$table)
print(xtable(ph9, type = "latex", tabular.environment="longtable"), file = "PH_9.
    tex")


#-----------------------------------------#
# Prognistic plots : quantiles plot   #
#-----------------------------------------#

## MODEL 6:

library("survminer")
```

```
Xs <- paste(a[[8]], collapse=" + ")
form <- as.formula(paste("Surv(time, status) ~",
paste(Xs, collapse="+")))

cox1 <- coxph(form, data =  dat.c, x=TRUE)
obj.pred <- predict(cox1,type = "lp")


# The prognistic scores diveded into quantiles ;
groups <- factor(cut(obj.pred,c(-Inf,
quantile(obj.pred, 0.20),
quantile(obj.pred, 0.40),
quantile(obj.pred, 0.60),
quantile(obj.pred, 0.80),
Inf)),
labels =c("Q1", " Q2", "Q3", "Q4","Q5"))

fit <- survfit(Surv(time, status)~ groups  ,data =dat.c )

ggsurvplot(
fit ,
size = 1 , # change line size
palette = c("red","blue", "#009999","yellow","orange") ,#
#conf.int = TRUE , # Add confidence interval
risk.table = TRUE , # Add risk table
censor = FALSE ,
pval = FALSE ,
#risk.table.col = " strata ", # Risk table color by groups
xlab = " Time in years since 1 VT  ",
ylab = " Probability of recurrence ",
break.time.by = 2,
legend.labs = c("Q1", " Q2", "Q3", "Q4","Q5"),
risk.table.height = 0.35 ,
surv.median.line = "hv",
fun = "event",
ggtheme = theme_classic() # Change ggplot2 theme
)



## MODEL 7

Xs <- paste(a[[14]], collapse=" + ")
form <- as.formula(paste("Surv(time, status) ~",
paste(Xs, collapse="+")))

cox1 <- coxph(form, data =  dat.c, x=TRUE)
obj.pred <- predict(cox1,type = "lp")


# The prognistic scores diveded into quantiles ;
groups <- factor(cut(obj.pred,c(-Inf,
quantile(obj.pred, 0.20),
quantile(obj.pred, 0.40),
quantile(obj.pred, 0.60),
quantile(obj.pred, 0.80),
Inf)),
labels =c("Q1", " Q2", "Q3", "Q4","Q5"))

fit <- survfit(Surv(time, status)~ groups  ,data =dat.c )

ggsurvplot(
```

```
fit ,
size = 1 , # change line size
palette = " jco ", #
#conf.int = TRUE , # Add confidence interval
risk.table = TRUE , # Add risk table
censor = FALSE ,
pval = FALSE ,
#risk.table.col = " strata ", # Risk table color by groups
xlab = " Time in years since 1 VT  ",
ylab = " Probability of recurrence ",
break.time.by = 2,
legend.labs = c("Q1", " Q2", "Q3", "Q4","Q5"),
risk.table.height = 0.35 ,
surv.median.line = "hv",
fun = "event",
ggtheme = theme_classic() # Change ggplot2 theme
)

## MODEL 8

Xs <- paste(a[[7]], collapse=" + ")
form <- as.formula(paste("Surv(time, status) ~",
paste(Xs, collapse="+")))

cox1 <- coxph(form, data =  dat.c, x=TRUE)
obj.pred <- predict(cox1,type = "lp")


# The prognistic scores diveded into quantiles ;
groups <- factor(cut(obj.pred,c(-Inf,
quantile(obj.pred, 0.20),
quantile(obj.pred, 0.40),
quantile(obj.pred, 0.60),
quantile(obj.pred, 0.80),
Inf)),
labels =c("Q1", " Q2", "Q3", "Q4","Q5"))

fit <- survfit(Surv(time, status)~ groups  ,data =dat.c )

ggsurvplot(
fit ,
size = 1 , # change line size
palette = " jco ", #
#conf.int = TRUE , # Add confidence interval
risk.table = TRUE , # Add risk table
censor = FALSE ,
pval = FALSE ,
#risk.table.col = " strata ", # Risk table color by groups
xlab = " Time in years since 1 VT  ",
ylab = " Probability of recurrence ",
break.time.by = 2,
legend.labs = c("Q1", " Q2", "Q3", "Q4","Q5"),
risk.table.height = 0.35 ,
surv.median.line = "hv",
fun = "event",
ggtheme = theme_classic() # Change ggplot2 theme
)



## Model 9 :
```

```
Xs <- paste(a[[6]], collapse=" + ")
form <- as.formula(paste("Surv(time, status) ~",
paste(Xs, collapse="+")))

cox1 <- coxph(form, data =  dat.c, x=TRUE)
obj.pred <- predict(cox1,type = "lp")


# The prognistic scores diveded into quantiles ;
groups <- factor(cut(obj.pred,c(-Inf,
quantile(obj.pred, 0.20),
quantile(obj.pred, 0.40),
quantile(obj.pred, 0.60),
quantile(obj.pred, 0.80),
Inf)),
labels =c("Q1", " Q2", "Q3", "Q4","Q5"))

fit <- survfit(Surv(time, status)~ groups  ,data =dat.c )

ggsurvplot(
fit ,
size = 1 , # change line size
palette = " jco ", #
#conf.int = TRUE , # Add confidence interval
risk.table = TRUE , # Add risk table
censor = FALSE ,
pval = FALSE ,
#risk.table.col = " strata ", # Risk table color by groups
xlab = " Time in years since 1 VT  ",
ylab = " Probability of recurrence ",
break.time.by = 2,
legend.labs = c("Q1", " Q2", "Q3", "Q4","Q5"),
risk.table.height = 0.35 ,
surv.median.line = "hv",
fun = "event",
ggtheme = theme_classic() # Change ggplot2 theme
)


## model 10:

Xs <- paste(a[[1]], collapse=" + ")
form <- as.formula(paste("Surv(time, status) ~",
paste(Xs, collapse="+")))

cox1 <- coxph(form, data =  dat.c, x=TRUE)
obj.pred <- predict(cox1,type = "lp")


# The prognistic scores diveded into quantiles ;
groups <- factor(cut(obj.pred,c(-Inf,
quantile(obj.pred, 0.20),
quantile(obj.pred, 0.40),
quantile(obj.pred, 0.60),
quantile(obj.pred, 0.80),
Inf)),
labels =c("Q1", " Q2", "Q3", "Q4","Q5"))

fit <- survfit(Surv(time, status)~ groups  ,data =dat.c )

ggsurvplot(
fit ,
```

```
size = 1 , # change line size
palette = " jco ", #
#conf.int = TRUE , # Add confidence interval
risk.table = TRUE , # Add risk table
censor = FALSE ,
#risk.table.col = " strata ", # Risk table color by groups
xlab = " Time in years since 1 VT  ",
ylab = " Probability of recurrence ",
break.time.by = 2,
legend.labs = c("Q1", " Q2", "Q3", "Q4","Q5"),
risk.table.height = 0.35 ,
surv.median.line = "hv",
fun = "event",
ggtheme = theme_classic() # Change ggplot2 theme
)
```

8.8 Nomograms for survival analysis

```
2
#-----------------------------------#
4 # Nomogram plot by Harrell        #
#-----------------------------------#
6
library(rms)
8 library(survival)

10 # data fopr model C:
# model c_6 had the next predictors
12
K <- dat.c[,c("oper3mnd","gips3mnd","hormoon","zwan3mnd","locatie1varII","sex_J","
    time","status")]
14 dat.K <- data.frame(K)

16 # change the columnnames to english names;
colnames(dat.K) <- c("Surgery","Plaster_cast","Hormone_use","Pregnant","Location_
    VT","Gender","time","status")
18
dat.K[,c(1:6)] <- lapply(dat.K[,c(1:6)] , factor)# make the columns as factors
20
# change 0 to no and 1 to yes, and gender codeing too;
22 dat.K$Gender <- factor(dat.K$Gender,labels=c('female','male'))
dat.K$Plaster_cast <- ifelse(dat.K$Plaster_cast==1,"yes","no")
24 dat.K$Surgery <- ifelse(dat.K$Surgery==1,"yes","no")
dat.K$Hormone_use  <- ifelse(dat.K$Hormone_use==1,"yes","no")
26 dat.K$Pregnant <- ifelse(dat.K$Pregnant==1,"yes","no")
dat.K$Location_VT <- ifelse(dat.K$Location_VT==1,"proximal","distal")
28
# fit the model,
30 X<- c("Surgery","Plaster_cast","Hormone_use","Pregnant","Location_VT","Gender")
Xs <- paste(X, collapse=" + ")
32 form <- as.formula(paste("Surv(time, status) ~",
paste(Xs, collapse="+")))
34 cox.obj <- cph(form , data =  dat.K ,surv=TRUE)

36 # using the Harrell code to generate nomograms,
ddist <- datadist(dat.K)
38 options(datadist='ddist')
surv.cox <- Survival(cox.obj)
40
nom.cox <- nomogram(cox.obj, fun=list(function(x)
42 surv.cox(2, x), function(x) surv.cox(5, x)),
funlabel=c("2-Year Sur. Prob.","5-Year Sur. Prob."),lp=F)
44
plot(nom.cox)
46
# generate angle number: we need to change some probabilities
48 # because they were not visible on the plot.

50 labels <- seq(0.8,0.98,0.02)
mp <- barplot(1:12, axes = FALSE, axisnames = FALSE)
52 text(mp, par("usr")[3], labels = labels, srt = 70, adj = c(1.1,1.1), xpd = TRUE,
    cex=.9)
axis(2)
54

56 #-----------------------#
```

```
# nomogram version II #
#-----------------------#

#install.packages("regplot")


source("D:/Locker/D_documents/studie/master/Thesis/data/data_prep_factors.R")

library(survival)
library(regplot)

# data:

dat.C <- subset(dat, select = -c(fib ,AT3,PCC,F7C,f9aggem ,f2c ,F10C,F11,lognAPCsr,
factorV ,TFPI,Hgb,nETPLT,logf8aggem ,logvwf ,logpsfree ,
logddimer ,logWBC,logmoperc ,logrdw ,logHsCRP))

dat.c <- na.omit(dat.C)# remove messing values;

K <- dat.c[,c("oper3mnd","gips3mnd","hormoon","zwan3mnd","locatie1varII","sex_J","
    time","status" )]
dat.K <- data.frame(K)
colnames(dat.K) <- c("Surgery","Plaster_cast","Hormone_use","Pregnant","Location_
    VT","Gender","time","status" )


K$surgery <- ifelse(dat.K$oper3mnd==1,"yes","no")
K$plaster_cast <- ifelse(K$gips3mnd==1,"yes","no")
K$oper3mnd <- as.factor(K$oper3mnd)
K$gips3mnd <- as.factor(K$gips3mnd)


dat.K[,c(1:6)] <- lapply(dat.K[,c(1:6)] , factor)
dat.K$Gender <- factor(dat.K$Gender,labels=c('female','male'))


X<- c("Surgery","Plaster_cast","Hormone_use","Pregnant","Location_VT","Gender")
Xs <- paste(X, collapse=" + ")
form <- as.formula(paste("Surv(time, status) ~",
paste(Xs, collapse="+")))

cx.obj <- coxph(form , data = dat.K )
regplot(cx.obj, dummies = FALSE, observation= dat.K[10,],points = TRUE, failtime =
    c(1, 5), prfail = TRUE,
showP=FALSE)
```

8.9 Coefficients plots

```
library("arm")


varNames <- c("age","surgery","plaster-cast","hospital","pregnant","hormone","
    Cardio-dise","cerebro-dise",
"BMI","Location VT","gender","provokedov","dischr_J","PTS1","PTS2",
"blood-type","Leiden V","TypeVT2","TypeVT3" ,"factor IX","factor II","factor X","
    factor XI","APC-ratio",
"fibrinogen","antitrombine","protein C","factor VII","factor V","TFPI","Hemoglobin
    ",
"ETP-low-TF","factor VIII","VWF","protein S","D-dimer","white-blood","monocyte","
    red-cell","HsCRP")
varNames<- varNames[order(varNames)]


#-----------------------#
# backward selection  #
#-----------------------#


# Model A

B.a <- as.matrix(A.bw$backward_model$coefficients)

# change names:
rownames(B.a) <- c("surgery","hormone","gender","TypeVT2","TypeVT3", "fibrinogen",
    "protein C","factor X",
"factor XI","APC-ratio","factor VIII","D-dimer")
x <- rownames(B.a)


ad.a <- varNames[!varNames %in% x]
mat.a <- matrix(0,length(ad.a),1)
rownames(mat.a) <- ad.a
mat.a <- rbind(B.a,mat.a)
mat.A <- as.matrix(mat.a[order(rownames(mat.a)),])
A.mat <- as.vector(mat.A)


# Model C

rm(list = c("x","s"))


C.B <- as.matrix(C.bw$backward_model$coefficients)

# change names:
rownames(C.B)<- c("surgery","plaster-cast","pregnant", "hormone","Location VT","
    gender","TypeVT2","TypeVT3","Leiden V")
x <- rownames(C.B)

adc <- varNames[!varNames %in% x]
matc <- matrix(0,length(adc),1)
rownames(matc) <- adc
matc <- rbind(C.B,matc)
matC <- as.matrix(matc[order(rownames(matc)),])
v.matC <- as.vector(matC)
```

```
#------------------------#
# Percentile and lasso #
#------------------------#

# Model A

l.A <- as.matrix(app_mod$model)
# change names:
rownames(l.A)<- c("surgery","pregnant","hormone","gender","TypeVT2","TypeVT3",
"PTS2","Leiden V","factor VIII","VWF","D-dimer" )
s <- rownames(l.A)

ads <- varNames[!varNames %in% s]
mats <- matrix(0,length(ads),1)
rownames(mats) <- ads
mats <- rbind(l.A,mats)
mat1 <- as.matrix(mats[order(rownames(mats)),])
A.L <- as.vector(mat1)

# Plots for model A:

# Graph the regression coefficients
coefplot(A.mat, sd = rep(0, 40),CI=0, xlim=c(-1.5,1.5), pch=8, cex.pts = 1,
main = "Cox coefficient estimates ",varnames = varNames, col="red")


coefplot(A.L, sd = rep(0, 40), pch=16, add = TRUE, col.pts = "blue")
legend("bottomright", c("Backward", "LASSO" ), col = c("red", "blue"), pch = c(8,
    16), bty = "o")


# Model C

l.C <- as.matrix(app_mod.c$model)

# change names:
rownames(l.C) <- c("surgery","plaster-cast", "hospital","pregnant","hormone","
    Cardio-dise","Location VT", "gender",
"TypeVT2", "TypeVT3", "PTS1","PTS2", "blood-type","Leiden V" )
sC <- rownames(l.C)
adC <- varNames[!varNames %in% sC]
matt <- matrix(0,length(adC),1)
rownames(matt) <- adC
matt <- rbind(l.C,matt)
mat1c <- as.matrix(matt[order(rownames(matt)),])
v.LC <- as.vector(mat1c)

#----------------#
# closed testing #
#----------------#

# Model C6:

Xs <- paste(a[[8]], collapse=" + ")
form <- as.formula(paste("Surv(time, status) ~",
paste(Xs, collapse="+")))

cox1 <- coxph(form, data =  dat.c, x=TRUE)


modl <- cox1$coefficients
ct.C <- as.matrix(modl)
```

```
114
# change names:
116 rownames(ct.C) <- c("surgery","plaster-cast", "pregnant","hormone", "Location VT",
      "gender")
ct <- rownames(ct.C)
118
ad.ct <- varNames[!varNames %in% ct]
120 mat.ct <- matrix(0,length(ad.ct),1)
rownames(mat.ct) <- ad.ct
122 mat.ct <- rbind(ct.C,mat.ct)
mat.ct.c <- as.matrix(mat.ct[order(rownames(mat.ct)),])
124 ct.vC <- as.vector(mat.ct.c)

126 # plots for model C6

128 # Graph the regression coefficients
coefplot(v.matC, sd = rep(0, 40),CI=0, xlim=c(-1.5,1.5), pch=8, cex.pts = 1, cex.
    var=0.8,
130 main = "Cox coefficient estimates ",varnames = varNames, col="red")

132 coefplot(v.LC, sd = rep(0, 40), pch=16, add = TRUE, col.pts = "blue")
legend("bottomright", c("Backward", "LASSO" ), col = c("red", "blue"), pch = c(8,
    16), bty = "o")
134
coefplot(ct.vC, sd = rep(0, 40), pch=10, add = TRUE, col.pts = "green")
136 legend("bottomright", c("Backward", "LASSO","Closed Testing" ), col = c("red", "
    blue","green"), pch = c(8, 16,10),bty = "o")


138
#----------------#
140 # One SE lasso  #
#----------------#
142
l.BC <- as.matrix(app_mod.1se$model)
144
# change names:
146 rownames(l.BC) <- c("surgery", "hormone", "Location VT", "gender","TypeVT=3","
    Leiden V" )
sC <- rownames(l.BC)
148 adC <- varNames[!varNames %in% sC]
matt <- matrix(0,length(adC),1)
150 rownames(matt) <- adC
matt <- rbind(l.BC,matt)
152 mat1c <- as.matrix(matt[order(rownames(matt)),])
v.LC <- as.vector(mat1c)
154
coefplot(v.LC, sd = rep(0, 40), pch=16, add = TRUE, col.pts = "blue")
156 legend("bottomright", c("Backward", "LASSO" ), col = c("red", "blue"), pch = c(8,
    16), bty = "o")


158
cbind(mat1c,matC)
160 cbind(mat1c,mat)
```

8.10 Sensitivity of lasso to data changes

```
1
#-------------------------------------#
3 # The previous data preparation  #
#-------------------------------------#
5
# This was the original data preparation process.
7 # Lasso resulted in different models, whereas percentile lasso
# was very stable. This shows how lasso was very sensitive
9 # to small changes in data R coding.

11
library(haven)
13 dat <- read_dta("....Data source....")

15 # remove the next variables variables: because they were replaced by new variables
    :
# dischr_J -> dischr_S
17 # provokedov -> provokedov_S
# locatie1var -> locatie1var_S
19 # PTS1var -> PTS_J

21 dat <- subset(dat, select = -c(dischr_J,provokedov,locatie1var,PTS1var))

23 # Some data variables to be changed;

25 dat$DVT <- ifelse(dat$TypeVT==1 | dat$TypeVT==3,1,0)
dat$PE <- ifelse(dat$TypeVT==2 | dat$TypeVT==3,1,0)
27 dat$locatie1varII <- ifelse(dat$locatie1var_S==2,1,0)
dat$PTS_J <-  as.factor(dat$PTS_J)
29
# note that TypeVT has 3 categorie i.e: DVT=1,PE=2 and DVT+PE=3;
31 # let us use this only for statistical modelling,
# and for the data descrition we use the exact information.
33

35 # remove  TypeVT and locatie1var. due to redundancy.
# we remove SNPscore, because it has so many missing variables: 70%
37 dat <- subset(dat , select = -c(TypeVT,locatie1var_S,SNPscore))

39 # note that some variables are already coded as 0/1, thus
# there's no need to make them as factor in R.
41

43 # Peredictors:

45 clinical.f <- c("lft", "oper3mnd","gips3mnd","bedzk3mnd","zwan3mnd", "hormoon",
"discardio","discerebro","bmi","locatie1varII","sex_J","provokedov_S",
47 "DVT","dischr_S","PTS_J","PE")

49 genetics.f <- c("blgroup","fvleiden_J")

51 laboratory.f <- c("fib","AT3","PCC","F7C","f9aggem" ,"f2c","F10C","F11","lognAPCsr
    ",
"factorV","TFPI","Hgb","nETPLT","logf8aggem" ,"logvwf","logpsfree",
53 "logddimer","logWBC","logmoperc","logrdw","logHsCRP")

55

57 A.var.org <- c(clinical.f,genetics.f,laboratory.f)
```

```
C.var.org <- c(clinical.f,genetics.f)


# Survival analysis time and indecator

# recidieftime -> time to event
# recidiefunprov -> event

dat$time <- dat$recidieftime
dat$status <- dat$recidiefunprov

# remove the redundant variables
dat <- subset(dat , select = -c(recidieftime,recidiefunprov))



#-----------------------------------#
# lasso and percentile lasso II  #
#-----------------------------------#

# Model A:

# Remove missing values;
dat.a <- na.omit(dat)#
X <- A.var.org
df <- dat.a

# objects for percentile lasso;
Xs <- paste(X, collapse=" + ")
form <- as.formula(paste("Surv(time, status) ~",
paste(Xs, collapse="+")))

x <- model.matrix(form, df)
y <- Surv(df$time,df$status)

# Which percentile to choose:

# we will repeat (345times) per.lasso to check for its stability; thus to choose
    one model.
# note that percentile lasso will often provide you two models,
# models arranging between 11 and 14 variables. choose one model.
# we note that there is almost no difference in model performence (apparent_c-indx
    ),
# therfore we will choose model with 11 variables C=0.703485,and C=0.704815 for 14
     variables.--> see the next code;

A <- list()
for(i in 1:345){
A[[length(A)+1]] <- Per.Lasso(x,y,K=10,per=c(0.75,0.8,0.85,0.9,0.95),cv.rep = 10)$
    res.sum

}

# lambdas: 0.013985781 -->  11 variables
# lambdas: 0.011611241 --> 14 variables

# Model A: results
per.lam <- 0.01398578  # optimal percentile lambda;
app_mod <- lasso_app(A.var.org,dat.a,per.lam)
C_indx.A <- lasso.boot(200,A.var.org,dat.a, app_mod$C_ind_apparent ,per.lam)

```

```
A <- list()
for(i in 1:345){
A[[length(A)+1]] <- Per.Lasso(x,y,K=10,per=c(0.75,0.8,0.85,0.9,0.95),cv.rep = 10)$
    res.sum

}

h <- matrix(0,length(A),4)
for(i in 1:length(A)){
k <- A[[i]]
h[i,1] <- k[1,3] # variables by lasso
h[i,2] <- k[7,3] # variables by percentile
h[i,3] <- round(k[1,1],4) # extract the lambdas_min
h[i,4] <- round(k[7,1],4) # extract the lambdas at cvm
colnames(h) <- c("stand","per","lam.min","lam.cvm")

}

H <- data.frame(h)

# Lambdas
lam.min <- table(H$lam.min)
stand.l.min <- data.frame(lam.min)

#--------#
# plot   #
#--------#

library(ggplot2)

ggplot(stand.l.min, aes(x = Var1, y = Freq)  ) +
geom_bar(stat = "identity",fill = "yellow") +
geom_text(aes(label = Freq),vjust = -0.3,color ="red") + ###
labs( y = "Frequencies",x=expression(hat(lambda)),
title = "All possible tuning parameter by ordinary lasso")+
theme_bw()+
theme(plot.title = element_text(hjust = 0.5,size = rel(1.5)),
axis.text = element_text(size = rel(1.1),color = "black"),
axis.title.y = element_text(size = rel(1.3) ),
axis.title.x = element_text(size = rel(1.3) ),
axis.text.x = element_text(angle = 45, hjust = 1))


# non-zero coeffiecients

V <- table(H$stand)
V.df <- data.frame(V)

ggplot(V.df , aes(x = Var1, y = Freq), hjust = -0.2) +
geom_bar(stat = "identity",fill = "skyblue",width = 0.7) +
geom_text(aes(label = Freq), hjust = -0.2,color ="red") + ###
coord_flip()+
labs(x = "Non-zero coeffiecients", y = "Frequencies",
title = "All possible selected model by ordinary lasso")+
theme_bw()+
theme(plot.title = element_text(hjust = 0.5,size = rel(1.5)),
axis.text = element_text(size = rel(1.1),color = "black"),
axis.title.y = element_text(size = rel(1.3) ),
axis.title.x = element_text(size = rel(1.3) ))
```

```
177
#-------------------------#
179 # plot percentile lasso  #
#-------------------------#
181
# Lambdas:
183 lam.p <- table(H$lam.cvm)
lam.per <- data.frame(lam.p)
185
#plots:
187
ggplot(lam.per, aes(x = Var1, y = Freq)  ) +
189 geom_bar(stat = "identity",fill = "green",width = 0.6) +
geom_text(aes(label = Freq),vjust = -0.3,color ="red") + ###
191 labs( y = "Frequencies",x=expression(hat(lambda)),
title = "All possible tuning parameter by percentile-lasso")+
193 theme_bw()+
theme(plot.title = element_text(hjust = 0.5,size = rel(1.5)),
195 axis.text = element_text(size = rel(1.1),color = "black"),
axis.title.y = element_text(size = rel(1.3) ),
197 axis.title.x = element_text(size = rel(1.3) ),
axis.text.x = element_text(angle = 45, hjust = 1))
199

201 # non-zero coeffiecients plots
P <- table(H$per)
203 per.l <- data.frame(P)

205

ggplot(per.l, aes(x = Var1, y = Freq) ) +
207 geom_bar(stat = "identity",fill = "red",width = 0.7) +
geom_text(aes(label = Freq), vjust = -0.2,color ="blue") + ###
209 labs(x = "Non-zero coeffiecients", y = "Frequencies",
title = "All possible selected model by percentile-lasso")+
211 theme_bw()+
theme(plot.title = element_text(hjust = 0.5,size = rel(1.5)),
213 axis.text = element_text(size = rel(1.1),color = "black"),
axis.title.y = element_text(size = rel(1.3) ),
215 axis.title.x = element_text(size = rel(1.3) ))
```