Variable selection techniques for the Cox proportional hazards model: A comparative study

Simon Petersson and Klas Sehlstedt

University of Gothenburg

School of Bussiness,

Economics and Law

2018-02-21

Supervisor: Alexander Herbertsson

Abstract

In statistics different models are used to emulate real world processes. Variable selection refers to reduction of the number of parameters in the models in order to increase interpretability and model effectiveness. This thesis investigates performance of different variable selection methods for Cox proportional hazards models such as; all subset selection, backward elimination, best subset selections and least absolute shrinkage and selection operator. Data is simulated for 5 different models with coefficients reflecting large, moderate and small effects, with different sized data sets and simulations. The result are also partly compared to earlier reported results from Tibshirani (1997), Fan & Li (2002), Zhang & Lu (2007). Our findings indicate that the best subset selection methods is faster than all subset selection but slower than least absolute shrinkage and selection operator. On the other hand best subset selection provide more accurate results than least absolute shrinkage and selection operator. We have been unable to verify all of the results reported by Tibshirani (1997), Fan & Li (2002) and Zhang & Lu (2007) probably because difference in the least absolute shrinkage and selection operator's tuning parameter.

Keywords: All subset selection, Backward elimination, Best subset selection, BeSS, Cox proportional hazards model, least absolute shrinkage and selection operator, LASSO.

Acknowledgment: We appreciate valuable input from Alexander Herbertsson, Erik Thordenberg and Joel Nilsson.

Contents

1 Introduction

1.1 Background

The development of information technology have made the competitive environment tougher for almost every type of business. Whether the competition is for customers or retaining employees, the benefits of data analysis is clear. Corporations can nowadays store massive amounts of information and with the improvements in computational power that have been made in the last 40 years, the possibility to deeply analyze these datasets thoroughly have emerged.

One application of data analysis is survival analysis which is a branch of statistics concerned with expected time until one or more events occur. This type of procedure is frequently used in the field of medical research or in engineering, where the events can be the death of a biological organism or the failure of a system. Since the main point of interest here is the time until an event, the applications exceed these two fields by far, with the possibility to define an event in multiple ways. Possible applications include investigating why customers or employees choose to end their relationship with the company, which can provide corporations with great insights as to where resources should be allocated. More effective resource allocation in turn, makes the company more profitable and results in competitive advantage.

1.2 Problem analysis and purpose

Today, corporations around the world have massive datasets waiting to be analyzed for achieving customer insight and provide competitive advantages, but the massive datasets are problematic to analyze. The problems arise when the amount of possible variables are too large to create a comprehensible and interpretable model. A model with too many variables makes it difficult to understand what information that is most important and where the resources should be spent, making tools for variable selection needed.

Several studies have been conducted comparing different variable selection techniques (Tibshirani 1997, Fan & Li 2002, Zhang & Lu 2007), which will be further discussed in the theory section. We are interested in analyzing how well the different variable selection method performs, when used together with the Cox Proportional Hazards model (CoxPH). We will compare our findings with previous comparative studies to see whether the results can be replicated. Assessing how the different techniques behave will be done by creating datasets from known true

models, with some coefficients set to zero. In doing so, the efficiency of the model can be assessed by looking at the average number of correct coefficients set to zero but also how many coefficients incorrectly set to zero. Theory, variable selection, implementation and simulation of this will be further discussed in the Sections 2 to 5 and ending with conclusions in Section 6.

Thus, the purpose of this thesis is to evaluate the performance of different variable selection techniques, when used together with the Cox proportional hazards model. Furthermore, we will benchmark our findings against previous comparative studies which have used a similar methodology.

2 Survival analysis and the Cox PH model

This section will start with discussing the general concept of survival analysis in Subsection 2.1. Following the general concept of survival analysis, the Cox proportional hazards model is discussed in Subsection 2.2 and both the base line hazard rate as well as the partial likelihood function is discussed in Subsection 2.3.

2.1 Survival Analysis

Survival analysis is a branch of statistics concerned with analyzing the time until an event. The time until an event can be anything from years to days or even the age of an individual when the event occurs (Kleinbaum & Klein 2012, p. 4). Common events include death, failure of a machine or a customer terminating its relationship with a company, in other words, basically anything of interest which can be defined as either fulfilled or not (Kleinbaum & Klein 2012, p. 4). Survival analysis have many applications and it can be used for describing the survival times of members in a group, compare survival times between groups and describe the effect different variables have on survival time (Kleinbaum & Klein 2012, p. 4). Some examples of its applications can be found in (Nie, Rowe, Zhang, Tian & Shi 2011) who conducts churn analysis, attrition analysis (Van Den Poel & Larivire 2004) and many more.

Introducing some mathematical notations, let T denote the random variable for an observation's survival time and $T \in [0, \infty]$. Next, let t denote a specific value of the random variable T, which is used for evaluating whether an observation survives or not for t numbers of time units. Furthermore, let $P(.)$ be the underlying probability measure where $P(T > t)$ denotes the probability that the survival time of an individual exceed the value of t. Two quantitative measurements which are important to understand in survival analysis is the survival function, denoted $S(t)$, and the hazard function, denoted $h(t)$.

The survival function is defined as

$$S(t) = P(T > t) = 1 - F(t), \tag{1}$$

where $F(t)$ is the cumulative distribution function (Rice 2007, p. 380). The survivor function is used to evaluate the probability that an observation object survives longer than a time specified by t. In other words; it returns the probability that the random variable T exceeds the specified value t. This function is essential and makes it possible to provide summary information from

survival data. With t ranging from zero up to infinity, it could theoretically be graphed as a smooth curve starting at 1 and approaching 0 as $t \to \infty$. The properties $S(0) = 1$ and $\lim_{t\to\infty} S(t) = 0$ is intuitive because when $t = 0$ nobody has experienced the event yet and as $t \to \infty$ every object must eventually experience the event(Kleinbaum & Klein 2012, pp. 44-45). The function is closely related to the hazard function defined as

$$h(t) = \lim_{\Delta t \to 0} \frac{P(t \leq T < t + \Delta t | T \geq t)}{\Delta t}. \tag{2}$$

The hazard function is used to quantify the instantaneous risk that the event will occur at time t, given that the observation have survived up until time t. Note that the hazard function $h(t)$, in contrast to the survival function, focuses on failure instead of success and is defined as (Kleinbaum & Klein 2012, p. 45):

The hazard function which is often the focus in survival analysis, since it is usually more insightful, where one example could be investigating human mortality rates. In such cases the instantaneous risk of dying, i.e. $h(t)$ is known to be elevated in the first years of life, then being relatively low and flat for the majority of a persons life and rapidly increase once a person gets older. Thus, the hazard function can have an arbitrary shape as long as it stays non-negative. It is possible to show that

$$S(t) = P(T > t) = \exp\big(-\int_0^\infty h(s)ds\big), \tag{3}$$

see e.g. Wackerly & Scheaffer (2008, p. 219).

One frequent analytical problem that have to be considered when conducting survival analysis is a term called censoring (Kleinbaum & Klein 2012, pp.5-9). Censoring refers to when the value of an observation or measurement is only partially known and the exact survival time can not be determined. There are general forms that this phenomenon takes:

1. The observation object does not experience the event within the time frame of the study.
2. An observation is lost in the follow-up of the study.
3. An observation object withdraws from the study. An example could be because of death, if death is not the event of interest or a death caused by actions not included in the study.
4. An observation object enters in the middle of the study, e.g. a new customer.

Censoring is usually discussed in three different types of censoring. Two of these types of censoring terms depend on which "side" of the observation period the survival time is censored.

Let t_s be the start of a study, t_f be the end or follow up time and t_i be the survival time of individual i where $i = 1, ..., N$ and N is the number of individuals. Right-censored data are data which can have experienced any of the forms 1-3 listed above and the logic behind it is, that the data becomes incomplete at the right side of the observation period, that is $t_i > t_f$. For data of this type, the complete survival interval is not known since the data has been cut off at the right side of the survival time interval. (Kleinbaum & Klein 2012, pp.5-9).

Left-censoring occurs when the observations real survival time is less than or equal to the observed survival time, $t_i \leq t_s$. An example of this could be a study concerning whether someone is affected by a disease. If a test is conducted and returns a positive result, when the patient was exposed to the disease is not known and can only be before or while being tested. Thus, the observation is truncated from the left(Kleinbaum & Klein 2012, pp.5-9).

The last possible type of censoring is interval-censored. This situation occurs when there are two observation periods and in the first observation, the event was no experienced, but was observed in the second observation period. Following the previous notations, $t_s < t_i \leq t_f$ In this situation we know that the event occurred after the first period, but before the second observation period. Therefore, we do not know the true survival time but only in what interval the event occurred(Kleinbaum & Klein 2012, pp.5-9).

2.2 The Cox proportional hazards model

The Cox proportional hazards model is one of the models used in survival analysis. The model was originally developed by Cox (1972) and is a model for dynamic survival analysis. Let $h_i(t|x_i)$ be the Cox proportional hazards model with the general expression

$$h_i(t|x_i) = h_0(t) \exp(\boldsymbol{x}_i\boldsymbol{\beta}). \tag{4}$$

The first factor, $h_0(t)$ represents the baseline hazard function which is the same for the every object in the entire sample. The second factor, $exp(\boldsymbol{x}_i\boldsymbol{\beta})$, is what makes the hazard function individual for each observation in the sample, since each observations most likely will not have the same observed behavior. The observed behavior is the $\boldsymbol{x}_i$, which corresponds to a row vector with the values for each variable that are included in the model for observation object i. $\boldsymbol{\beta}$ is a column vector with regression coefficients, one for each variable in $\boldsymbol{x}_i$.

It is possible to include time varying covariates as well as discrete and continuous measurements in order to get a representation of the hazard function, given that we observe a certain behavior.

The Cox proportional hazards model is dominant when it comes to using dynamic survival models (Stare et al. 2001, Allison et al. 2010). Some explanations to its use might be that studies have proven it to be robust and requiring few assumptions (Kumar & Westberg 1996).

2.3 Estimate the base line and state partial likelihood function

One of the features with the Cox proportional hazards model is that the base line hazard function $h_0(t)$ is not needed to estimate the $\boldsymbol{\beta}$ parameters in the model (4). The reason for that is that it does not show up in the partial log likelihood expression. Let $\tilde{T}_i$ denote the minimum of the censoring time C_i and the survival time T_i. Furthermore, let $\boldsymbol{x}_i = \{\boldsymbol{x}_{i,1}, ..., \boldsymbol{x}_{i,p}\}$ be the 1-by-p vector of covariates associated with t_i for the ith object observation, where t_i is the observed value of $\tilde{T}_i$. Denote $\boldsymbol{\beta} = \{\boldsymbol{\beta}_i, ..., \boldsymbol{\beta}_p\}^\top$ the p-by-1 vector of coefficients and let $\delta_i = 1$ be the indicator variable, which is one if t_i is an event or zero if it is a censored time. Assume that data have no ties. Ties are observations with identical survival times. If that is the case, the partial likelihood (5) must be changed accordingly. Further let $i = \{1, 2, ..., N\}$ where N is the total number of observations and $\mathcal{R}(t) = \{i : t_i \geq t\}$ is the set of objects at risk at time t. Then from (Chen et al. 2009, Cox 1975, 1972) the partial likelihood of the model (4) can be expressed as:

$$L_p(\boldsymbol{\beta} \,|\, D_{obs}) = \prod_{i=1}^{N} \left[\frac{e^{\boldsymbol{x}_i \boldsymbol{\beta}}}{\sum_{j \in \mathcal{R}(t_i)} e^{\boldsymbol{x}_j \boldsymbol{\beta}}} \right]^{\delta_i}, \tag{5}$$

where $D_{obs} = \{(t_i, \delta_i, \boldsymbol{x}_i) : i = 1, 2, ..., N\}$ is the observed univariate right censored survival data (Chen et al. 2009). From (5) one can see that the baseline hazard function is absent. This is due to the fact that it exists both in the nominator and denominator and has been canceled out as $h_0(t) > 0$ for all $t > 0$. Taking the natural logarithm on both sides of (5) and denote $l_p(\boldsymbol{\beta}) = \ln(L_p(\boldsymbol{\beta}))$ gives the logarithm partial likelihood:

$$l_p(\boldsymbol{\beta} \,|\, D_{obs}) = \sum_{i=1}^{N} \delta_i \left[\boldsymbol{x}_i \boldsymbol{\beta} - \ln(\sum_{j \in \mathcal{R}(t_i)} e^{\boldsymbol{x}_j \boldsymbol{\beta}}) \right]. \tag{6}$$

For the completely observed data D_{obs}, the maximum partial likelihood estimate (MPLE) is defined as $\hat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}} L_p(\boldsymbol{\beta} \,|\, D_{obs})$ (Chen et al. 2009). Since $L_p(\boldsymbol{\beta} \,|\, D_{obs})$ and $l_p(\boldsymbol{\beta} \,|\, D_{obs})$ has maximum for the same $\boldsymbol{\beta}$, $\hat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}} l_p(\boldsymbol{\beta} \,|\, D_{obs})$ can be solved instead, because

$$\frac{d \ln(f(x))}{dx} = \frac{\frac{df(x)}{dx}}{f(x)}.$$

Solving $\hat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}} l_p(\boldsymbol{\beta} \mid D_{obs})$ can be done in the ordinary way, taking the partial derivatives of (6) with respect to $\boldsymbol{\beta}$, setting the derivatives' to zero, solving for $\boldsymbol{\beta}$ and check for maximum which will give $\hat{\boldsymbol{\beta}}$. One could of course solve for minimum by changing the sign on the right side in (6). However many of the statistical standard softwares have support for solving the MPLE in the first way above, i.e. in R one can use the package survival. R also have support for LASSO with at least two packages, i.e. glmnet or lars. In this thesis, glmnet and cv.glmnet is used.

A reasonable question to ask is does the MPLE always exist? The answer is no. Chen et al. (2009) gives a example when the MPLE does and does not exist. This is not an issue in this thesis with simulated data.

3 Variable Selection and previous comparative studies

In this section variable selections and findings from earlier articles are discussed. Subsections 3.1 to 3.6 outlines the general theory of variable selection and discuss common techniques used when trying to simplify regression models. After explaining the different techniques which is used in this thesis, Subsection 3.7 will present results from some of the previous comparative studies evaluating different selection techniques.

3.1 Variable selection

In essence, variable selection is concerned with removing redundant or irrelevant variables from the model which describes the data, by reducing variance on the expense of bias. The removal is controlled by a criterion, such as RSS, AIC or BIC described later. The model that provides the best criterion, e.g. minimum, is the selected one. Starting with an arbitrary linear model, let $i = 1, ..., N$ be the number of observations and $j = 1, ..., p$ be the number of predictors. The model in matrix notation is given by

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \epsilon$$

where $\boldsymbol{y}$ is a $N \times 1$ vector of outcome values, $\boldsymbol{X}$ is the $N \times p$ design matrix, $\boldsymbol{\beta}$ is a $p \times 1$ vector of unknown coefficients and ϵ is a $N \times 1$ vector of unexplained error. Estimating the unknown coefficients can be achieved by using the standard approach, ordinary least squares(OLS). In mathematical notations, assume $\boldsymbol{b}$ is a candidate for the parameter vector, let $RSS(\boldsymbol{b})$ be the residual sum of squares(RSS) for the model with estimated coefficients $\boldsymbol{b}$, defined by

$$RSS(\boldsymbol{b}) = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{b})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{b}).$$

The $\boldsymbol{b}$ vector which minimizes the $RSS(\boldsymbol{b})$ is called the OLS estimator of $\boldsymbol{\beta}$, which most commonly is denoted

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{b}} RSS(\boldsymbol{b}).$$

OLS will find a solution containing every predictor, without concern as to how much the predictor actually is beneficial when trying to predict the outcome. Here lies the first problem statisticians face in building a model. When trying to generalize a model for predictive purposes, the OLS procedure might read to much into the dataset used for building the model, meaning that the ϵ influences the $\hat{\boldsymbol{\beta}}$ in such a way that the model performs worse on future

data. The general term for this is overfitting the model to the data and is what statisticians aim to reduce, also known as reduction of variance (James et al. 2014). Continuing with the notations above and let $\hat{\boldsymbol{y}}$ be the vector containing predicted values of $\boldsymbol{y}$ by the estimated model. Furthermore, introducing the notation $pMSE = E[(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})^2]$ which refers to the expected prediction mean square error and consist of the three properties

$$pMSE = Bias(\hat{\boldsymbol{y}})^2 + Var(\hat{\boldsymbol{y}}) + Var(\boldsymbol{\epsilon}). \tag{7}$$

The $pMSE$ refers to the average mean square error if repeatedly estimating $\boldsymbol{\beta}$ using a large number of datasets (James et al. 2014). Here, the $Bias(\hat{\boldsymbol{y}})$ is a measure of how much simplification assumptions contribute to making the error larger in predictive models(James et al. 2014). An example could be trying to estimate a non-linear relationship with a linear model, which will induce error in the estimates. The $Var(\hat{\boldsymbol{y}})$ is the variance of the model, in other words, how much the estimated $\boldsymbol{\beta}$ changes using a different dataset (James et al. 2014). Lastly, the $Var(\boldsymbol{\epsilon})$ is the variance of the irreducible error. Finding a method which lowers the bias and variance is important to make the model more stable, leading to better predictions.

The second problem with too many parameters is that the interpretability suffers as a result(James et al. 2014). However, there are techniques in place to handle the problem of overfitting to data and increasing interpretability. In this thesis we discuss two common practices which is divided into regularized regression, further discussed in Section 3.2, and selecting the best model from a large set of candidate models, based on a criterion. There are several criterion which can be used for determining the best model like R-squared, prediction error or various information criterion. The focus in this thesis will be on the latter and more specifically the Akaike information criterion (AIC) as well as the Bayesian information criterion (BIC).

The AIC was developed by Hirotugu Akaike and proposed in Akaike (1974). It is likelihood based and founded on information theory. The AIC criterion not provide any information about how well the model fits the data, but rather how well the model behaves compared to the other models. Xu et al. (2009) implements the AIC procedure in the Cox PH model with the formula

$$\text{AIC} = -2pl(y|\hat{\boldsymbol{\beta}}(y)) + 2p, \tag{8}$$

where $pl(.)$ is the log partial likelihood and p is the dimension of β. This formula will be used when we implement our variable selection techniques that are based on selecting a model from a wide range candidate models. This will be explained further in the subsections for All subset selection, Stepwise regression and Best subset selection.

Another selection criterion is the Bayesian information criterion (BIC). As the name implies, it is Bayesian based, e.g. based on Bayes' theorem and was proposed by (Schwarz 1978) as a criterion for model selection. Just like AIC it makes use of the likelihood function and differs only in how the penalty for including more variables behave. The difference makes BIC being harsher on incorporating more coefficients, thus tending to choose smaller models. Since this thesis makes use of the Cox proportional hazards model, the formula which will be implemented was outlined by Volinsky & Raftery (2000) and given by

$$\text{BIC} = -2pl(y|\hat{\boldsymbol{\beta}}(y)) + p \cdot \ln(d), \tag{9}$$

where $pl(.)$ is the log partial likelihood, p is the number of coefficients in $\boldsymbol{\beta}$ and d is the number of events that have occurred. This formula will be used when the BIC score is calculated for finding the model that most accurately describes the data in all subset selection and backwards elimination.

3.2 The LASSO

One possible way of solving the variable selection problem is to let the LASSO do the selection for you. The method was developed by Tibshirani (1996), which was a proposal for a new way of estimating the parameters in a linear regression. The main goal of the technique, as laid out in the original article, is to increase prediction accuracy and boost the interpretability of the model which are two common concerns for data analysts. Fitting several variables with standard ordinary least squares (OLS) technique tend to produce several small coefficients and the estimates tend to have low bias but high variance. Tibshirani (1996) argues that this can often be remedied by setting some coefficients to zero, which sacrifices some bias in order to reduce the variance of the predicted values, that can in turn increase the overall prediction accuracy of the model. The second problem with standard OLS is that the interpretability decreases as the amount of variables increases. Thus, often investigators are interested in a smaller set of variables which show stronger effects on the outcome.

The LASSO minimizes the residual sum of squares with the constraint of the sum of the absolute value of the coefficients being less than a constant, C, that is

$$(\hat{\alpha}, \hat{\beta}) = \arg\min\{\sum_{i=1}^{N}(y_i - \alpha - \sum_j \beta_j x_{ij})^2\} \quad \text{under the constraint } \sum_j |\beta_j| \leq C. \tag{10}$$

Here, the parameter C controls the shrinkage and the way of finding out what level C should be set to will be discussed below. If we denote the full least squares estimates $\hat{\beta}_j^0$ and let $C_0 = \sum |\hat{\beta}_j|$, then for any value of $C < C_0$ the LASSO solution will shrink towards 0, with some coefficients being exactly 0. The idea behind the LASSO constraint comes from the non-negative garotte proposal by Breiman (1993, 1995).

Tibshirani (1996) argues that the drawback of non-negative garrotte is that it depends on the magnitude of the OLS estimates and the sign of the coefficients. This means that when variables are highly correlated the OLS estimates behave poorly and the garrotte method might suffer. Thus, the LASSO is proposed as a better alternative since it avoids the explicit use of OLS estimated coefficients. The corresponding equation to solve when using LASSO with the Cox PH model then becomes (Hastie et al. (2015, p. 43)):

$$\arg\min_{\beta} \left\{ -\sum_{i=1}^{N} \delta_i \left[\boldsymbol{x}_i^\top \boldsymbol{\beta} - \ln\Big(\sum_{j \in \mathcal{R}(t_i)} e^{\boldsymbol{x}_j^\top \boldsymbol{\beta}} \Big) \right] + \lambda ||\boldsymbol{\beta}||_1 \right\}. \tag{11}$$

Equation (11) is the Lagrangian optimization problem for estimating the LASSO $\boldsymbol{\beta}$ coefficients, where $||\boldsymbol{\beta}||_1 = \sum_j |\beta_j|$ and is also known as the L_1-norm. The δ_i in (11) is the censoring indicator, which is 1 if the observed time is an event and 0 if the observed time is censored. The λ in (11) is the the LASSO tuning parameter, which is determined by cross-validation, and will be discussed in the Subsection 3.3.

The LASSO method does not however stand without criticism. The critique comes both from advocates of other variable selection techniques as well as researchers wanting to expand the technique. Fan & Li (2002) argues that the LASSO technique potentially can produce coefficients with substantial bias, if λ is chosen too big. Instead, Fan & Li (2002) proposes another method called Smoothly clipped absolute deviation (SCAD) penalty which they state provide better theoretical properties compared to that of the LASSO. However, as pointed out by Zhang & Lu (2007), the non-convex penalty form of SCAD makes it difficult to use in practice and could also suffer from numerical instability.

3.3 Cross-validation

In this section we discuss the general idea with cross-validation, then cross-validation in previous studies and ending with the cross-validation used in this thesis. There are several different and

commonly used types of cross-validation (CV). In this study the focus is on K-fold CV and the reason to use it is to determine the LASSO tuning parameter λ (Hastie et al. 2015, p. 13).

The basic idea is to divide the data set into K equally sized random data subsets, denoted folds, without replacement, using $K-1$ subsets for training and the remaining subset for validation, while repeating this K times. This means that all K subsets are used for validation once. By selecting K as an integer such as $K = \frac{N}{N_s}$, where N_s is the number of observations in the equally sized subsets and N the total number of observations in the data set, e.g. K is the number of folds to use in the cross-validation. Thus all observations are used both for training and validation. In each iteration over the K folds an arbitrarily statistic is computed, such as mean, standard error or partial likelihood deviance as a function of an interval of some variable.

(Tibshirani 1997) use generalized cross-validation described by Wahba (1985), Craven & Wahba (1979), given by

$$GCV(s) = \frac{1}{N} \frac{-l_s}{N\left[1 - \frac{p(s)}{N}\right]^2}, \tag{12}$$

where l_s is the partial log likelihood as a function of the variable s and $p(s)$ is a function that relates s to the number of effective, e.g. non-zero, coefficients basically via an approximation of $\boldsymbol{\beta}$ and the LASSO tuning parameter. Fan & Li (2002) and Zhang & Lu (2007) uses similar definitions as in Equation (12) for their generalized cross-validations.

For Cox proportional hazards model applied to the LASSO method the statistics used in this thesis is an approximation of the partial likelihood deviance as a function of the logarithm of an interval of the LASSO tuning parameter λ, e.g $\lambda \in [0, 1]$ in 100 steps. For each iteration over the K folds and the specific λ value the statistic partial likelihood deviance is computed and stored. When all iterations over the K folds and the specific λ value is completed further statistics such as mean and standard error of partial likelihood deviance is computed and stored. The the process is repeated with the next specific λ value. When all the specific λ values in the range been exhausted, the λ values yielding the minimum and 1 standard error of the partial likelihood deviance are identified, denoted lambda.min and lambda.1se, respectively. In parallel with each specific λ value the effective number of estimated $\boldsymbol{\beta}$ coefficients, e.g. non-zero coefficients is computed and stored.

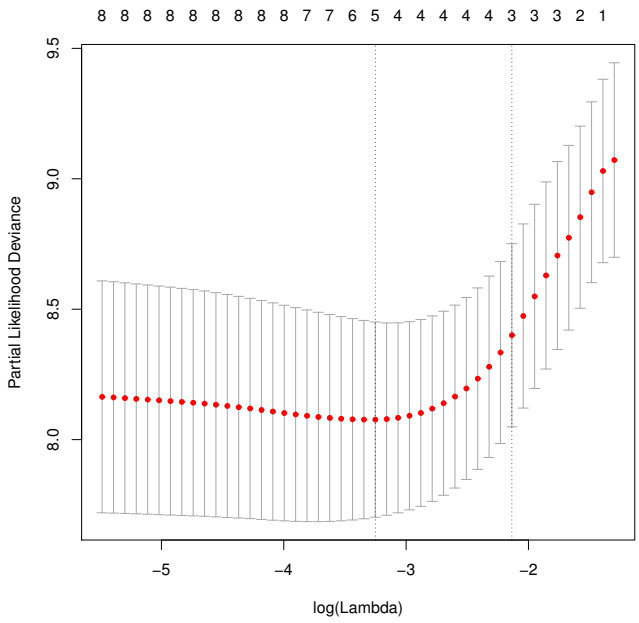

Figure 1: Lasso 3 fold cross-validation for sample size 100,model 1, censoring ratio approximately 30%. Lower x-axis is log(Lambda). Upper x-axis show numbers of estimated non-zero coefficients in model, e.g. model size. The y-axis is the partial likelihood deviance. Left dotted vertical line is lambda.min and right dotted vertical line is lambda.1se

When the cross-validation is done the result can be plotted and Figure 1 gives an example of the output from the cross-validation. The lower x-axis has the logarithm of the λ values and the upper x-axis has the number of effective model parameters. The left dotted vertical indicates the lambda.min and the right dotted line the lambda.1se values, respectively. The y-axis is the partial likelihood deviance, the dots are the average values and the adjoining bars the plus minus one standard error of the partial likelihood deviance.

In this study cv.glmnet is used for selection of the LASSO tuning parameter λ. It is unclear if cv.glmnet provides the same cross-validation, e.g. Equation (12) on Page 15, as Tibshirani (1997), Fan & Li (2002) and Zhang & Lu (2007) uses. This will complicate the comparison of the results.

3.4 All subset selection

When trying to find the model which most accurately describes the outcome an alternative would be to enumerate every model, for each possible model size, and select the model which performs

best. This is what all subset selection refers to. The method is to perform an exhaustive search for all possible subsets, which in practice means that for p number of predictors, enumerating $p, \binom{p}{2}, ..., \binom{p}{p-1}, \binom{p}{p}$ models for model sizes $1, 2, ..., p-1, p$ (James et al. 2014). For every model size, the AIC is calculated for each model in the subset and the model leading to the lowest score of AIC is considered the best model. The last step of all subset selection is to compare the AIC scores of every best model at every given model size, picking the one that has the smallest AIC overall. It is clear to see that as the number of predictors grow large, the task quickly becomes computational intensive. The formula for calculating AIC is given by Equation (8). There have been several suggestions for algorithms which can perform this task for linear regressions in R, but not as many which are compatible with the Cox proportional hazards model. The algorithm used in this thesis is called glmulti and was developed by Calcagno & Mazancourt (2010). The algorithm conducts an exhaustive search over the possible subsets and finds the best model from an information criterion, which in this case will be AIC.

3.5 Stepwise regression

Stepwise regression has similarities with all subset selection in the sense that it enumerates every possible subset at any given size, but differs in the number of possible subsets. Instead of conducting an exhaustive search for each model size, where all combinations of predictors are available, it considers the variables for inclusion or elimination one at a time depending on what type of routine that is implemented (James et al. 2014). In this thesis, the stepwise regression term is used as a collective term for the three different methods of adding or subtracting variables one at a time. The three different methods consist of Forward selection, Backward elimination or a method based on a combination of the two, sometimes referred to as stepwise regression, which is why the clarifying statement above had to be made(James et al. 2014). In this thesis, the backward elimination method is used and will be discussed further below.

Backward elimination starts with all variables included in the model, calculating the information criterion, i.e. AIC 8 or BIC 9, saving them for future comparison. In the next step, the algorithm tries to drop one of the variables at a time, calculating and saving the information criterion for each model at the model size $p-1$. With all the criterion calculated, the variable which is dropped is the one that produced the model with lowest criterion score at $p-1$ and is no longer considered for inclusion, which is why this procedure evaluates fewer subsets. The procedure keeps doing this until the information criterion does not improve by

dropping additional variables. A comparison between all subset selection for p number of predictors, evaluating $p, \binom{p}{2}, ..., \binom{p}{p-1}, \binom{p}{p}$ subsets, backward elimination only have to evaluate $\binom{p}{p}, \binom{p}{p-1}, \binom{p-1}{p-2}, \binom{p-2}{p-3}, ...$ until the stopping criterion is met or the model runs out of variables to drop. The used algorithm for this procedure is outlined in 4.2.2.

The stepwise procedure have been subject to criticism on many counts. One of the most common noted downside is that the procedure is not guaranteed to reveal the best subset of any given size. Furthermore, the restriction to add or delete one variable at a time can possibly create a situation where a great model might be overlooked, as demonstrated by Mantel (1970). With this said, it is a popular procedure which easily can be implemented and is easy to understand.

3.6 Best subset selection

The best subset selection problem have been proposed to be solved in several different ways. In the linear case many applications are built on a branch and bound algorithm, created by Furnival & Wilson (1974), with packages such as leaps and bestglm for R implementation. In this thesis, we evaluate a new strategy for solving the best subset selection problem. The strategy is built on the primal dual active set (PDAS), implemented in R under the package name BeSS (Best subset selection) and was developed by Wen, Zhang, Quan & Wang (2017). Through extensive simulations studies, the BeSS package is proven to solve problems with high dimensional data in matter of seconds on a standard computer, with the number of observations being 1000 and variables ranging all the way up to 10000 (Wen et al. 2017). The idea behind the algorithm is to, instead of enumerating all possible subsets, find the best subset by minimizing a convex loss function, further discussed below.

The BeSS algorithms can be dived into two main parts. The first part deduct the primal dual formulation, active set and Cox proportional hazards model, the second part Determination of the optimal k determine the best model, e.g. the best model with the model size k.

The primal dual formulation, active set and Cox proportional hazards model

The outlining below follows the notation and computations by Wen et al. (2017, pp. 3-4). The general optimization problem to be solved, with the subset size k, can be written as

$$\underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min}\, l(\boldsymbol{\beta}) \quad \text{such that} \quad ||\boldsymbol{\beta}||_0 = k, \tag{13}$$

where $l(\boldsymbol{\beta})$ is a convex loss function of the model coefficients $\boldsymbol{\beta} \in \mathbb{R}^p$ and k is a positive integer. In contrast to LASSO, BeSS uses the L_0-norm $||\boldsymbol{\beta}||_0 = \sum_{j=1}^{p} |\beta_j|_0 = \sum_{j=1}^{p} 1_{\beta_j \neq 0}$, counting the number of non zeros in $\boldsymbol{\beta}$. Wen et al. (2017) states that the known solution to (13) is necessarily a coordinate minimizer, denoted $\boldsymbol{\beta}^{\diamond}$. For each coordinate $j = 1, 2, ..., p$, write $l_j(t) = l(\beta_1^{\diamond}, ..., \beta_{j-1}^{\diamond}, t, \beta_{j+1}^{\diamond}, ..., \beta_p^{\diamond})$ while fixing the other coordinates. Let

$$g_j(t) = \frac{\partial l_j(t)}{\partial t} \tag{14}$$

and

$$h_j(t) = \frac{\partial^2 l_j(t)}{\partial^2 t}, \tag{15}$$

be the first and second derivatives of $l_j(t)$ with respect to t. Then by Taylor expansion of (13), the local quadratic approximation of $l_j(t)$ around $\beta_j^{\diamond}$ is given by

$$l_j^Q(t) = l_j(\beta_j^{\diamond}) + g_j(\beta_j^{\diamond})(t - \beta_j^{\diamond}) + \frac{1}{2} h_j(\beta_j^{\diamond})(t - \beta_j^{\diamond})^2. \tag{16}$$

By expanding the products, reorganize the resulting terms and factors, then by completing the square and further reorganization in (16) gives

$$l_j^Q(t) = \frac{1}{2} h_j(\beta_j^{\diamond}) \left(t - \beta_j^{\diamond} + \frac{g_j(\beta_j^{\diamond})}{h_j(\beta_j^{\diamond})} \right)^2 + l_j(\beta_j^{\diamond}) - \frac{(g_j(\beta_j^{\diamond}))^2}{2 h_j(\beta_j^{\diamond})}. \tag{17}$$

In (17) substitute $-\frac{g_j(\beta_j^{\diamond})}{h_j(\beta_j^{\diamond})}$, which denotes the standard gradient at $\beta_j^{\diamond}$, with $\gamma_j^{\diamond}$. Hence (17) can be written as

$$l_j^Q(t) = \frac{1}{2} h_j(\beta_j^{\diamond})(t - (\beta_j^{\diamond} + \gamma_j^{\diamond}))^2 + l_j(\beta_j^{\diamond}) - \frac{(g_j(\beta_j^{\diamond}))^2}{2 h_j(\beta_j^{\diamond})}. \tag{18}$$

It is clear that minimizing the function $l_j^Q(t)$ in (18) with respect to t is obtained at $t_j^* = \beta_j^{\diamond} + \gamma_j^{\diamond}$. (Wen et al. 2017)

The constraint in (13) suggest that there are $(p - k)$ elements of $\left\{ t_j^*, j = 1, ..., p \right\}$ that will be set to zero. To determine them, consider the sacrifice of $l_j^Q(t)$ if t_j^* is switched from $\beta_j^{\diamond} + \gamma_j^{\diamond}$ to zero, given by

$$\boldsymbol{\Delta}_j = \frac{1}{2} h_j(\beta_j^{\diamond})(\beta_j^{\diamond} + \gamma_j^{\diamond})^2. \tag{19}$$

Among all the candidates, we may impose those t_j^*'s to zero if they contribute the least total sacrifice to the overall loss. To determine those, let $\boldsymbol{\Delta}_{(1)} \geq ... \geq \boldsymbol{\Delta}_{(p)}$ denote the sacrifice vector as a decreasing sorted vector $\boldsymbol{\Delta}_j$ for $j = 1, ..., p$, then truncate the ordered sacrifice vector at

position k, that is drop all coefficients $\beta_j^\diamond + \gamma_j^\diamond$ where $\boldsymbol{\Delta}_j < \boldsymbol{\Delta}_{(k)}$ from the active set (model). Consequently, upon the quadratic approximation of (18), the coordinate minimizer $\boldsymbol{\beta}$ is shown to satisfy the following primal-dual condition:

$$\beta_j^\diamond = \begin{cases} \beta_j^\diamond + \gamma_j^\diamond, \text{ if } \boldsymbol{\Delta}_j \geq \boldsymbol{\Delta}_{(k)} \\ 0, \text{ othewise,} \end{cases} \qquad \text{for } j = 1, ..., p. \tag{20}$$

In (20)[1], treat $\boldsymbol{\beta} = (\beta_1, ..., \beta_p)$ as primal variables, $\boldsymbol{\gamma} = (\gamma_1, ..., \gamma_p)$ as dual variables and $\boldsymbol{\Delta} = (\Delta_1, ..., \Delta_p)$ as reference sacrifices. Primal and dual variables here refers to the theory of convex optimization and the duality theorems, e.g. an Lagrangian optimization problem can be viewed from two perspectives. In the first perspective the the primal variables is the solution and in the second perspective the dual variables is the lower bound solution, for details see duality in Boyd (2004, Chapter 5). These quantities are key elements in the forthcoming description when applied to the Cox proportional hazards model. (Wen et al. 2017)

Equation (13) is the general case and we will now show how Wen et al. (2017) use this technique for the Cox proportional hazards model. Consider the Cox PH model $h(t|\boldsymbol{x}) = h_0(t) \exp(\boldsymbol{x}^\top \boldsymbol{\beta})$, with an unspecified baseline hazards $h_0(t)$ and $\boldsymbol{x} \in \mathbb{R}^p$. Given the survival data $(T_i, \delta_i, \boldsymbol{x}_i) : i = 1, ..., N$ with observations of survival times T_i and censoring indicator δ_i. The model parameters $\boldsymbol{\beta}$ can be obtained by minimizing the convex loss, partial likelihood function (Cox 1972)

$$l(\boldsymbol{\beta}) = -\sum_{i:\delta_i=1} \left(\boldsymbol{x}_i^\top \boldsymbol{\beta} - \ln \left(\sum_{i':T_{i'} \geq T_i} \exp(\boldsymbol{x}_{i'}^\top \boldsymbol{\beta}) \right) \right) \tag{21}$$

For a given $\boldsymbol{\beta}$, write

$$\omega_{i,i'} = \frac{\exp(\boldsymbol{x}_i^\top \boldsymbol{\beta})}{\sum_{i':T_{i'} \geq T_i} \exp(\boldsymbol{x}_{i'}^\top \boldsymbol{\beta})}, \tag{22}$$

then it can be verified that

$$g_j(\beta_j) = -\sum_{i:\delta_i=1} \left(\boldsymbol{x}_{i,j} - \sum_{i':T_{i'} \geq T_i} \omega_{i,i'} \boldsymbol{x}_{i',j} \right) \tag{23}$$

by derive (21) with respect to given $\boldsymbol{\beta}$ rearrange and substitute with (22) and derive (23) with respect to given $\boldsymbol{\beta}$ rearrange and substitute again to get

$$h_j(\beta_j) = -\sum_{i:\delta_i=1} \sum_{i':T_{i'} \geq T_i} \omega_{i,i'} \left(\boldsymbol{x}_{i',j} - \sum_{i':T_{i'} \geq T_i} \omega_{i,i'} \boldsymbol{x}_{i',j} \right)^2, \tag{24}$$

[1]It is strange to have the same notation on left and right hand in the equation, but this is the notation used by Wen et al. (2017)

so that

$$\gamma_j = -\frac{g_j(\beta_j)}{h_j(\beta_j)} \tag{25}$$

and

$$\Delta_j = \frac{1}{2} h_j(\beta_j)(\beta_j + \gamma_j)^2 \tag{26}$$

for $j = 1, ..., p$. (Wen et al. 2017)

With the above the active set can be defined as the following. For the best subset problem in (13, the active set is defined as $\mathcal{A} = \{j : \beta_j \neq 0\}$ with cardinality k and the inactive set $\mathcal{I} = \{j : \beta_j = 0\}$ with cardinality $p - k$. For the coordinate-wise minimizer $\boldsymbol{\beta}^\diamond$ satisfying the primal-dual condition (20), we have that

- When $j \in \mathcal{A}$, $\quad \beta_j^\diamond \neq 0, \gamma_j^\diamond = 0$ and $\Delta_j = \frac{1}{2} h_j(\beta_j^\diamond)(\beta_j^\diamond)^2$
- When $j \in \mathcal{I}$, $\quad \beta_j^\diamond = 0, \gamma_j^\diamond = -g_j(0)/h_j(0)$ and $\Delta_j = \frac{1}{2} h_j(0)(\gamma_j^\diamond)^2$
- $\Delta_j \geq \Delta_{j'}$ whenever $j \in \mathcal{A}$ and $j' \in \mathcal{I}$.

Obviously, the primal variables β_j's and the dual variables γ_j's have complementary supports. Complementary support here means that when $\beta_j^\diamond \neq 0$ then $\gamma_j^\diamond = 0$ and when $\beta_j^\diamond = 0$ then $\gamma_j^\diamond \neq 0$. The active set $\mathcal{A}$ plays a crucial role in the best subset problem; indeed if $\mathcal{A}$ is known a priori, we may estimate the k non-zero primal variables by standard convex optimization:

$$\min l(\boldsymbol{\beta}_{\mathcal{A}}) = \min_{\boldsymbol{\beta}_{\mathcal{I}}=0} l(\boldsymbol{\beta}), \qquad \text{where} \qquad \mathcal{I} = \mathcal{A}^{\complement}. \tag{27}$$

We may use an iterative procedure to determine the active set $\mathcal{A}$. Suppose at the m-th iteration with the current estimate $\mathcal{A}^m$, we may estimate $\boldsymbol{\beta}^m$ by (27) and derive (γ^m, δ) as discussed above and then update the active set by

$$\mathcal{A}^{m+1} = \{j : \Delta_j^m \geq \Delta_(k)^m\} \text{ and } \mathcal{I}^{m+1} = \{j : \Delta_j^m < \Delta_{(k)}^m\}. \tag{28}$$

From the above the Primal-dual active set (PDAS) algorithm can be created (Wen et al. 2017). The renaming issue is now how should the k be selected.

Determination of optimal k

The model size k is normally unknown and has to be determined from the data. Cross-validation is one way, e.g. in a similar fashion as for LASSO, but cross-validation is computationally

expensive. An alternative is to execute the PDAS algorithm over a sequence of small to large k values and identify an optimal choice according to some criteria such as AIC or BIC, while another alternative is to use a golden section method. The basic idea of the golden section method is to identify the knee in the loss function, see for example in Figure 2. The knee is were the likelihood function levels out or in other words, if the model size is increased the change in the likelihood is small. On the other hand, if the model size is decreased the change in likelihood is larger. Thus, when a true predictor is added, the loss function drops dramatically and the previously added predictors are adjusted accordingly. Consequently the k value at the knee is the optimum one. (Wen et al. 2017)

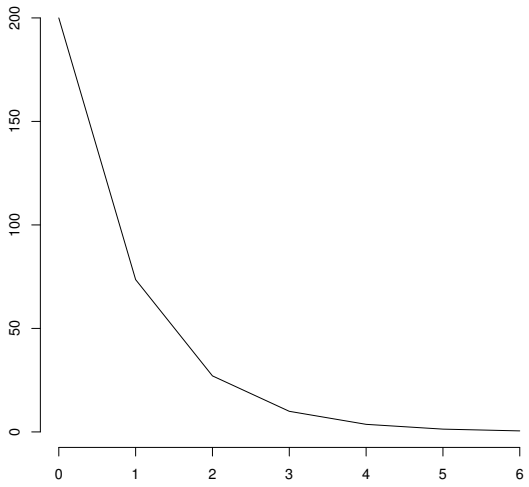

Figure 2: The likelihood as a function of the model size k levels out at around model size 3, so the optimal k-value used in Equation (13) will be $k = 3$.

For details about the algorithms see Wen et al. (2017).

3.7 Previous comparative studies

This study is not the first to examine how well different variable selection methods performs and how they compare with each other. However, the results are not consistent and different studies come up with varying selection techniques that manage to pick the true model to the highest extent.

Tibshirani (1997) evaluated the LASSO method for the Cox proportional hazards model against stepwise regression. The methods are examined by simulating 50 datasets, each containing 50 observations, with censoring percentage set to zero. The correlation between the observations x_i and x_j is set to $\rho^{|i-j|}$ where $\rho = 0.5$. Two different models are evaluated, with the difference being the coefficient's values, using the same number of datasets and observations. The different models are used in order to evaluate the performance of the techniques in models with large effects as well as small effects. Tibshirani (1997) does not provide details of the average number of correct or incorrect coefficients set to zero, but reports the average number of zero coefficients. Without an exact number, it is difficult to assess which model actually performs better than the other in terms of finding the correct model. The results does however provide the median mean square error (MMSE), as a metric for how well the selected models fit the data. The results show that the LASSO, on average, sets more coefficients to zero compared to the stepwise regression and this holds true for both models with different coefficient values. In each case the LASSO also proved to have better MMSE compared to stepwise regression, suggesting that the LASSO tends to find models which explains the data to a higher degree. Furthermore, the stepwise regression is shown to have much higher variability in terms of selected variables and estimated coefficients, while LASSO seems more stable.

Fan & Li (2002) conducted a study on 100 simulated datasets with 75 and 100 observations with a censoring rate of 30% comparing the SCAD, LASSO, Best subset and HARD methods. The dataset consisted of 8 variables where only 3 where included in the true model (see model 1 in Table 1), having a $\beta \neq 0$, while the others should be dropped by the variable selection method. The variables in each column have a correlation of 0.5 with variables in the column next to it, which is decreasing the more columns that are between the two variables. The comparisons are made by examining the average number of correct and non-correct zero coefficients as well as comparing median relative model error (MRME) in percent. For a sample size of 75, their findings show that HARD and Best subset regression have an average of 4.87 and 4.75 correct zero coefficients, beating both LASSO and SCAD. Although HARD is the closest to the correct number of five, it also have the highest number of incorrect coefficients set to zero, at 0.29 on average. Comparing the MRME across the different selection techniques, the SCAD outperforms the other and LASSO ending up performing worst. Much of the same hold true when the sample size is increased from 75 to 100. The the main difference is that now less variables are incorrectly set to zero but the metric MRME is worse for all methods except

SCAD which actually improves its MRME. Their results suggest that LASSO possibly could impose to light of a penalty, since it on average prefers to use larger models. On the other hand, it seems like HARD is too strict, setting the most coefficients to zero but producing poor results of MRME, suggesting that important predictors might be dropped as a result of harsher restrictions on the model size. Their findings show that the SCAD estimator outperforms the others, both in terms of average number of correct zero coefficients as well as the incorrect zero coefficients.

A more recent study by Zhang & Lu (2007) proposed a new LASSO method called Adaptive LASSO (ALASSO) and conducted extensive simulation studies comparing the techniques SCAD, LASSO, MLE against ALASSO. The comparisons have similarities with Fan & Li (2002) but are more extensive, comparing more sample sizes and two different censoring percentages. The censoring rates are set at 25% and 40% and correlation between the variables are the same as Fan & Li (2002) and Tibshirani (1997) uses, with sample sizes being 100, 200 and 300. Furthermore, they use two different models with varying coefficient sizes, see models 2 and 3 in Table 1. The first model represents variables with strong effects and the second model use smaller coefficients, which represents small effects on the outcome. Zhang & Lu (2007) and Fan & Li (2002) counts the average number of correct and incorrect zero coefficients which are compared between the outcome of methods and models. For the model with large effects, ALASSO proved to be the best model in terms of mean square error as well as choosing the right model. Their findings also show that the mean square error of LASSO was smaller compared to that of SCAD for a sample size of 100, but as sample size increased, the SCAD started outperforming LASSO in this metric as well. The LASSO performed well compared to the other methods when it came to not setting any coefficients incorrectly to zero but fell short in setting every zero-coefficient to zero correctly. The performance of the selection techniques decreased across the board when trying to select the model with small coefficients. The most notable difference from their findings with the other model is that now both ALASSO and SCAD did substantially worse in the sense of setting more non-zero coefficients to zero than before. This hold true for LASSO as well but not to the same extent. The results once again suggest that LASSO tends to, on average, include more variables in the final model.

4 Procedures to achieve the estimations

This section describes the procedures that are used to estimate the $\boldsymbol{\beta}$ coefficients in the models and the resulting prediction efficiency. It is divided into the following Subsections: Simulation of data sets, Variable selections with criterion and estimations, The main algorithm and ending with Prediction accuracy metrics, e.g. Subsections 4.1 to 4.6.

4.1 Simulation of data sets

In the effort to partly compare the various variable selection procedures used in Tibshirani (1997), Fan & Li (2002), Zhang & Lu (2007) and added methods; such as all subset selection, backward elimination and best subset selection, the input data has to correspond to what Tibshirani (1997), Fan & Li (2002), Zhang & Lu (2007) used. Consequently this subsection describes simulation of the data sets used in this study, in the following order True models and covariates and then Survival times and status.

True models and covariates

Tibshirani (1997), Fan & Li (2002) and Zhang & Lu (2007) use different true models $\boldsymbol{\beta}$, displayed in Table 1. In Table 1 we show the 5 different models with two different model sizes, e.g. 8 and 9. Thus both the true coefficients $\beta_1, \beta_2, ..., \beta_p$ and true model size denoted p differ between

Table 1: True models, the model column holds the model number

Model	Article	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9
1	Fan & Li (2002)	0.8	0	0	1	0	0	0.6	0	
2	Zhang & Lu (2007)	-0.7	-0.7	0	0	0	-0.7	0	0	0
3	Zhang & Lu (2007)	-0.4	-0.4	0	0	0	-0.2	0	0	0
4	Tibshirani (1997)	-0.35	-0.35	0	0	0	-0.35	0	0	0
5	Tibshirani (1997)	0.1	0.1	0	0	0	0.1	0	0	0

what Tibshirani (1997), Fan & Li (2002) and Zhang & Lu (2007) uses. Let the matrix $\boldsymbol{x}$, a $N \times p$ matrix represent the N observations of the p covariates. Therefore $\boldsymbol{x}_l$ denotes the lth observation with the p covariates. Let $\boldsymbol{x}_{l,i}$ denote the lth observation of covariate i. Tibshirani (1997), Fan & Li (2002) and Zhang & Lu (2007) used marginally standard normally distributed covariates with pairwise correlation between observation $\boldsymbol{x}_{l,i}$ and $\boldsymbol{x}_{l,j}$ with $\rho^{|i-j|}$ with the same

$\rho = 0.5$, where $i, j = \{1, 2, ..., N\}$. Consequently the correlation matrix $\boldsymbol{\rho}$ is a $p \times p$ correlation matrix where the diagonal elements are one, because $\rho^{|j-j|} = \rho^0 = 1$ and the off diagonal values are $\rho^{|i-j|}$, with $\rho = 0.5$. The correlation matrix is depicted as a correlation heat map in Figure 3.

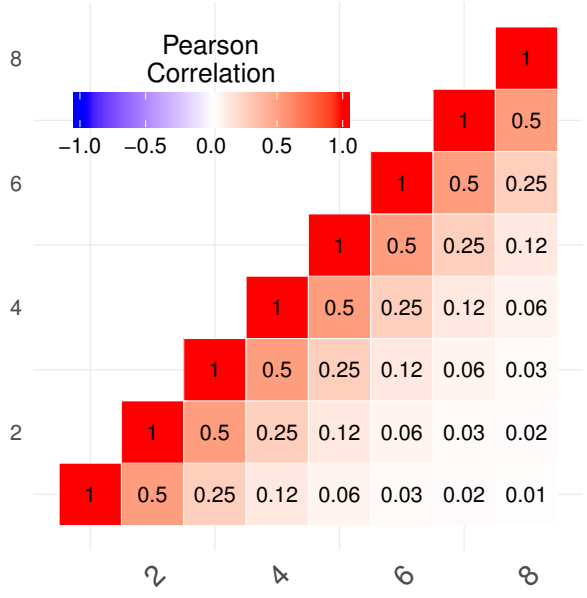

Figure 3: Input correlation, e.g. $0.5^{|i-j|}$ for simulation of covariates. X and Y axis holds the number of the covariates.

This results in moderate to strong effects for the non-zero regressors for model 4 and small effect for model 5 in Table 1 according to Tibshirani (1997). Thus models on row 1 to 3 gives strong to moderate effects.

Two other differences between Tibshirani (1997), Fan & Li (2002) and Zhang & Lu (2007) are the samples sizes and the number of simulations they did. Tibshirani (1997) used sample size 50 and did 50 simulations, Fan & Li (2002) used two samples sizes, 75, 100 and did 100 simulations. Finally Zhang & Lu (2007) uses three sample sizes 100, 200, 300 and did 100 simulations. This study use the following sample sizes 50, 75, 100 and 200 and simulations 50 and 100. All simulations in this study is done with "the software package R".

Neither (Tibshirani 1997, Fan & Li 2002) and Zhang & Lu (2007) provides any details regarding if the input correlation should reflect the simulated samples or the population. Note that the

population covariance $\boldsymbol{\Sigma}$ is needed to calculate MSE given by

$$MSE = (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta_0})\boldsymbol{\Sigma}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta_0})^\top, \tag{29}$$

where $\hat{\boldsymbol{\beta}}$ is the estimated coefficients of the model, $\boldsymbol{\beta_0}$ is the coefficients of the true model and $\boldsymbol{\Sigma}$ is the population covariance matrix according to Tibshirani (1997). We have interpreted $\boldsymbol{\Sigma}$ as the population covariance matrix of the marginally standard normal covariates and is computed according to Equation (30) and that the simulated covariates are marginally standard normal, which gives that each covariates have the standard deviation, $\sigma = 1$ and mean, $\mu = 0$ with the correlation $\boldsymbol{\rho}$. Consequently we interpret and define

$$\boldsymbol{\Sigma} = \boldsymbol{I}, \tag{30}$$

where $\boldsymbol{I}$ is the $p \times p$ identity matrix of the covariates standard deviation.

In this study the empirical parameter is true in the call to mvrnorm in R so that the correlation matrix of the simulated covariates reflects the input correlation matrix. Given from the above information we can simulate the covariates $\boldsymbol{x}$ for the different data sets. The first step is to create the correlation matrix and in the second step creating the $\boldsymbol{x}$ matrix for the different data sets needed, e.g. by using the function mvrnorm in R. In order to complete the data set response variables are needed, such as survival times and status.

Survival times and status

There are several ways to simulate random data such as rejection sampling (Casella et al. 2004), ziggurat algoritm (Marsaglia & Tsang 2000) and inverse transform sampling, e.g. used by Bender et al. (2006), which is used in this case because it is easy to create the inverse of the cumulative distribution function included in the survival function (1). Consequently the survival times are created according to the following deduction.

Let $\boldsymbol{\beta}$ be a $p \times 1$ vector representing the parameters in the true model, e.g. one of the models in Table 1. Then let $\boldsymbol{X}$ denote the $N \times p$ random variables and $\boldsymbol{X}_i \sim N(0, \boldsymbol{\rho})$ denote the p random variables of the observation i, correspondingly $\boldsymbol{x}$ are observations of the random variables $\boldsymbol{X}$ and $\boldsymbol{x}_i$ the ith observation. By using Bender et al.'s (2006) description and the definitions in the two previous sentences, the following will be done to simulate the exponentially distributed survival times in Cox proportional hazards model given in (4), which is a death intensity or rate function. It has the survival function $S(t|\boldsymbol{x}) = e^{-H_0(t)e^{\boldsymbol{x}\boldsymbol{\beta}}}$, where the cumulative baseline hazard function $H_0(t) = \int_0^t h_0(s)e^{\boldsymbol{x}\boldsymbol{\beta}}ds$ and $\boldsymbol{x}$ are independent of t. Here, $h_0(t)$ is called the baseline

hazard function. Consequently the cumulative distribution function of the Cox proportional hazards model (4) is given by

$$F(t|\boldsymbol{x}) = 1 - e^{-H_0(t)e^{\boldsymbol{x}\boldsymbol{\beta}}}. \tag{31}$$

Next, let Y be a random variable with distribution function F, then $U = F(Y)$ follows a uniform distribution in the interval $[0, 1]$, i.e. $U \sim Uni[0, 1]$, because $F(Y) \in [0, 1]$. This also entails that $(1-U)$ is $Uni[0, 1]$. Thus, let the survival time T have hazard rate $h_0(t) \exp \boldsymbol{x}\boldsymbol{\beta}$. Equation (31) and the above observations then implies

$$U = e^{-H_0(T)e^{\boldsymbol{x}\boldsymbol{\beta}}} \sim Uni[0, 1]. \tag{32}$$

Further, let $H_0(t) = h_0 t$, where h_0 is a positive constant, so that $H_0^{-1}(s) = \frac{s}{h_0}$. Then, using (32) the survival times T of the Cox proportional hazards model can be expressed as:

$$T = H_0^{-1}\left(-\frac{\ln(U)}{e^{\boldsymbol{x}\boldsymbol{\beta}}}\right) = -\frac{\ln(U)}{h_0 e^{\boldsymbol{x}\boldsymbol{\beta}}} \tag{33}$$

where U is a random variable $U \sim Uni[0, 1]$. This also corresponds with the results of Bender et al. (2006, Table II). Thus, to simulate different survival times T_i for observations $i = 1, ..., N$ with different covariates $\boldsymbol{x}_i$ and $h_0 = 1$, since Tibshirani (1997), Fan & Li (2002) and Zhang & Lu (2007) uses $h_0 = 1$, we use Equation (33) so that

$$T_i = -\frac{\ln(U_i)}{e^{\boldsymbol{x}_i\boldsymbol{\beta}}}, \quad \text{where} \quad U_i \sim Uni[0, 1].$$

Here, U_i can be simulated with R's function runif or with R's function rexp because $-\ln(U) \sim Exp(1)$. Consequently the latent survival times are

$$T_i \sim \frac{\text{Exp}(1)}{\exp(\boldsymbol{x}_i\boldsymbol{\beta})}. \tag{34}$$

Fan & Li (2002) have approximately 30% censored times, while Zhang & Lu (2007) have 25% and 40% censored times. Furthermore, they also claim that the censoring schemes is non informative. Fan & Li (2002) assumes that latent survival times,e.g. T_i and latent censored times, e.g. C_i are conditionally independent given the covariates $\boldsymbol{x}$. The below four quotes emphasize the censoring mechanisms used by Fan & Li (2002) and Zhang & Lu (2007):

> *Let T, C and $\boldsymbol{x}$ be respectively the survival time, the censoring time and their associated covariates. Correspondingly, let $Z = \min\{T, C\}$ be the observed time and $\delta = I(T \leq C)$ be the censoring indicator. It is assumed that T and C are conditionally independent given $\boldsymbol{x}$ and that the censoring mechanism is noninformative.*
> (Fan & Li 2002, p. 79)

The distribution of the censoring time is an exponential distribution with mean $Uexp(\boldsymbol{x}\top\boldsymbol{\beta}_0)$*, where* U *is randomly generated from the uniform distribution over* $[1, 3]$ *for each simulated data set so that about 30% data are censored. Here* $\boldsymbol{\beta}_0 = \boldsymbol{\beta}$ *which is regarded as a known constant so that the censoring scheme is noninformative.* (Fan & Li 2002, p. 89)

Censoring times are generated from a $Un(0, c_0)$ *distribution,where* c_0 *is chosen to obtain the desired censoring rate.* (Zhang & Lu 2007, p. 696)

Define $\tilde{T}_i = min(T_i, C_i)$ *and* $\delta_i = I(T_i \leq C_i)$*. We use* $z_i = (z_{i1}, ..., z_{id})^\top$ *to denote the vector of covariates for the ith individual, Assume that* T_i *and* C_i *are conditionally independent given* z_i*, and that the corresponding censoring mechanism is noninformative.* (Zhang & Lu 2007, p. 692)

Both Fan & Li (2002) as well as Zhang & Lu (2007) selects the pairwise minimum between the latent survival time T_i and latent censored time C_i, which becomes the survival time, e.g. $\tilde{T}_i = \min(T_i, C_i)$.

Fan & Li's (2002) statement about a censoring rate about 30% yields an average censoring rate greater than 35% in eight cases of 100 simulations with sample size 100. Instead we are using U randomly generated from the random uniform distribution over interval [0,0.95] for Fan & Li (2002) data sets.

If one use the Zhang & Lu (2007) censoring description, the selection of c_0 seems to vary with the sample size and selected censoring rate. Thus c_0 seems to be a function. The c_0 can be calculated by using the distributions of T_i and C_i, given that T_i and C_i are conditionally independent of the covariates and the known censoring rate. Instead of calculating c_0, we used trial and error to determine these parameters c_0. The approximations are stated in Table 2, which was used for the Zhang & Lu (2007) data sets.

Table 2: c_0 values for different censor rates and sample sizes used for Zhang & Lu (2007) data sets.

Censor rate (%)	25	25	40	40
Sample size	100	200	100	200
c_0	6.35	4.7	2.6	2.3

The observed survival times for Fan & Li (2002) and Zhang & Lu (2007) are $t_i = \tilde{T}_i$ and Tibshirani (1997) have no censored times, so observed times are $t_i = T_i$.

The indicator variable status, denoted δ_i marks if the observed survival time is a latent survival time or a latent censored time. To sum this up the simulated variables, e.g. t_i, δ_i and $\boldsymbol{x}_i$ for $i = 1, 2, ...N$ are combined into data sets suitable for the Cox proportional hazards model with different true models. They use different model sizes, sample sizes, censoring rates (0%, 25%, 30%, 40%) in each of the articles by Tibshirani (1997), Fan & Li (2002) and Zhang & Lu (2007).

4.2 Variable selections with criterion and estimations

The below discussion describes the different Cox proportional hazards model variable selection methods used in this study, such as backward elimination, all subset selection, best subset selection and LASSO, which was presented in Section 3. All of these use different criterion for selecting the model, e.g. AIC, BIC and LASSO penalty. In R each of these methods are realized by calling different functions from different R packages, such as glmulti, glmnet and BeSS. To provide a generic interface for these functions in order to handle different input and output for these functions they are enclosed in wrapper functions. The functionality of these wrapper functions are to re-organize data and results in a uniform way and add different, but necessary parameters for each method function. By the aid of the wrapper functions, regardless of method, data, parameters and results all method functions can be called in a standard way and like wise return the result in a uniform way, e.g. estimated model coefficients and the fit to be used for further processing. The fit is an R standard that is used to return results from linear regression optimizing.

The different methods with estimations are organized in sub-subsections starting with All subset selection, then Backward elimination, Best subset selection and Least absolute shrinkage and selection operator. Lastly, the main algorithm is outlined and this section ends with the Prediction accuracy metrics.

4.2.1 All subset selection

This method uses an exhaustive search of all possible models as described in Subsection 3.4, that is starting with using all covariates in the model and then all possible combinations of

that model and selects the model according to the lowest value of the AIC criteria. This is implemented by using R' function glmulti. Glmulti is straightforward to use, just provide the data set for each simulation and the maximum sized model, that is using all p covariates. The glmulti returns the best model according to the criterion. One reason to use this function is that it has support for Cox proportional hazards model directly. One drawback of the glmulti is that it can only handle 32 covariates (predictors). Although this does not pose a problem in this thesis, it is important to keep in mind that it can not solve high dimensional problems. The wrapper function returns the coefficients of the selected model and the fit.

4.2.2 Backward elimination

Recall from Subsection 3.5, that the idea behind backward elimination is to start with a full model, then repeatedly removing one of the covariates from the model, selecting the best of these models and then repeating this until all covariates are exhausted. The selection criterion can be arbitrarily selected, e.g. AIC or BIC. We have not found any function that directly can be implemented with Cox PH in R, but that does not mean that none exist. There are functions in R that can do various stepwise selections, however it is unclear if these have support for Cox proportional hazards model. Consequently it was implemented according to the following pseudo algorithm from Jörnsten (2017), which is slightly modified to use R's coxph function and support AIC and BIC criterion. The wrapper function returns the coefficients of the selected model and the fit.

1. Start with the full model and solve for the estimation of the $\boldsymbol{\beta}$ coefficients by the aid of R's function coxph.
2. Calculate the criteria, e.g. AIC or BIC according to equations (8) and (9) in Subsection 3.1 for the Cox proportional hazards model and save this as $Crit_{complex}$ together with the coefficients and the fit.
3. Let $p_{current} = p$.
4. Create the active set A of variables to $\{z_1, z_2, ..., z_{p-1}\}$ where z_k, $k = \{1, 2, ..., p-1\}$.
5. Repeat
   (a) Solve in the same way as described in step 1 for each combination by dropping one of the variables of the $p_{current} - 1$ subset models in the active set A.

(b) Calculate the criteria, for example using AIC or BIC and save as $Crit_k$ for each $k \in [1, p_{current} - 1]$.

(c) Determine the variable $z_* \in A$ with the minimum $Crit_* = \min_{k \in A} Crit_k$.

(d) If $Crit_* < Crit_{complex}$.

i. drop z_* from the model and update active set $A = A \setminus z_*$.

ii. Let $Crit_{complex} = Crit_k$ and $p_{current} = p_{current} - 1$

iii. If $p_{current} = 1$ go to step 6. No more variables to drop, Cox proportional hazards model's intercept is in the baseline function.

iv. Go to 5

(e) Else if If $Crit_* > Crit_{complex}$ go to step 6

6. Save winning model with coefficients and criteria, e.g. the fit.

4.3 Best subset selection

The R package BeSS provides three different methods, we are are using BeSS.gs and BeSS.seq. Both of these have support for Cox proportional hazards model and the criteria AIC or BIC. We use all four combinations and the functions are straightforward to use. The wrapper function returns the coefficients of the selected model and the fit. A difference from the all subset selection and backward elimination methods is that the BeSS functions return the solution path in a similar way as LASSO. The solution paths is a diagram of the curves of the estimated $\boldsymbol{\beta}$ coefficients as a function of the best model for that model size.

4.4 Least absolute shrinkage and selection operator

We are using R's package glmnet, because it supports the Cox proportional hazards model. A difficulty with LASSO is the selection of the criterion, e.g the LASSO tuning parameter λ, which controls the shrinkage of the coefficients which also ultimately determines which coefficients in $\boldsymbol{\beta}$ that are set to 0. The tuning parameter is normally selected by cross-validation see Figure 4.

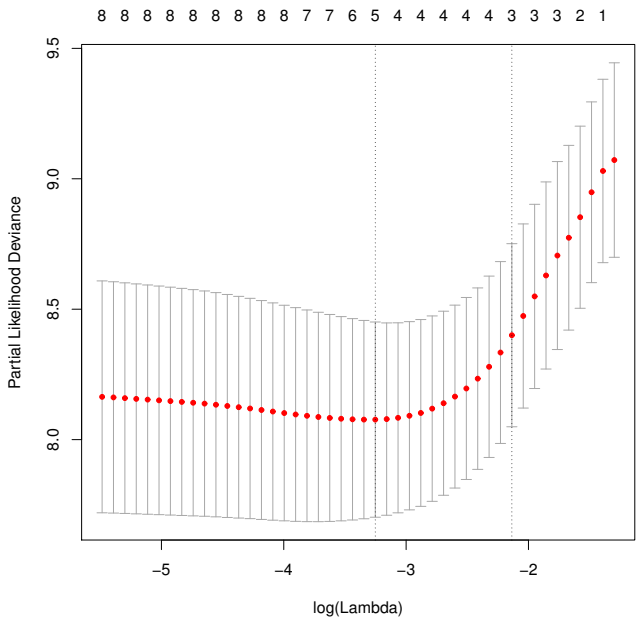

Figure 4: Lasso 3 fold cross-validation for sample size 100,model 1, censoring ratio approximately 30%. Lower x-axis is log(Lambda). Upper x-axis show numbers of estimated non-zero coefficients in model, e.g. model size. The y-axis is the partial likelihood deviance. Left dotted vertical line is lambda minimum and right dotted vertical line is lambda 1 standard deviation.

Note that λ corresponds to Lambda in Figure 4. Two ordinary choices for λ are lambda.min or lambda.1se, for details see (Hastie et al. 2015, p. 45). Lambda.min gives the minimum partial likelihood deviance and lambda.1se is one standard error above the lambda.min, which results in more coefficients set to zero, see upper x-axis in Figure 4. Another parameter that influence the lambda.min and lambda.1se in the glmnet cross-validation in the glmnet package is the number of folds to use. Recall the fold discussion in Subsection 3.3. The standard value is 10 however if the sample size is small each fold will have few observations which have an impact on the cross-validation results, e.g. partial likelihood, lambda and number of coefficients in the model. Consequently the number of folds in cross-validation is changed to 3 if sample size is less than 100, 5 if sample size is 100 and 10 if sample size larger than 100. In this study we are using $\lambda = (lambda.min + lambda.1se)/2$ as the tuning parameter. The wrapper function, see Subsection 4.2 returns the coefficients of the selected model, the fit and the cross-validation.

Tibshirani (1997), Fan & Li (2002) and Zhang & Lu (2007) uses generalized cross-validation,

however it is unclear if the cross-validation in the glmnet package gives the same tuning parameter λ value as the $\arg\min_s$ of Equation (12).

4.5 The main algorithm

Each of the estimating methods, e.g. all subset selection, backward elimination, BeSS.gs, BeSS.seq and LASSO are surrounded by a wrapper function, described previously. The reason for this is to organize the input data and output data so the different methods can be used in a generic way from the main algorithm. To cover for the different settings of parameters such as, true model, criteria, sample size, censoring ratio, simulations and methods these are organized into an object with the wrapper function for each configuration. Each such configuration object is then added to a set, denoted M. The main pseudo-algorithm is available in Appendix E.

4.6 Prediction accuracy metrics

Let $\hat{\boldsymbol{\beta}}$ be the estimated coefficients in the model and let $\boldsymbol{\beta}_0$ be the model parameters we want to estimate. To verify how close $\hat{\boldsymbol{\beta}}$ are $\boldsymbol{\beta}_0$ one can use different types of "error" measure. We use the median of the mean square errors ($MMSE$). The $MMSE$ is calculated as the median of the mean square errors in Equation (29).

We also estimates the mean, e.g. $\bar{\hat{\beta}}_k$, of each coefficient for the simulations and the standard error, e.g $\hat{\beta}_{Se_k}$, of each coefficient versus the true model according to:

$$\bar{\hat{\beta}}_k = \frac{1}{r}\sum_{i=1}^{r} \hat{\boldsymbol{\beta}}_{\boldsymbol{k}i} \tag{35}$$

and

$$\hat{\beta}_{Se_k} = \sqrt{\frac{1}{r-1}\sum_{i=1}^{r} \left(\hat{\boldsymbol{\beta}}_{\boldsymbol{k}i} - \boldsymbol{\beta}_k\right)^2}, \tag{36}$$

where $\hat{\boldsymbol{\beta}}_{\boldsymbol{k}i}$ is the estimated coefficient for all simulations for the current object in the set M and r is the number of simulations, $k = \{1, 2, ..., p\}$ is the sequence of coefficients in the current object's true model and p is the current object's true model size.

The analysis of our results and conclusions are reported in Sections 5 and 6.

5 Numerical results of the simulations and estimations

In the following subsections we will report the results from the simulations as well as the estimations with the different methods and models. Recall the numbering of the models in Table 1. All models uses 100 simulations except models 4 and 5 (Tibshirani 1997), which only uses 50, because Tibshirani (1997) only reports for 50 simulations. All simulations and estimations are done in R. In this section's subsections, where applicable, e.g. for the LASSO methods, the LASSO tuning parameter λ is the mean of lambda.min and lambda.1se. The lambda.min and lambda.1se values are the results from the cross-validation in the LASSO method,e.g the cv.glmnet function in R. The Subsections of this section are organized in this order; Simulated data, Solutions paths and cross-validation for BeSS and LASSO, Median mean square errors and average number of correct and incorrect coefficients, Means and standard error of the estimated $\boldsymbol{\beta}$ coefficients and ending with a comparison of zero, non-zero coefficients and $MMSE$ results versus earlier papers. e.g. Subsections 5.1 to 5.5.

5.1 Simulated data

To verify that the simulated data have the right distributions for covariates and simulated times, three figures are presented of the covariates distribution, the correlation of the covariates and the distribution of the simulated times. The average actual censoring rate is stated in Table 3.

Covariates distribution

Figure 5 shows one of the covariates from one of the data sets for model 1, see Table 1, with 100 observations. It has a few gaps, occasionally overshots and some slight skewness compared to the normal curve. If the number of observations are increased, for example to 10000, these anomalies vanish. It is approximately normally distributed. The remaining covariates histograms for true model 1 are similar and can be found in the Appendix A. Similar behaviors are observed for these covariates as well, but just like before, increasing the sample size removes the observed anomalies. Thus, we conclude that the covariates are marginally normally distributed.

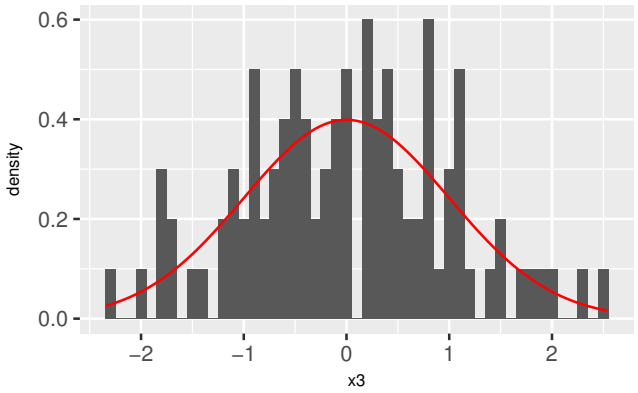

Figure 5: Histogram of simulated covariate X3, normalized, with 100 observations, true model 1, normalized. The line depicts the normal curve.

Covariates correlation

The left and right plots in Figure 6 displays the input correlation matrix and the resulting covariates correlation matrix respectively, at sample size 100. Analyzing the two plots reveals that the covariates correlation is approximately equal to the input correlation. Consequently, the mvrnorm with the extra parameter empirical set to true, the output correlation approximately becomes a replica of the input correlation matrix. All data sets for all models in Table 1 use the same input correlation.

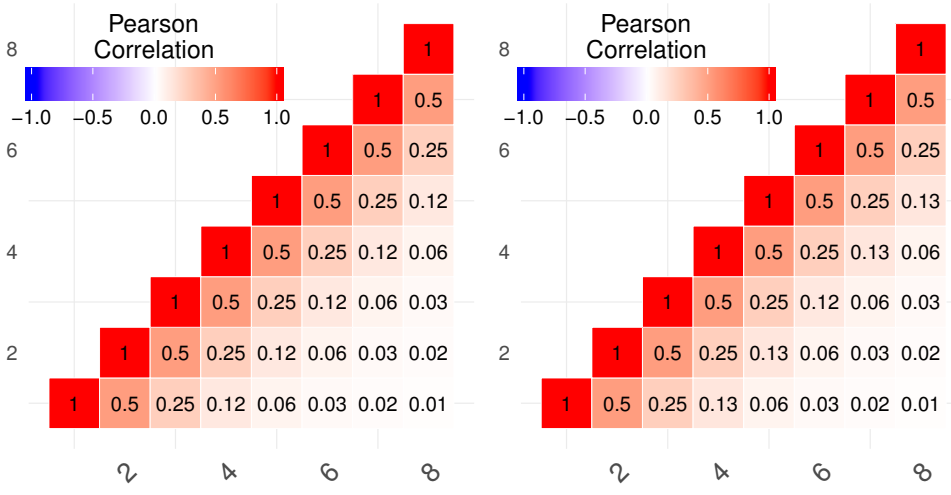

Figure 6: Input correlation, e.g. $0.5^{|i-j|}$ for covariate simulation to the left, resulting correlation for simulated covariates to the right, where i and j are the numbers, e.g. x and y axis, of the p (model size) covariates in the model.

Simulated times

Simulated times for model 1, including both event and censored times, after normalization, is here presented in Figure 7. It approximately possess the characteristics of an exponential distribution. There are gaps and overshots compared to exponential curve, but as before the sample size is small. If the sample size is increased in the similar way as before, but as the simulated times also contains censored times of different distribution versus the event times, the simulated times becomes approximately exponentially distributed. The censoring scheme's parameters are different from what Fan & Li (2002) and Zhang & Lu (2007) uses. For example Fan & Li (2002) use $U \sim Uni[1, 3]$ and our simulations use $U \sim Uni[0, 0.95]$, since the former gives a censoring rate of approximately 65%. Zhang & Lu's (2007) parameter c_0 is unknown and has been determined by repeated trial and error, thus they may deviate from what Zhang & Lu (2007) uses. A reason to these differences may be a misinterpretation of Fan & Li (2002), Zhang & Lu (2007) papers.

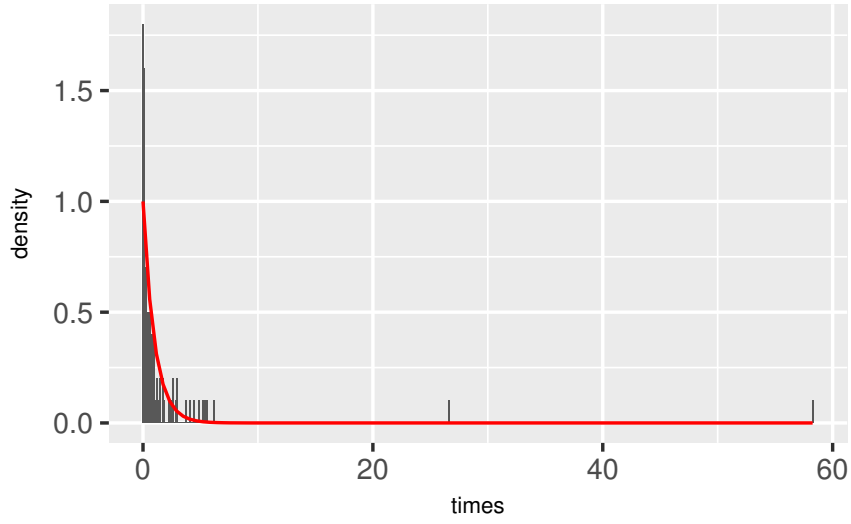

Figure 7: Histogram of 100 simulated observation times with approximately 30% censoring rate, true model 1, normalized.

5.2 Solution paths and cross-validations for BeSS and LASSO

This subsection discuss plots of the results from the BeSS and LASSO methods. Both the BeSS methods and LASSO provides solutions paths, e.g. each curve's estimated $\boldsymbol{\beta}$ coefficient's values as a function of model size. The BeSS methods provides likelihood and the coefficients' estimates as a function of model size in the same Figure 8. LASSO uses two Figures 9 and 10

to provide similar information. The first LASSO Figure 9, shows the LASSO cross-validation, e.g. log likelihood deviance as a function of the logarithm of lambda on the lower x-axis and models size on the upper x-axis. The second LASSO Figure 10, displays the LASSO solution paths as function of model size on the upper x-axis and the logarithm of lambda on the lower x-axis.

Figure 8 displays the solution path and likeliyhood for the BeSS method applied to model 1. The x-axis in Figure 8 represents the model size and the upper half in Figure 8 displays the likelihood as a function of estimated coefficients values at that model size. The lower half in Figure 8 gives the solution paths, e.g. each curve's estimated $\boldsymbol{\beta}$ coefficient's values as a function of model size. The number to the right and above each curve corresponds to the coefficient in the model. The vertical dashed line show the optimum of the selected criterion, in this plot it is the minimum AIC.

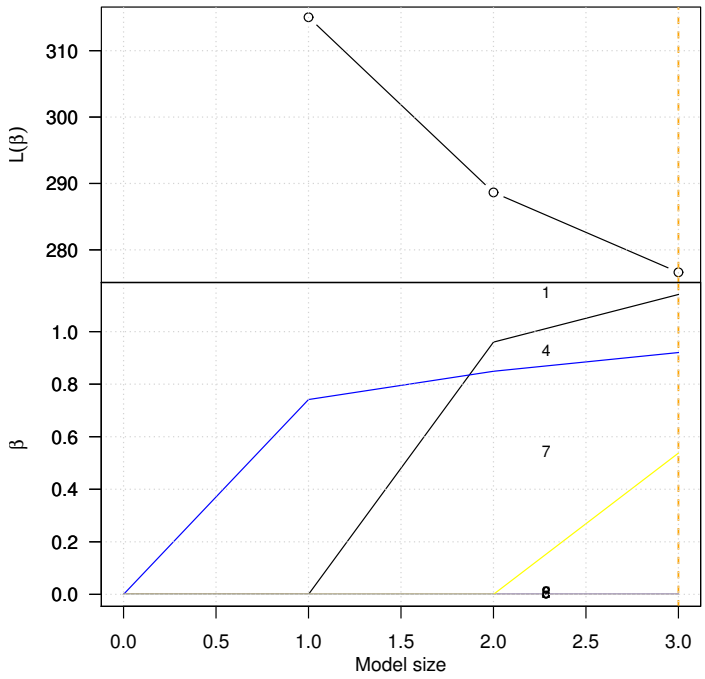

Figure 8: BeSS sequential solution paths for samples size 100, model 1, censoring ratio approximately 30%, model size on x-axis, coefficient value on lower y-axis. Each coefficient's curve show the coefficient's value for that model size. Likelihood value on upper y-axis. The likelihood curve show the likelihood value for that models size. The dotted vertical line indicates the model size given by the AIC criterion.

Figure 9 shows the LASSO cross-validation for model 1 in Table 1, where the y-axis provides the partial likelihood deviance as a function of the logarithm of lambda on the lower x-axis and model size on the upper x-axis, where lambda is the LASSO tuning parameter λ. The lower x-axis represents the logarithm of lambda, while the upper x-axis is the model size corresponding to the logarithm of lambda. The left vertical dotted line in Figure 9 shows what is called lambda.min, that is the lambda value that gives the minimum partial likelihood deviance. The right vertical dotted line display what is called lambda.1se, which gives the minimum partial likelihood deviation plus one standard error of the minimum partial likelihood deviation. Thus one can in a similar way as with the BeSS method select the size of the model and their respective coefficients.

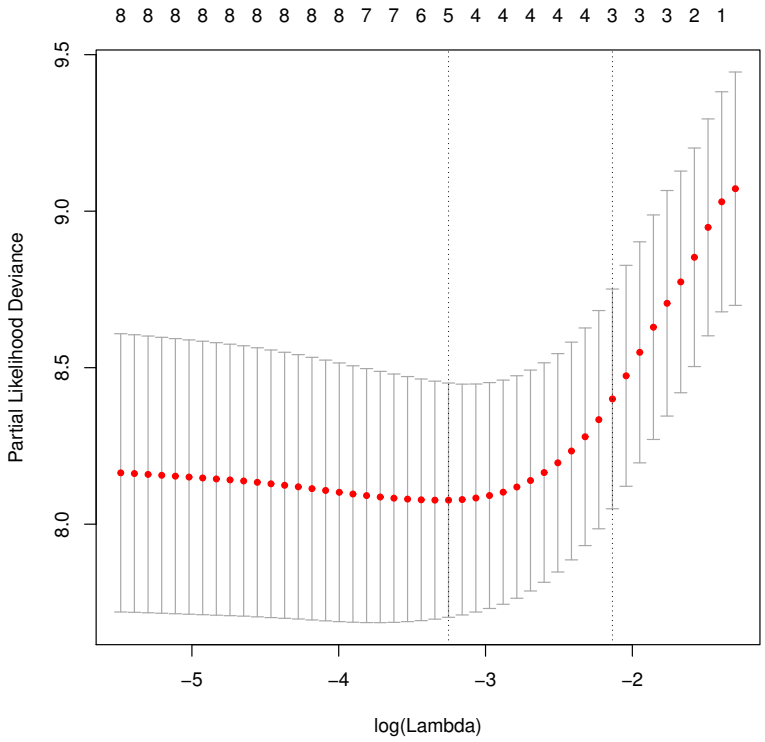

Figure 9: Lasso 3 fold cross-validation for sample size 100, model 1, censoring ratio approximately 30%, lower x-axis is log(Lambda), upper x-axis is numbers of estimated non-zero coefficients in model, partial likelihood deviance on y-Axis, dots represents mean of partial likelihood deviance, Vertical line ending with a short horizontal line indicates $\pm$ 1 standard error of partial likelihood deviance. Left dotted vertical line is lambda at minimum partial likelihood deviance, denoted lambda.min and right dotted vertical line is lambda at plus 1 standard deviation of partial likelihood deviance, denoted lambda.1se.

Figure 10 displays the solution paths diagram for LASSO. The y-axis gives the values of the estimated $\boldsymbol{\beta}$ coefficients as a function of the L_1-norm. The lower x-axis states the L_1-norm and the upper x-axis the model size, which corresponds to the L_1-norm on the lower x-axis. The coefficient's number of the model is at the right of each curve. Thus, by selecting the L_1-norm or model size one can determine the coefficients value. The different values of the L_1-norm is a consequence of letting the LASSO tuning parameter λ span over an interval of values, for example $\lambda \in [0, 1]$, see Equation (11) on Page 14.

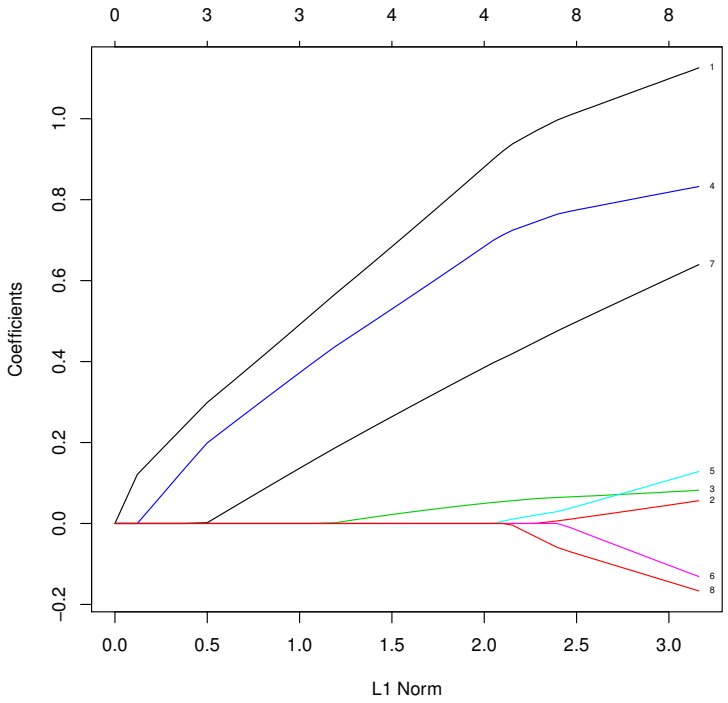

Figure 10: Lasso solution paths for samples size 100, model 1, censoring ratio approximately 30%, L_1-norm on x-axis, coefficient value on y-axis. Each curve show the particular coefficient's value for the corresponding L_1-norm value.

In Figures 8 to 10, we used the same model (model 1), same sample size, same censoring rate and sam data sets, but two different methods; The Bess method in Figure 8 and the LASSO in Figures 9 and 10. Summarizing the results in Figures 8 to 10, we conclude that at minimum AIC BeSS gives a model with three coefficients (1,4,7) and LASSO, by selecting the tuning parameter λ as the mean of lambda.min and lambda.1se, also returns a model with three coefficients (1,4,7). Normally LASSO seems to shrink the coefficients more than BeSS do at the same model size,resulting in a different prediction performance.

5.3 Median mean square errors and average number of correct and incorrect coefficients

Table 3 contains the median mean square error ($MMSE$), average number of correct zero and incorrect zero coefficients together with the fitting method, true model, model selection criteria, sample sizes, average censor rates and the execution times in seconds for each data set over the simulations for the different configurations. Let's focus on the rows 2 to 9, where sample size is 100 with approximately 30% censor rate for model 1 that have 5 zero and 3 non-zero coefficients.

Table 3: Partly mimics of examples in Fan & Li (2002), Zhang & Lu (2007). Showing average over 100 simulations. "Method" is fitting method. "Model" refers to model in Table 1. "Criterion" is the model selection criterion. MMSE is the median of mean squared error. "Correct $\beta = 0$" is average correct zero coefficients. "Incorrect $\beta \neq 0$" is average incorrect non-zero coefficients. "Censored" is average censored observations and "Execution time" is execution time per data set in seconds.

	Method-Model	Criterion	N	MMSE	Correct $\beta = 0$	Incorrect $\beta \neq 0$	Censored (%)	Execution time (s/dataset)
1	LASSO-1	lambda.bet	75	0.45	3.44	0.17	29.7	0.05
2	All subset-1	aic	100	0.15	3.37	0.01	31.6	1.24
3	Back elim-1	aic	100	0.32	4.66	0.41	31.6	0.13
4	Back elim-1	bic	100	0.40	4.92	0.85	31.6	0.14
5	BeSS.gs-1	aic	100	0.17	4.04	0.03	31.6	0.49
6	BeSS.gs-1	bic	100	0.08	4.84	0.08	31.6	0.52
7	BeSS.seq-1	aic	100	0.16	4.19	0.01	31.6	0.51
8	BeSS.seq-1	bic	100	0.09	4.81	0.05	31.6	0.51
9	LASSO-1	lambda.bet	100	0.35	3.55	0.03	31.6	0.08
10	LASSO-2	lambda.bet	100	0.23	4.99	0.00	24.5	0.07
11	LASSO-2	lambda.bet	100	0.26	5.02	0.00	39.8	0.07
12	LASSO-3	lambda.bet	200	0.09	5.34	0.43	24.2	0.15
13	LASSO-3	lambda.bet	200	0.10	5.51	0.53	40.0	0.15
14	LASSO-4	lambda.bet	50	0.24	5.28	0.88	0.0	0.04
15	LASSO-5	lambda.bet	50	0.03	5.32	2.66	0.0	0.04

The $MMSE$ quantity in Table 3 for rows 2 to 9 varies between 0.09 and 0.40 for the different methods. Furthermore we see that the BeSS sequential BIC method is best within this metric and the Backward eliminations and LASSO methods does not perform so well. The BeSS methods and All subset selection performs quite well.

Continuing with the average correct zero $\boldsymbol{\beta}$ coefficients metric in Table 3 for rows 2 to 9, we conclude that it spans over the interval [3.37, 4,92], where 5 is the theoretically correct value for model 1. The Backward eliminations and BeSS BIC methods are the best ones and the All subset selection and LASSO is the worst for this metric.

The average incorrect non-zero coefficients quantity in Table 3 for rows 2 to 9, it is in the range within [0.01, 0.85], where 0 is the theoretically correct value for all models in Table 1. The BeSS, All subset selection and LASSO methods perform almost equally well and note that Backward elimination methods perform rather bad. Putting these three metrics together we conclude in the following order that BeSS methods have overall best performance, then the All subset selection, LASSO and backwards eliminations methods. By increasing the LASSO tuning parameter λ towards lambda.1se or even more, the quantity of average correct zero coefficients improves, see Figure 9. This also shrinks the estimated $\boldsymbol{\beta}$ coefficients, see Figure 10. This ultimately increase the $MMSE$ quantity and may increase average incorrect non-zero coefficients. This also visualize the challenge with the variable selection problem, e.g. the trade-off between bias and variance discussed in Subsection 3.1 and Equation (7). Furthermore it envision the crucial selection of the LASSO tuning parameter λ. This is somewhat easier with the BeSS methods, were one can specify the number of non-zero $\boldsymbol{\beta}$ coefficients.

In Table 3 we have also displayed execution times. The LASSO is the fastest,e.g. 12 times faster than the slow All subset selection method. That the All subset selection method is slowest was expected because it evaluates all possible models. Backward elimination is almost as fast as LASSO. The BeSS methods are approximately twice as fast as the All subset selection method. If these relations will hold on high dimensional data is not determined by this study. What is clear is that All subset selection will always be slower due to the fact that is analyzes all possible models. The designers (Wen et al. 2017) of the BeSS methods claims that BeSS is faster for Cox proportional hazards model than LASSO, e.g. glmnet in this case, when data is high dimensional, e.g. very large p. If that holds then the BeSS is a contender to LASSO at least for Cox proportional model. The BeSS methods are somewhat simpler to use than LASSO because the one can select the model complexity, e.g. numbers of coefficients with out

having the LASSO hassle to determine the tuning parameter to reduce the model complexity. During the simulations we have observed that the values for LASSO varies from run to run even if the same dataset are used. This is caused by the cross-validation which randomize which observations that goes into each fold. The consequence are that the lambda.min and lambda.1se changes, which have impact on our LASSO tuning parameter λ and thus the model selections and shrinking of the estimated coefficients are different from run to run with the same data sets.

5.4 Means and standard error of the estimated coefficients

Table 4 displays the mean and the standard error of the estimated coefficients for the nonzero coefficients in all investigated models and are for each model based on 100 simulations. The Values for all coefficients are available in the Table 6 in the Appendix C. The means and standard errors are calculated according to Equations (35) and (36) respectively. Focusing on the rows 2 to 9 in Table 4, e.g. were the true model is 1, the sample size is 100 and the censoring rate is approximately 30% applied to the 8 different methods.The All subset selection and the BeSS methods do quite well estimating β_1, β_4 and β_7. The same is valid for standard error. This also holds for the zero coefficients. Backward elimination methods slightly underestimates β_1, β_7 and β_4: This is also reflected in the $MMSE$ metric. LASSO on row 9 shrinks the coefficients even more compared to Backward elimination but produces a better $MMSE$.

Table 4: Mean and standard error of the non-zero parameters for each method and model, showing average over 100 simulations, "Method" is fitting method, "Model" refers to model in Table 1.

	Method-	β_1		β_2		β_4		β_6		β_7	
	model	$\bar{\hat{\beta}}$	$\hat{\beta}_{Se}$	$\bar{\hat{\beta}}$	$\hat{\beta}_{Se}$	$\bar{\hat{\beta}}$	$\hat{\beta}_{Se}$	$\bar{\hat{\beta}}$	$\hat{\beta}_{Se}$	$\bar{\hat{\beta}}$	$\hat{\beta}_{Se}$
1	LASSO-1	0.45	0.41	0.02	0.07	0.61	0.47	0.02	0.08	0.27	0.38
2	All subset-1	0.86	0.20	-0.00	0.15	1.05	0.24	-0.01	0.13	0.63	0.21
3	Back elim-1	0.74	0.29	-0.00	0.08	0.97	0.28	-0.01	0.09	0.45	0.36
4	Back elim-1	0.64	0.36	0.00	0.03	0.91	0.30	0.01	0.06	0.24	0.49
5	BeSS.gs-1	0.85	0.19	0.00	0.16	1.05	0.24	0.00	0.14	0.63	0.22
6	BeSS.gs-1	0.83	0.19	0.00	0.07	1.03	0.21	0.00	0.05	0.60	0.23
7	BeSS.seq-1	0.86	0.20	-0.01	0.15	1.05	0.24	-0.01	0.13	0.63	0.21
8	BeSS.seq-1	0.84	0.18	0.01	0.10	1.04	0.21	0.00	0.08	0.61	0.22
9	LASSO-1	0.50	0.34	0.03	0.07	0.66	0.38	0.02	0.06	0.33	0.32
10	LASSO-2	-0.49	0.26	-0.48	0.26	-0.01	0.03	-0.42	0.31	-0.02	0.06
11	LASSO-2	-0.47	0.28	-0.47	0.29	-0.01	0.03	-0.38	0.36	-0.01	0.04
12	LASSO-3	-0.26	0.17	-0.26	0.18	-0.00	0.01	-0.05	0.16	-0.01	0.02
13	LASSO-3	-0.24	0.19	-0.25	0.19	-0.00	0.02	-0.03	0.18	-0.00	0.01
14	LASSO-4	-0.13	0.25	-0.11	0.27	-0.00	0.09	-0.11	0.28	-0.01	0.02
15	LASSO-5	0.01	0.11	0.00	0.10	0.01	0.05	0.01	0.10	0.01	0.05

The interpretation of the Table 6 is much easier to grasp by using box-plots in the Figures 11, 12 and 13 which corresponds to rows 2, 6 and 9 in the Table 6. The Appendix B displays the box-plots for all rows in Table 6. Note that the standard errors in the box-plots are based on the estimated coefficients' mean, $\bar{\hat{\beta}}$, which is different from $\hat{\beta}_{Se}$ in Table 6. The plots in Figures 11 to 13 shows all coefficients true values along the x-axis. The estimated $\boldsymbol{\beta}$ coefficient's means are just over the x-axis. The small dots indicates the locations of the means, which gives a good visual view of the estimations. The larger dots indicates outliers. For example Figure 13 show very clearly how much the smaller the coefficients are compared to the true model, which is quite the opposite in the Figures 11 and 12.

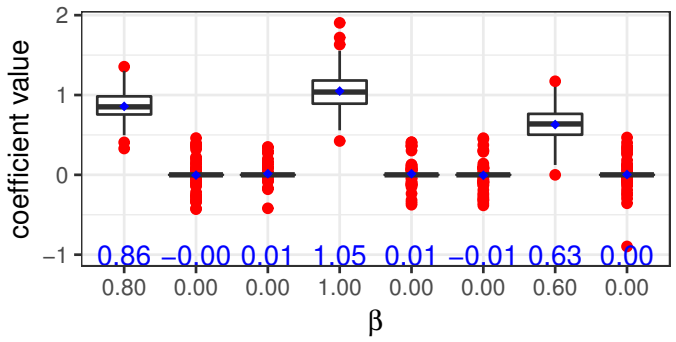

Figure 11: Box plot of $\boldsymbol{\beta}$ over 100 simulations, method All subset selection, true model 1, criterion is AIC, censoring 30% and larger dots are outliers. Smaller dots are estimated means of the $\boldsymbol{\beta}$ coefficients-

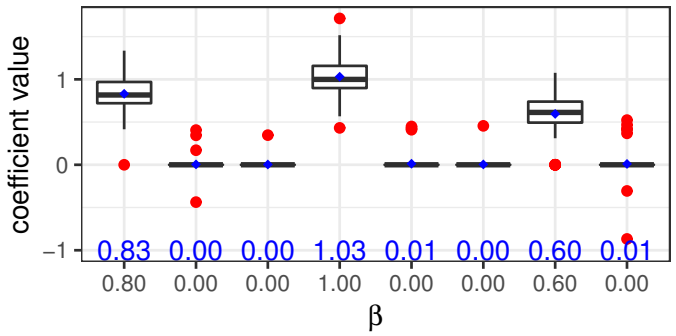

Figure 12: Box plot of $\bar{\hat{\boldsymbol{\beta}}}$ over 100 simulations, method BeSS.gs, true model 1, criterion is BIC, censoring 30% and larger dots are outliers. Smaller dots are estimated means of the $\boldsymbol{\beta}$ coefficients-

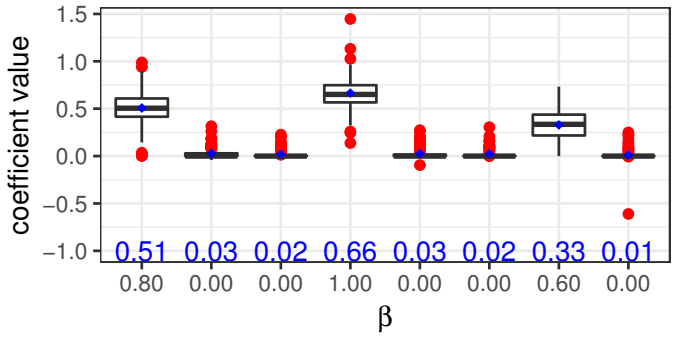

Figure 13: Box plot of $\bar{\hat{\boldsymbol{\beta}}}$ over 100 simulations, method LASSO, true model 1, criterion is λ, e.g. the mean of lambda.min and lambda.1se, censoring 30% and dlarger dots are outliers. Smaller dots are estimated means of the $\boldsymbol{\beta}$ coefficients-

5.5 Comparison of zero, non-zero coefficients and MMSE results versus earlier papers

Table 5 reports the data that is to be compared against Tibshirani (1997), Fan & Li (2002) and Zhang & Lu (2007) articles. A larger version of this table is available in Table 7 in Appendix D. In this section comparison is organized in the following order, first rows 14 and 15 versus Tibshirani (1997) then row 1 and 9 towards Fan & Li (2002) findings, then rows 10 to 13 against Zhang & Lu (2007) and lastly rows 2 to 8 which uses the same model 1 and data set as Fan & Li (2002).

Tibshirani (1997) only reports the average number of coefficients set to zero and that value is larger than the number of coefficients that are zero in the true models, e.g. 4 and 5 in Table 1 on Page 25. So our reported value is the sum of the average number of the correct coefficient set to zero and the incorrect coefficients set to zero. Looking in Table 5 at row 14 average number of zero coefficients deviates with 8% and the $MMSE$ departs with just under 7%. If we instead look at row 15 in Table 5 average number of zero coefficients is off by 2.3% Regarding our $MMSE$ is much better than Tibshirani's (1997), but the relative error is 80%. This last result is surprising because looking at the estimated non-zero coefficients in Table 6 at row 15 they are quite small compared to the true model 5 in Table 1. The selected tuning parameter λ plays a big role in this comparison. Models 4 and 5 in Table 1 and data sets do not have any censoring so censoring can be ruled out as source of error. But if our interpretation and implementation are correct our findings are in line or even better than Tibshirani's (1997).

Row 1 in Table 5 corresponds to the Fan & Li model (2002, Table 1, p. 9) with 75 observations. The average correct zero coefficients deviates with 15% and the average incorrect have a relative difference of 70 %, which is quite much. On the other hand the $MMSE$ is within 1% which can be considered as good. Comparing row 9 in Table 5. The average correct zero coefficients deviates with 11% and the average incorrect have a relative difference of 50 %, but the absolute incorrect values are small e.g. 0.03 and 0.02. The $MMSE$ relative difference is almost 23%, but our $MMSE$ is better than Fan & Li (2002) $MMSE$. Hence our results does not math well to the corresponding results in Fan & Li (2002).

Table 5: Comparing average number of correct and incorrect coefficients and median of MSE over 100 simulations (Tibshirani 50 simulations). First two columns is the same as in Tables 3 and 6. Column 3 is our average number of correct coefficients, e.g. $\beta = 0$, column 4 is value to compare against column 3. Column 5 is the absolute relative error in percent between columns 3 and 4. Column 6 is our average number of incorrect coefficients, e.g. $\beta \neq 0$, column 7 is value to compare against column 7. Column 8 is the absolute relative error in percent between columns 6 and 7. Column 8 is our estimated $MMSE$, column 9 is value to compare against column 8. Column 10 is the absolute relative error in percent between columns 8 and 9. Column 11 is the source of the values to compare against, e.g. F=Fan & Li (2002), Z=Zhang & Lu (2007) and T=Tibshirani (1997). Empty cells indicate indicate that value compared against is zero or missing.

	Method-Model	Cor. $\beta = 0$	Cmp. cor. $\beta = 0$	Rel. error (%)	Incor. $\beta \neq 0$	Cmp. incor. $\beta \neq 0$	Rel. error (%)	MMSE	CMMSE	Rel. error (%)	Ref to
1	LASSO-1	3.44	4.05	15.06	0.17	0.10	70.00	0.45	0.46	1.00	F
2	All subset-1	3.37	3.99	15.54	0.01	0.02	50.00	0.15	0.46	66.22	F
3	Back elim-1	4.66	3.99	16.79	0.41	0.02	1950.00	0.32	0.46	30.50	F
4	Back elim-1	4.92	3.99	23.31	0.85	0.02	4150.00	0.40	0.46	12.51	F
5	BeSS.gs-1	4.04	3.99	1.25	0.03	0.02	50.00	0.17	0.46	63.16	F
6	BeSS.gs-1	4.84	3.99	21.30	0.08	0.02	300.00	0.08	0.46	82.82	F
7	BeSS.seq-1	4.19	3.99	5.01	0.01	0.02	50.00	0.16	0.46	64.60	F
8	BeSS.seq-1	4.81	3.99	20.55	0.05	0.02	150.00	0.09	0.46	80.96	F
9	LASSO-1	3.55	3.99	11.03	0.03	0.02	50.00	0.35	0.46	22.78	F
10	LASSO-2	4.99	4.87	2.46	0.00	0.00		0.23	0.19	20.29	Z
11	LASSO-2	5.02	4.67	7.49	0.00	0.00		0.26	0.20	30.54	Z
12	LASSO-3	5.34	5.43	1.66	0.43	0.08	437.50	0.09	0.08	7.89	Z
13	LASSO-3	5.51	5.32	3.57	0.53	0.08	562.50	0.10	0.43	76.71	Z
14	LASSO-4	6.16	6.70	8.06				0.24	0.26	6.89	T
15	LASSO-5	7.98	7.80	2.31				0.03	0.15	80.00	T

Now comparing our outcome with the corresponding ones in Zhang & Lu (2007) on the rows 10 to 13 in Table 5. Recall that models 2 and 3 in Table 1 on Page 25 have 6 zero coefficients and 3 non-zero coefficients. Our average number of correct zero coefficients results are slightly better or in line with Zhang & Lu (2007) findings. Regarding average number of incorrect

coefficients shows approximately the same results for row 10 and 11 in Table 5, but row 12 and 13 shows very large relative error, e.g. 437% and 562%, respectively. Our $MMSE$ results for row 10 and 11 are slightly larger than Zhang & Lu's (2007) outcomes. On the other hand our $MMSE$ results on row 12 is in line with Zhang & Lu's (2007) and the result on row 13 is slightly better. This difference between our results and Zhang & Lu's (2007) is probably caused by different LASSO tuning parameter or censoring in our simulations versus Zhang & Lu's (2007) simulations.

Now going back to row 2 to 8 in Table 5, we see that backwards elimination stands out concerning average incorrect coefficients set to zero, but the absolute $MMSE$ is better than Fan & Li's (2002) reported result. This also holds for all subset selection and even more so for best subset selection methods methods.

Five of 8 of our applied LASSO methods have better $MMSE$ values when compared to earlier papers (Tibshirani 1997, Fan & Li 2002, Zhang & Lu 2007). Even that our results partly shows better result than earlier papers the spread in the results are worrying. There could be a number of reasons why, such as censoring, different cross-validation, yielding another LASSO tuning parameter, miss interpretation of the $\boldsymbol{\rho}, \boldsymbol{\Sigma}, MSE$ and $MMSE$ or mistakes in our implementation.

6 Conclusion

In this thesis our LASSO results does not fully agree with the results reported by Tibshirani (1997), Fan & Li (2002) and Zhang & Lu (2007). Our added methods such as All subset selection, the BeSS methods and Backward elimination performs rather well regarding estimating average number of correct $\boldsymbol{\beta}$ coefficients set to zero and $MMSE$, but the spread of the estimated average number of incorrect $\boldsymbol{\beta}$ coefficients is a warning sign that the simulations are not perfect. This spread is also valid for our applied LASSO methods. There are several possibilities why the number differs. For example, the correlation for population or sample, the censoring scheme, the cross-validation and the selection of the LASSO tuning parameter λ, estimation of MSE and $MMSE$ and miss interpretation of the metrics versus Tibshirani (1997), Fan & Li (2002) and Zhang & Lu (2007) metrics and of course errors in the implementation.

Our correlation interpretation may be wrong compared to the papers by Tibshirani (1997), Fan & Li (2002) and Zhang & Lu (2007), but as the methods uses identically data sets when the methods', model, censoring rate, sample size and number of simulations are the same, this systematic error ought to have equal impact on all the methods. However this will have an impact when results are compared to results in earlier papers.

The error caused by our different parameters in censoring schemes compared to the articles by Fan & Li (2002) and Zhang & Lu (2007) seems to have different systematic impact on the different methods with the same data sets. Especially LASSO because it probably have an impact on the cross-validation and thus the selected LASSO tuning parameter.

Our cross-validation is different from the general cross-validations used by Tibshirani (1997), Fan & Li (2002) and Zhang & Lu (2007), thus our selection of tuning parameter λ is the one thing that likely matters most, because the other methods seems to give better estimates of the coefficients versus the true model and the tuning parameter λ is the thing that is different from the other methods. The problem with this is if the tuning parameter is changed to give more coefficients set to zero it also shrinks the non-zero coefficients more, thus either average number of correct zero and non correct zero coefficients changes accordingly but the coefficients' values will depart more from the true model and this increase our $MMSE$ metric.

Our estimation of $MMSE$ may be wrong but this would have the same impact on all methods, because it is the same computation, only the estimated $\boldsymbol{\beta}$ coefficients differ. Even if the Σ is wrong it would create the same type of systematic error on all methods. This as previously

mentioned would make the comparison versus Tibshirani (1997), Fan & Li (2002) and Zhang & Lu (2007) results completely wrong because Fan & Li (2002), Zhang & Lu (2007) both refers to Tibshirani's (1997) Equation (29) for mean square errors, but Fan & Li (2002) seems to modify Tibshirani's equation to be a relative metric. Consequently we may have miss interpreted this.

It is of course possible that we have made implementation errors but apart from the LASSO tuning parameter the implementation does not differ very much compared to the other methods that provides better results.

There are several possibilities to potentially improve the the results. For example, firstly, do the same cross-validation to select the tuning parameter λ as Tibshirani (1997), Fan & Li (2002) and Zhang & Lu (2007). Secondly, improve the interpretation and implementation of $MMSE$. Thirdly, use Fan & Li's (2002) and Zhang & Lu's (2007) censoring scheme and lastly, verify the implementation, e.g. do the calculation mentioned above Table 2 in Section 4 to verify that our different parameters are valid.

We have not been able to reproduce the LASSO results reported by Tibshirani (1997), Fan & Li (2002) and Zhang & Lu (2007), although 5 of 8 of our applied LASSO methods have better $MMSE$ values compared to Tibshirani (1997), Fan & Li (2002) and Zhang & Lu (2007). However our added methods, all subset selection, backward elimination and the BeSS methods performs rather well although they are slower than LASSO for all used data sets and data set sizes in this study.

References

Akaike, H. (1974), 'A new look at the statistical model identification', *Annals of the Institute of Statistical Mathematics* **19**(6).

Allison, P. D., (e-book collection), B. & Books24x7, I. (2010), *Survival Analysis Using SAS: A Practical Guide, Second Edition*, 2nd;2; edn, SAS Press [Imprint], Cary.

Bender, R., Augustin, T. & Blettner, M. (2006), 'Generating survival times to simulate cox proportional hazards models by ralf bender, thomas augustin and maria blettner, statistics in medicine 2005; 24 :17131723', *Statistics in Medicine* **25**(11), 1978–1979.

Boyd, S. P. (2004), *Convex optimization*, Cambridge University Press, Cambridge.

Breiman, L. (1993), 'Better subset regression using the non-negative garrote'. This ariticle was unpublised 1993, it was publsihed (1995), see acknowledgements in Tibshirani (1996).

Breiman, L. (1995), 'Better subset regression using the nonnegative garrote', *Technometrics* **37**(4), 373–384.

Calcagno, V. & Mazancourt, C. d. (2010), 'glmulti : An r package for easy automated model selection with (generalized) linear models', *Journal of Statistical Software* **34**(12).

Casella, G., Robert, C. P. & Wells, M. T. (2004), 'Mixture models, latent variables and partitioned importance sampling', *Statistical Methodology* **1**(1), 1–18.

Chen, M.-H., Ibrahim, J. G. & Shao, Q.-M. (2009), 'Maximum likelihood inference for the cox regression model with applications to missing covariates', *Journal of Multivariate Analysis* **100**(9), 2018–2030.

Cox, D. (1972), 'Regression models and life-tables (with discussion)', *Journal of the Royal Statistical Society, Series B, Methodological* **34**, 187.

Cox, D. (1975), 'Partial likelihood', *Biometrika* **62**(2), 269–276.

Craven, P. & Wahba, G. (1979), 'Smoothing noisy data with spline functions', *Numerische Mathematik* **31**(4), 377–403.

Fan, J. & Li, R. (2002), 'Variable selection for cox's proportional hazards model and frailty model', *Ann. Statist.* **30**(1), 74–99.

Furnival, G. M. & Wilson, R. W. (1974), 'Regressions by leaps and bounds', *Technometrics* **16**(4), 499–511.

Hastie, T., Tibshirani, R., Wainwright, M. & (e-book collection), C. (2015), *Statistical learning with sparsity: the lasso and generalizations*, Vol. 143, 1 edn, CRC Press, Boca Raton.

James, G., Witten, D., Hastie, T., Tibshirani, R., service), S. O. & (e-book collection), S. (2014), *An Introduction to Statistical Learning: with Applications in R*, Vol. 103, Springer New York, New York, NY.

Jörnsten, R. (2017), 'Msg500/mve190 linear models -lecture 6', Online. Accessed 2018-01-15. **URL:** *http://www.math.chalmers.se/Stat/Grundutb/GU/MSG500/A17/Lecture6.pdf*

Kleinbaum, D. G. & Klein, M. (2012), *Parametric Survival Models*, Springer New York, New York, NY, pp. 289–361.

Kumar, D. & Westberg, U. (1996), 'Proportional hazards modeling of time-dependent covariates using linear regression: a case study [mine power cable reliability]', *IEEE Transactions on Reliability* **45**(3), 386–392.

Mantel, N. (1970), 'Why stepdown procedures in variable selection', *Technometrics* **12**(3), 621–625.

Marsaglia, G. & Tsang, W. (2000), 'The ziggurat method for generating random variables', *Journal of Statistical Software* **5**, 1–7.

Nie, G., Rowe, W., Zhang, L., Tian, Y. & Shi, Y. (2011), 'Credit card churn forecasting by logistic regression and decision tree', *Expert Systems with Applications* **38**(12), 15273–15285.

Rice, J. A. (2007), *Mathematical statistics and data analysis*, Duxbury advanced series, 3. ed. edn, Thomson Brooks/Cole, Belmont, Calif.

Schwarz, G. (1978), 'Estimating the dimension of a model', *The Annals of Statistics* **6**(2), 461–464.

Stare, J., Harrell, F. & Heinzl, H. (2001), 'Bj: an s-plus program to fit linear regression models to censored data using the buckley-james method', *Computer Methods And Programs In Biomedicine* **64**(1), 45–52.

Tibshirani, R. (1996), 'Regression shrinkage and selection via the lasso', *Journal of the Royal Statistical Society. Series B (Methodological)* **58**(1), 267–288.

Tibshirani, R. (1997), 'The lasso method for variable selection in the cox model', *Statistics in Medicine* **16**(4), 385–395.

Van Den Poel, D. & Larivire, B. (2004), 'Customer attrition analysis for financial services using proportional hazard models', *European Journal of Operational Research* **157**(1), 196–217.

Volinsky, C. T. & Raftery, A. E. (2000), 'Bayesian information criterion for censored survival models', *Biometrics* **56**(1), 256–262.

Wackerly, M. & Scheaffer (2008), *Mathematical statistics with applications*, 7. ed., international student ed.. edn, Southbank : Thomson Learning, Southbank.

Wahba, G. (1985), 'A comparison of gcv and gml for choosing the smoothing parameter in the generalized spline smoothing problem', *The Annals of Statistics* **13**(4), 1378–1402.

Wen, C., Zhang, A., Quan, S. & Wang, X. (2017), 'Bess: An r package for best subset selection in linear, logistic and coxph models', *arXiv preprint arXiv:1709.06254* . Forthcoming.

Xu, R., Vaida, F. & Harrington, D. P. (2009), 'Using profile likelihood for semiparametric model selection with application to proportional hazards mixed models', *Statistica Sinica* **19**(2), 819–842.

Zhang, H. H. & Lu, W. (2007), 'Adaptive lasso for cox's proportional hazards model', *Biometrika* **94**(3), 691–703.

Appendices

The following Sections contains additional histograms of covariates, box plots of the estimated coefficients and enlarged tables, e.g. of Tables 4 and 5.

A Histogram covariates

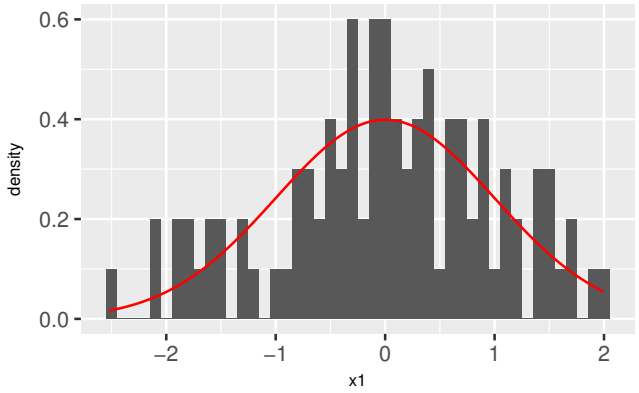

Figure 14: Histogram of simulated covariate X1, true model 1, normalized

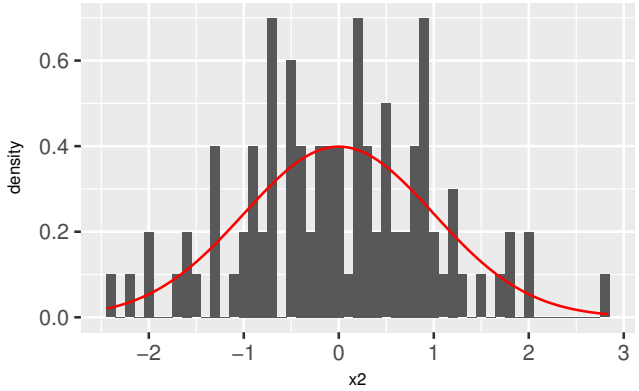

Figure 15: Histogram of simulated covariate X2, true model 1, normalized

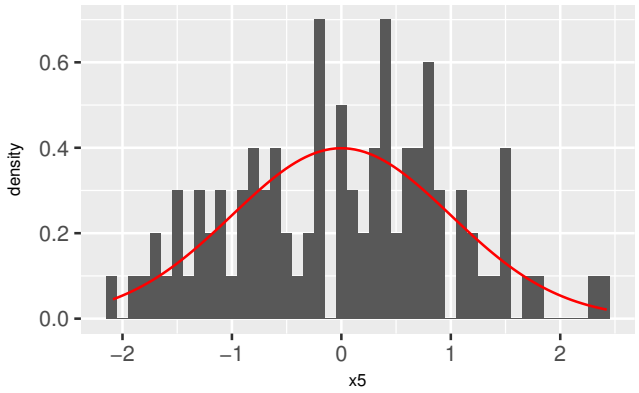

Figure 17: Histogram of simulated covariate X5, true model 1, normalized

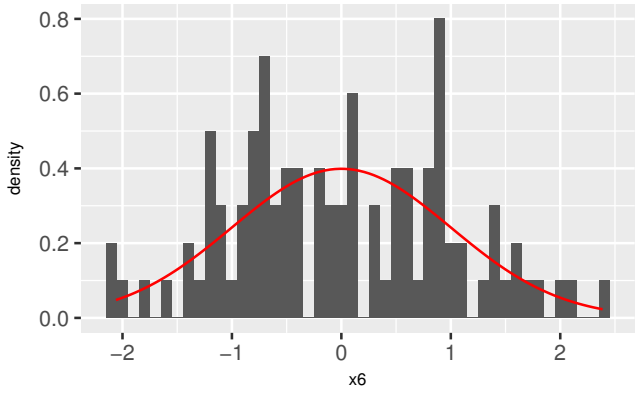

Figure 18: Histogram of simulated covariate X6, true model 1, normalized

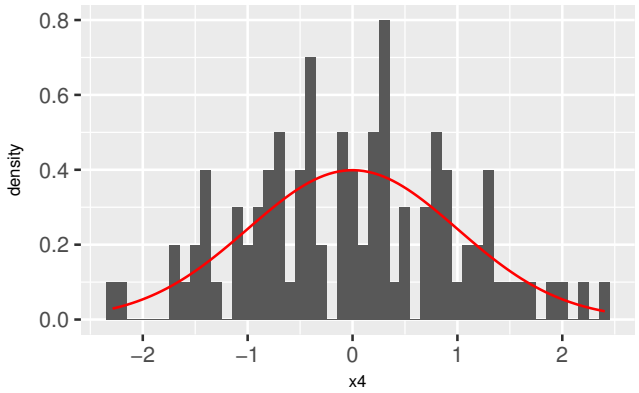

Figure 16: Histogram of simulated covariate X4, true model 1, normalized

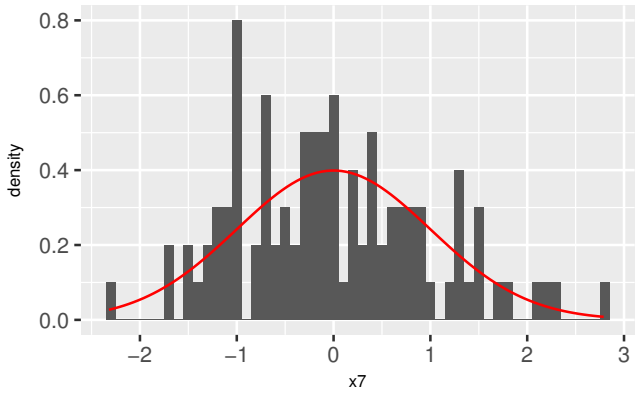

Figure 19: Histogram of simulated covariate X7, true model 1, normalized

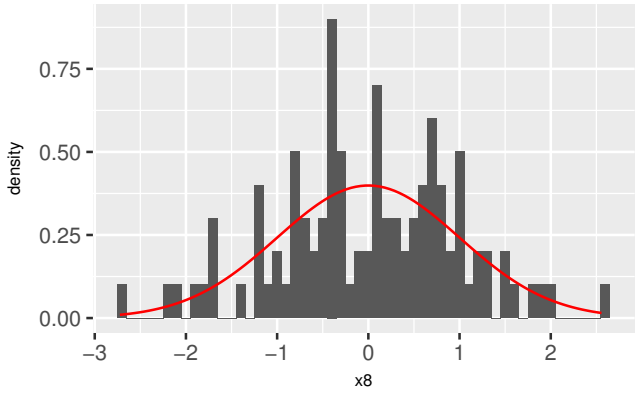

Figure 20: Histogram of simulated covariate X8, true model 1, normalized

B Boxplots of estimated coefficients

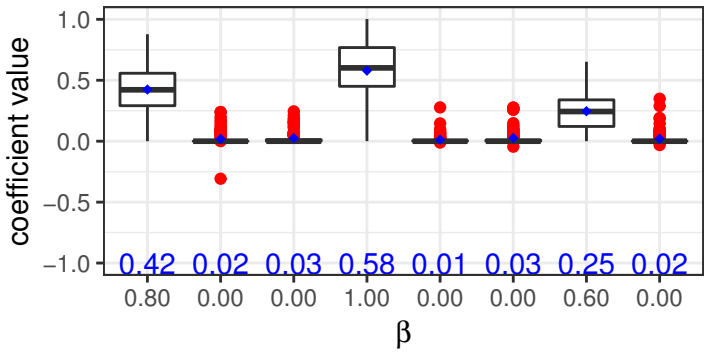

Figure 21: Box plot of $\bar{\hat{\boldsymbol{\beta}}}$ over 100 simulations, sample size 75, method LASSO, true model 1, criterion is λ, censoring 30% and dots (red)are outliers. The x-axis holds the true model coefficients. The small dot (blue) marks the mean of the estimated coefficient. The blue text just over the x-axis is the mean of the estimated coefficient.

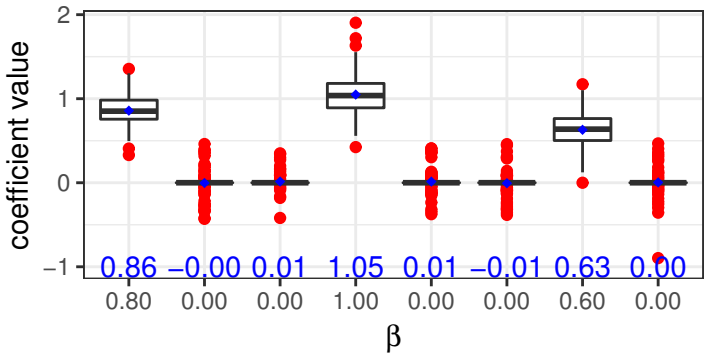

Figure 22: Box plot of $\boldsymbol{\beta}$ over 100 simulations, method All subset selection, true model 1, criterion is AIC, censoring 30% and dots (red)are outliers. The x-axis holds the true model coefficients. The small dot (blue) marks the mean of the estimated coefficient. The blue text just over the x-axis is the mean of the estimated coefficient.

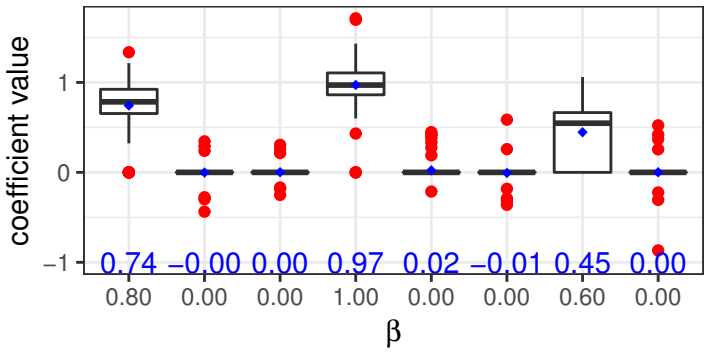

Figure 23: Box plot of $\bar{\hat{\beta}}$ over 100 simulations, method Backward elimination, true model 1, criterion is AIC,censoring 30% and dots (red)are outliers. The x-axis holds the true model coefficients. The small dot (blue) marks the mean of the estimated coefficient. The blue text just over the x-axis is the mean of the estimated coefficient.

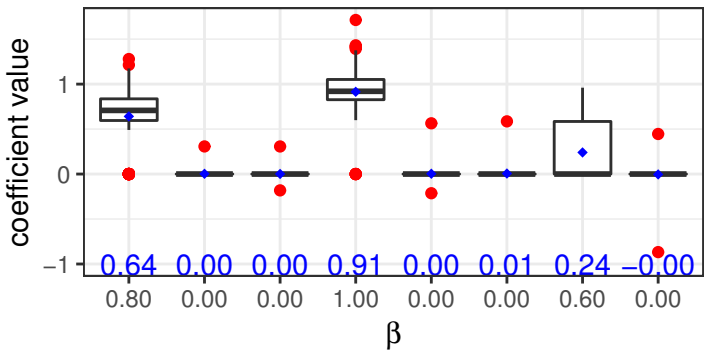

Figure 24: Box plot of $\bar{\hat{\beta}}$ over 100 simulations, method Backward elimination, true model 1, criterion is BIC, censoring 30% and dots (red)are outliers. The x-axis holds the true model coefficients. The small dot (blue) marks the mean of the estimated coefficient. The blue text just over the x-axis is the mean of the estimated coefficient.

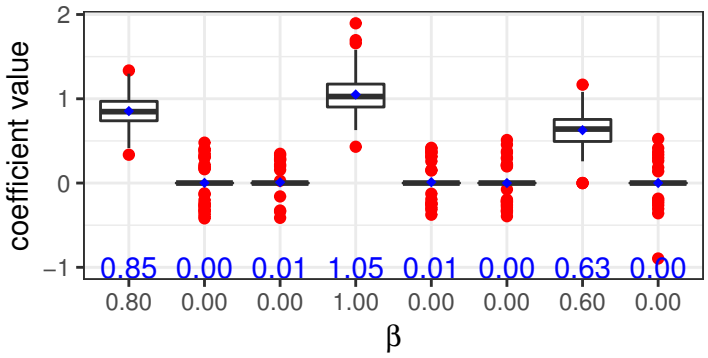

Figure 25: Box plot of $\bar{\hat{\boldsymbol{\beta}}}$ over 100 simulations, method BeSS.gs, true model 1, criterion is AIC, censoring 30% and dots (red)are outliers. The x-axis holds the true model coefficients. The small dot (blue) marks the mean of the estimated coefficient. The blue text just over the x-axis is the mean of the estimated coefficient.

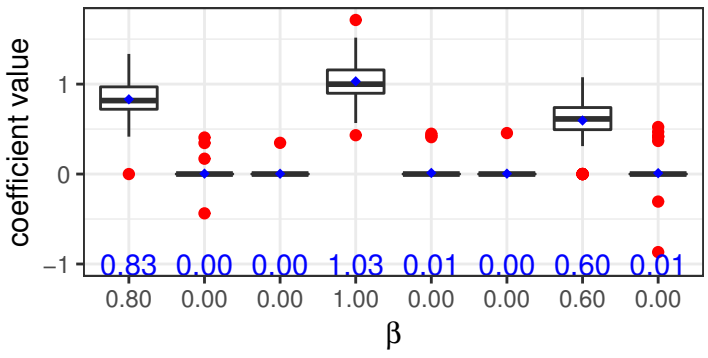

Figure 26: Box plot of $\bar{\hat{\boldsymbol{\beta}}}$ over 100 simulations, method BeSS.gs, true model 1, criterion is BIC, censoring 30% and dots (red)are outliers. The x-axis holds the true model coefficients. The small dot (blue) marks the mean of the estimated coefficient. The blue text just over the x-axis is the mean of the estimated coefficient.

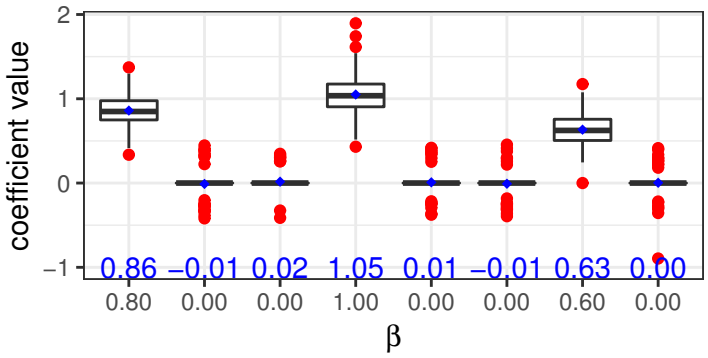

Figure 27: Box plot of $\bar{\hat{\boldsymbol{\beta}}}$ over 100 simulations, method BeSS.seq, true model 1, criterion is AIC, censoring 30% and dots (red)are outliers. The x-axis holds the true model coefficients. The small dot (blue) marks the mean of the estimated coefficient. The blue text just over the x-axis is the mean of the estimated coefficient.

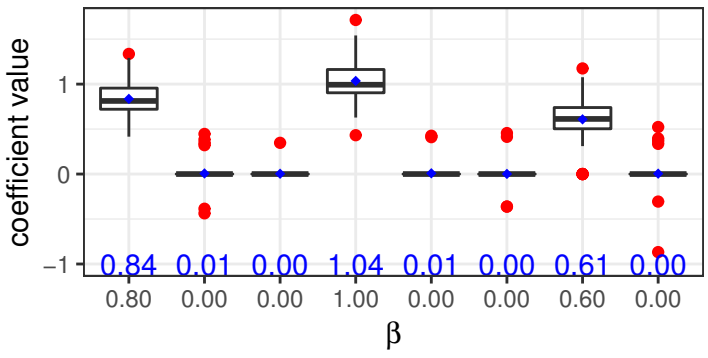

Figure 28: Box plot of $\bar{\hat{\boldsymbol{\beta}}}$ over 100 simulations, method BeSS.seq, true model 1, criterion is BIC, censoring 30% and dots (red)are outliers. The x-axis holds the true model coefficients. The small dot (blue) marks the mean of the estimated coefficient. The blue text just over the x-axis is the mean of the estimated coefficient.

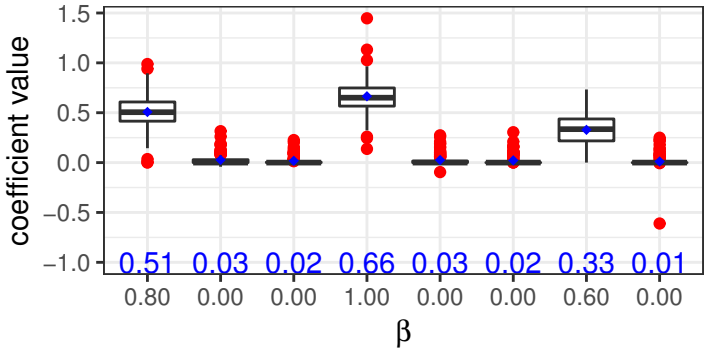

Figure 29: Box plot of $\bar{\hat{\boldsymbol{\beta}}}$ over 100 simulations, method LASSO, true model 1, criterion is λ, censoring 30% and dots (red)are outliers. The x-axis holds the true model coefficients. The small dot (blue) marks the mean of the estimated coefficient. The blue text just over the x-axis is the mean of the estimated coefficient.

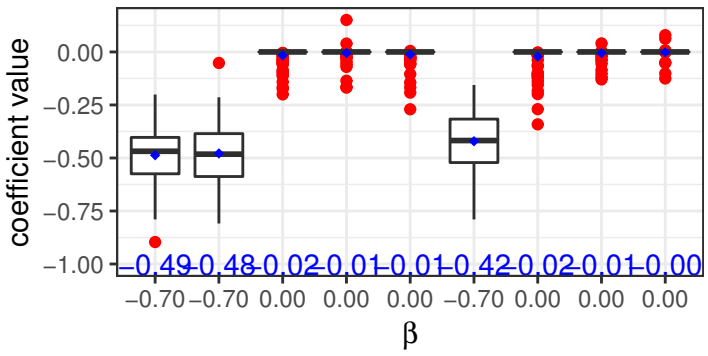

Figure 30: Box plot of $\bar{\hat{\boldsymbol{\beta}}}$ over 100 simulations, method LASSO, true model 2, criterion is λ, censoring 25% and dots (red)are outliers. The x-axis holds the true model coefficients. The small dot (blue) marks the mean of the estimated coefficient. The blue text just over the x-axis is the mean of the estimated coefficient.

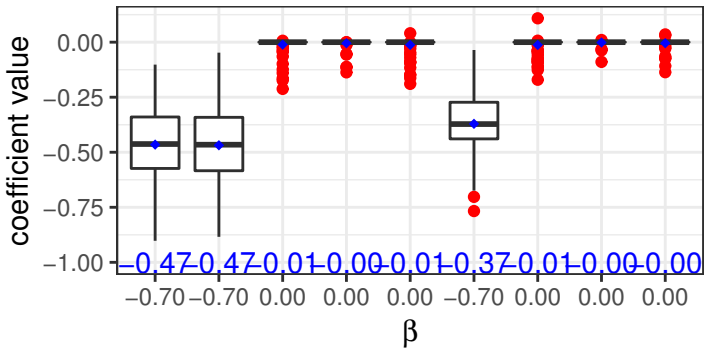

Figure 31: Box plot of $\bar{\hat{\boldsymbol{\beta}}}$ over 100 simulations, method LASSO, true model 2, criterion is λ, censoring 40% and dots (red)are outliers. The x-axis holds the true model coefficients. The small dot (blue) marks the mean of the estimated coefficient. The blue text just over the x-axis is the mean of the estimated coefficient.

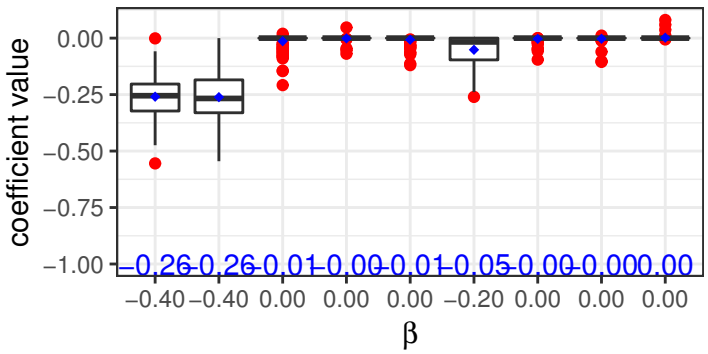

Figure 32: Box plot of $\bar{\hat{\boldsymbol{\beta}}}$ over 100 simulations, sample size 200, method LASSO, true model 3, criterion is λ, censoring 25% and dots (red)are outliers. The x-axis holds the true model coefficients. The small dot (blue) marks the mean of the estimated coefficient. The blue text just over the x-axis is the mean of the estimated coefficient.

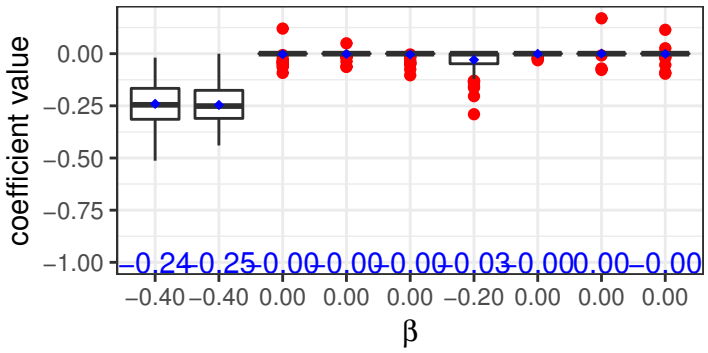

Figure 33: Box plot of $\bar{\hat{\beta}}$ over 100 simulations, sample size 200, method LASSO, true model 3, criterion is λ, censoring 40% and dots (red)are outliers. The x-axis holds the true model coefficients. The small dot (blue) marks the mean of the estimated coefficient. The blue text just over the x-axis is the mean of the estimated coefficient.

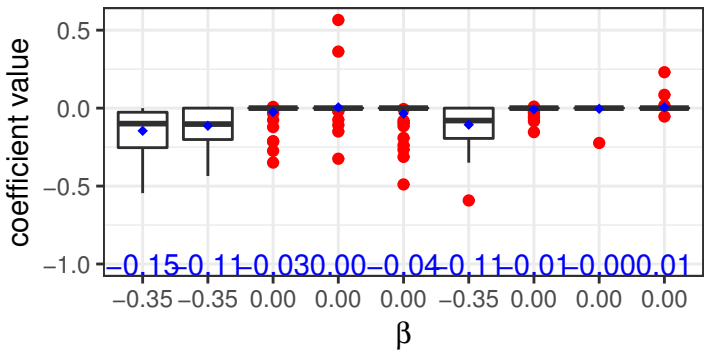

Figure 34: Box plot of $\bar{\hat{\beta}}$ over 50 simulations, sample size 50, method LASSO, true model 4, criterion is λ, censoring 0% dots (red)are outliers. The x-axis holds the true model coefficients. The small dot (blue) marks the mean of the estimated coefficient. The blue text just over the x-axis is the mean of the estimated coefficient.

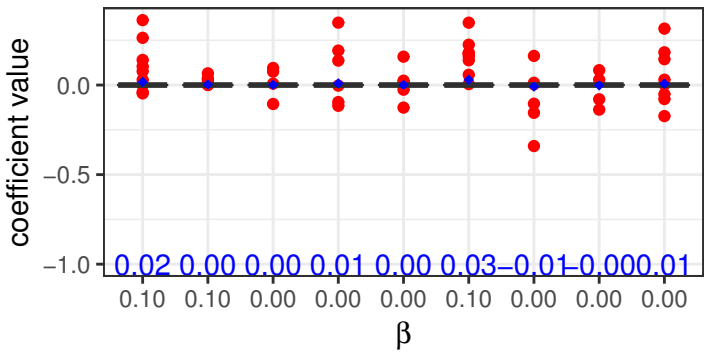

Figure 35: Box plot of $\bar{\hat{\beta}}$ over 50 simulations, sample size 50, method LASSO, true model 5, criterion is λ, censoring 0% dots (red)are outliers. The x-axis holds the true model coefficients. The small dot (blue) marks the mean of the estimated coefficient. The blue text just over the x-axis is the mean of the estimated coefficient.

C Enlarged table, means and standard error of the estimated coefficients

Table 6: Mean and standard error of the parameters for each method and model, Showing average over 100 simulations, Method is fitting method, Model refers to model in Table 1 and the means.

	Method-	β_1		β_2		β_3		β_4		β_5		β_6		β_7		β_8		β_9	
	model	$\bar{\hat{\beta}}$	$\hat{\beta}_{Se}$	$\bar{\hat{\beta}}$	$\hat{\beta}_{Se}$	$\bar{\hat{\beta}}$	$\hat{\beta}_{Se}$	$\bar{\hat{\beta}}$	$\hat{\beta}_{Se}$	$\bar{\hat{\beta}}$	$\hat{\beta}_{Se}$	$\bar{\hat{\beta}}$	$\hat{\beta}_{Se}$	$\bar{\hat{\beta}}$	$\hat{\beta}_{Se}$	$\bar{\hat{\beta}}$	$\hat{\beta}_{Se}$	$\bar{\hat{\beta}}$	$\hat{\beta}_{Se}$
1	LASSO-1	0.45	0.41	0.02	0.07	0.03	0.06	0.61	0.47	0.02	0.05	0.02	0.08	0.27	0.38	0.02	0.06		
2	All subset-1	0.86	0.20	-0.00	0.15	0.01	0.09	1.05	0.24	0.01	0.13	-0.01	0.13	0.63	0.21	0.00	0.15		
3	Back elim-1	0.74	0.29	-0.00	0.08	0.00	0.06	0.97	0.28	0.02	0.10	-0.01	0.09	0.45	0.36	0.00	0.13		
4	Back elim-1	0.64	0.36	0.00	0.03	0.00	0.04	0.91	0.30	0.00	0.06	0.01	0.06	0.24	0.49	-0.00	0.10		
5	BeSS.gs-1	0.85	0.19	0.00	0.16	0.01	0.10	1.05	0.24	0.01	0.13	0.00	0.14	0.63	0.22	0.00	0.15		
6	BeSS.gs-1	0.83	0.19	0.00	0.07	0.00	0.03	1.03	0.21	0.01	0.07	0.00	0.05	0.60	0.23	0.01	0.14		
7	BeSS.seq-1	0.86	0.20	-0.01	0.15	0.02	0.10	1.05	0.24	0.01	0.13	-0.01	0.13	0.63	0.21	0.00	0.15		
8	BeSS.seq-1	0.84	0.18	0.01	0.10	0.00	0.03	1.04	0.21	0.01	0.06	0.00	0.08	0.61	0.22	0.00	0.12		
9	LASSO-1	0.50	0.34	0.03	0.07	0.02	0.05	0.66	0.38	0.03	0.06	0.02	0.06	0.33	0.32	0.01	0.07		
10	LASSO-2	-0.49	0.26	-0.48	0.26	-0.02	0.05	-0.01	0.03	-0.01	0.05	-0.42	0.31	-0.02	0.06	-0.01	0.02	0.00	0.02
11	LASSO-2	-0.47	0.28	-0.47	0.29	-0.01	0.04	-0.01	0.03	-0.01	0.04	-0.38	0.36	-0.01	0.04	-0.00	0.01	-0.01	0.03
12	LASSO-3	-0.26	0.17	-0.26	0.18	-0.01	0.04	-0.00	0.01	-0.01	0.02	-0.05	0.16	-0.01	0.02	-0.00	0.02	0.00	0.01
13	LASSO-3	-0.24	0.19	-0.25	0.19	-0.00	0.02	-0.00	0.02	-0.01	0.02	-0.03	0.18	-0.00	0.01	-0.00	0.02	0.00	0.02
14	LASSO-4	-0.13	0.25	-0.11	0.27	-0.02	0.06	-0.00	0.09	-0.03	0.10	-0.11	0.28	-0.01	0.02	0.00	0.01	0.00	0.02
15	LASSO-5	0.01	0.11	0.00	0.10	0.01	0.04	0.01	0.05	0.00	0.00	0.01	0.10	0.01	0.05	-0.00	0.03	0.00	0.03

D Enlarged table, comparison versus earlier papers

Table 7: Larger version of Table 5

	Method-Model	Cor. $\beta = 0$	Cmp. cor. $\beta = 0$	Rel. error (%)	Incor. $\beta \neq 0$	Cmp. incor. $\beta \neq 0$	Rel. error (%)	MMSE	CMMSE	Rel. error (%)	Ref to
1	LASSO-1	3.44	4.05	15.06	0.17	0.10	70.00	0.45	0.46	1.00	F
2	All subset-1	3.37	3.99	15.54	0.01	0.02	50.00	0.15	0.46	66.22	F
3	Back elim-1	4.66	3.99	16.79	0.41	0.02	1950.00	0.32	0.46	30.50	F
4	Back elim-1	4.92	3.99	23.31	0.85	0.02	4150.00	0.40	0.46	12.51	F
5	BeSS.gs-1	4.04	3.99	1.25	0.03	0.02	50.00	0.17	0.46	63.16	F
6	BeSS.gs-1	4.84	3.99	21.30	0.08	0.02	300.00	0.08	0.46	82.82	F
7	BeSS.seq-1	4.19	3.99	5.01	0.01	0.02	50.00	0.16	0.46	64.60	F
8	BeSS.seq-1	4.81	3.99	20.55	0.05	0.02	150.00	0.09	0.46	80.96	F
9	LASSO-1	3.55	3.99	11.03	0.03	0.02	50.00	0.35	0.46	22.78	F
10	LASSO-2	4.99	4.87	2.46	0.00	0.00		0.23	0.19	20.29	Z
11	LASSO-2	5.02	4.67	7.49	0.00	0.00		0.26	0.20	30.54	Z
12	LASSO-3	5.34	5.43	1.66	0.43	0.08	437.50	0.09	0.08	7.89	Z
13	LASSO-3	5.51	5.32	3.57	0.53	0.08	562.50	0.10	0.43	76.71	Z
14	LASSO-4	6.16	6.70	8.06				0.24	0.26	6.89	T
15	LASSO-5	7.98	7.80	2.31				0.03	0.15	80.00	T

E The main pseudo algorithm

The objects in the set M are the methods and the characteristics such as, true model, sample size, censoring rate, criterion and total simulations.

1. Create the objects for the set M.
2. Sort the objects in set M in the following order sample size, model, method, censor ratio. The reason for this is to be able to use the same data sets where applicable.
3. Set the variable *de* to false. The variable *de* marks if data sets are available.
4. For each object in the set M.
   (a) Initialize result variables at the place holder for the current configuration object, e.g counters for zero, non-zero coefficients, average of counters, MSE, $MMSE$, $\bar{\hat{\boldsymbol{\beta}}}$ and $\hat{\boldsymbol{\beta}}_{Se}$.
   (b) Check if data sets are available, e.g. de = true, in memory or on disk and the correct one, e.g. same sample size, model, censoring rate, simulations as previous object in the set M, if so set *de* to true otherwise set *de* to false.
   (c) For simulation from 1 to objects M's total simulations
      i. Initialize result data for the current object and simulation place holder, e.g. frequency counters for zero and non-zero coefficients.
      ii. if *de* is false then generate a sample data set according to the object's sample size, true model, censoring rate and simulation.
      iii. if *de* is true use existing data set.
      iv. Compute the correlation of $\boldsymbol{x}$ and print a correlation heat map, if enabled.
      v. Plot and save histogram of the covariates in the sample data set, if enabled.
      vi. Execute the current object's wrapper function with the current simulation data set and save the estimated coefficients of the winning model and the fit into the current object's simulation place holder.
      vii. Determine which coefficients that are zero and increment their respective counters, e.g. zero and non-zero counters.

viii. Compute the difference between the estimated coefficients and true model coefficients. Then compute the MSE according to Equation (29) and save to the object and simulation placeholder.

(d) if *de* is false set *de* to true and save the data sets to disk.

(e) Save the aggregated counters to the current object's placeholder

(f) Compute the median of the MSE, e.g. $MMSE$ and save to the current object's place holder.

(g) Compute the $\bar{\hat{\boldsymbol{\beta}}}$ according to Equation (35) and save to current object's place holder.

(h) Compute the $\hat{\boldsymbol{\beta}}_{Se}$ according to Equation (36) and save to current object's place holder.

(i) Plot and save a box plots of the mean of then estimated coefficients, e.g. $\bar{\hat{\boldsymbol{\beta}}}$.

(j) Compute difference between these results and (Tibshirani 1997, Fan & Li 2002) and Zhang & Lu (2007)

(k) Organize the results of the object into tables.

5. print and save tables of the results.