Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

Martin Blindheimsvik

Functional form for covariates in parametric accelerated failure time models using nonparametric exponential regression

Master's thesis in MTFYMA
Supervisor: Bo Henry Lindqvist
June 2019

NTNU
Kunnskap for en bedre verden

Martin Blindheimsvik

Functional form for covariates in parametric accelerated failure time models using nonparametric exponential regression

Master's thesis in MTFYMA
Supervisor: Bo Henry Lindqvist
June 2019

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

NTNU
Norwegian University of
Science and Technology

Abstract

This thesis looks at how the functional form of potentially misspecified covariates in an accelerated failure time model, can be estimated using two methods applied to the Cox-Snell residuals of the model. Two methods are looked at. One using Loess to smooth the Cox-Snell residuals. The other uses a method for nonparametric exponential regression called the covariate order method, to estimate the hazard for Cox-Snell residuals. We simulate data and do various simulations and calculations in R to showcase and illustrate the methods for estimating the functional form, and their effectiveness. We also apply the methods and analyze two real datasets. One regarding the post-election survival times of popes. The other containing data from a study on the fatal chronic liver disease PBC. We conclude with both methods being able to estimate the functional form of a covariate, and we see that even if a there are no clear underlying functional forms, we can still get ideas on how to improve a model through the estimate of the functional.

Sammendrag

Denne oppgaven undersøker hvordan funksjonell formen til en potensielt misspesifisert kovariat i en akselerert levetids modell kan bli estimert ved hjelp av to forskjellige metoder anvendt på Cox-Snell residualene til modellen. Den ene metoden bruker Loess for å glatte residualene i modellen. Den andre benytter en metode for ikke-parametrisk eksponentiell regresjon, kalt covariate order method, på Cox-Snell residualene. R blir brukt til diverse simuleringer og kalkulasjoner for å analysere og illustrere metodene som blir presentert. To virkelige datasett blir også analysert. Det ene datasettet inneholder hvor lenge paver lever etter de har blitt valgt inn som pave. Det andre består av levetider fra en kjent studie som omhandlet leversykdommen PBC. Vi konkluder med at begge metodene presentert er i stand til å estimere funksjonell formen til en misspesifisert kovariat, og vi har sett at dersom det ikke er en tydlig underliggende funksjonell form, så kan estimatet av denne likevel gi hint til forbedringer av den tilpassede modellen.

Preface

This project is written as a part of TMA4900, Industrial Mathematics, master's thesis, and builds on the project written in TMA4500. I would like to express my utmost gratitude to my project supervisor, Bo Henry Lindqvist, for the help he has given me, and all our great meetings. Having him as my supervisor has been a pleasure.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In survival analysis the cox-hazard model is widely used to model the relationship between event times and covariates. One of the drawbacks of this model is that it requires proportional hazards, and the number of probability distributions that can be model with it is some what limited. An alternative for modelling the relationship between event times and covariates is what is called the accelerated failure time model. While a proportional hazard model makes the assumption that the effect of a covariate is to multiply the hazard of the lifetime by some constant, an accelerated failure time model (AFT model) assumes that the effect of a covariate is to accelerate or decelerate the lifetime. An AFT model is a parametric regression model that is applied in various fields, including economics, reliability engineering, and biostatistics. The AFT model can be written on the form

$$\log Y = f(\mathbf{X}) + \sigma W. \tag{1.1}$$

Y is the lifetime or event time; $\boldsymbol{X} = (X_1, X_2, \ldots, X_p)$ is a vector of covariates, which is called a covariate vector; $f(.)$ is a function determining the effect of the covariates on the lifetime, which will be referred to as the functional form of the covariates; σW is an error term where σ is a positive scale parameter, and W is assumed to follow a fully specified standard distribution such as the standard Gumbel of the smallest extreme, standard normal distribution, or standard logistic distribution. The distribution of W gives the distribution of the lifetimes. If W is distributed according to the standard Gumbel of the smallest extreme, then Y is Weibull distributed. In case W follows the standard normal distribution or the standard logistic distribution, then Y is log-normal or log-logistic distributed, respectively.

In this project we initially assume that $f(.)$ is parametric and on the linear form

$$f(\mathbf{X}) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p. \tag{1.2}$$

β_0 is an intercept term, and $\beta_1, \ldots, \beta_p$ are coefficients of a regression model that can be estimated using maximum likelihood. The idea that this project will look at is that the true form of $f(.)$ is not necessarily linear as in equation (1.2), but

the covariates in $\mathbf{X}$ can have a more general, non-linear effect on the lifetimes in equation (1.1). By first fitting and estimating a potentially misspecified linear AFT model to data, we will look at using methods applied to the Cox-Snell residuals of this model to check and estimate the functional form, and potentially suggest better functional forms $f(X_i)$ of the covariates. One of these methods uses some of the estimated regression parameters of the potentially misspecified model, along with an estimate of the expectation of the fitted residuals conditional on the covariate, to estimate the functional form for said covariate. The other method is based on what is called the covariate order method, which is a method to do censored nonparametric exponential regression, to find an estimate for the functional form of a covariate. [13]

First we will present some general theory in statistics and survival analysis, necessary to understand how to estimate the functional forms. Then we will simulate four datasets of $n = 100$ Weibull distributed lifetimes, with different degrees of censoring, in which one of the covariates has an underlying logarithmic functional form. We will then do residual analysis and try to produce an estimate of this functional form using two different methods. Afterwards, we will investigate two more real datasets for any underlying functional forms. The first of these real datasets describes the post-election survival times of popes, and the two covariates we will look at is the year the popes where elected, and at what age they were elected. The second real dataset is from a well-known study on the fatal chronic liver disease PBC. The dataset contains 18 variables, but we will limit ourselves to looking at the effect of five covariates.

Chapter 2

Theory

2.1 Survival Analysis

Survival Analysis is a field in Statistics focused on analyzing and modelling the "lifetimes", commonly denoted by T, or survival of an item or an individual. The term lifetime does not necessarily mean the duration of time a person is alive. The term is also commonly used to denote the time until failure for a mechanical component or an item of some sort. We always have that $T \geq 0$. Lifetimes are modelled using probability density functions, and we can have both continuous and discrete lifetimes. For continuous lifetimes the probability density function (PDF), $f_T(t)$, must integrate to 1

$$\int_0^\infty f_T(t)dt = 1 \tag{2.1}$$

The cumulative distribution function (CDF) for the lifetime T is

$$F_T(t) = P(T \leq t) = \int_0^t f_T(u)du, \tag{2.2}$$

and gives the probability that a subject/item on test has failed prior to time t. Another central function in survival analysis is what is called the reliability or survival function. The survival function of T is

$$R_T(t) = 1 - F_T(t) = P(T > t) = \int_t^\infty f_T(u)du, \tag{2.3}$$

and it gives the probability that the subject/item on test has not failed at time t.

2.2 Poisson Distribution, HPP, and NHPP

An experiment that yields the number of outcomes, X, during a time interval or specified region is called a Poisson experiment. A Poisson experiment is derived

from what is called a Poisson process. The orthodox case is what is called a homogeneous Poisson process (HPP), which fulfills the following 3 properties.

1) Independent events. The number of events in a given time interval is independent of the number of events in a disjoint time interval.
2) The number of events occurring in a time interval is proportional to the length of the time interval and independent of events occurring outside of the interval in question.
3) The probability that two events happens simultaneously is negligible. $P(X(t, t+h) \geq 2) = o(h)$

The random variable X, modelling the number of events that happens during a Poisson experiment is said to follow the Poisson distribution. The Poisson distribution is a discrete probability distribution, and the probability mass function for a Poisson distributed random variable X is given as

$$p(x;t) = \frac{e^{-\lambda t}(\lambda t)^x}{x!}, \quad x = 0, 1, 2, ... \tag{2.4}$$

where λ is the average number of events per time unit. A HPP is a stationary point process for which the number of events is in an interval is only depending on the length of the interval. Alternatively to the HPP, it is also possible to have a non homogeneous Poisson process (NHPP). For the NHPP, the intensity or hazard, λ, varies as a function of time. Non homogeneous Poisson processes can for example be used to model repairable systems, and they are extensively used since they can model trends in the rate of failures. For more details on HPP and NHPP see [1].

2.3 Exponential Distribution

The exponential distribution is one of the most well known probability distributions in math, and it is the most applied distribution in survival analysis. The probability density function of the exponential distribution is

$$f_T(t) = \lambda e^{-\lambda t}, \quad t > 0, \lambda > 0. \tag{2.5}$$

λ is often referred to as the rate parameter. If the rate parameter equals 1 then variables are said to be unit exponentially distributed. The mean and variance of the exponential is

$$\begin{aligned} \mathrm{E}[T] &= \frac{1}{\lambda}, \\ \mathrm{Var}[T] &= \frac{1}{\lambda^2}. \end{aligned} \tag{2.6}$$

The Survival function is

$$R_T(t) = e^{-\lambda t}, \quad t > 0, \lambda > 0. \tag{2.7}$$

An important and useful property of the exponential distribution is what is referred to as the memoryless property. The memoryless property can be seen as a result from computing the conditional probability of the exponential.

$$\begin{aligned} R_T(s|t) = Pr(T > s+t|T > t) &= \frac{Pr(T > s+t)}{Pr(T > t)} \\ &= \frac{e^{-\lambda(s+t)}}{e^{-\lambda t}} = e^{-\lambda s} = R_T(s). \end{aligned} \tag{2.8}$$

This shows that an item's time to failure at time s is independent of how long it has been on test/active. In other news we say that the item is basically "as good as new". The memoryless property is part of what makes the exponential distribution easy to work with, but on the other hand it is not realistic for an item that has been operation for an extended duration of time to be "as good as new", so it can also be viewed as weakness of the exponential.

2.4 The Gumbel distribution of the smallest extreme

Assume an independent set of identically distributed lifetimes, T_i for n components with ordered values such that $T_{(1)} < T_{(2)} < \cdots < T_{(n)}$. $T_{(1)}$ is then the minimum lifetime in the set. For large n, $T_{(1)}$ is approximately Weibull distributed. [6] This is a motivation for the widespread use of the Weibull to model lifetimes in survival analysis. Assume now that the $T_i s$ have support $(-\infty, \infty)$ and are no longer lifetimes. For a normalized version of $T_{(1)}$, the limiting distribution will be equal to the CDF of a random variable Y

$$F_Y(y) = 1 - e^{-e^{\frac{y-\mu}{\sigma}}}, \qquad -\infty < y < \infty. \tag{2.9}$$

Here, $\mu > 0$ and σ are constants called the mode and scale, respectively. This is the PDF for what is called the Gumbel distribution of the smallest extreme. The Gumbel distribution is an important asymmetric distribution due to its extreme value behaviour.

An important case of $Y \sim \text{Gumbel}(\mu, \sigma)$ is the standard Gumbel distribution, $W \sim \text{Gumbel}(0, 1)$. It follows from (2.9) that the CDF of the standard Gumbel is

$$G(w) = 1 - e^{-e^w}, \qquad -\infty < w < \infty. \tag{2.10}$$

From the relation $R(w) = 1 - G(w)$ it follows that the reliability function of the standard Gumbel is

$$R(w) = e^{-e^w}, \qquad -\infty < w < \infty. \tag{2.11}$$

The PDF is

$$g(w) = e^w e^{-e^w}, \qquad -\infty < w < \infty. \tag{2.12}$$

2.5 The Weibull distribution

The Weibull distribution is one of the most utilized distributions in survival analysis. Its PDF, mean, and variance are given as

$$\begin{aligned} f_T(t) &= \frac{\alpha}{\theta}\left(\frac{t}{\theta}\right)^{\alpha-1} e^{-(\frac{t}{\theta})^{\alpha}}, \qquad t>0, \quad \theta>0, \\ E[T] &= \theta\cdot\Gamma\left(\frac{1}{\alpha}+1\right), \\ Var[T] &= \theta^2\left(\Gamma\left(\frac{2}{\alpha}+1\right)-\Gamma^2\left(\frac{1}{\alpha}+1\right)\right). \end{aligned} \tag{2.13}$$

$\alpha > 0$ and θ are known as the shape and scale parameters, respectively. $\Gamma(.)$ is the well-known Gamma function defined by the integral $\Gamma(a) = \int_0^\infty u^{a-1}e^{-u}du$. From the PDF in (2.13) it can be seen that $\alpha = 1$ gives the PDF for the exponential distribution. Thus, the exponential distribution is a special case of the Weibull. The Reliability function for the Weibull is

$$R_T(t) = e^{-(\frac{t}{\theta})^{\alpha}}, \quad \text{for } t>0. \tag{2.14}$$

2.6 Censoring

A lifetime is said to be censored if the failure time is not observed directly. The most common forms of censoring is right, left, and interval censoring. If it is known when a subject is put on test but not when it fails, then the lifetime for the subject is said to be right censored. If the time of failure is known while the time the subject is put on test is unknown, then we have left censoring. Assume the exact time a subject fails is unknown, but it is known that the subject fails sometime between time t_1 and t_2, the lifetime for the subject is then said to be interval censored.

2.7 Accelerated failure time model

Observations in this project are assumed to be realizations of the random vector $(\mathbf{X}, W, C)$. $\mathbf{X}$ is a vector of covariates that can take both discrete and continuous form. W is an error that is distributed according to some probability distribution function $\phi(.)$, with a corresponding cdf $\Phi(.)$. W is assumed independent of X and in addition, $\phi(u) > 0\ \forall u \in (-\infty, \infty)$. C denotes the censoring time of the observation, which is an absolutely positive random variable that is distributed according to some distribution that can depend on $\mathbf{X}$.

An individual Y has a true lifetime given by

$$\log Y = f(\boldsymbol{X}) + \sigma W, \tag{2.15}$$

where σ is a positive scale parameter, and $f(\mathbf{X})$ is a parametric function of the covariate vector $\mathbf{X}$. The observed lifetimes are given by $T = \min(Y, C)$. It follows that the censoring indicator is given by $\Delta = I(Y \leq C)$.

Let $h(.|.)$ and $H(.|.)$ denote the PDF and CDF of Y conditional on $\mathbf{X}$, respectively. Assume further an observed i.i.d. sample $(t_i, \delta_i, \mathbf{x_i})$ of $(T, \Delta, \mathbf{X})$. Under the assumption that the distributions of $\mathbf{X}$ and C are independent of the parameters of $h(.|.)$ we have the following likelihood for survival analysis. [9]

$$\prod_{i=1}^{n} \{h(t_i|\boldsymbol{x_i})\}^{\delta_i} \{H(t_i|\boldsymbol{x_i})\}^{1-\delta_i}, \tag{2.16}$$

the parameters of $h(.|.)$ is here the scale parameter and the specification of $f(\mathbf{X})$.

2.8 Residuals for AFT models

Standardized residuals are a common and natural type of residuals for Accelerated failure time models. These residuals can be found by solving (2.15) for W. It then follows that

$$S = \frac{\log T - f(\boldsymbol{X})}{\sigma}. \tag{2.17}$$

It can then be seen that conditionally on $\mathbf{X}$, S follows a distribution $\Phi(.)$. For observed data $\{(t_i, \delta_i, x_i),\ i = 1, \ldots, n\}$, the standardized residuals are defined by $(\hat{s}_i, \delta_i), i = 1, \ldots, n$

$$\hat{s_i} = \frac{\ln t_i - \hat{f}(\boldsymbol{x_i})}{\hat{\sigma}}, \tag{2.18}$$

where $\hat{f}(.)$, and $\hat{\sigma}$ are satisfactory estimates of the underlying functional form $f(.)$, and the scale parameter σ in the model. [14] These estimates are normally computed based on maximum likelihood estimation. The idea behind this form of residuals is that if the specified model is good, then $(\hat{s_i}, \delta_i)$ will behave like a censored sample from the distribution function of the error W in (2.15). If there are right censored observations t_i, then this will correspond to the standardized residuals $\hat{s_i}$ becoming "small".

Cox-Snell residuals are another commonly applied form of residuals in survival analysis. Like Standardized residuals, Cox-Snell residuals are mainly used for model checking. The basis for Cox-Snell is that for a lifetime Y, where $G(t) = P(Y > t)$ is the corresponding survival function, the random variable $-\log G(t)$ will be unit exponentially distributed. Note from (2.15) that

$$G(t|\boldsymbol{X}) = P(Y > t|\boldsymbol{X}) = 1 - \Phi\left(\frac{\log t - f(\boldsymbol{X})}{\sigma}\right). \tag{2.19}$$

It further follows by taking $-\log$ of (2.19), that

$$R = -\log G(Y|\boldsymbol{X}) = -\log\left(1 - \Phi\left(\frac{\log Y - f(\boldsymbol{X})}{\sigma}\right)\right), \tag{2.20}$$

should be unit exponential given $\mathbf{X}$. Thus, the Cox-Snell residuals of a fitted model is given by $(\hat{r}_i, \delta_i)$, for $i = 1, \ldots, n$, where

$$\hat{r_i} = -\log\left(1 - \Phi\left(\frac{\log t_i - \hat{f}(\mathbf{x}_i)}{\hat{\sigma}}\right)\right). \tag{2.21}$$

If the fitted model is good for the data $\{(t_i, \delta_i, x_i), i = 1, \ldots, n\}$, then (2.21) will behave akin to a censored sample from the unit exponential distribution. From (2.17) and (2.20) it is seen that

$$\begin{aligned} R &= -\log(1 - \Phi(S)), \\ S &= \Phi^{-1}(1 - e^{-R}). \end{aligned} \tag{2.22}$$

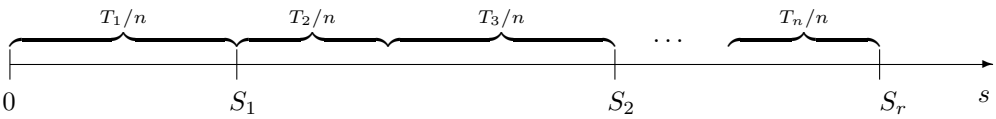
When calculating residuals for a fitted model, censored data needs to be handled. An often used method to account for censoring is by simply adding the expected value of the residual distribution to the residuals corresponding to censored observations, and then proceed as if the set of data is non-censored. Cox-Snell residuals are expected to follow the unit exponential distribution. Thus, censoring can simply be handled by adding 1 to the censored residuals due to the memoryless property of the exponential distribution. This is called 1-adjusted residuals. An alternative way of handling censoring for Cox-Snell is by $\log 2$-adjusted residuals. This follows from the mean residual life of a unit exponentially distributed random variable being equals to $\log 2$. The $\log 2$-adjusted residuals are computed by adding $\log 2$ to the censored residuals.

To analyze the residuals you can plot the residuals versus the covariate components of the covariate vector $\mathbf{X}$. If there are censored residuals this can be slightly misleading due to small censored residuals. The introduction of censored residuals is in hopes of mitigating this. Adjusted residuals might work well unless there is a high degree of censoring, in which case the effect of the covariates on the residuals might become blurred due to adjusting. A common way of presenting residual plots is by plotting the logarithm of the residuals as a function of covariates due to better symmetry of the residuals. For a good fit the logarithm of Cox-Snell residuals should be symmetric with respect to the covariate axis.

2.9 Covariate Order Method

The covariate order method is a nonparametric method for censored exponential regression. It can be shown that this method leads to a consistent estimator of the hazard rate as a function of the covariate. [13] The covariate order method will be used in this project as a means to get an estimate of the functional form of an AFT model with Weibull distributed lifetimes.

Assume n independent observations $(T_1, \delta_1, \mathbf{X_1}), ..., (T_n, \delta_n, \mathbf{X_n})$, where $T = \min(Y, C)$ is the observation time, $\delta = I(Y \leq C)$ is the censoring indicator, and $\mathbf{X}$ is the covariate vector. For given $\mathbf{X} = \mathbf{x}$, Y is assumed to be exponentially distributed with an unknown hazard rate $\lambda(\mathbf{x})$ such that the pdf of Y is $f_Y(t|\mathbf{x}) = \lambda(\mathbf{x}) \exp(-\lambda(\mathbf{x})t)$. C follows some unknown censoring distribution $f_C(t|\mathbf{x})$ which could be dependent on $\mathbf{X}$, but is independent of Y. C is called the censoring time, and Y is called the lifetime. Furthermore, assume that $\mathbf{X}$, which is a subset $\boldsymbol{\chi}$ of $\mathbb{R}^m$, remains continuous over time, and that $\lambda(\mathbf{x})$ is continuous on $\boldsymbol{\chi}$.



For the 1-dimensional case of the method, start with sorting the set of observations $\{(T_j, \delta_j, \mathbf{X_j}), j = 1, \ldots, n\}$, such that $X_1 \leq X_2 \leq \cdots \leq X_n$. If there are a small number of ties in the covariate data, then this can be dealt with by arranging the corresponding observations in random order. Proceed by diving the observation times with the number of observations, n. Then treat $\frac{T_1}{n}, \frac{T_2}{n}, \ldots, \frac{T_n}{n}$ as inter-arrival times of an artificial Poisson process illustrated in the above figure (figure provided by Bo Lindqvist). Let the endpoints of the intervals that correspond to uncensored observations be considered as events that occur at times $S_1, S_2, \ldots, S_r$, while censored observations are not considered as events. We have that $r = \sum_{j=1}^{n} \delta_j$. Formally we have that

$$S_i = \sum_{j=1}^{k(i)} \frac{T_j}{n}, \quad k(i) = \min(s | \sum_{j=1}^{s} \delta_j = 1) \tag{2.23}$$

The covariate order method as described by Kvaløy and Lindqvist, uses density estimation to estimate the intensity of the artificial point process, $\rho(s)$, which can then be transformed into an estimator of the intensity $\lambda(x)$ at given values of x by inverting

$$\hat{\rho}(s) = n\hat{\lambda}(X(s)). \tag{2.24}$$

The key to this estimate is the relationship between $X_1, \ldots, X_n$ on the covariate axis, and the process $S_1, \ldots, S_r$ on the s-axis. This relationship can be estimated by for example a step-function

$$\bar{s}(x) = \frac{1}{n} \sum_{i=1}^{j} T_i, \quad X_j \leq x \leq X_{j+1}, \tag{2.25}$$

called the correspondence function. The correspondence function can be replaced with more sophisticated estimators to get a smoother estimate, but in many cases the step-function should prove sufficient. Now define

$$\hat{\lambda}(x) = \rho(\bar{s}(x)) \tag{2.26}$$

The main motivating idea behind the method is that if $\lambda(x) = \lambda$ is constant, then the process of the artificial process $S_1, \ldots, S_r$ is a homogeneous Poisson process. So, if $\lambda(x)$ does not vary too much, then the process $S_1, \ldots, S_r$ can be imagined to be a nearly non-homogeneous Poisson process, and the intensity can be estimated by combining the the estimate of the correspondence function in (2.25) with the kernel estimate (2.24). For more details and theory on the covariate order method see Kvaløy and Lindqvist (2004). [13]

Chapter 3

Simulated data

3.1 Simulating data and Cox-Snell Residuals

To get an idea of how well our methods for estimating the functional form performs, we start off by simulating some datasets with different degrees of censoring, and then apply the methods for estimating the functional form of covariates. We look at simulated data since we can then simulate a covariate with a clear functional form, and see how well our methods can recover that functional form from a misspecified linear AFT model. Looking at data with different degrees of censoring should give an idea of how censoring affects the estimate of the functional form.

It is known that the log-location scale model,

$$\log T = \boldsymbol{\beta}^T \mathbf{Z} + f(\mathbf{X}) + \sigma W, \tag{3.1}$$

models Weibull distributed lifetimes when $W \sim \text{Gumbel}(0, 1)$. Here $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_k)$ is a vector of coefficients, $\mathbf{Z} = (1, z_1, \ldots, z_k)$ is a vector of covariates. It follows that Weibull distributed lifetimes can be simulated from

$$T = e^{\boldsymbol{\beta}^T \mathbf{Z} + f(\mathbf{X}) + \sigma W}. \tag{3.2}$$

To simulate Weibull lifetimes using (3.2), it is necessary to simulate W from the standard Gumbel distribution of the smallest extreme. An algorithm to do this can be developed by means of the inverse transformation method described in [15]. By using the fact that the CDF of a random variable takes values in $[0, 1]$, we can simulate $u \sim \text{Unif}[0, 1]$. Then set u equals to the CDf (2.10) and solve for w.

$$\begin{aligned} 1 - e^{-e^w} &= u, \\ -e^w &= \log(1 - u), \\ w &= \log[-\log(1 - u)], \\ w &= \log[-\log(u)]. \end{aligned} \tag{3.3}$$

w in (3.3) is a realization from the standard Gumbel distribution if $u \sim \text{Unif}[0,1]$. The last transition in equation (3.3) follows from $u \sim \text{Unif}[0,1]$.

We have that Φ is the CDF of the residual distribution of our model. Hence, Φ is the CDF of the standard Gumbel (2.10), and an expression for the Cox-Snell residuals of a Weibull AFT model can be found by solving for W in (3.2) and inserting into (2.20). It follows that the Cox-Snell residuals for a Weibull AFT model is

$$\begin{aligned} \hat{R}_i &= -\log[1 - F_W(W_i)], \\ &= -\log(e^{-e^{W_i}}) = e^{W_i}, \\ &= e^{\frac{\log T_i - \boldsymbol{\beta}^T \boldsymbol{z_i} - f(x_i)}{\sigma}}. \end{aligned} \tag{3.4}$$

By utilizing the relation between Cox-Snell and standardized in (2.22), the standardized residuals are found to be

$$\hat{S}_i = \log \hat{R}_i = \frac{\log T_i - \boldsymbol{\beta}^T \boldsymbol{z_i} - f(x_i)}{\sigma}. \tag{3.5}$$

To simulate lifetimes with censoring, let $\Psi(.)$ be the CDF of the censoring times C. $\Psi(.)$ can be dependent on the covariate $\mathbf{X}$, but in this project assume that C and $\mathbf{X}$ are independent. For this project the censoring times C are simulated according to the Exponential distribution described in section 2.3. It follows that censored lifetimes can be simulated as

$$Y_i = \min(T_i, C_i),$$

where T_i is simulated according to (3.2). Furthermore, the censoring indicator is

$$\delta_i = 1 \text{ if } T_i < C_i,$$

$$\delta_i = 0 \text{ if } T_i \geq C_i.$$

We will simulate four datasets of $n = 100$ observations, with different degrees of censoring. The motivation behind doing this is to explore how our two methods for estimating the functional form performs under controlled circumstances with various censoring, where the true functional form is known. 1 uncensored and 3 censored data sets with 20%, 60%, and 80% censoring are simulated. The parameters for the simulation are set to

$$\begin{aligned} \beta_0 &= 0, \\ \beta_1 &= 1, \\ f(x) &= \log x, \\ \sigma &= 1. \end{aligned} \tag{3.6}$$

Thus, T_i is simulated according to

$$T_i = e^{\beta_1 z_i + \log x_i + \sigma W_i} \tag{3.7}$$

z_i is simulated from the standard normal distribution, while x_i is simulated from the exponential distribution with rate parameter $\lambda = 1/2$.

After simulating a set of failure times, 3 sets of censoring times $\{C_i, i = 1, .., n\}$ were simulated independently from the exponential distribution with rate parameters $\lambda = [9, 0.77, 0.2]$ to give 3 censored data sets with 20%, 60%, and 80% censoring respectively. The simulated uncensored data, and the censored data are given in Appendix A.1, while the code used to simulate the data are given as Appendix B.1.

3.2 Estimation of covariate functions

For a lifetime T assume that the correct model is given by

$$\log T = \beta_0 + \boldsymbol{\beta}^T \boldsymbol{Z} + f(X) + \sigma W. \tag{3.8}$$

X is a component of the covariate vector $\boldsymbol{X}$ in section 2.7, the remaining components of $\boldsymbol{X}$ is denoted as $\boldsymbol{Z}$. Thus, $\boldsymbol{X}$ is more formally denoted as $\boldsymbol{X} = (X, \boldsymbol{Z})$. Given the data $\{(t_i, \delta_i, x_i, z_i); i = 1, ..., n\}$, the goal is now to derive an expression for the functional form for the covariate X, $f(X)$.

To begin with fit the linear model

$$\log T = \beta_0 + \boldsymbol{\beta}^T \boldsymbol{Z} + \gamma X + \sigma W. \tag{3.9}$$

Using maximum likelihood, where the likelihood function is (2.16), the estimated, potentially misspecified, model (3.9) is

$$\log T = \hat{\beta}_0 + \hat{\boldsymbol{\beta}}^T \boldsymbol{Z} + \hat{\gamma} X + \hat{\sigma} W. \tag{3.10}$$

By inserting the estimated model (3.10) into formula (2.18) it follows that the standardized residuals are

$$\hat{s_i} = \frac{\log t_i - \hat{\beta}_0 - \hat{\boldsymbol{\beta}}^T \boldsymbol{z_i} - \hat{\gamma} x_i}{\hat{\sigma}}. \tag{3.11}$$

The theory on residuals in section 2.8 says that if model (3.9) is correctly specified, and $f(x)$ is linear in x, then the standardized residuals should behave as a sample from the distribution $\Phi(.)$. If on the other hand $f(x)$ is not linear in x, Kvaløy and Lindqvist has shown that the standardized residuals can be used as a means to infer the functional form. [14] If a model is defined by parameters that could be false, $(\beta_0^*, \boldsymbol{\beta}^*, \gamma^*, \sigma^*)$, the theoretical standardized residuals are

$$S^* = \frac{\log T - \beta_0^* - \boldsymbol{\beta}^{*T} \boldsymbol{Z} - \gamma^* X}{\sigma^*}. \tag{3.12}$$

Inserting the true model (3.8) for $\log T$ in (3.12) gives

$$S^* = \frac{\sigma}{\sigma^*}W + \frac{(\beta_0 - \beta_0^*) + (\boldsymbol{\beta} - \boldsymbol{\beta}^*)^T \boldsymbol{Z} + f(X) - \gamma^* X}{\sigma^*}. \tag{3.13}$$

If $f(x)$ is in fact linear, then it can be seen from (3.13) that S^* conditional on $(X, \boldsymbol{Z})$ is distributed according to W. Solving for $f(X)$ gives

$$f(X) = -\sigma W - (\beta_0 - \beta_0^*) - (\boldsymbol{\beta} - \boldsymbol{\beta}^*)^T \boldsymbol{Z} + \gamma^* X + \sigma^* S^* \tag{3.14}$$

Taking the conditional expectation given $X = x$ yields

$$f(x) = -\sigma E(W) - (\beta_0 - \beta_0^*) - (\boldsymbol{\beta} - \boldsymbol{\beta}^*)^T E(\boldsymbol{Z}|X = x) + \gamma^* x + \sigma^* E(S^*|X = x). \tag{3.15}$$

Assume that X and Z are independent, it then follows that $-\sigma E(W) - (\beta_0 - \beta_0^*) - (\boldsymbol{\beta} - \boldsymbol{\beta}^*)^T E(\boldsymbol{Z}|X = x)$ is independent of x and is just a displacement of the curve $f(x)$. This leads to the equation

$$f(x) = const + \gamma^* x + \sigma^* E(S^*|X = x), \tag{3.16}$$

where const denotes some displacement of the curve. As a consequence, f can be estimated by

$$\hat{f}(x) = \hat{\gamma} x + \hat{\sigma}\hat{H}(x), \tag{3.17}$$

where $\hat{H}(x)$ is an estimate of

$$H(x) = E(S^*|X = x), \tag{3.18}$$

and can be found by smoothing $\{(x_i, \hat{s}_i); i = 1, ..., n\}$. Observe that if the potentially misspecified model in equation (3.10) is indeed a good model, then $E(S^*|X = x)$ is approximately zero and consequently

$$\hat{f}(x) \approx \hat{\gamma} x. \tag{3.19}$$

In practice we will mainly work with Cox-Snell residuals. Assuming Weibull distributed lifetimes it follows that $\hat{H}(x)$ can instead be estimated by smoothing the adjusted Cox-Snell residuals $r_i = \log s_i$. We will be working with 1-adjusted Cox-Snell residuals as described in section 2.8

Moving on we will apply equation (3.17) to a linear model fitted to the simulated data in Appendix A.1. This is done by using the *survreg* function in the *survival* library in R. The code for fitting the models to the uncensored, and censored data sets are given in Appendix B.6. The model

$$\log Y_i = \beta_0 + \beta_1 x_{1i} + \gamma x_{2i},$$

was fitted to the data with no censoring, 20%, 60%, and 80% censoring, and gave the parameter estimates in Table 3.1

censoring	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\gamma}$	$\hat{\sigma}$	$\log \hat{\sigma}$
0	$-0.6235(0.2145)$	$0.9531(0.1152)$	$0.4443(0.0976)$	1.08	$0.0746(0.0782)$
20%	$-0.6654(0.2384)$	$1.0613(0.1478)$	$0.4913(0.1148)$	1.11	$0.1068(0.0882)$
60%	$-1.536(0.344)$	$1.260(0.241)$	$1.219(0.247)$	1.13	$0.122(0.120)$
80%	$-1.0603(0.4786)$	$1.3195(0.3424)$	$0.8818(0.3074)$	1.02	$0.0195(0.1735)$

Table 3.1: Parameter estimates for the misspecified linear Weibull AFT model $\log T = x_1 + x_2 + \sigma W$, that was fitted to the simulated datasets in Appendix A.1 using survreg. The underlying correct model for the simulated data is $\log T = x_1 + \log(x_2) + \sigma W$.

x_1 is what has previously been referred to as z, while x_2 is the misspecified covariate which has been referred to as x. From Table 3.1 we can read off the parameter estimates that we will use in (3.17) to estimate functional form. The Cox-Snell residuals can also be calculated by inserting the parameter estimates into (3.4). In order to investigate whether the fitted models provide good fits for their respective datasets, we look at plots of the logarithm of the Cox-Snell residuals against the covariates. Plots of the logarithm of the Cox-Snell against the covariates are illustrated in Figure 3.1, 3.2, 3.3, and 3.4 for the uncensored dataset, and the data with 20%, 60%, and 80% censoring, respectively. The Cox Snell residuals for the models fitted to the censored datasets have been adjusted by adding 1 as described in section 2.8

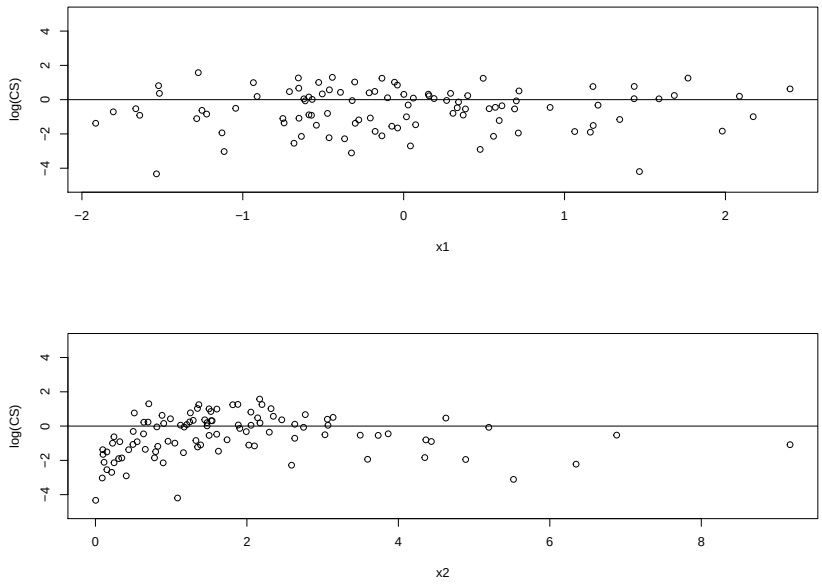


Figure 3.1: Log of Cox-Snell residuals as a function of the covariates for the misspecified model corresponding to the uncensored dataset

From Figure 3.1 there appears to be no dependency between the residuals and x_1 as the logarithm of the Cox-Snell residuals appears to be symmetric around 0 with no apparent patterns as a function of x_1. For x_2 most of the residuals can be found between 0 and 2 on the covariate axis, and the value of the residuals look like they increase as a function of x_2. Thus, it is clear from Figure 3.1 that the fitted linear model to the uncensored data set is not a good fit as there is a pattern for x_2.

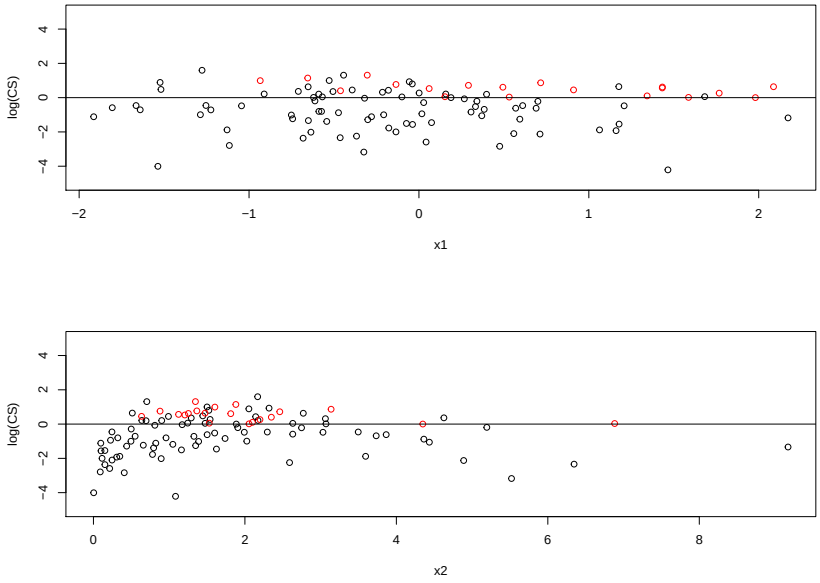


Figure 3.2: Log of 1-adjusted Cox-Snell residuals as a function of the covariates for the misspecified model corresponding to the dataset with 20% censoring. The red dots are censored residuals which have been adjusted by adding 1, while the black dots are uncensored residuals.

Similarly to Figure 3.1, Figure 3.2 shows no dependency between the residuals and x_1. For x_2 the same pattern present in Figure 3.1 appears. So the regardless of the 20% censoring the pattern in the residuals as a function of x_2 is still clear.

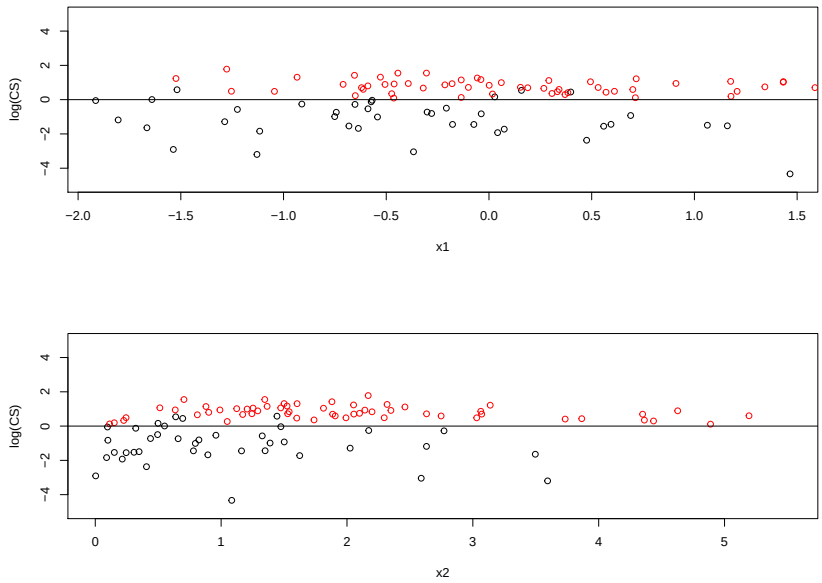


Figure 3.3: Log of 1-adjusted Cox-Snell residuals as a function of the covariates for the misspecified model corresponding to the dataset with 60% censoring. The red dots are censored residuals which have been adjusted, while the black dots are uncensored residuals.

In Figure 3.3 there is large number of censored residuals that have been adjusted. For x_1 there appears to be no pattern like the two previous cases but we can see that for the residual axis, the observations above 0 are primarily censored residuals that have been adjusted, while most uncensored residuals are less than 0. For x_2 the pattern present in the previous two cases appears to have disappeared to a large extent. While most residuals are still located between 0 and 3 along the covariate axis for x_2, it does not look as clear that the value of the residuals is increasing as x_2 increases. This might be a result of the high degree of censoring in the simulated data, effectively masking the functional form.

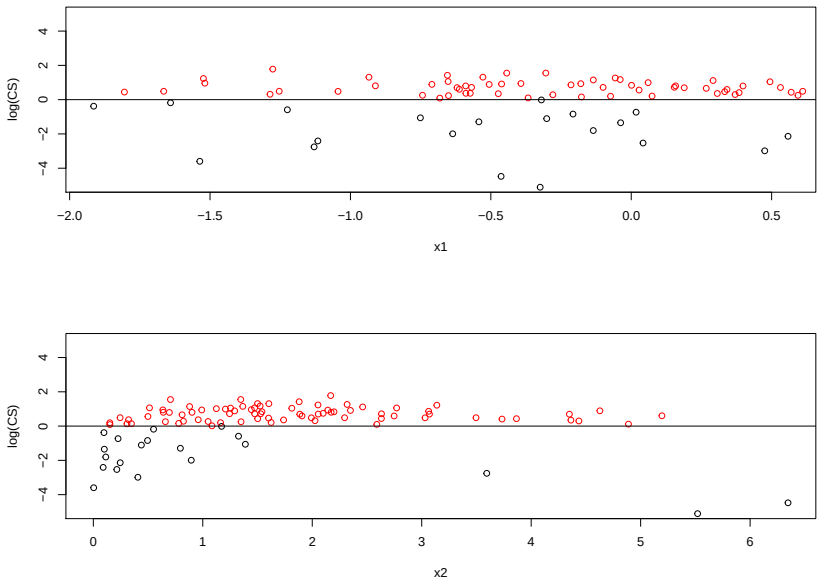


Figure 3.4: Log of 1-adjusted Cox-Snell residuals as a function of the covariates for the misspecified model corresponding to the dataset with 80% censoring. The red dots are censored residuals, while the black dots are uncensored residuals.

For Figure 3.4 like the 3 previous cases there appears to be no dependency between the residuals and x_1. For x_2 there appears to be more of a pattern present than in the case for 60% censoring, and we observe that between 0 and around 1.5 on the covariate axis, the unadjusted, uncensored residuals increase as x_2 increases. However, the pattern does appear to largely disappear as in the previous case with 60% censoring.

From all 4 residual plots there looks like there is no dependency between the residuals and the covariate x_1 for any level of censoring. For the misspecified covariate x_2 on the other hand, there is a clear pattern and dependency between the covariate and the residuals for the first two cases, but this disappears to a large extent for the last two cases. It can be observed from Figure 3.2, 3.3, and 3.4 that 1-adjusting the censored residuals contribute to "blurring" the pattern present in the residual plot for the misspecified covariate.

We now move on to estimating the functional form of the misspecified covariate x_2. Using LOESS in R, with span = 0.75, to smooth the logarithm of the adjusted Cox-Snell residuals we get an estimate of $\hat{H}(x_2)$. Inserting the estimated values for $\hat{\gamma}$ and $\hat{\sigma}$ given in the Table 3.1, along with the estimate for $\hat{H}(x_2)$, into equation (3.17) gives the estimated functional form of x_2 in Figure 3.5 for the 4 models.

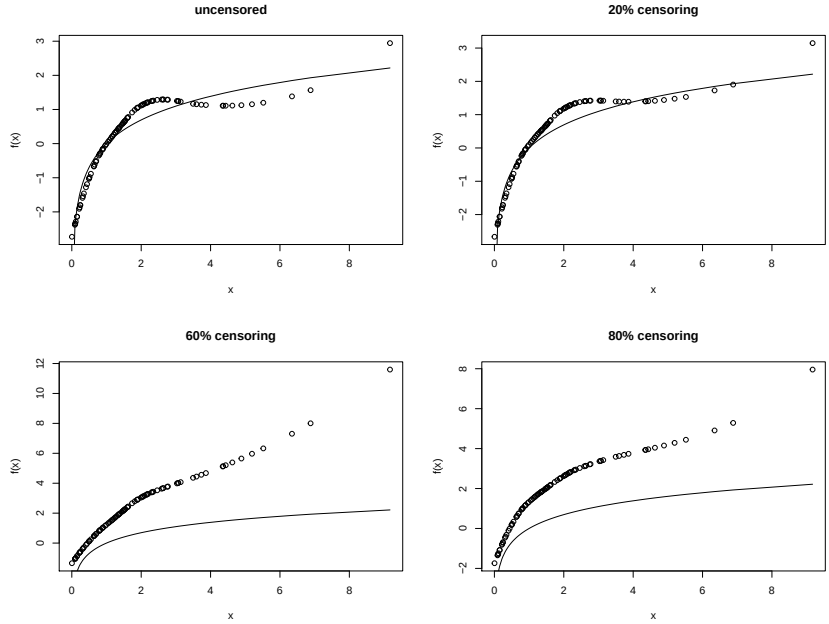


Figure 3.5: Functional form of the misspecified covariate x_2 using equation (3.17), for the data with no censoring, 20%, 60%, and 80% censoring. The solid lines show the true functional form $f(x_2) = \log x_2$, while the dots is the estimated functional form.

For the case with no censoring and 20% censoring the functional form appears to have been recovered and it looks clear that it is $f(x_2) = \log x_2$. For the case with 60% censoring the functional form appears to be a bit more linear, and at the very least its hard to say that the functional form has been recovered. For the case with 80% censoring the functional form appears to be more similar to the logarithmic one than the case with 60% censoring. This might be a result of chance, as you would expect the estimate of the functional form the become gradually worse the more censoring is present. Also, observe that the displacement of the curve increases for higher censoring. This is likely due to the estimated regression parameters being further from the true underlying regression parameters for the case higher censoring cases, which leads to a larger displacement of the curve by equation (3.15).

3.3 Covariate Order Method

The goal now is to implement and apply the covariate order method described in section 2.9 to estimate the functional form of the misspecified covariate.

Let Cox-Snell residuals $\hat{r}_i$ and the data be as given in section 3.2. The main idea is to consider a synthetic data set $\{(\hat{r}_i, \delta_i, x_i), \quad i = 1, \ldots n)\}$, where x_i denotes the value for a specific covariate at time y_i, where we impose an exponential model

with hazard rate $\lambda(x)$ for $\hat{r}$ given x. Now we apply exponential regression to the synthetic data set in order to estimate the hazard rate $\lambda(.)$. The estimate is denoted by $\hat{\lambda}(.)$.

By applying the connection between Cox-Snell and standardized residuals in (2.22) to $H(x) = E(S^*|X = x)$, $H(x)$ can be written in terms of the Cox-Snell residuals as

$$H(x) \equiv E(\Phi^{-1}(1 - \exp(-R^*))|X = x). \tag{3.20}$$

Assuming the exponential model for the Cox-Snell residuals given the parameterization in section 2.3, the expected value of R^* for a given x is $\frac{1}{\hat{\lambda}(x)}$. Thus, by replacing R^* by its expected value in (3.21), the estimate $\hat{H}(x)$ of $H(x)$ is

$$\hat{H}(x) = \Phi^{-1}(1 - \exp(1/\hat{\lambda}(x))). \tag{3.21}$$

$\Phi^{-1}(1 - e^{-r})$ is concave for the Weibull model, thus the right-hand side of (3.21) is convex in $\hat{\lambda}(x)$. Jensen's inequality says that for any concave function

$$E[f(X)] \leq f(E[X]).$$

Thus, it follows from Jensen's inequality that

$$E(\hat{H}(x)) \geq \Phi^{-1}(1 - \exp(1/E(\hat{\lambda}(x)))).$$

This indicates a possibility of overestimating. In addition, under the given assumptions, if $\hat{\lambda}(x)$ is a consistent estimator for the hazard rate, then $\hat{H}(x)$ is also a consistent estimator for $H(x)$ [14].

For Weibull AFT models we have that

$$\Phi^{-1}(x) = \log(-\log(1 - x)), \quad \text{for } 0 < x < 1. \tag{3.22}$$

It follows from (3.22) that $H(x) = E(\log R^*|X = x)$. $\hat{H}(x)$ can thus be found by smoothing the points $(x_i, \log \hat{r}_i)$. In addition, it follows that $\hat{H}(x)$ can also be written as

$$\hat{H}(x) = -\log \hat{\lambda}(x), \tag{3.23}$$

By inserting (3.23) into (3.17) this leads to

$$\hat{f}(x) = \hat{\gamma}x - \hat{\sigma} \log \hat{\lambda}(x). \tag{3.24}$$

As mentioned in section 2.9, the covariate order method is a nonparametric method for censored exponential regression. Thus, by applying the covariate order method to the synthetic data $(\hat{r}_i, \delta_i, x_i)$, the hazard $\lambda(x)$ can be estimated, assuming the Cox-Snell residuals are approximately exponentially distributed.

To implement an algorithm to estimate $\lambda(x)$ via the covariate order method, start by sorting the set of observations as described in section 2.9, and computing the times $\{S_i, \quad i = 1, \ldots, r\}$, where $r = \sum_{j=1}^{n} \delta_j$.

Let $K(.)$ denote a positive kernel function that integrates to 1 and disappears outside the interval $[-1, 1]$. Furthermore, let h_s be a smoothing parameter that can either be constant or varying along the s-axis.

$$\hat{\lambda}(x) = \frac{1}{nh_s} \sum_{i=1}^{r} K\left(\frac{\hat{s}(x) - S_i}{h_s}\right); \qquad x \in \boldsymbol{\chi} \tag{3.25}$$

$\hat{\lambda}(x)$ in equation (3.25) was proven by Kvaløy and Lindquist to be a uniformly consistent estimator of $\lambda(x)$. [13] $\hat{s}(x)$ is the correspondence function as described in section 2.9.

The performance of a kernel is measured by MISE (mean integrated squared error) or AMISE (asymptotic MISE). Any kernel $K(.)$ that satisfies the assumptions can be used, but we will use the Epanechnikov kernel, since this produces the minimal MISE for kernels of order (0,2) [11]. The Epanechnikov kernel is given as

$$K(x) = \frac{3}{4}(1 - x^2) I_{[-1,1]}. \tag{3.26}$$

where $I_{[-1,1]}$ is an index function in the interval $[-1, 1]$. In practice the estimator in (3.25) will be downward biased near the endpoints. In order to handle this we can implement a boundary kernel or use the reflection method. We can for example use the boundary kernel

$$K_c(x) = \begin{cases} \frac{12}{(1+c)^4}(1+x)\left[x(1-2c) + \frac{3c^2-2c+1}{2}\right], & -1 \le x \le c, \\ 0, & \text{otherwise.} \end{cases} \tag{3.27}$$

from Zhang and Karunamuni, which is a natural continuation of the Epanechnikov kernel (3.26). [19]

The reflection method on the other hand is based on reflecting the data points around both endpoints, and is what we will use in this project to handle problems in the kernel estimation near the endpoints. By using the reflection method the estimator (3.25) becomes

$$\hat{\lambda}(x) = \frac{1}{nh_s} \sum_{i=1}^{r} \left[K\left(\frac{\hat{s}(x) - S_i}{h_s}\right) + K\left(\frac{\hat{s}(x) + S_i}{h_s}\right) + K\left(\frac{\hat{s}(x) + S_i - 2S}{h_s}\right)\right], \tag{3.28}$$

where $S = \sum_{j=1}^{n} T_j/n$. The parameter h_s is a smoothing parameter that corresponds to smoothing over a certain amount of data along the s-axis. On the covariate axis a corresponding parameter h_x, which covers approximately the same data, is defined via the relation between the points on the s-axis and the covariate axis. Generally, if one of the smoothing parameters is held constant, then the other will be varying. Using a constant h_s corresponds to ordinary density estimation on the s-axis. By using a constant value for h_x, (3.28) becomes

$$\hat{\lambda}(x) = \frac{1}{nh_s(\hat{s}(x))} \sum_{i=1}^{r} \left[K\left(\frac{\hat{s}(x) - S_i}{h_s(\hat{s}(x))} \right) + K\left(\frac{\hat{s}(x) + S_i}{h_s(\hat{s}(x))} \right) + K\left(\frac{\hat{s}(x) + S_i - 2S}{h_s(\hat{s}(x))} \right) \right]. \tag{3.29}$$

While Epanechnikov should be a good choice of kernel, the choice of kernel is not as important as the choice of smoothing parameter. The function implementing the covariate order method is given in Appendix B.2.

3.4 Finding the smoothing parameters using cross-validation, and estimating the functional form using the covariate order method

There are two main approaches to finding the smoothing parameter, the plug-in approach, and the classical approach. We will focus on trying to use cross-validation, primarily leave-one-out cross-validation (loocv), which falls under the classical approach. The Covariate Order method is implemented so that it is possible to smooth along either the covariate or the event axis. Since x_1 is simulated from the standard normal distribution, while x_2 is simulated from the exponential distribution with rate $\lambda = 1/2$, and it would be preferable to avoid a constant smoothing parameter on this axis with a varying number of observations in each interval. Since, we will look using the reflection method to handle the boundaries in the density estimation instead of using the boundary kernel in the covariate order method, the cross-validation algorithms we will look at uses reflection when it calls the covariate order function.

The likelihood function for censored survival data without truncation is given by

$$L(\theta; x, \delta_i) = \prod_{i=1}^{n} [f(x_i; \theta)]^{\delta_i}, [H(x_i; \theta)]^{1-\delta_i}. \tag{3.30}$$

where $f(.)$ is the PDF and $H(.)$ is the Survival function. The Cox-Snell residuals are denoted as $\hat{R}_i$, while X_i denotes the covariate used in the Covariate Order method. It follows that the likelihood of the Cox-Snell residuals is

$$L(\lambda(.)) = \prod_{i=1}^{n} [\lambda(X_i) e^{-\lambda(X_i)\hat{R}_i}]^{\delta_i} [e^{-\lambda(X_i)\hat{R}_i}]^{1-\delta_i}. \tag{3.31}$$

This gives the log-likelihood

$$l(\lambda(.)) = \sum_{i=1}^{n} [\delta_i \log \lambda(X_i) - \lambda(X_i)\hat{R}_i]. \tag{3.32}$$

The idea behind using loocv is to use all of the data except for observation i to estimate the hazard function $\lambda(x)$. We then proceed by using using this estimate

of the hazard rate to find the hazard rate of the left out observation i. Let $\lambda^{-i}(x|h)$ denote the hazard rate for the left out observation, this gives the likelihood loocv criterion

$$lCV(h) = \sum_{i=1}^{n} [\delta_i \log \hat{\lambda}^{-i}(X_i|h) - \hat{\lambda}^{-i}(X_i|h)\hat{R}_i]. \tag{3.33}$$

The idea is that the value of h that gives the largest value of the lCV criterion in equation (3.33) should be the "optimal" value of the smoothing parameter h for use in the Covariate Order method function. The code for calculating the lCV criterion for a specified value of h can be found in Appendix B.3.

It was found that the function in Appendix B.3 seems to break down for values of h smaller than 0.3 when applied to the Cox-Snell residuals corresponding to the model fitted to the uncensored data. Using the covariate order method to estimate the functional form of x_2, based on the uncensored dataset, for 4 different values of h is plotted in Figure 3.6.

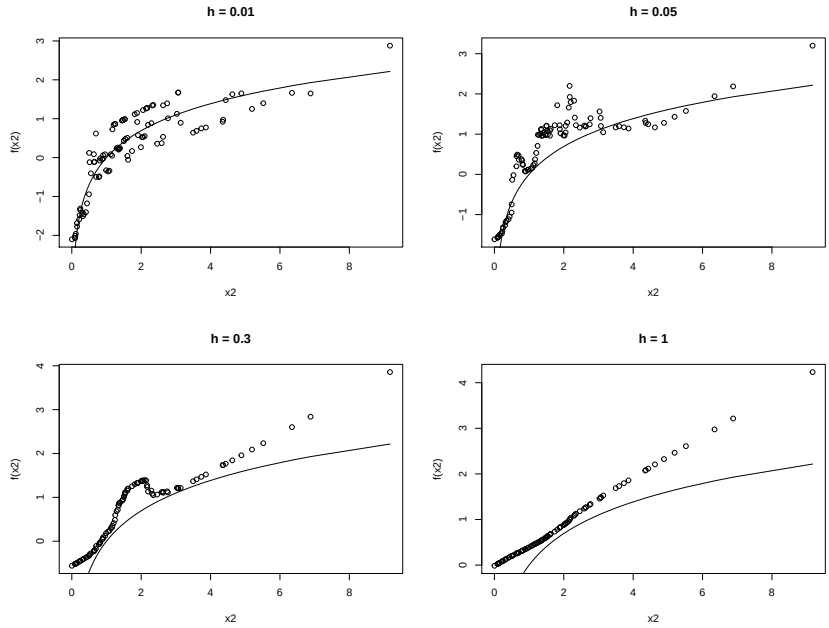


Figure 3.6: Estimated functional form of x_2 using various values for h in the Covariate order function. The points are the estimated functional form, whereas the lines is the true functional form $\log(x_2)$.

From Figure 3.6 we can see that the two upper plots do a good job of finding the functional form of covariate x_2, whereas the two lower plots do not recover the functional form. Thus, it appears that a small value of h is necessary in this case.

Since our implementation of the aforementioned likelihood cross-validation method breaks down for $h < 0.4$, we try a different criterion to find a good value of h.

We will attempt to implement a method that uses the Kernel density estimate to produce a criterion for choosing h instead of using that Cox-Snell residuals follow an exponential distribution if they provide an adequate fit to the model, to propose the aforementioned likelihood criterion. From equation (2) in Ximing Wu (2018), the likelihood cross-validation criterion

$$lCV(h) = \max_h \frac{1}{n} \sum_{i=1}^{n} \ln \hat{f}_i(h), \tag{3.34}$$

is provided. $\hat{f}_i(h) = 1/(n-1)\sum_{j \neq i} K_h(X_i - X_j)$ is here the leave-one-out density estimate. [18] Where observation i is left out of the density estimation. We can apply equation (3.34) to the kernel estimate of the S_i referred to in section 2.9. We estimate the density r times, where r is number of S values. n in (3.34) is the number of observations in the dataset. It follows that the Kernel estimate of the S_i values using the reflection method becomes

$$\hat{f}_i(h) = \frac{1}{(n-1)h_s} \sum_{j \neq i} \left[K\left(\frac{S_i - S_j}{h_s}\right) + K\left(\frac{S_i + S_j}{h_s}\right) + K\left(\frac{S_i + S_J - 2S}{h_s}\right) \right]. \tag{3.35}$$

h as a function of the lCV criterion for covariate x_2 in (3.34) is plotted for the four datasets in Figure 3.7.

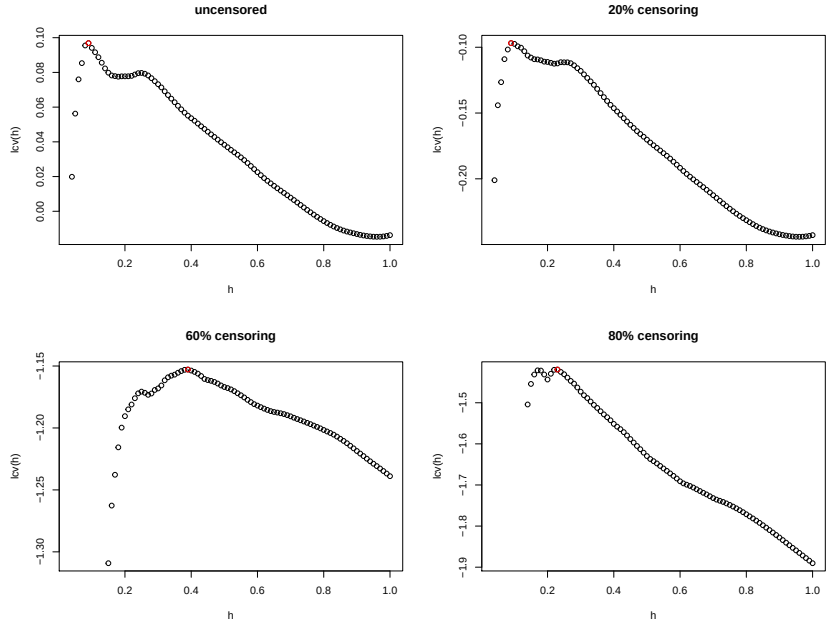


Figure 3.7: cross-validation criterion $lCV(h)$ against h for the covariate order method applied to the four simulated datasets. The red dots are the maximum lCV values.

The maximum value of the cross-validation criterion and the corresponding h for x_2 is given in Table 3.2

	uncensored	20%	60%	80%
$lCV(h)$	0.096869771	0.114686311	0.005902621	0.157389683
h	0.09	0.08	0.09	0.04

Table 3.2: Maximum value for $lCV(h)$ with corresponding h for x_2 with no censoring, 20%, 60%, and 80% censoring.

Using the values of h in Table 3.2 in the covariate order function to estimate the hazard rates and using this estimate of the hazard rate in equation (3.24), gives the functional form of x_2 in Figure 3.8.

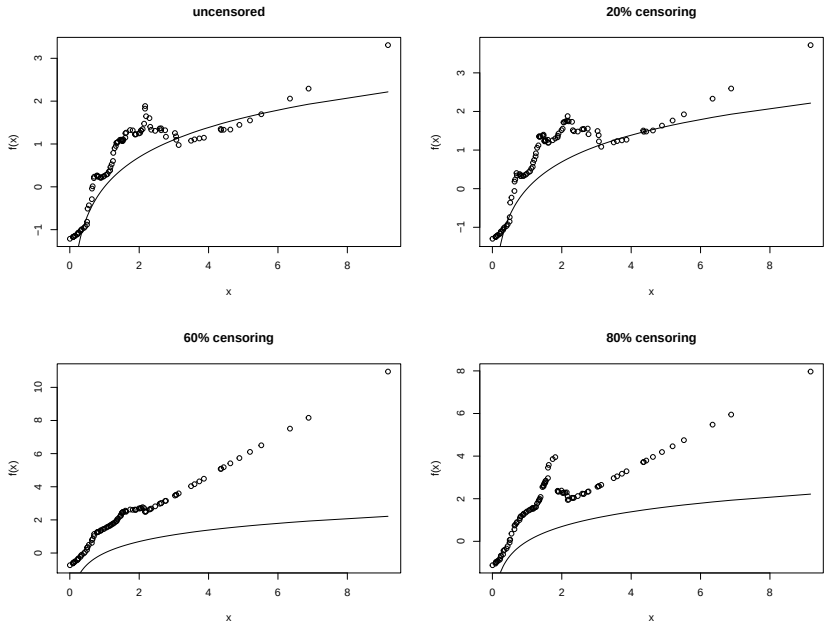


Figure 3.8: Estimated functional form of x_2 for the data with no censoring, 20%, 60%, and 80% censoring. The points are the estimated functional form, whereas the solid lines are the true functional form $\log(x_2)$.

Figure 3.8 indicates that the functional form of x_2 for the uncensored case and the case with 20% censoring is $\log(x_2)$. From the two lower plots showing the functional form for the case with 60% and 80% censoring, it is hard to say that the true functional form is logarithmic, as was the case for the method of estimating the functional form presented in section 3.2. For the two high censoring cases the plot of the functional form appears to be fairly linear for values of x_2 over 2. The reason for why the functional form appears to be linear for $x_2 > 2$ for the two high censoring cases in Figure 3.8 might not only be the high number of censored observations, but due to the fact that most of the uncensored residuals in Figure 3.3 and Figure 3.4 correspond to $x_2 < 2$. For $x_2 < 2$ in the high censoring cases in Figure 3.8, the curve is slightly displaced, but the functional form does look as if it could be logarithmic.

Looking at the estimated hazard $\hat{\lambda}(x_2)$ computed by the covariate order method can give an indication of model fit and trends in the hazard. Figure 3.9 shows the estimated hazard against x_2.

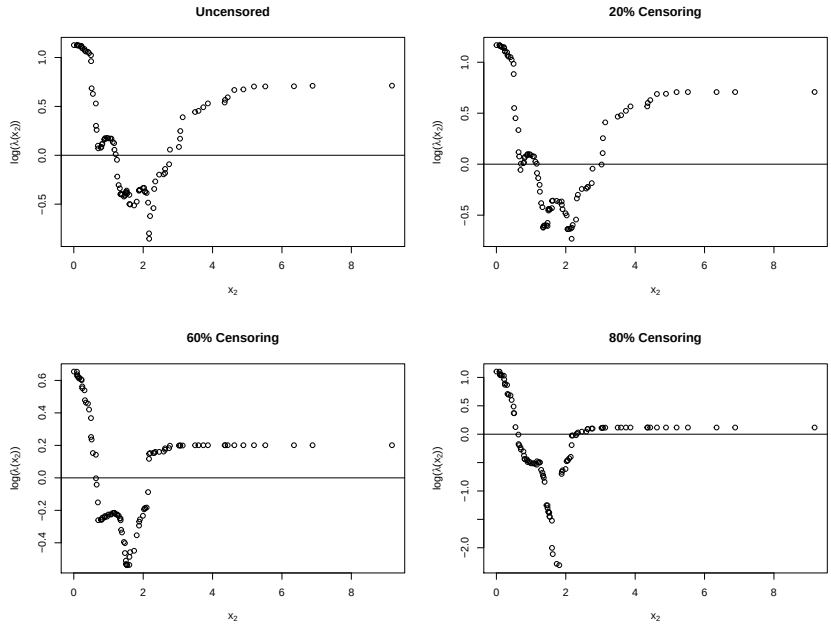


Figure 3.9: Log of the estimated hazard, $\hat{\lambda}(x_2)$, as a function of the covariate x_2 for the data with no censoring, 20%, 60%, and 80% censoring.

If the fitted linear model is correct, the logarithm of the hazard would be expected to be around zero. For no censoring and 20% censoring there is a decreasing trend in the hazard until approximately 2 on the covariate axis. For $x_2 > 2$ there is an increasing trend until the hazard levels out and is approximately constant. For the two high censoring cases there is a similar pattern, but the hazard is not as turbulent and it levels out closer to zero and becomes constant.

3.5 Testing for covariate effect

If there was no trend in the hazard ($\lambda(x) \equiv \lambda$), there would be no covariate effect, and the artificial point process $S_1, S_2, \ldots, S_2$ is a homogeneous Poisson process. This suggests that covariate effect can be tested by any statistical test that tests for the Null hypothesis of a HPP versus the alternative hypothesis of any variant of NHPP. In Kvaløy (2002), Kvaløy outlines various statistical tests, constructed based on the covariate order method, that tests for covariate effect in survival data.[12] Kvaløy recommends to use an Anderson Darling type test since it shows to have good properties for both monotonic and non-monotonic alternatives to a constant hazard. The AD test for trend constructed based on the covariate order method is

$$AD = -\frac{1}{\hat{r}}\left[\sum_{i=1}^{\hat{r}}(2i-1)(\ln\frac{S_i}{S} + \ln(1-\frac{S_{\hat{r}+1-i}}{S}))\right] - \hat{r}, \tag{3.36}$$

where $S = \sum_{i=1}^{n} T_i/n$, and $\hat{r}$ is defined as

$$\hat{r} = \begin{cases} r & \text{if } S_r < S \\ r-1 & \text{if } S_r = S \end{cases} \tag{3.37}$$

The Anderson Darling statistic for the estimated hazard $\hat{\lambda}(x_2)$ in the four datasets is computed by equation (3.36) to be

$$\begin{aligned} AD_{x_2} &= 7.677 \text{ for uncensored,} \\ AD_{x_2} &= 7.432 \text{ for } 20\% \text{ censoring,} \\ AD_{x_2} &= 3.003 \text{ for } 60\% \text{ censoring,} \\ AD_{x_2} &= 6.968 \text{ for } 80\% \text{ censoring.} \end{aligned} \tag{3.38}$$

For datasets with few lifetimes Kvaløy states that the level properties of the AD test can be improved by using resampling techniques such as bootstrap or permutation methods. [12] However, using the asymptotic null distribution of equation (3.36) gives a more conservative result for small sample sizes than one would expect to get by using resampling methods. Still, the asymptotic distribution is a good approximation to the real distribution for sample sizes as small as $n = 10$. The asymptotic null distribution of the AD statistic was derived by Anderson and Darling (1952). [8] The null hypothesis of no covariate effect is rejected at a 5% significance level if $AD \geq 2.492$. It follows that the null hypothesis is rejected for all the four cases, and thus we conclude that there is a significant trend in the hazard. It follows that the Cox-Snell residuals are dependent on the covariate x_2 and we can conclude that the fitted model is not correct.

Chapter 4

Popes data

We now turn to a real dataset that we will refer to as the popes data. The dataset consists of post-election survival times for all 62 popes starting from and including Pope Innocent VII, who began his pontificate in 1404, up until, and not including, Pope Francis who began his pontificate in 2013 and is at the time of writing this project still the pope. [17] The dataset can be found in Appendix A.2. The 15th century was chosen as the starting point since date of birth, age of election, year of election, and death (resignation) is accurately documented from this point on wards. [17] All popes aside from Pope Gregory XII who resigned in 1415 and died in 1417, and Pope Emeritus Benedict XVI who resigned on 28 February 2013, died in office. Pope Emeritus Benedict XVI is still alive at the time of writing this project, and he is the only censored observation in the dataset. The version of the dataset that is being analyzed is from the 25th of December 2016, and at that time Pope Emeritus Benedict XVI has a post-election survival time of 11.7 years.

We will look at fitting a model including 2 explanatory variables. The first being at what age were the popes elected (Age.Election), and the second being in which year were they elected (Year.Elected). The median age of election is 63.5 years, while the median post election survival time of the 61 popes (Benedict XVI excluded) is 9 years.

4.1 Analysis of popes data

Fitting a linear Weibull AFT model to the popes data with *Age.Election* and *Year.Elected* denoted as the covariates x_1 and x_2, respectively, gives the summary

```
Call:
survreg(formula = surv_obj ~ x1 + x2, data = popes, dist = "weibull")
                Value Std. Error     z      p
(Intercept)  0.975920   1.226590  0.80 0.4262
x1          -0.027200   0.011484 -2.37 0.0179
x2           0.001807   0.000661  2.73 0.0063
Log(scale)  -0.133331   0.111860 -1.19 0.2333
```

```
Scale= 0.875

Weibull distribution
Loglik(model)= -196.6   Loglik(intercept only)= -201.5
Chisq= 9.74 on 2 degrees of freedom, p= 0.0077
Number of Newton-Raphson Iterations: 7
n= 62
```

From the summary we observe that both the covariates x_1 and x_2 are statistically significant at a 5% significance level. By using equation (2.21) we calculate the Cox-Snell residuals and plot them as a function of the covariates in Figure 4.1.

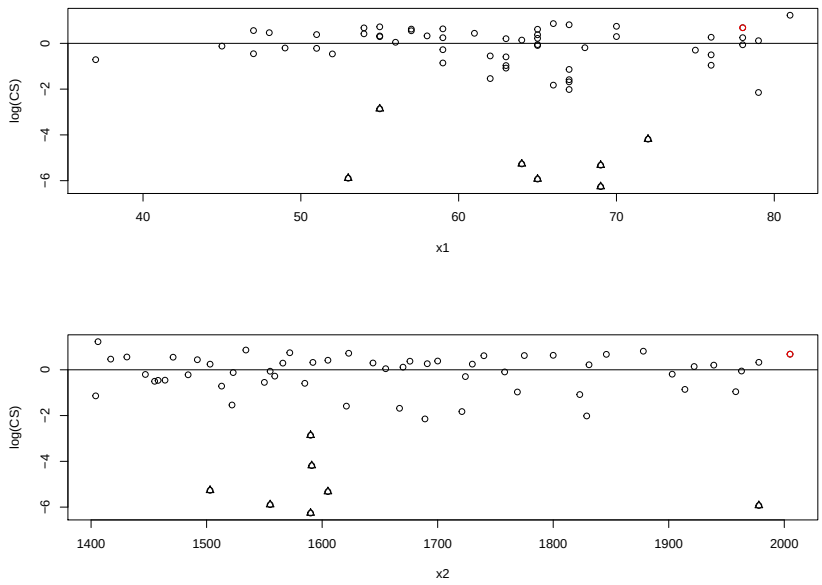


Figure 4.1: Log of Cox-Snell residuals versus the covariates. The red dot is the residual corresponding to Pope Emeritus Benedict XVI, which is censored and has been 1-adjusted. The triangles correspond to the popes that died withing 1 year of election.

Looking at the Residual plots in Figure 4.1 the triangles are clear outliers. These Residuals correspond to popes that died within a year of being elected. Aside from Pope John Paul I who died in 1978, all the popes that died in less than a year were popes in the 17th and 18th century. Pope John Paul I died of a heart attack September 1978, 33 days after being elected pope. [3] Pope Leo XI died at the age of 70, of fatigue 27 days after being elected in 1605. The 5 other popes who died in the 17th century are reported to have died of an illness of some sort. Pope Gregory

who was pope for approximately 10 months and died at age 56 from complications with gallstones. [2] Pope Innocent IX died of a fever 2 months after being elected pope, whereas the 3 other short lived popes of the 17th century all died within a month of being elected. Aside from the outliers, there does not appear to be much of pattern present in the residuals.

To estimate a functional form for the covariates, we first look at using the method in Section 3.2, more specifically equation (3.17). Using the estimates in the summary $\hat{\sigma} = 0.875$, $\hat{\beta}_2 = -0.0272$, and $\hat{\beta}_3 = 0.001807$, along with an estimate $\hat{H}(x)$ and inserting into equation (3.17), gives the estimated functional forms in Figure 4.2 for the two covariates. $\hat{\beta}_2$ and $\hat{\beta}_3$ correspond to γ for x_1 and x_2 in equation 3.17, respectively. $\hat{H}(x)$ is estimated by smoothing log of the fitted Cox-Snell residuals versus the covariates by using the *loess* function in R with span = 2.

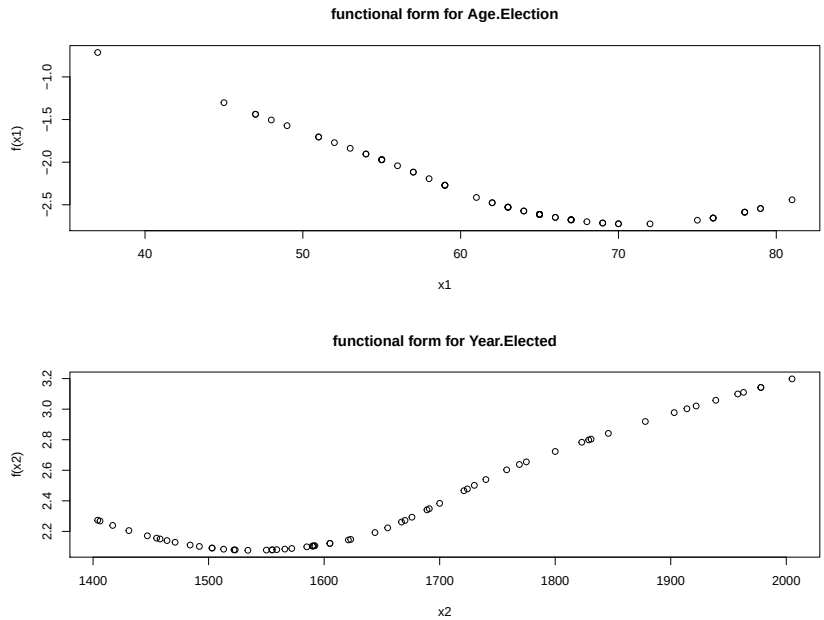


Figure 4.2: Estimated functional form for covariates x_1 (top) and x_2 (bottom) using equation (3.17).

The estimated functional form of x_1 in Figure 4.2 is linearly decreasing until approximately 70, after which the functional form becomes close to constant. For x_2 the functional form appears to be close to constant until around 1600, and then it linearly increases.

Moving on we look at using the covariate order method on the Cox-Snell residuals of the fitted model in order to get an estimate of the hazard $\hat{\lambda}(x)$, which can in turn be used to estimate the functional form by equation (3.24). To choose a smoothing parameter h, calculate and plot the lCV criterion in equation (3.35) as

a function of h.

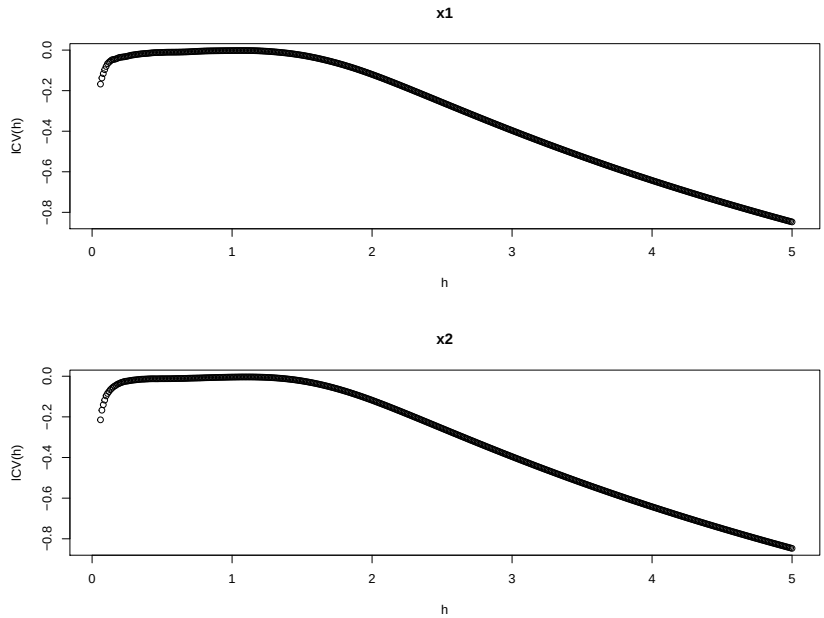


Figure 4.3: cross-validation criterion $lCV(h)$ against h for the 2 covariates x_1 (top) and x_2 (bottom).

In Figure 4.3 the value of h corresponding to the maximum value of lCV for the two covariates is found to be 1.08 and 1.09 respectively. Observe that the curve in Figure 4.3 is fairly flat for values of h in the interval 0.3 to around 1.3. Since the value of the cross validation criterion does not change much in this interval we choose a smaller value of h on this interval in addition to ones corresponding to the maximum of the cross-validation criterion. The reason for this is that a smaller value of h will not smooth as much as a larger value of h and might give some valuable insight into the functional form.

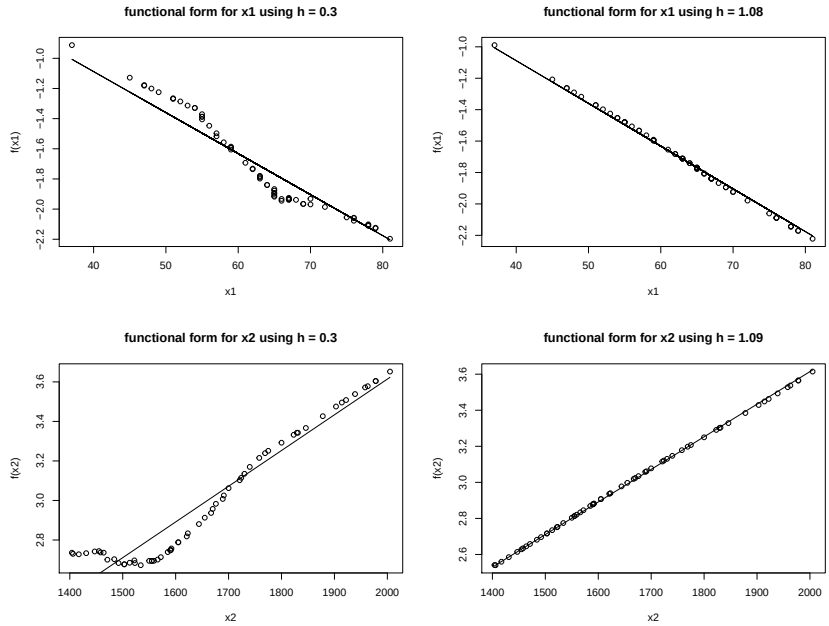


Figure 4.4: Estimated functional form for covariates x_1 (top) and x_2 (bottom). The solid line is the estimated linear covariate function $\hat{\beta}_2 x_1$ for the upper plots, and $\hat{\beta}_3 x_2$ for the bottom plots. on the left-hand side, the functional forms are estimated using $h = 0.3$ in the covariate order function to estimate the hazard. On the right-hand side, the functional forms are estimated using the values of h corresponding to the maximum values of lCV mentioned previously.

Figure 4.4 shows the estimated functional form of x_1 and x_2. The functional form is estimated by using the covariate order method to estimate the hazard rate and inserting this estimate into equation (3.24) along with the estimated parameter values in the summary for the model fitted to the popes data. We use the reflection method to handle the boundaries in the kernel density estimation in the covariate order, and we smooth along the event-axis using $h = 0.3$ and h equals to the value corresponding to the maximum value of the lCV.

Figure 4.4 shows that for the values of h corresponding to the maximum value of the cross validation criterion, the functional form appears to be linear. The estimated functional forms for the covariates coincides with the solid lines $\hat{\beta}_{i+1} x_i$, where $\hat{\beta}_{i+1}$ is the estimated regression coefficient corresponding to covariate x_i. According to equation (3.17) you would expect that if the fitted model is a good fit, then $\hat{H}(x) \approx 0$, and the estimated functional form is approximately the one estimated in the fitted model, $\hat{f}(x_i) \approx \hat{\beta}_{i+1} x_i$ for covariate x_i.

For the case with smoothing parameter $h = 0.3$ in Figure 4.4 the true functional forms of x_1 and x_2 could be close to linear. However, looking at the functional form of x_2 there might a better functional form than the linear one. Observe that for

x_2 the functional form is approximately constant prior to 1600, and it looks to linearly increasing after this point. A suggestion for the functional form of x_2 could be something along the lines of

$$f(x_2) = \begin{cases} \alpha, & \text{for } 0 \leq x_2 \leq 1600 \\ \alpha + \beta(x_2 - 1600), & \text{for } 1600 < x_2. \end{cases} \tag{4.1}$$

α and β in equation 4.1 are constants. To check whether or not equation 4.1 is better than the estimated linear covariate function for x_2 we can transform the data and fit a new linear model in *survreg* and analyze this. But first we look for any trend in the estimated hazards.

Figure 4.5 shows plots of the log of the estimated hazards $\hat{\lambda}(x_1)$ and $\hat{\lambda}(x_2)$ versus the covariates x_1 and x_2, respectively.

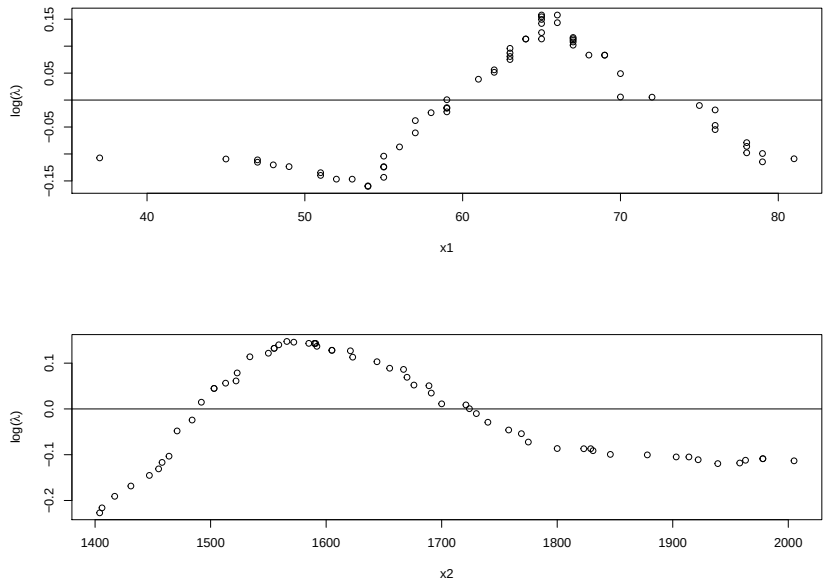


Figure 4.5: Log of the estimated hazard, λ, as a function of the covariates x_1 (top) and x_2 (bottom).

From Figure 4.5 there looks as if there is an increasing trend for values of x_1 from 54 until approximately 66. For values of x_1 larger than 66 there appears to be a decreasing trend. However, the values on the $\log(\lambda)$ axis span -0.15 to 0.15, so $\log(\lambda)$ is fairly close to zero which hints at a good model fit. Similarly for x_2 the points lie close to zero, which hints at a good model. For x_2 there looks as if there is a small increasing trend from 1400 until around 1570/1580. After this point there is a small decreasing trend until approximately 1800. To test more rigorously for

covariate effect we apply the Anderson Darling test for covariate effect presented in section 3.5.

Using equation (3.36) the AD statistic for x_1 and x_2 are computed to be

$$\begin{aligned} AD_{x_1} &= 0.5355, \\ AD_{x_2} &= 0.4204. \end{aligned} \tag{4.2}$$

The null hypothesis of no significant trend in the hazard is rejected at a 5% significance level if $AD_{x_i} \geq 2.492$. It follows that we keep the null hypothesis for both covariates since the computed statistics in equation (4.2) are significantly smaller than the one for the null hypothesis. Thus, the conclusion is that there are no significant trends in the two hazards. It follow that the Cox-Snell residuals for the model fitted to the popes data are independent of the covariates, and the fitted linear model is believed to be a good model.

While the linear model appears to be a good model, applying some transformations to the covariates and fitting a new model could potentially produce a better model. For example the functional form for x_2 (Year.elected) proposed in equation (4.1), might improve upon the linear AFT model. To check whether or not the transformation will yield a better model we can for example look at the Akaike information criterion (AIC) of the original linear model, and the model where x_2 is transformed by equation (4.1). The AIC value for a model is

$$\text{AIC} = 2k - 2\log(\hat{L}), \tag{4.3}$$

where k is the number of estimated parameters in the model, and $\log(\hat{L})$ is the estimated log-likelihood of the model. Since the original linear model and the model where x_2 is transformed by equation (4.1), will have the same estimated parameters k, comparing the model using AIC is reduced to choosing the model with The largest value of $\log(\hat{L})$.

The summary of an AFT model using survreg contains an estimated value $\log(\hat{L})$. Fitting a new model where x_2 is transformed by equation (4.1), gives the following summary.

```
> summary(popes_model2)

Call:
survreg(formula = surv_obj2 ~ x1 + x2T, data = popes, dist = "weibull")
               Value Std. Error     z       p
(Intercept)  3.739718   0.726501  5.15 2.6e-07
x1          -0.027767   0.011633 -2.39  0.0170
x2T          0.002671   0.000944  2.83  0.0047
Log(scale)  -0.140063   0.111794 -1.25  0.2103

Scale= 0.869

Weibull distribution
```

```
Loglik(model)= -196   Loglik(intercept only)= -201.5
Chisq= 11.01 on 2 degrees of freedom, p= 0.0041
Number of Newton-Raphson Iterations: 7
n= 62
```

From the summary of the original model in the beginning of this section we observe the value $\log(\hat{L}) = -196.6$. The estimated log-likelihood of the new model is $\log(\hat{L}) = -196$. Since the log-likelihood for the new model is slightly higher than the original model, we conclude that the new model is slightly better than the old. Thus, we observe that even if there is no significant trend in the estimated hazard, estimating the functional form can still lead to clues on how to improve upon a model.

Chapter 5

PBC data

Moving on we will have a look at a well-known dataset from the Mayo Clinic trial on Primary biliary cirrhosis (PBC) of the liver. The trial was conducted between 1974 and 1984. The trial consists of 424 patients who between the 10 year interval met the eligibility criteria for this randomized, double-blinded, placebo controlled trial of the drug D-penicillamine (DPCA). [5] The first 312 patients in the dataset agreed to take part in the trial, and for these patients. histologic, clinical, serologic, and biochemical parameters were recorded. The remaining 112 patients did not agree to take part in the trial, but they did however agree to have basic measurements recorded and to be followed for survival. Six of these 112 individuals were lost in the follow up and are excluded from the data. The data consists of the 312 patients that took part in the trial, along with the 106 patients that agreed to take basic measurements. For the 312 patients that took part in the trial the follow-up lasted until July 1986. At which point 125 of the 312 had died, and only 11 of these deaths were not due to PBC. 8 of these 312 were lost in the follow-up and are censored in the data, while 19 of them received a liver transplant.

PBC is a rare type of fatal chronic liver disease, estimated to only occur in approximately 50 per 1 million population. For a patient with PBC the immune system mistakenly attacks the bile ducts, which leads to the bile ducts becoming injured or damaged. This causes bile to build up in the liver and can lead to scarring of the liver (cirrhosis), or liver failure if the illness is left untreated. [4] The Mayo clinic trial on DPCA established that DPCA is not an effective treatmeant of PBC. Until recent times treatment for PBC was limited to supportive care. Today, PBC is mainly treated with a drug called UDCA, and in severe casses it requires a live transplant. While DPCA did not show to be effective, the data that was gathered during the trial is still very valuable to specialists working with liver disease, and here in this project we will look at trying to find the functional forms of some of the variables in the dataset.

5.1 Analysis of PBC data

The PBC dataset can be found by typing *pbc* in the *survival* library in R. The Table 5.1 contains all the variables in the full dataset, with a short description of them.

Variable	Description
1. age	in years
2. albumin	serum albumin (g/dl)
3. alk.phos	alkaline phosphotase (U/liter)
4. ascites	presence of ascites
5. ast	aspartate aminotransferase (U/ml)
6. bili	serum bilirubin (mg/dl)
7. chol	serum cholesterol (mg/dl)
8. copper	urine copper (ug/day)
9. edema	0 no edema, 0.5 untreated or successfully treated, 1 edema despite diuretic therapy
10. hepato	presence of hepatomegaly or enlarged liver
11. id	case number
12. platelet	platelet count
13. protime	standardised blood clotting time
14. sex	m/f
15. spiders	blood vessel malformations in the skin
16. stage	histologic stage of disease (needs biopsy)
17. status	status at endpoint, 0/1/2 for censored, transplant, dead
18. time	number of days between registration and the earlier of death, transplantion, or study analysis in July, 1986
19. trt	1/2/NA for D-penicillmain, placebo, not randomised
20. trig	triglycerides (mg/dl)

Table 5.1: Variables in the PBC dataset

The final model proposed by Fleming and Harrington (1991) consists of 5 of the variables in 5.1 that were shown to be significant. [10] Thus, we will limited ourselves to only looking at models including these five variables. The covariates in the final model proposed by Fleming and Harrington was age, edema, log(bilirubin),

log(protime), and log(albumin). We will start with a linear Weibull AFT model consisting of the covariates age, edema, bilirubin, protime, and albumin. In addition, reduce the dataset to only contain the 312 individuals that agreed to take part in the trial, since these are complete observations where no variable values are missing. The fitted model is

$$\begin{aligned}\log Y = \beta_0 + \beta_{age} \cdot x_{age} + \beta_{edema} \cdot x_{edema} + \beta_{bili} \cdot x_{bili} \\ + \beta_{protime} \cdot x_{protime} + \beta_{albumin} \cdot x_{albumin} + \sigma W.\end{aligned} \tag{5.1}$$

The parameter estimates in equation (5.1) are found in the following summary of the fitted model.

```
> summary(pbc_model)

Call:
survreg(formula = surv_obj ~ PBC$age + PBC$edema + PBC$bili +
    PBC$protime + PBC$albumin, data = PBC, dist = "weibull")
               Value Std. Error     z       p
(Intercept)  8.90246    0.78129 11.39 < 2e-16
PBC$age     -0.02120    0.00602 -3.52 0.00043
PBC$edema   -0.58436    0.19863 -2.94 0.00326
PBC$bili    -0.07388    0.00913 -8.09 5.7e-16
PBC$protime -0.18144    0.04859 -3.73 0.00019
PBC$albumin  0.77625    0.14695  5.28 1.3e-07
Log(scale)  -0.42850    0.07237 -5.92 3.2e-09

Scale= 0.651

Weibull distribution
Loglik(model)= -1105.4   Loglik(intercept only)= -1188.8
Chisq= 166.79 on 5 degrees of freedom, p= 3.5e-34
Number of Newton-Raphson Iterations: 6
n= 312
```

To compute the Cox-Snell residuals for the fitted model, insert the lifetimes Y_i along with the parameter estimates for the regression coefficients $\hat{\beta}_i$, and the scale parameter $\hat{\sigma}$, in the above summary into equation (3.4). Of the 312 observations used to fit the model, only 125 are uncensored, corresponding to 60% censoring. We plot the log of the 1-adjusted Cox-Snell residuals as a function of the covariates to get an idea of the model fit, and if the covariates are appropriately represented in the model.

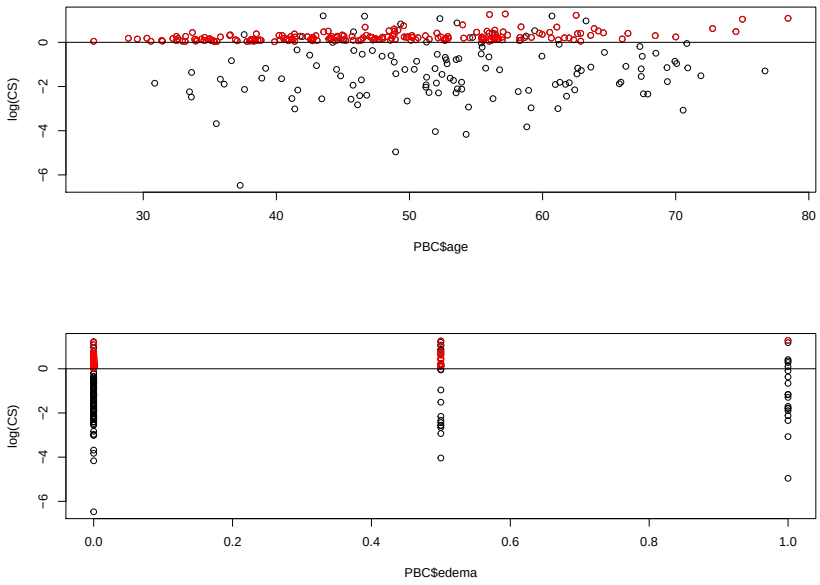


Figure 5.1: Log of 1-adjusted Cox-Snell residuals versus the covariates age (top), and edema (bottom). The red dots show the censored residuals which have been adjusted by adding 1.

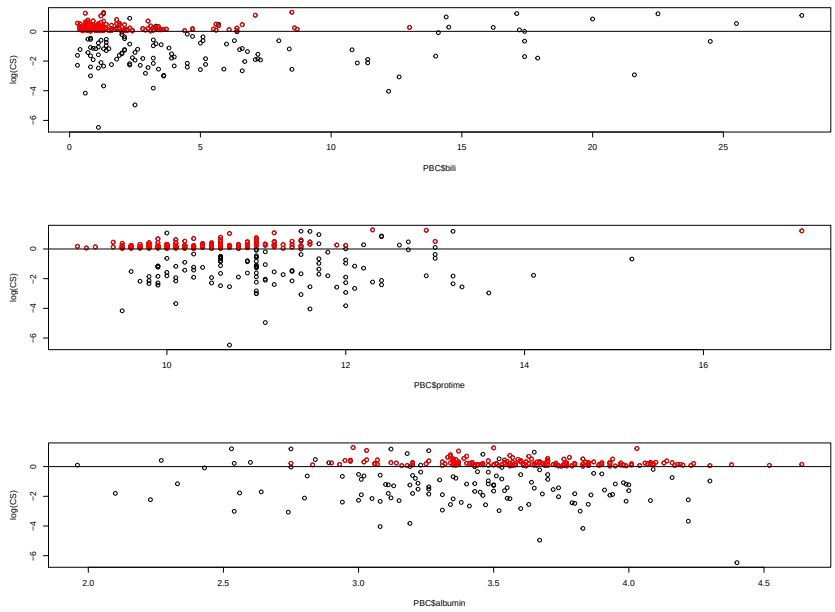


Figure 5.2: Log of 1-adjusted Cox-Snell residuals versus the covariates bilirubin (top), protime (middle), and albumin (bottom). The red dots show the censored residuals which have been adjusted by adding 1.

From the top residual plot in Figure 5.1 we observe that the value of the log of the 1-adjusted Cox-Snell residuals are fairly low. However, there is no apparent pattern in them. Thus, we believe that the age covariate is well represent as a linear term in the model. In the bottom residual plot of 5.1 we observe the effect of the discrete covariate edema on the residuals. From this plot there is no indication that the residual distributions for the three covariate values of edema deviate from one another. Thus, it is believed that the edema covariate is not misspecified in the fitted model.

Looking at the residual plot for the bilirubin covariate (bili) in Figure 5.2, observe that most residuals are found for lower values of bilirubin, and it appears that there could be a pattern in log of the Cox-Snell residuals as a function of bilirubin. In the middle residual plot in Figure 5.2 there appears to be a some outliers for higher values of the protime covariate, and while it is difficult to observe a clear pattern in the 1-adjusted residuals, the residuals do not appear to completely symmetric with respect to zero on the residual-axis. Consequently, protime could potentially be modelled slightly better. For the lower residual plot of the albumin covariate it is hard to observe any pattern or asymmetries.

We will try to estimate the functional form of the age, bilirubin, protime, and albumin covariates, using the covariate order method. To start we compute the cross-validation criterion in equation (3.35) for a vector of h values, for each of the

four covariates. This is to get an idea of what smoothing parameters to use in the covariate order function in each case. Figure 5.3 shows plots of the cross-validation criterion as a function of h for the four covariates. Table 5.2 shows the maximum value of the cross-validation criterion with the corresponding value of h for the four covariates. Since the edema covariate was found to be well represented in the fitted model, and the covariate order method cannot be used for discrete covariates, we choose to not estimate the functional form for this covariate. To see how to estimate the functional form for a discrete covariate such as edema, see Kvaløy, Lindqvist, and Aaserud (2015). [14]

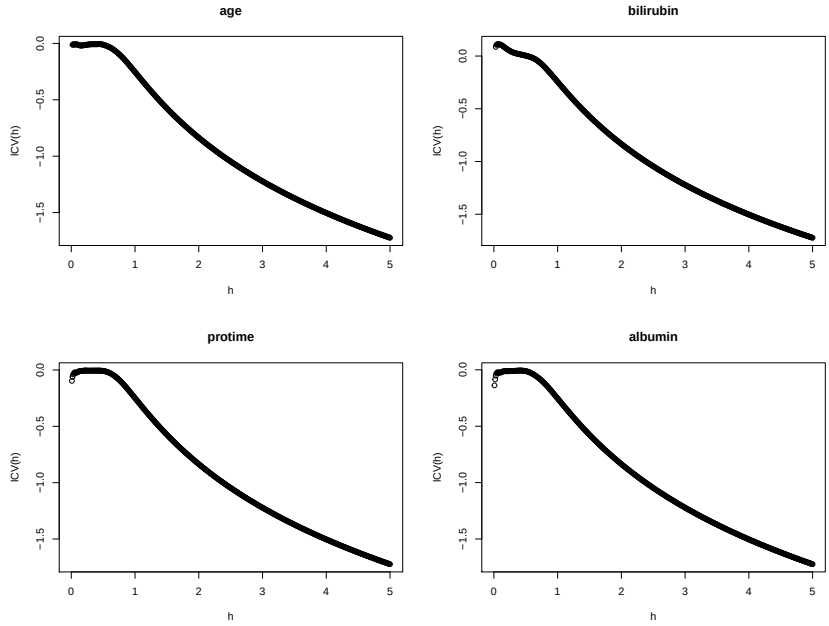


Figure 5.3: cross validation criterion (lCV) as a function of h for the four covariates age, bilirubin, protime, and albumin.

	age	bilirubin	protime	albumin
$lCV(h)$	-0.006167469	0.113054127	-0.004831749	-0.006493977
h	0.03	0.07	0.22	0.41

Table 5.2: Maximum value for $lCV(h)$ with corresponding h for the four covariates age, bilirubin, protime, and albumin.

Observe from Figure 5.3 that $lCV(h)$ is flat for a large number of small values of h, for all covariates but bilirubin, which has a clear global maximum before

$lCV(h)$ decreases. Thus, choose the value $h = 0.07$ corresponding to the maximum value of lCV as a smoothing parameter for the bilirubin covariate in the covariate order function. For the age covariate we also select the smoothing parameter corresponding to the value of lCV, $h = 0.03$. In the case of the protime and albumin covariates we observe the values $h = 0.22$ and $h = 0.41$ in Table 5.2, respectively. Since the curves in Figure 5.3 are flat for values of h from approximately 0.05 until around $h = 0.5$, we choose to use $h = 0.05$ instead of the values corresponding to the global maximum for these two covariates, this because we want to capture more of the variance in the estimated functional form. We choose the aforementioned values of h as smoothing parameters in covariate order function, where we smooth along the event-axis, and use reflection to handle boundary problems in the kernel estimation. The resulting estimates of the corresponding hazards are inserted into equation (3.24) along with the estimated parameter values in the summary for the fitted model. This gives the estimated functional forms for the four covariates in Figure 5.4

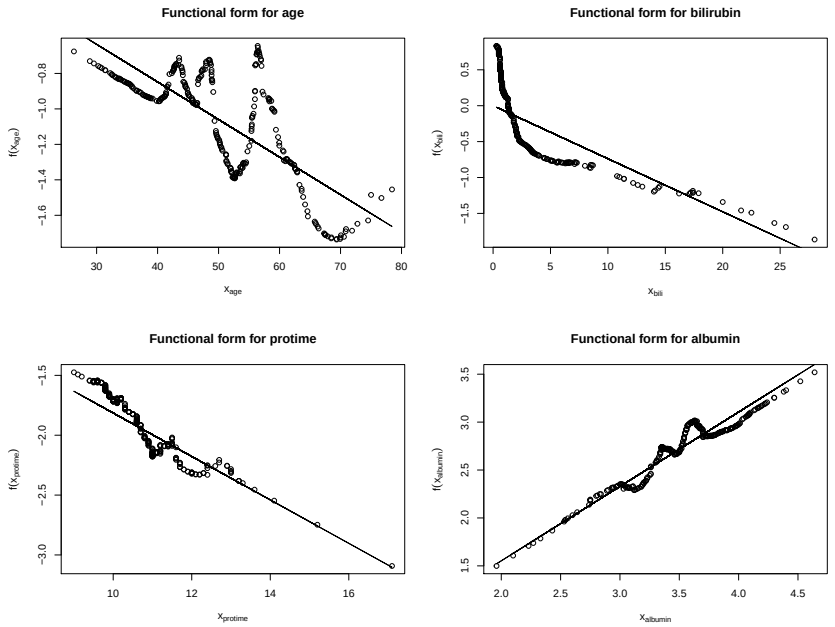


Figure 5.4: The dotted plots are the estimated functional forms for the four covariates age, bilirubin, protime, albumin, using the covariate order method. The solid black lines are the linear lines $\hat{\beta}_i x_i$, $i \in \{\text{age, bili, protime, albumin}\}$.

The estimated functional form for age in Figure 5.4 looks to fluctuate over and under the linear solid line after age 40, but it hard to see if there is a clear underlying functional form for this covariate, or if the linear functional form is sufficient. For the bilirubin covariate the functional form clearly appears to be logarithmic. In the

case of the protime and albumin covariates, their linear functional forms appear to be sufficient. To look more into whether or not the covariates are well represented in the fitted model, we plot the logarithm of the estimated hazard rates as a function of the covariates in Figure 5.5. We also compute the Anderson Darling test statistic for covariate effect in equation (3.36). Table 5.3 contains the computed AD test statistics for the four covariates.

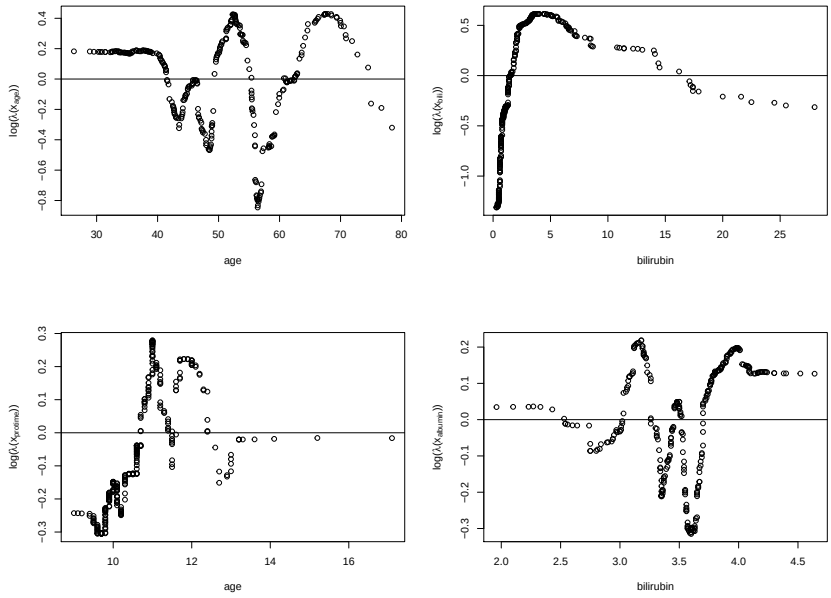


Figure 5.5: Plot of the log of the estimated hazard rates of the Cox-Snell residuals in the linear model versus the covariates.

	age	bilirubin	protime	albumin
AD statistic	0.5789	8.569	0.9311	0.3511

Table 5.3: Maximum value for $lCV(h)$ with corresponding h for the four covariates age, bilirubin, protime, and albumin.

From Figure 5.5 there is a large increasing trend for the bilirubin covariate. For the 3 other covariates there are both some increasing and decreasing trends. From Table 5.3, all of the statistics aside from bilirubin is smaller than 2.492. Thus, only the trend for the bilirubin covariate is significant at a 5% significance level.

Since we discovered that the functional form of bilirubin appears to be logarithmic we fit a new model

$$\begin{aligned}\log Y = \beta_0 + \beta_{age} \cdot x_{age} + \beta_{edema} \cdot x_{edema} + \beta_{bili} \cdot \log(x_{bili}) \\ + \beta_{protime} \cdot x_{protime} + \beta_{albumin} \cdot x_{albumin} + \sigma W,\end{aligned} \tag{5.2}$$

using *survreg* in R. *survreg* only takes linear covariates, thus we apply a logarithmic transformation to the bilirubin covariate in the data set and fit a linear Weibull AFT model to the data. The summary for the fitted model is as follows.

```
> summary(pbc_model2)

Call:
survreg(formula = surv_obj ~ PBC$age + PBC$edema + PBC$bili +
    PBC$protime + PBC$albumin, data = PBC, dist = "weibull")
              Value Std. Error     z       p
(Intercept)  9.4719     0.8219 11.52 < 2e-16
PBC$age     -0.0207     0.0054 -3.83 0.00013
PBC$edema   -0.5585     0.1809 -3.09 0.00201
PBC$bili    -0.5314     0.0586 -9.07 < 2e-16
PBC$protime -0.1628     0.0529 -3.08 0.00208
PBC$albumin  0.5744     0.1419  4.05 5.2e-05
Log(scale)  -0.4731     0.0709 -6.68 2.5e-11

Scale= 0.623

Weibull distribution
Loglik(model)= -1088.7   Loglik(intercept only)= -1188.8
Chisq= 200.05 on 5 degrees of freedom, p= 2.8e-41
Number of Newton-Raphson Iterations: 6
n= 312
```

To see if the new model is an improvement upon the first we can compare the hazards for $x_{bilirubin}$, and $\log(x_{bilirubin})$. The hazard for $\log(x_{bilirubin})$ is estimated using the covariate order method with $h = 0.16$ found by using the cross-validation criterion in equation (3.35). Figure 5.6 shows the estimated hazard rates for bilirubin and log of bilirubin.

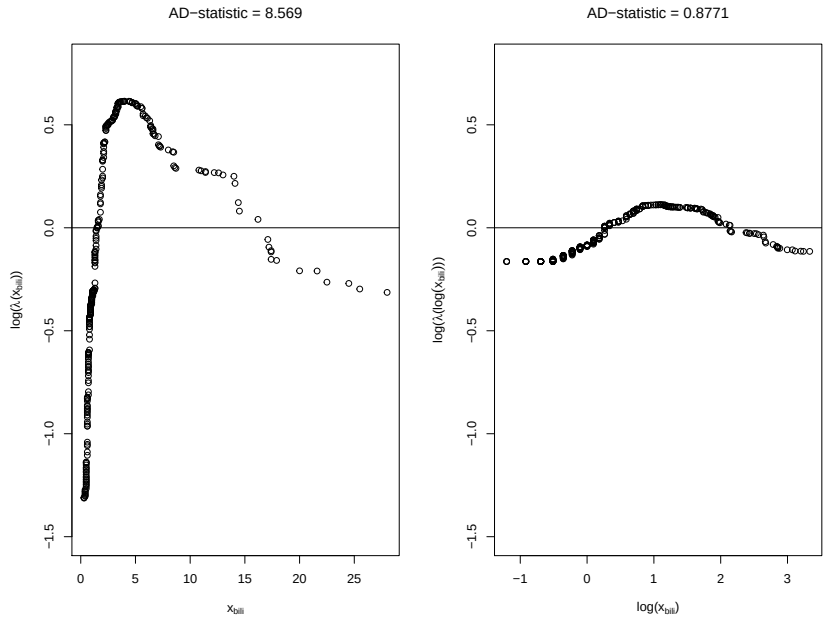


Figure 5.6: Plot of the log of the estimated hazard rates of the Cox-Snell residuals against the first model using only bilirubin on its regular scale (left), and a model using log of bilirubin (right).

From Figure 5.6, observe from the right-hand side that most of the trend has disappeared, and the covariate is much better modelled by using log(bilirubin) in the model. From the AD-statistic for log(bilirubin) in Figure 5.6 observe that it is smaller than 2.492, so the trend is not significant at a 5% significance level. Thus, the bilirubin covariate is much better modelled as log(bilirubin).

In the final model proposed by Fleming and Harrington (1991), the five variables age, edema, bilirubin, protime, and albumin, where as mentioned previously, modelled as the covariates age, edema, log (bilirubin), log (protime), log (albumin). While we did not observe any clear functional forms of any covariates aside from bilirubin in Figure 5.4, and the Anderson-darling test only showed significant trend in bilirubin, we can check if models including log (protime) and log (albumin) gives better fits by the same approach we just used for bilirubin. Fit two models, one where the covariates are age, edema, bilirubin, log (protime), and albumin. The other where the covariates are age, edema, bilirubin, protime, and log (albumin). The hazards for the two models are again estimated using the covariate order method with reflection to handle kernel boundaries, and smoothing along the event-axis with $h = 0.05$. $h = 0.05$ was chosen since its the same value used for the protime and albumin in the original AFT model with covariates age, edema, bilirubin, protime, and albumin. Figure 5.7 shows the estimated hazards for protime and albumin in the original model, and the hazards for log (protime) and log (albumin)

in the two new models.

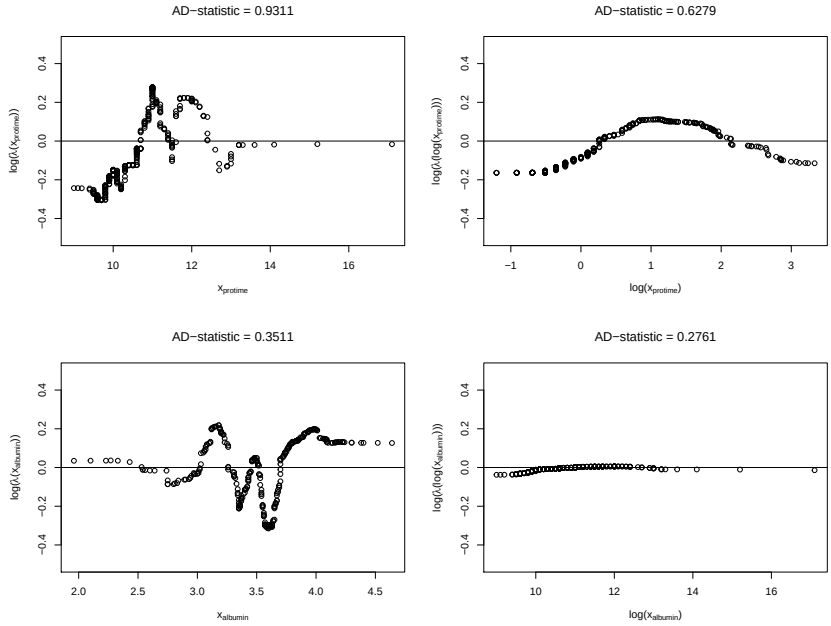


Figure 5.7: Plot of the log of the estimated hazard rates of the Cox-Snell residuals against the first model using protime and albumin on their regular scales (left), and two new models with log of protime (upper right) and log of albumin (lower right).

From Figure 5.7 we observe that the hazards for log (protime) and log (albumin) are less significant than the ones for protime and albumin. The AD-statistics also shows less significant trend for log (protime) and log (albumin) than protime and albumin, less so for albumin. Thus, we can conclude that modelling protime and albumin as log (protime) and log (albumin) is an improvement. The model with the covariates age, edema, log (bilirubin), log (protime), log (albumin) is a better model than the one where the variables are on its original scale, or where only one of bilirubin, protime, and albumin is applied the log transform.

Chapter 6

Conclusion

In this project we have shown two ways of estimating the functional form for a potentially misspecified accelerated failure time model. One where you can use a smoothing of the Cox-Snell residuals for the model, along with the model's estimated regression parameters, to estimate the functional form as shown in section 3.2. The other method uses the covariate order method to estimate the hazard for the Cox-Snell residuals.

For the first method of estimating the functional form of the simulated data in section 3.2 we saw that for uncensored and low censoring data the functional form is recovered. If the data contains a large degree of censoring then the estimate of the functional form becomes gradually worse, and the displacement of the curve increases. From section 3.3 we saw that the choice of smoothing parameter largely influences the estimated hazard produced by the covariate order method. While the cross-validation criterion gave an idea of what smoothing parameter to use, it would be interesting to look at more ways of finding a good choice for h. From the functional form estimate produced through the covariate order method, we saw that while it captured the functional form for low values of the covariate, it did not for higher values. This is probably due to most of the uncensored Cox-Snell residuals corresponding to lower values of the covariate which lead to few events for higher covariate values in the covariate order method. If there were more observations in the data, then we would expect the covariate order method to give a better estimate even if the degree of censoring remained the same.

For the Popes data we did not know if there was any underlying functional form unlike the simulated data. While, we did not find any glaring functional here, we saw how the functional form estimate can still be used to make improvements upon an already adequate model. We also saw the usefulness of comparing hazards of covariates to see if changing the functional form is an improvement. Our final model suggests to model the post election lifetimes with the covariates Age.Elected and f(Year.Elected), where $f(.)$ is given by equation (4.1).

The analysis of the PBC data showed a clear logarithmic functional form for bilirubin. Furthermore, we again saw the value of estimating and analyzing hazards to see if an alternate functional form is an improvement for a covariate.

log(protime) and log(albumin) were shown to be improvements over a model with protime and albumin.

Bibliography

[1] Poisson processes. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-262-discrete-stochastic-processes-spring-2011/course-notes/MIT6_262S11_chap02.pdf. Accessed: 22-05-2019.

[2] Pope gregory xiv. http://www.papalartifacts.com/portfolio-item/pope-gregory-xiv/. Accessed: 20-05-2019.

[3] Pope john paul i. http://http://www.papalartifacts.com/portfolio-item/pope-john-paul-i/. Accessed: 20-05-2019.

[4] Primary biliary cholangitis(primary biliary cirrhosis). https://www.nhs.uk/conditions/primary-biliary-cirrhosis-pbc/. Accessed: 06-06-2019.

[5] Mayo clinic primary biliary cirrhosis data. https://stat.ethz.ch/R-manual/R-devel/RHOME/library/survival/html/pbc.html, 2000.

[6] Tma4275 lifetime analysis slides 4: Gumbel distribution. log-location-scale families. https://www.math.ntnu.no/emner/TMA4275/2015v/TMA4275-Slides4-2015.pdf, 2015.

[7] S. Aaserud. Residuals and functional form in accelerated life regression models. 2011.

[8] T. W. Anderson, D. A. Darling, et al. Asymptotic theory of certain" goodness of fit" criteria based on stochastic processes. *The annals of mathematical statistics*, 23(2):193–212, 1952.

[9] D. Collet. *Modelling Survival Data in Medical Research*. Chapman and Hall, 2003.

[10] T. Fleming and D. Harrington. Counting processes and survival analysis john wiley & sons. *Inc. New York*, 1991.

[11] R. J. Karunamuni and T. Alberts. On boundary correction in kernel density estimation. *Statistical Methodology*, 2(3):191–212, 2005.

[12] J. T. Kvaløy. Covariate order tests for covariate effect. *Lifetime data analysis*, 8(1):35–51, 2002.

[13] J. T. Kvaløy and B. H. Lindqvist. The covariate order method for nonparametric exponential regression and some applications in other lifetime models. In *Parametric and Semiparametric Models with Applications to Reliability, Survival Analysis, and Quality of Life*, pages 221–237. Springer, 2004.

[14] B. H. Lindqvist, J. T. Kvaløy, and S. Aaserud. Residual plots to reveal the functional form for covariates in parametric accelerated failure time models. *Lifetime data analysis*, 21(3):353–378, 2015.

[15] K. Sigman. Inverse transform method. `http://www.columbia.edu/~ks20/4404-Sigman/4404-Notes-ITM.pdf`, 2010. Accessed: 23-03-2019.

[16] J. Stander, L. D. Valle, and M. Cortina-Borja. A bayesian survival analysis of a historical dataset: How long do popes live? *The American Statistician*, 72(4):368–375, 2018.

[17] J. Stander, L. D. Valle, and M. Cortina-Borja. A bayesian survival analysis of a historical dataset: How long do popes live? *The American Statistician*, pages 1–8, 2018.

[18] X. Wu. Robust likelihood cross-validation for kernel density estimation. *Journal of Business Economic Statistics*, pages 1–10, 2018.

[19] S. Zhang and R. J. Karunamuni. On kernel density estimation near endpoints. *Journal of statistical Planning and inference*, 70(2):301–316, 1998.

Appendices

Appendix A

Datasets

A.1 Simulated datasets with Weibull lifetimes, with various degrees of censoring

	t	y20	y60	y80	delta20	delta60	delta80	x1	x2	W
1	1.37	1.37	1.37	0.22	1	1	0	0.40	0.69	0.28
2	2.78	2.78	0.29	0.00	1	0	0	-0.61	5.19	-0.01
3	1.49	1.49	0.26	0.10	1	0	0	0.34	1.91	-0.58
4	0.11	0.11	0.11	0.11	1	1	1	-1.13	3.59	-2.34
5	8.41	3.67	1.54	0.07	0	0	0	1.43	1.25	0.47
6	3.38	0.04	0.45	0.22	0	0	0	1.98	4.35	-2.23
7	0.10	0.10	0.10	0.00	1	1	0	-0.37	2.59	-2.87
8	0.44	0.44	0.29	0.22	1	0	0	-1.04	3.03	-0.88
9	3.17	3.17	0.36	0.25	1	0	0	0.57	3.87	-0.77
10	3.32	1.03	0.88	0.22	0	0	0	-0.14	1.37	1.02
11	15.35	11.61	0.97	0.05	0	0	0	2.40	0.88	0.46
12	2.54	2.54	0.10	0.03	1	0	0	-0.04	1.52	0.55
13	1.12	1.12	1.12	0.09	1	1	0	0.69	1.50	-0.98
14	0.49	0.49	0.49	0.22	1	1	0	0.03	0.50	-0.04
15	0.08	0.08	0.08	0.02	1	1	0	-0.74	0.66	-1.34
16	1.59	1.59	0.39	0.10	1	0	0	0.19	1.89	-0.36
17	0.14	0.14	0.14	0.13	1	1	0	-1.80	2.63	-1.10
18	0.04	0.04	0.04	0.00	1	1	0	1.47	1.08	-4.81
19	1.72	0.05	0.14	0.23	0	0	0	0.15	1.53	-0.03
20	2.32	2.32	0.11	0.18	1	0	0	2.17	1.05	-1.38
21	0.04	0.04	0.04	0.04	1	1	1	0.48	0.41	-2.70
22	3.53	3.53	0.41	0.08	1	0	0	-0.71	4.63	0.44
23	1.81	1.81	0.03	0.11	1	0	0	0.61	2.30	-0.85
24	1.31	0.76	0.51	0.26	0	0	0	-0.93	1.60	0.73
25	0.09	0.09	0.04	0.03	1	0	0	-1.25	0.24	0.28

26	3.13	2.48	0.00	0.46	0	0	0	0.29	2.46	-0.05
27	1.95	1.95	1.06	0.07	1	0	0	-0.44	0.70	1.46
28	1.48	1.48	0.88	0.02	1	0	0	0.00	1.54	-0.04
29	0.24	0.24	0.24	0.14	1	1	0	0.07	1.62	-1.97
30	0.54	0.54	0.43	0.06	1	0	0	-0.59	0.90	0.07
31	0.61	0.61	0.61	0.09	1	1	0	-0.57	1.47	-0.32
32	0.05	0.05	0.02	0.05	1	0	1	-0.14	0.11	-0.67
33	0.35	0.35	0.15	0.01	1	0	0	1.18	0.15	-0.35
34	0.75	0.75	0.01	0.01	1	0	0	-1.52	2.05	0.52
35	0.46	0.46	0.46	0.09	1	1	0	0.59	1.35	-1.66
36	0.90	0.90	0.54	0.28	1	0	0	0.33	1.60	-0.91
37	0.23	0.23	0.23	0.08	1	1	0	1.06	0.35	-1.47
38	2.21	2.19	0.01	0.58	0	0	0	-0.30	1.35	0.80
39	2.08	2.08	0.49	0.42	1	0	0	0.37	4.44	-1.13
40	0.94	0.94	0.15	0.26	1	0	0	0.27	0.81	-0.12
41	0.09	0.09	0.09	0.09	1	1	1	-0.54	0.80	-1.62
42	2.90	2.90	1.33	0.28	1	0	0	1.21	1.99	-0.83
43	0.24	0.24	0.24	0.03	1	1	0	1.16	0.31	-1.41
44	3.28	3.28	0.39	0.92	1	0	0	0.70	2.75	-0.52
45	6.38	0.07	0.27	0.15	0	0	0	1.59	2.05	-0.45
46	0.10	0.10	0.10	0.10	1	1	1	0.56	0.25	-1.44
47	2.27	2.27	1.73	0.20	1	0	0	-1.28	2.17	1.32
48	0.13	0.13	0.13	0.12	1	1	0	-0.57	0.32	-0.30
49	0.12	0.12	0.12	0.12	1	1	1	-1.22	1.33	-1.17
50	1.00	1.00	0.35	0.03	1	0	0	-0.47	4.36	-1.00
51	1.22	1.22	0.60	0.20	1	0	0	-0.62	3.07	-0.30
52	0.03	0.03	0.03	0.03	1	1	1	0.04	0.21	-1.90
53	0.72	0.72	0.72	0.43	1	1	0	-0.91	2.17	-0.19
54	1.05	1.05	1.05	0.49	1	1	0	0.16	0.64	0.34
55	2.59	1.51	0.41	0.97	0	0	0	-0.65	1.88	0.97
56	29.71	2.59	3.91	0.21	0	0	0	1.77	2.20	0.84
57	7.40	7.35	0.16	0.42	0	0	0	0.72	3.14	0.14
58	1.04	0.99	0.14	0.88	0	0	0	0.91	0.64	-0.42
59	2.26	2.26	0.18	0.34	1	0	0	0.38	3.73	-0.89
60	6.02	6.02	0.56	0.13	1	0	0	1.68	1.24	-0.11
61	0.04	0.04	0.04	0.04	1	1	1	-0.64	0.89	-2.40
62	1.81	0.46	0.39	0.16	0	0	0	-0.46	2.35	0.20
63	3.67	3.02	2.62	0.38	0	0	0	1.43	1.12	-0.25
64	5.27	5.27	0.33	0.29	1	0	0	-0.65	9.17	0.10
65	0.17	0.17	0.17	0.17	1	1	1	-0.21	0.50	-0.85
66	0.91	0.91	0.64	0.45	1	0	0	-0.39	0.99	0.30
67	0.63	0.63	0.51	0.63	1	0	1	-0.32	1.17	-0.31
68	0.17	0.17	0.17	0.01	1	1	0	-0.28	0.82	-1.32
69	7.35	1.74	2.42	0.12	0	0	0	0.49	1.81	0.91
70	0.09	0.09	0.09	0.08	1	1	0	-0.18	0.78	-2.02

71	0.84	0.84	0.07	0.43	1	0	0	-0.51	1.29	0.08
72	1.41	0.50	0.17	0.72	0	0	0	1.34	2.10	-1.74
73	2.62	2.62	0.60	0.16	1	0	0	-0.21	3.06	0.06
74	1.96	1.96	1.56	0.08	1	0	0	-0.18	2.14	0.09
75	1.77	1.77	0.94	0.05	1	0	0	-0.10	2.63	-0.30
76	1.14	1.14	0.60	0.01	1	0	0	0.71	4.89	-2.17
77	0.16	0.16	0.16	0.09	1	1	0	-0.07	1.16	-1.93
78	0.09	0.09	0.09	0.09	1	1	1	-0.04	0.10	-0.06
79	0.02	0.02	0.02	0.00	1	1	0	-0.68	0.15	-1.38
80	0.16	0.16	0.16	0.16	1	1	1	-0.32	5.52	-3.21
81	1.07	0.67	0.43	0.31	0	0	0	0.06	1.21	-0.18
82	0.18	0.18	0.18	0.10	1	1	0	-0.59	0.96	-1.08
83	10.73	0.60	0.09	0.06	0	0	0	0.53	6.88	-0.09
84	0.36	0.36	0.36	0.19	1	1	0	-1.52	1.44	0.12
85	0.66	0.66	0.10	0.41	1	0	0	0.31	1.74	-1.28
86	0.00	0.00	0.00	0.00	1	1	1	-1.54	0.00	0.47
87	0.11	0.11	0.11	0.11	1	1	1	-0.30	0.44	-1.08
88	1.86	1.86	0.49	0.10	1	0	0	-0.53	1.50	0.74
89	2.03	2.03	2.03	0.01	1	1	0	-0.65	2.77	0.34
90	4.26	4.26	0.02	0.36	1	0	0	-0.06	2.32	0.66
91	0.02	0.02	0.02	0.02	1	1	1	-1.91	0.10	0.36
92	4.71	4.71	0.25	0.04	1	0	0	1.18	0.51	1.04
93	0.29	0.29	0.29	0.03	1	1	0	-1.66	3.50	-0.81
94	0.53	0.53	0.19	0.53	1	0	1	-0.46	6.35	-2.02
95	0.01	0.01	0.01	0.01	1	1	1	-1.12	0.09	-1.39
96	0.15	0.15	0.15	0.15	1	1	1	-0.75	1.39	-1.48
97	9.35	8.55	2.32	0.18	0	0	0	2.09	1.47	-0.24
98	0.21	0.21	0.03	0.21	1	0	1	0.02	0.23	-0.11
99	0.12	0.12	0.12	0.07	1	1	0	-1.29	2.03	-1.56
100	0.05	0.05	0.05	0.05	1	1	1	-1.64	0.55	-0.69

Table A.1: t are the uncensored lifetimes, y_i are lifetimes with i% censoring. delta$_i$ is the censoring indicator for the lifetimes with i% censoring. x_1 and x_2 are simulated covariates, while W is simulated from the standard Gumbel distribution of the smallest extreme. How these data are simulated is described in section 3.1.

A.2 Post-election survival times of popes

	Common.name	Year.Elected	Age.Election	Years.as.Pope	Survival	Censored
1	Benedict XVI	2005.00	78.00	7.87	11.70	1.00
2	John Paul II	1978.00	58.00	26	26.00	0.00
3	John Paul I	1978.00	65.00	<1	0.09	0.00
4	Paul VI	1963.00	65.00	15	15.00	0.00
5	John XXIII	1958.00	76.00	4	5.00	0.00

6	Pius XII	1939.00	63.00	19	19.00	0.00
7	Pius XI	1922.00	64.00	17	17.00	0.00
8	Benedict XV	1914.00	59.00	7	8.00	0.00
9	Pius X	1903.00	68.00	11	11.00	0.00
10	Leo XIII	1878.00	67.00	25	26.00	0.00
11	Pius IX	1846.00	54.00	31	31.00	0.00
12	Gregory XVI	1831.00	65.00	15	15.00	0.00
13	Pius VIII	1829.00	67.00	1	2.00	0.00
14	Leo XII	1823.00	63.00	5	5.00	0.00
15	Pius VII	1800.00	59.00	23	24.00	0.00
16	Pius VI	1775.00	57.00	24	24.00	0.00
17	Clement XIV	1769.00	63.00	5	5.00	0.00
18	Clement XIII	1758.00	65.00	10	10.00	0.00
19	Benedict XIV	1740.00	65.00	17	18.00	0.00
20	Clement XII	1730.00	78.00	9	9.00	0.00
21	Benedict XIII	1724.00	75.00	5	6.00	0.00
22	Innocent XIII	1721.00	66.00	3	2.00	0.00
23	Clement XI	1700.00	51.00	20	20.00	0.00
24	Innocent XII	1691.00	76.00	9	9.00	0.00
25	Alexander VIII	1689.00	79.00	1	1.00	0.00
26	Innocent XI	1676.00	65.00	12	13.00	0.00
27	Clement X	1670.00	79.00	6	7.00	0.00
28	Clement IX	1667.00	67.00	2	2.00	0.00
29	Alexander VII	1655.00	56.00	12	12.00	0.00
30	Innocent X	1644.00	70.00	10	10.00	0.00
31	Urban VIII	1623.00	55.00	20	21.00	0.00
32	Gregory XV	1621.00	67.00	2	2.00	0.00
33	Paul V	1605.00	54.00	15	16.00	0.00
34	Leo XI	1605.00	69.00	<1	0.07	0.00
35	Clement VIII	1592.00	55.00	13	14.00	0.00
36	Innocent IX	1591.00	72.00	<1	0.17	0.00
37	Gregory XIV	1590.00	55.00	<1	0.86	0.00
38	Urban VII	1590.00	69.00	<1	0.03	0.00
39	Sixtus V	1585.00	63.00	5	5.00	0.00
40	Gregory XIII	1572.00	70.00	12	13.00	0.00
41	Pius V	1566.00	55.00	12	13.00	0.00
42	Pius IV	1559.00	59.00	6	7.00	0.00
43	Paul IV	1555.00	78.00	4	5.00	0.00
44	Marcellus II	1555.00	53.00	<1	0.06	0.00
45	Julius III	1550.00	62.00	5	5.00	0.00
46	Paul III	1534.00	66.00	15	15.00	0.00
47	Clement VII	1523.00	45.00	11	11.00	0.00
48	Adrian VI	1522.00	62.00	1	2.00	0.00
49	Leo X	1513.00	37.00	8	8.00	0.00
50	Julius II	1503.00	59.00	9	10.00	0.00

51	Pius III	1503.00	64.00	<1	0.07	0.00
52	Alexander VI	1492.00	61.00	11	11.00	0.00
53	Innocent VIII	1484.00	51.00	7	8.00	0.00
54	Sixtus IV	1471.00	57.00	13	13.00	0.00
55	Paul II	1464.00	47.00	6	7.00	0.00
56	Pius II	1458.00	52.00	5	6.00	0.00
57	Calixtus III	1455.00	76.00	3	3.00	0.00
58	Nicholas V	1447.00	49.00	8	8.00	0.00
59	Eugene IV	1431.00	47.00	15	16.00	0.00
60	Martin V	1417.00	48.00	13	14.00	0.00
61	Gregory XII	1406.00	81.00	8	10.90	0.00
62	Innocent VII	1404.00	67.00	2	2.00	0.00

Table A.2: Dataset of post-election survival times of popes. Some unused columns such as date pontificate start, end, age death, were deleted. For a version containing these columns see [16].

Appendix B

R-code

B.1 Simulating data sets

```
#gumbel_sim simulates n elements from the standard gumbel distribtuion
gumbel_sim <- function(n){
  u <- runif(n)
  y <- log(-log(u))
  return(y)
}

set.seed(1)
n <- 100
beta1 <- 1
beta2 <- 1
sigma <- 1
gumb <- gumbel_sim(n)
#lambdas <- c(0.57, 0.22, 0.05)
lambdas <- c(9, 0.77, 0.2)

x1 <- rnorm(n)
x2 <- rexp(n , rate = 1/2)

t <- exp(beta1*x1 + beta2*log(x2) + sigma*gumb)
censorings <- rep(0,3)

#20% censoring
c20 <- rexp(n,rate = 1/lambdas[1])
delta20 <- as.numeric(t < c20)
#censor indicator, 1=true lifetime, 0 = censored
y20 <- pmin(t, c20)
censorings[1] <- sum(delta20==0)/n

#60% censoring
c60 <- rexp(n,rate = 1/lambdas[2])
delta60 <- as.numeric(t < c60)
#censor indicator, 1=true lifetime, 0 = censored
y60 <- pmin(t, c60)
censorings[2] <- sum(delta60==0)/n

```

```
#80% censoring
c80 <- rexp(n,rate = 1/lambdas[3])
delta80 <- as.numeric(t < c80)
#censor indicator, 1=true lifetime, 0 = censored
y80 <- pmin(t, c80)
censorings[3] <- sum(delta80==0)/n

W <- gumb

dataset <- data.frame(t, y20, y60, y80, delta20, delta60, delta80, x1,
     x2, W)

print(censorings)


#Write CSV in R
#write.csv(dataset, file = "simData_n100_v2.csv")

#dat <- read.table('simData_n100.csv', header = TRUE, sep=',', row.
    names = 1)
```

B.2 Covariate order method

Covariate Order method code based on code by Jan Terje Kvaløy. [7]

```
library(survival)
library(MASS)

#sx is a function that calculates corrspondance function for a
    specified grid, xgrid.
sx <- function(xgrid,Xdata,dat){
  n <- length(xgrid) #length of covariate vector
  sx <- vector(length=n)
  for (i in 1:n) {
    sx[i] <- tail(c(0,dat[Xdata<xgrid[i]]),1)

  }
  return(sx)
}

#Epanechnikov kernel
epK <- function(u){
  ifelse(abs(u)<1, 3/4 * (1-u^2), 0)
}

#boundary kernel from Zhang and Karunamuni ( JSPI , 1998)
epbK <-function(t,c){
  ifelse( abs(t)<1 & t<c,  12/(1+c)^4  * (1+t) * ( t*(1-2*c) +(3*c^2-2
    *c+1)/2 ), 0)
}

#Covariate order function
CovOrder<- function(survtimes, x, delta, h, xgrid = 1, splot = F, edge
     = "BK", smoothing = "s"){
  #input arguments
  #x is the covariate vector
  #survtimes is a vector of survival times
```

```
#delta is the censoring indicator; 0 = censored, 1 = not censored
#edge handles boundaries in the density estimation. BK = boundary kernel is default, R = reflection.
#smooth specifies wheter to smooth over the s-axis ("s"), or the covariate axis ("x")

n <- length(x) #number of observations

#sorting data
ordered <- order(x) #indices corresponding to ordered x
x <- x[ordered]
delta <- delta[ordered]
survtimes <- survtimes[ordered]
#calculate basic quantities
V <- survtimes/n #S is realizations of the poisson point process
V <- cumsum(V)
Vend <- V[n] #endpoint
S <- V[delta==1] #only including points for non censored incidents, these are the observations in the poisson process.
xs <- x[delta==1] #x-vals corresponding to the observations in the poisson process.
K <- length(S) #number of realizations in the point process

tildesx <- V #correspondance function as a step-function
#if(splot == TRUE){
 # plot(x,tildesx)
#}

#calculating correspondance function using sx if xgrid is specified, if not using the default step-function
lambdaest <- vector(length=n) #initialize vector to hold lambda estimates
if(length(xgrid) > 1){
  xordered <- order(xgrid)
  unordered <- order(xordered)
  xgrid <- xgrid[xordered]
  tildesx <- sx(xgrid, x, V)
}

if(length(xgrid) == 1){
  xgrid <- x
  unordered <- order(ordered)
  tildesx <- V
}

########
len_xgrid <- length(xgrid)
if(edge != "R"){
  if(smoothing == "s"){
    #smoothing along s-axis
    for (i in 1:len_xgrid) {
      if(h > Vend/2){
      h <- Vend/2
      }
      #look at kernel cases depending on where on the grid we are
      if(tildesx[i]<h){
```

```
80            lambdaest[i] <- sum( epbK( (tildesx[i]-S)/h, tildesx[i]/h )
    )/(n*h)
81          }
82            if(tildesx[i]>=h & tildesx[i]<=Vend-h){
83              lambdaest[i] <- sum( epK( (tildesx[i]-S)/h) )/(n*h)
84          }
85            if(tildesx[i]>Vend-h){
86              lambdaest[i] <- sum( epbK( -(tildesx[i]-S)/h, (Vend-tildesx[
    i])/h ) )/(n*h)
87          }
88          }
89        }
90
91        else{
92          #smoothing along x-axis
93          for (i in 1:len_xgrid) {
94            h_sx <- sx(xgrid[i] + h/2, x, V) - sx(xgrid[i] - h/2, x, V)
95            if(h_sx > Vend/2){
96              h_sx <- Vend/2
97            }
98            #look at kernel cases depending on where on the grid we are
99            if(tildesx[i]<h_sx){
100             lambdaest[i] <- sum( epbK( (tildesx[i]-S)/h_sx, tildesx[i]/h
    _sx ) )/(n*h_sx)
101           }
102           if(tildesx[i]>=h_sx & tildesx[i]<=Vend-h_sx){
103             lambdaest[i] <- sum( epK( (tildesx[i]-S)/h_sx) )/(n*h_sx)
104           }
105           if(tildesx[i]>Vend-h_sx){
106             lambdaest[i] <- sum( epbK( -(tildesx[i]-S)/h_sx, (Vend-
    tildesx[i])/h_sx ) )/(n*h_sx)
107           }
108         }
109       }
110     }
111     ####
112     #using reflection method to handle boundaries.
113     if(edge == "R"){
114       if(smoothing == "s"){
115         for (i in 1:len_xgrid) {
116           lambdaest[i] <- (1/(n*h))*( sum( epK((tildesx[i]-S)/h) ) +
      sum( epK((tildesx[i]+S)/h) ) + sum( epK((tildesx[i]+S-2*Vend)/h) )
       )
117         }
118       }
119
120       else{
121         for (i in 1:len_xgrid) {
122           h_sx <- sx(xgrid[i] + h/2, x, V) - sx(xgrid[i] - h/2, x, V)
123           lambdaest[i] <- (1/(n*h_sx))*( sum( epK((tildesx[i]-S)/h_sx) )
       +  sum( epK((tildesx[i]+S)/h_sx) ) + sum( epK((tildesx[i]+S-2*
    Vend)/h_sx) ) )
124         }
125       }
126     }
127     #####
128
```

```
129
130   if(splot == TRUE){
131     plot(xs, S, type="s")
132     points(xs, S)
133   }
134
135   out <- list(x=xgrid, x_unsorted = xgrid[unordered], lambdaest=
        lambdaest, lambdaest_unsorted = lambdaest[unordered], s=S)
136   return(out)
137 }
```

B.3 Leave-one-out likelihood cross-validation

```
1  loocv <- function(h, data, CS, x, delta = 'uncensored'){
2    source('CovOrderAlt.R')
3    library(splines)
4    # Inputs:
5    #h is the value for smoothing paramater we wish to
6    #data is the dataset
7    #CS is the cox-snell for which we use loocv on through the CovOrder
       function
8    #x is the covarite vector for the CovOrder function
9    #delta is the name of censoring indicator in data as a string,
       uncensored is default
10
11
12   #output:
13   #lcv = this is the value of the likelihood cross-validation
       criterion.
14
15   n <- length(CS)
16   if(delta != 'uncensored'){
17     delta <- data[[delta]]
18   }
19   else{
20     delta = rep(1,n)
21     }
22
23   lcv <- 0
24   for (i in 1:n) {
25     #esitimate lambda without observation i and add to lcv criterion
26     out <- CovOrder(CS[-i], x[-i], delta[-i], h, xgrid = 1, splot = F,
        edge = "R", smoothing = "s")
27     #interpSpline gives piecewise interpolation representation.
28     #polySpline gives polynomial representation of the spline
29     spline <- polySpline(interpSpline(out$x, out$lambdaest))
30     predSpline <- predict(spline,x[i])
31     lambdaPredict <- predSpline$y
32     lcv <- lcv + delta[i]*log(lambdaPredict) - lambdaPredict*CS[i]
33   }
34   return(lcv)
35   }
```

B.4 Cross-validation criterion equation (3.35)

```
#sx is a function that calculates corrspondance function for a
    specified grid, xgrid.
sx <- function(xgrid,Xdata,dat){
  n <- length(xgrid) #length of covariate vector
  sx <- vector(length=n)
  for (i in 1:n) {
    sx[i] <- tail(c(0,dat[Xdata<xgrid[i]]),1)

  }
  return(sx)
}

#Epanechnikov kernel
epK <- function(u){
  ifelse(abs(u)<1, 3/4 * (1-u^2), 0)
}

#boundary kernel from Zhang and Karunamuni ( JSPI , 1998)
epbK <-function(t,c){
  ifelse( abs(t)<1 & t<c,  12/(1+c)^4  * (1+t) * ( t*(1-2*c) +(3*c^2-2
    *c+1)/2 ), 0)
}

# loocv, only handling the reflection method for boundary problems

loocv_CovOrder <- function(h, survtimes, x, delta, xgrid = 1,
    smoothing = "s"){

  lcv <- 0
  n <- length(x)
  #sorting data
  ordered <- order(x) #indices corresponding to ordered x
  x <- x[ordered]
  delta <- delta[ordered]
  survtimes <- survtimes[ordered]
  #calculate basic quantities
  V <- survtimes/n #S is realizations of the poisson point process
  V <- cumsum(V)
  Vend <- V[n] #endpoint
  S <- V[delta==1] #only including points for non censored incidents,
    these are the observations in the poisson process.
  xs <- x[delta==1] #x-vals corresponding to the observations in the
    poisson process.

  ########
  len_xgrid <- length(xgrid)
  lcv <- 0 #loocv-criterion that will be the return parameter
  k <- sum(delta)

  counter <- 0 #number of g elements
  g <- rep(0,k)
  for (j in 1:k) {
    counter <- counter + 1
    #remove elements j corresponding to the S_j
    Si <- S[-j]
```

```
51
52   if(smoothing == "s"){
53       g[counter] <- (1/((n-1)*h))*( sum( epK((S[j]-Si)/h) ) +  sum(
       epK((S[j]+Si)/h) ) + sum( epK((S[j]+Si-2*Vend)/h) ) )
54   }
55
56   else{
57       h_sx <- sx(xgrid[i] + h/2, x, V) - sx(xgrid[i] - h/2, x, V)
58       g[counter] <- (1/((n-1)*h_sx))*( sum( epK((S[j]-Si)/h_sx) ) +
       sum( epK((S[j]+Si)/h_sx) ) + sum( epK((S[j]+Si-2*Vend)/h_sx) ) )
59   }
60
61   }
62   lcv <- sum(log(g))/k
63   return(lcv)
64 }
```

B.5 AD test

```
1  ADtest <- function(x, T, delta) {
2    #x is the covariate, T is the observation times, delta is the
       censoring indicator
3    ordered <- order(x)
4    x <- x[ordered]
5    T <- T[ordered]
6    delta <- delta[ordered]
7
8    n <- length(x)
9    S <- cumsum(T)/n #S is realizations of the poisson point process
10   Smax <- S[n]
11   S <- S[delta==1]
12
13   if(delta[n] == 1){
14     rhat <- sum(delta) - 1
15   }
16   else{
17     rhat <- sum(delta)
18   }
19
20   AD <- 0
21
22   for (i in 1:rhat) {
23     AD <-  AD + (2*i-1)*( log(S[i]/Smax) + log(1 - S[rhat+1-i]/Smax) )
24   }
25   AD <- -1/rhat*AD - rhat
26   return(AD)
27 }
```

B.6 Simulated data analysis

```
1  library(survival)
2  library(MASS)
3  source('loocv_kernel.R', echo=FALSE)
4  source('CovOrderAlt.R', echo=FALSE)
5  source('ADtest.R', echo=FALSE)
```

```
6
7
8  data <- read.table('simData_n100_v2.csv', header = TRUE, sep=',', row.
       names = 1)
9
10 #fit model for uncensored data
11 n <- 100
12 surv <-Surv(data$t)
13 x1 <- data$x1
14 x2 <- data$x2
15 delta <- rep(1,n)
16 t <- data$t
17 fit <- survreg(surv~x1 + x2, data, dist = "weibull")
18
19 #fitting models to censored data
20 #20% censoring
21 y20 <- data$y20
22 delta20 <- data$delta20
23 surv20 <- Surv(y20, delta20, type = "right")
24 fit20 <- survreg(surv20~x1 + x2, data, dist = "weibull")
25
26 #60% censoring
27 y60 <- data$y60
28 delta60 <- data$delta60
29 surv60 <- Surv(y60, delta60, type = "right")
30 fit60 <- survreg(surv60~x1 + x2, data, dist = "weibull")
31
32 #80% censoring
33 y80 <- data$y80
34 delta80 <- data$delta80
35 surv80 <- Surv(y80, delta80, type = "right")
36 fit80 <- survreg(surv80~x1 + x2, data, dist = "weibull")
37
38 #calculating CS residuals
39 CS <- exp((log(t) - fit$coefficients[[1]] - fit$coefficients[[2]]*x1 -
       fit$coefficients[[3]]*x2)/fit$scale)
40 CS20 <- exp((log(y20) - fit20$coefficients[[1]] - fit20$coefficients
       [[2]]*x1 - fit20$coefficients[[3]]*x2)/fit20$scale)
41 CS60 <- exp((log(y60) - fit60$coefficients[[1]] - fit60$coefficients
       [[2]]*x1 - fit60$coefficients[[3]]*x2)/fit60$scale)
42 CS80 <- exp((log(y80) - fit80$coefficients[[1]] - fit80$coefficients
       [[2]]*x1 - fit80$coefficients[[3]]*x2)/fit80$scale)
43
44 #adjusting the censored CS residuals
45 #CS20[delta20==0] <- CS20[delta20==0] + 1
46 #CS60[delta60==0] <- CS20[delta60==0] + 1
47 #CS80[delta80==0] <- CS20[delta80==0] + 1
48
49 h_vals <- seq(0.04,1,0.01)
50 counter <- 1
51 lcv <- length(h_vals)
52 lcv20 <- lcv
53 lcv60 <- lcv
54 lcv80 <- lcv
55
56 for (h in h_vals) {
57   lcv[counter] <- loocv_CovOrder(h, CS, x2, delta = rep(1,n))
```

```
  lcv20[counter] <- loocv_CovOrder(h, CS20, x2, delta20)
  lcv60[counter] <- loocv_CovOrder(h, CS60, x2, delta60)
  lcv80[counter] <- loocv_CovOrder(h, CS80, x2, delta80)
  counter <- counter + 1
}


lcv_max <- c(max(lcv), max(lcv20), max(lcv60), max(lcv80)) #holds the best h value for the 4 data sets
h_max <- c(h_vals[lcv==lcv_max[1]], h_vals[lcv20==lcv_max[2]], h_vals[lcv60==lcv_max[3]], h_vals[lcv80==lcv_max[4]])
par(mfrow=c(2,2))

plot(h_vals, lcv, xlab = "h", ylab = "lcv(h)", main = "uncensored")
points(h_max[1], lcv_max[1], col = "red")

plot(h_vals, lcv20, xlab = "h", ylab = "lcv(h)", main = "20% censoring")
points(h_max[2], lcv_max[2], col = "red")

plot(h_vals, lcv60, xlab = "h", ylab = "lcv(h)", main = "60% censoring")
points(h_max[3], lcv_max[3], col = "red")

plot(h_vals, lcv80, xlab = "h", ylab = "lcv(h)", main = "80% censoring")
points(h_max[4], lcv_max[4], col = "red")

cov_obj <-  CovOrder(CS, x2, delta, h_max[1], xgrid = 1, splot = F, edge = "R", smoothing = "s")
cov_obj20 <-  CovOrder(CS20, x2, delta20, h_max[2], xgrid = 1, splot = F, edge = "R", smoothing = "s")
cov_obj60 <-  CovOrder(CS60, x2, delta60, h_max[3], xgrid = 1, splot = F, edge = "R", smoothing = "s")
cov_obj80 <-  CovOrder(CS80, x2, delta80, h_max[4], xgrid = 1, splot = F, edge = "R", smoothing = "s")

fx2 <- fit$coefficients[[3]]*cov_obj$x - fit$scale*log(cov_obj$lambdaest)
fx2_20 <- fit20$coefficients[[3]]*cov_obj20$x - fit20$scale*log(cov_obj20$lambdaest)
fx2_60 <- fit60$coefficients[[3]]*cov_obj60$x - fit60$scale*log(cov_obj60$lambdaest)
fx2_80 <- fit80$coefficients[[3]]*cov_obj80$x - fit80$scale*log(cov_obj80$lambdaest)

par(mfrow=c(2,2))
plot(cov_obj$x, fx2, xlab = "x", ylab = "f(x)", main = "uncensored", title())
lines(sort(x2), log(sort(x2)))

plot(cov_obj20$x, fx2_20, xlab = "x", ylab = "f(x)", main = "20% censoring")
lines(sort(x2), log(sort(x2)))

plot(cov_obj60$x, fx2_60, xlab = "x", ylab = "f(x)", main = "60% censoring")
```

```
lines(sort(x2), log(sort(x2)))

plot(cov_obj80$x, fx2_80, xlab = "x", ylab = "f(x)", main = "80%
    censoring")
lines(sort(x2), log(sort(x2)))

par(mfrow=c(2,2))
plot(cov_obj$x,log(cov_obj$lambdaest), xlab = expression(x[2]), ylab =
     expression(paste("log(",lambda,"(",x[2],"))")), main = "
    Uncensored" )
abline(0,0)
plot(cov_obj20$x,log(cov_obj20$lambdaest), xlab = expression(x[2]),
    ylab = expression(paste("log(",lambda,"(",x[2],"))")), main = "20%
     Censoring" )
abline(0,0)

plot(cov_obj60$x,log(cov_obj60$lambdaest), xlab = expression(x[2]),
    ylab = expression(paste("log(",lambda,"(",x[2],"))")), main = "60%
     Censoring" )
abline(0,0)
plot(cov_obj80$x,log(cov_obj80$lambdaest), xlab = expression(x[2]),
    ylab = expression(paste("log(",lambda,"(",x[2],"))")), main = "80%
     Censoring" )
abline(0,0)

AD1 <- ADtest(x2, CS, delta)
AD2 <- ADtest(x2, CS20, delta20)
AD3 <- ADtest(x2, CS60, delta60)
AD4 <- ADtest(x2, CS80, delta80)
print(AD1)
print(AD2)
print(AD3)
print(AD4)
```

B.7 Data analysis popes

```
library(readr)
library(survival)
library(MASS)

source('loocv_kernel.R', echo=FALSE)
source('CovOrderAlt.R', echo=FALSE)
source('ADtest.R', echo=FALSE)
###script to do analysis on the popes data

popes <- read_csv("popes_25_December_2016.csv")
#Removing pope Francis from data
popes <- popes[2:63,]
#popes <- popes[popes$Years.as.Pope!="<1",] #to remove outliers

#died or resigned at age represents the lifetime in this case.
t <- popes$Survival
#delta is censoring indicator; 0=censored, 1=not censored
popes$Censored <- as.integer(popes$Censored == 0)
delta <- popes$Censored

```

```
x1 <- popes$Age.Election
x2 <- popes$Year.Elected

surv_obj <- Surv(t, delta, type = "right")
popes_model <- survreg(surv_obj~ x1 + x2, data=popes, dist = "weibull")
CS <- exp((log(t) - popes_model$coefficients[[1]] - popes_model$coefficients[[2]]*x1 - popes_model$coefficients[[3]]*x2)/popes_model$scale)
CS[delta==0] <- CS[delta==0] + 1 #Adjusting the censored residuals. This line is "removed" when computing anything based on the covariate order method

par(mfrow=c(2,1))
plot(x1, log(CS))
points(x1[delta==0],log(CS[delta==0]), col = "red")
points(x1[popes$Years.as.Pope=="<1"],log(CS[popes$Years.as.Pope=="<1"]), pch = 24)
abline(0,0)

plot(x2, log(CS))
points(x2[delta==0],log(CS[delta==0]), col = "red")
points(x2[popes$Years.as.Pope=="<1"],log(CS[popes$Years.as.Pope=="<1"]), pch = 24)
abline(0,0)

h_vals <- seq(0.04,5,0.01)
counter <- 1
lcv1 <- rep(0,length(h_vals))
lcv2 <- lcv1
for (h in h_vals) {
  lcv1[counter] <- loocv_CovOrder(h, CS, x1, delta)
  lcv2[counter] <- loocv_CovOrder(h, CS, x2, delta)
  counter <- counter + 1
}

par(mfrow=c(2,1))
plot(h_vals, lcv1, xlab = "h", ylab = "lCV(h)", main = "x1")
plot(h_vals, lcv2,  xlab = "h", ylab = "lCV(h)", main = "x2")

lcv_max <- c(max(lcv1), max(lcv2))
h_max <- c(h_vals[lcv1==lcv_max[1]], h_vals[lcv2==lcv_max[2]])

#out <- CovOrder(t,x1,delta, splot = T, lambdaplot = T)
out1 <- CovOrder(CS, x1, delta, 0.3, edge = "R", splot = F)
out11 <- CovOrder(CS, x1, delta, 1.08, edge = "R", splot = F)

#small h values gives result more similar to old method than the h found using cv.
fx1 <- popes_model$coefficients[[2]]*out1$x - popes_model$scale*log(out1$lambdaest)
fx11 <- popes_model$coefficients[[2]]*out11$x - popes_model$scale*log(out11$lambdaest)

out2 <- CovOrder(CS, x2, delta, 0.3, edge = "R", splot = F)
out22 <- CovOrder(CS, x2, delta, 1.09, edge = "R", splot = F)

```

```
fx2 <- popes_model$coefficients[[3]]*out2$x - popes_model$scale*log(
    out2$lambdaest)
fx22 <- popes_model$coefficients[[3]]*out22$x - popes_model$scale*log(
    out22$lambdaest)

par(mfrow=c(2,2))
plot(out1$x,fx1, xlab ="x1", ylab="f(x1)", main = "functional form for
     x1 using h = 0.3")
lines(x1, popes_model$coefficients[[2]]*x1)
plot(out11$x,fx11, xlab ="x1", ylab="f(x1)", main = "functional form
    for x1 using h = 1.08")
lines(x1, popes_model$coefficients[[2]]*x1)

plot(out2$x,fx2, xlab ="x2", ylab="f(x2)", main = "functional form for
     x2 using h = 0.3")
lines(x2, popes_model$coefficients[[3]]*x2)
plot(out22$x,fx22, xlab ="x2", ylab="f(x2)", main = "functional form
    for x2 using h = 1.09")
lines(x2, popes_model$coefficients[[3]]*x2)

i = order(x1)
smt1 <- loess(log(CS) ~ x1, pch=19, cex=0.1,span = 2)
smt1 <- popes_model$coefficients[[2]]*x1[i] + popes_model$scale*smt1$
    fitted[i]

j = order(x2)
smt2 <- loess(log(CS) ~ x2, pch=19, cex=0.1, span = 2)
smt2 <- popes_model$coefficients[[3]]*x2[j] + popes_model$scale*smt2$
    fitted[j]

par(mfrow=c(2,1))
plot(x1[i],smt1,lwd=1, xlab ="x1", ylab="f(x1)", main = "functional
    form for Age.Election")
plot(x2[j],smt2,lwd=1, xlab ="x2", ylab="f(x2)", main = "functional
    form for Year.Elected")

par(mfrow=c(2,1))
plot(out1$x,log(out1$lambdaest), xlab = "x1", ylab = expression(paste(
    "log(",lambda,")")) )
abline(0,0)
plot(out2$x,log(out2$lambdaest), xlab = "x2", ylab = expression(paste(
    "log(",lambda,")")) )
abline(0,0)


#Anderson Darling test for covariate effect
AD_x1 <- ADtest(x1,CS,delta)
AD_x2 <- ADtest(x2,CS,delta)
print(AD_x1)
print(AD_x2)

##testing to see if the transformation to year.elected proposed in the
     project will improve the model

popes$Year.Elected[35:62] <- 0
popes$Year.Elected[1:34] <- popes$Year.Elected[1:34] - 1600

```

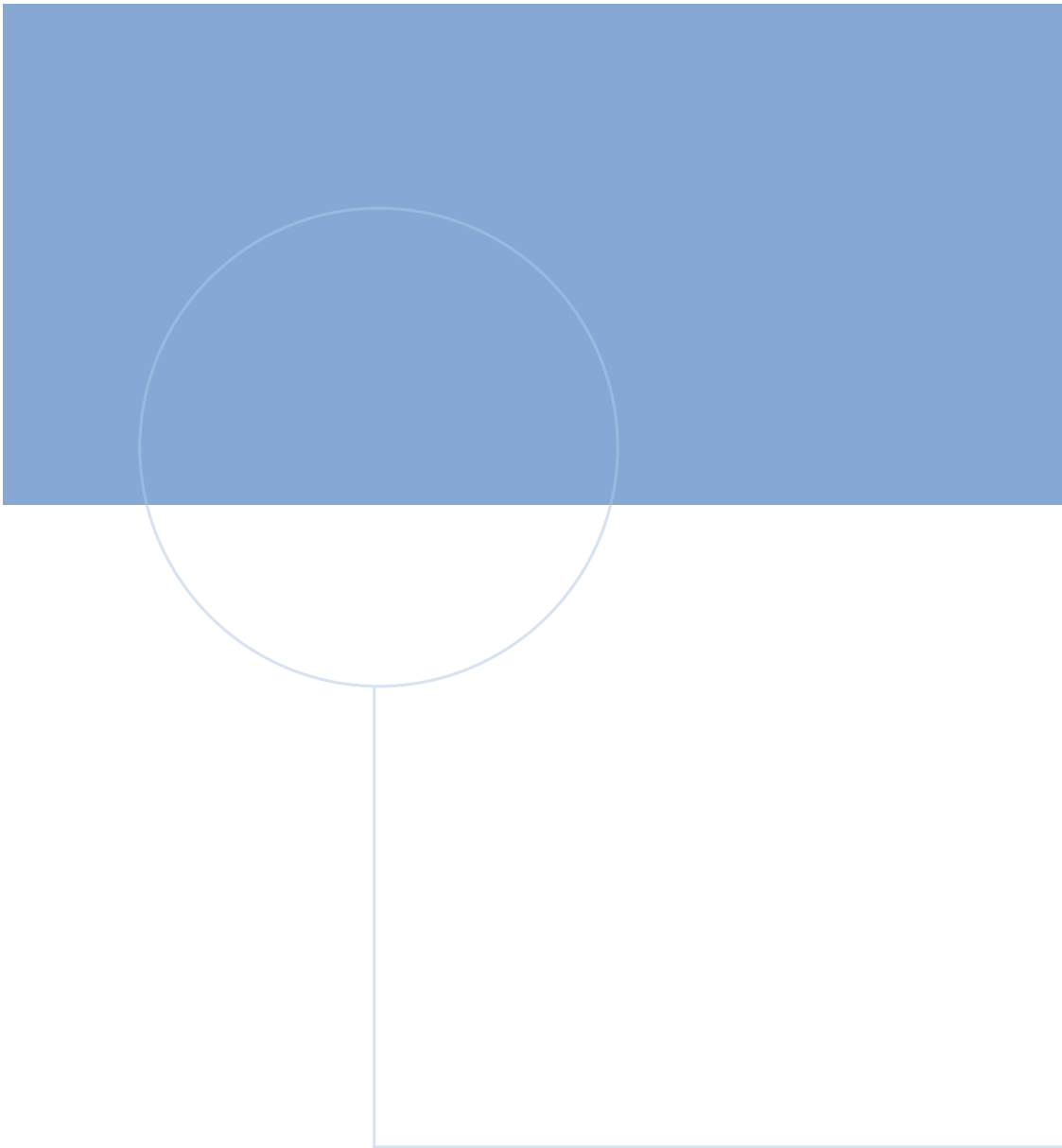
```
x2T <- popes$Year.Elected

surv_obj2 <- Surv(t, delta, type = "right")
popes_model2 <- survreg(surv_obj2~ x1 + x2T, data=popes, dist = "
    weibull")
CS2 <- exp((log(t) - popes_model2$coefficients[[1]] - popes_model2$
    coefficients[[2]]*x1 - popes_model2$coefficients[[3]]*x2T)/popes_
    model2$scale)
CS2[delta==0] <- CS2[delta==0] + 1 #adjusting the censored residuals

par(mfrow=c(2,1))
plot(x2, log(CS))
points(x2[delta==0],log(CS[delta==0]), col = "red")
points(x2[popes$Years.as.Pope=="<1"],log(CS[popes$Years.as.Pope=="<1"
    ]), pch = 24)
abline(0,0)

plot(x2T, log(CS2))
points(x2T[delta==0],log(CS2[delta==0]), col = "red")
points(x2T[popes$Years.as.Pope=="<1"],log(CS2[popes$Years.as.Pope=="<1
    "]), pch = 24)
abline(0,0)

summary(popes_model2)
```



NTNU

Kunnskap for en bedre verden