UHASSELT
KNOWLEDGE IN ACTION

Maastricht University

Faculty of Sciences
School for Information Technology
Master of Statistics and Data Science

Master's thesis

Non-parametric maximum likelihood based method to handle a left-censored covariate in a regression model.

Inez De Batselier

Thesis presented in fulfillment of the requirements for the degree of Master of Statistics and Data Science, specialization Biostatistics

SUPERVISOR :

Prof. dr. Roel BRAEKERS

2022
2023

Transnational University Limburg is a unique collaboration of two universities in two countries: the University of Hasselt and Maastricht University.

UHASSELT
KNOWLEDGE IN ACTION

www.uhasselt.be
Universiteit Hasselt
Campus Hasselt:
Martelarenlaan 42 | 3500 Hasselt
Campus Diepenbeek:
Agoralaan Gebouw D | 3590 Diepenbeek

Maastricht University

Faculty of Sciences
School for Information Technology
Master of Statistics and Data Science

Master's thesis

Non-parametric maximum likelihood based method to handle a left-censored covariate in a regression model.

Inez De Batselier

Thesis presented in fulfillment of the requirements for the degree of Master of Statistics and Data Science, specialization Biostatistics

SUPERVISOR :

Prof. dr. Roel BRAEKERS

Acknowledgement

This endeavor would not have been possible without Prof. dr. Roel Braekers. I would like to thank him for allowing me to work on this project and for his excellent supervision. During the process of this thesis he allowed me to grow in my knowledge and interest for the subject. He was always available for answering my questions and providing constructive feedback. Thanks should also go to all professors who I encountered in my bachelor in mathematics and master of statistics at the University of Hasselt. They all provided a little piece of my knowledge and managed to spark my interests. I am also grateful to my classmates for their insightful collaborations on multiple projects in my academic career. Lastly I would like to mention the continuous support received from my parents, grandparents, friends and family. Without their encouragement and emotional support I would have not been able to finish my studies.

Abstract

In many study fields left-censored data occurs regularly due to the limit of detection of measurement equipment. In a regression context many methods have been developed in the past to analyze situations where the response variable is left-censored. The situation where also the covariate becomes left-censored complicates things. In this situation they often resort to methods such as a complete case analysis or imputation methods. These methods are often not recommended due to inefficient and biased parameter estimates. To step away from these methods Tran et al. developed a maximum likelihood based method in which a parametric assumption had to be made for the left-censored covariate. This paper proposes a maximum likelihood based method for which no parametric assumption is made on the covariate. In a simulation study the performance of the proposed method was investigated for different amounts of censoring in both X and Y. The proposed method was also compared to a complete case analysis and the substitution method. The simulations have shown that the method introduced in this paper delivers sufficiently unbiased and efficient estimates for all the parameters of the regression model. It was also shown that the method outperformed the complete case analysis and the substitution method.

Contents

1 Introduction

Left-censoring is a common problem in environmental, epidemiological, biological and biomedical studies. Left-censored data occurs when an observation is known only to be less than some value. It is often the result of the presence of a limit of detection or quantification due to measurement instrument sensitivity. (Sattar et al., 2012). The limit of detection (LOD) is, according to a paper written by Tran et. al., defined as the "smallest measured concentration of an analyte from which it is possible to deduce the presence of the analyte in the test sample with acceptable certainty". While the limit of quantification (LOQ) is defined as the "smallest measured content of an analyte above which the determination can be made with the specified degree of accuracy and precision". (Tran et al., 2021) As a consequence the data is only known up to the LOD or LOQ which leaves the data left-censored.

An example of a study for which left-censoring occurs is a study conducted by the National Institute for Environmental Health Sciences (NIEHS). This study investigated the health of the workers and volunteers who participated from April to December of 2010 in the response and cleanup of the oil release after the Deepwater Horizon explosion in the Gulf of Mexico. An important part of the study was an exposure assessment to investigate the exposure-disease relationship. For this purpose exposure measurements for a variety of contaminants were collected. A large amount of these measurements was below the limit of detection which resulted in a large amount of left-censoring. Assessing the effect of the exposure measurements on a disease, results in a regression problem where the covariate is left-censored. (Huynh et al., 2014)

In the previous example the covariate was left-censored. The next example, from a paper written by Zaffora et al., shows a regression problem where both the covariate and the response are left-censored. (Zaffora et al., 2017) High-energy particle accelerators operated by e.g. the European Organization for Nuclear Research (CERN) produce radioactive waste. The characterization of the activity of the chemical compounds in the waste is quite important to ensure an appropriate disposal. Detection of the compounds ranges from easy to impossible. The limitations of the equipment to measure the concentration of some compounds results in left-censored data. The purpose of this article was to predict the activity of the hard to measure compounds by looking at the activity of the easy measurable compounds. This translates into a regression problem where both the response and covariate are concentrations of chemical compounds, which are subject to left-censoring due to equipment limitations.

When using left-censored data in a regression model to investigate the effect of a covariate on a response variable it is important to take the censoring into account. In the past most studies have focused on developing methods to handle regression models with left-censoring in the response alone. In this setting several methods where proposed based on adaptions of the methods for right-censored data. (Helsel, 2011) When both the covariate and the response become left-censored it complicates the situation. One of the conventional and easy applicable approaches to handle the censoring is the substitution method. For this method the censored observations are imputed by the LOD/LOQ or a fraction of this value. Normal analysis is then preformed on the imputed dataset. This method often leads to biased parameter estimates and underestimation of the variability in the data, especially if large proportions of data are censored. (Tran et al., 2021) Another commonly used method is to remove the censored values and perform an analysis on the remaining data. This is called a complete case analysis. This method results in loss of information, which leads to inefficient parameter estimates. If the censoring is non-informative (i.e censoring is independent from X and Y) this method does produce unbiased results. (Tran et al., 2021) Due to the drawbacks of these conventional methods other methods were proposed. One of these methods was a parametric maximum likelihood estimate approach where an assumption on the distribution of the covariate is made. This parametric method produces unbiased and efficient estimates given the correct assumption is made. (Tran et al., 2021) In this paper a method which does not make any assumptions on the distribution of the covariate is proposed. The goal is

to develop a non-parametric maximum likelihood based method to asses the effect of a left-censored covariate on a left-censored response. The method will be developed for data with one non-negative left-censored covariate and one non-negative left-censored response with multiple LOD's.

This paper is structured as follows. In section 2 the proposed method is theoretically explained and the possible drawbacks are discussed. In section 3 a simulation study is done to investigate the performance of the proposed method. In section 4 the ethical concepts and stakeholders are discussed. In section 5 a conclusion is given for this paper. Finally, in section 6 ideas for future research are discussed.

2 Methods

In this section the non-parametric likelihood function for the left-censored response and covariate will be deduced. This likelihood will be used to reach a maximum likelihood estimate for the effect of the covariate on the response without making any assumptions on the distribution of the covariate. Before explaining the proposed method, some other theoretical concepts need to be touched upon. Firstly the Riemann-Stieltjes integral for step-functions is discussed. Then a non-parametric method to estimate the cumulative density function of a left censored variable, namely the Kaplan-Meier estimate, is discussed. After that a parametric method to handle a regression model with a left censored response, namely an accelerated failure time model, is discussed. Finally all these theories discussed before are used to develop the likelihood function.

2.1 Riemann-Stieltjes integral of step function

Theorem 2.1. *If g is Riemann integrable w.r.t F on $[a,b]$ and F has a continuous derivative $F' \equiv f$ on $[a,b]$. Then the Riemann integral, $\int_a^b g(x)f(x)dx$, exists and we have:*

$$\int_a^b g(x)f(x)dx = \int_a^b g(x)dF(x)$$

(Apostol, 1974)

Theorem 2.2. *Let $F : [a,b] \to \mathbb{R}$ be a step function with discontinuities at $x_1 < \ldots < x_d$, where $a \leq x_1$ and $x_d \leq b$. Let $g : [a,b] \to \mathbb{R}$ be continuous at each x_j , $1 \leq j \leq n$. Then g is Riemann integrable w.r.t F on $[a,b]$ and*

$$\int_a^b g(x)dF(x) = \sum_{j=i}^{d} g(x_j)[F(x_j^+) - F(x_j^-)]$$

Where $F(x_i^-) = \lim_{\substack{x \to x_i \\ x < x_i}} F(x)$, $F(x_i^+) = \lim_{\substack{x \to x_i \\ x > x_i}} F(x)$
and $F(x_1^+) = a$ if $x_1 = a$, $F(x_d^+) = b$ if $x_d = b$ (Apostol, 1974)

2.2 Kaplan-Meier for left-censored data

Let M be a value larger than the maximum of a left censored variable X. The left-censored variable can be transformed into a right-censored variable by subtracting each value of X from M ($\tilde{x}_i = M - x_i$). Indeed, let x_i be a left censored value, we thus have for the real value x: $x < x_i \Rightarrow M - x > M - x_i$. The choice of the value of M does not influence the results as long as it is large enough. This value is used to flips the time axis but does not change the estimated values of the survival probability. The shift is solely done to make sure standard implemented software for right censored data can be used. (Helsel, 2011)

The survival function of the transformed variable $\tilde{X}$ is given as follows: (Helsel, 2011)

$$S(\tilde{x}) = P(\tilde{X} > \tilde{x}) = P(M - X > \tilde{x}) = P(X < M - \tilde{x}) = F_X(M - \tilde{x}) \tag{1}$$

The survival function of the transformed variable $\tilde{X}$ is thus the cumulative distribution function of X. The cumulative distribution function of X can be estimated by estimating the survival function of $\tilde{X}$ with the Kaplan-Meier estimator.

The Kaplan-Meier estimator with ties is given by (She, 1997): $\hat{S}(\tilde{x}) = \prod_{i=1}^{k} \frac{n_i - d_i}{n_i}$ for $\tilde{x}_k \leq \tilde{x} < \tilde{x}_{k+1}$ with

- n_i the 'risk set' or the number of transformed observations greater than or equal to $\tilde{x}_i$.
- d_i the number of uncensored observations occurring at $\tilde{x}$

Note that if the smallest observation x_0 of X is censored we have that: $\forall x \leq x_0,\ F_X(x) > 0$ (She, 1997). To have a correctly defined cumulative density function we define $F_X(x) = 0\ \forall x \leq x_0$ and $F_X(x) = 1\ \forall x \geq x_n$ with x_n the largest observation.

2.3 Accelerated failure time models

An accelerated failure time (AFT) model is a parametric model used to model the effect of an observed covariate on a censored variable. For this model a specific distribution is assumed on the censored variable.

In general the accelerated failure time model for a non-negative response Y can be expressed as: (Liu, 2018)

$$\log(\mathbf{Y}) = X\boldsymbol{\beta} + \sigma\boldsymbol{\epsilon} \text{ with } \boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_n), \epsilon_i \text{ i.i.d } f_{\epsilon_i}(z), E[\epsilon_i] = 0, Var(\epsilon_i) = 1$$
$$\Leftrightarrow \boldsymbol{\epsilon} = g^{-1}(\boldsymbol{Y}) = \frac{\log(\boldsymbol{Y}) - X\boldsymbol{\beta}}{\sigma}$$

Where $\boldsymbol{\beta} = (\beta_0, \beta_1, ...\beta_p)$ are the regression coefficients, $\boldsymbol{X}$ is the design matrix and σ is a scale parameter.

The probability density function of Y given X can be written as: (Liu, 2018)

$$f_{\boldsymbol{Y}}(y) = f_{\boldsymbol{\epsilon}}(g^{-1}(y))|J| \text{ with } J = \frac{\partial g^{-1}(\boldsymbol{Y})}{\partial y} = \frac{1}{\sigma y} \tag{2}$$

Several different density functions can be assumed for the error term. The most often used distributions are the standard Gumbel and normal distribution. These distributions lead respectively to a Weibull and log-normal distribution for the response Y. The following example displays how the Weibull AFT model is build. For this model we have:

$$\epsilon \sim Gumbel(0, 1) \Rightarrow f_\epsilon(x) = \exp(-(x + \exp(-x)))$$

According to formula 2, the density function of Y is given as follows: (Liu, 2018)

$$f_Y(y) = \frac{1/\sigma}{\exp(X\boldsymbol{\beta})} \left(\frac{y}{\exp(X\boldsymbol{\beta})}\right)^{\frac{1}{\sigma}-1} \exp\left[-\left(\frac{y}{\exp(X\boldsymbol{\beta})}\right)^{\frac{1}{\sigma}}\right] = \gamma\rho^{-1}(t\rho^{-1})^{\gamma-1}\exp[-(t\rho^{-1})^\gamma] \tag{3}$$

In this density the density function of the Weibull distribution with parameters ρ and γ can be recognized: (Liu, 2018)

$$Y \sim Weibull(\rho, \gamma) \text{ with } \rho = \exp(X\boldsymbol{\beta}) \text{ and } \gamma = \frac{1}{\sigma} \tag{4}$$

The cumulative density function for a variable following a Weibull distribution is given as follows: (Liu, 2018)

$$F_Y(y) = 1 - \exp\left[-(y\rho^{-1})^\gamma\right] \tag{5}$$

2.4 Log-likelihood for left censored response and covariate

Suppose both X and Y are non-negative left-censored variables. The results can be easily generalized to variables on the real line. Let there be n data pairs for X and Y. Let δ_{ix} and δ_{iy} be the censoring indicators for X and Y respectively. δ_{ix} (δ_{iy}) is equal to 1 if x_i (y_i) is observed and 0 if it is censored. Let $f(x, y)$ be the joint density function of X and Y. The likelihood for the data can be split into 4 parts, depending on which value is censored. (Tran et al., 2021)

$$\mathcal{L} = \prod_{i=1}^{n} f(x_i, y_i)$$
$$= \prod_{i=1}^{n} f(x_i, y_i)^{\delta_{ix}\delta_{iy}} \times \left(\int_0^{x_i}\int_0^{y_i} f(x, y)dxdy\right)^{(1-\delta_{ix})(1-\delta_{iy})} \times \left(\int_0^{x_i} f(x, y_i)dx\right)^{(1-\delta_{ix})\delta_{iy}} \times \left(\int_0^{y_i} f(x_i, y)dy\right)^{\delta_{ix}(1-\delta_{iy})}$$

The joint density function of X and Y can be split into a conditional density function and a marginal density function: $f(x,y) = f_{Y|X}(y|x)f_X(x)$. The main goal in this section is to eliminate the distributional assumption on the marginal density function of X.

As explained in section 2.2, the cumulative distribution function of a left-censored variable X can be estimated by use of the Kaplan-Meier estimator of the transformed variable. As a result of this estimation, the cumulative distribution function of X is estimated by a step function. By using the theory of the Riemann-Stieltjes integrals explained in section 2.1 the integrals in the likelihood function can be re-written as follows:

- Integral for censored X:

$$\int_0^{x_i} f(x,y_i)dx = \int_0^{x_i} f_{Y|X}(y_i|x)f_X(x)dx \overset{Th.\ 2.1}{=} \int_0^{x_i} f_{Y|X}(y_i|x)dF_X(x) \overset{Th.\ 2.2}{=} \sum_{x_j \leq x_i} f_{Y|X}(y_i|x_j)[F(x_j^+) - F(x_j^-)]$$
$$\equiv \sum_{x_j \leq x_i} f_{Y|X}(y_i|x_j)w_j$$

- Integral for censored Y:

$$\int_0^{y_i} f(x_i,y)dy = \int_0^{y_i} f_{Y|X}(y|x_i)f_X(x_i)dy = f_X(x_i)\int_0^{y_i} f_{Y|X}(y|x_i)dy = f_X(x_i)F_{Y|X}(y_i|x_i)$$

- Integral for censored X and Y:

$$\int_0^{x_i}\int_0^{y_i} f(x,y)dxdy = \int_0^{x_i}\int_0^{y_i} f_{Y|X}(y|x)f_X(x)dxdy = \int_0^{x_i} f_X(x)\left(\int_0^{y_i} f_{Y|X}(y|x)dy\right)dx = \int_0^{x_i} f_X(x)F_{Y|X}(y_i|x)dx$$
$$\overset{Th.\ 2.1}{=} \int_0^{x_i} F_{Y|X}(y_i|x)dF_X(x) \overset{Th.\ 2.2}{=} \sum_{x_j \leq x_i} F_{Y|X}(y_i|x_j)[F(x_j^+) - F(x_j^-)] \equiv \sum_{x_j \leq x_i} F_{Y|X}(y_i|x_j)w_j$$

The likelihood can now be written as:

$$\mathcal{L} = \prod_{i=1}^n f_{Y|X}(y_i|x_i)f_X(x_i)^{\delta_{ix}\delta_{iy}} \times \left(\sum_{x_j \leq x_i} F_{Y|X}(y_i|x_j)w_j\right)^{(1-\delta_{ix})(1-\delta_{iy})} \times \left(\sum_{x_j \leq x_i} f_{Y|X}(y_i|x_j)w_j\right)^{(1-\delta_{ix})\delta_{iy}}$$
$$\times \left(f_X(x_i)F_{Y|X}(y_i|x_i)\right)^{\delta_{ix}(1-\delta_{iy})}$$

The log-likelihood can than be written as:

$$\ell = \log(\mathcal{L}) = \sum_{i=1}^n \delta_{ix}\delta_{iy}\log(f_{Y|X}(y_i|x_i)) + \delta_{ix}\delta_{iy}(\log(f_X(x_i)) + (1-\delta_{ix})(1-\delta_{iy})\log\left(\sum_{x_j \leq x_i} F_{Y|X}(y_i|x_j)w_j\right)$$
$$+ (1-\delta_{ix})\delta_{iy}\log\left(\sum_{x_j \leq x_i} f_{Y|X}(y_i|x_j)w_j\right) + \delta_{ix}(1-\delta_{iy})\log(f_X(x_i)) + \delta_{ix}(1-\delta_{iy})\log(F_{Y|X}(y_i|x_i))$$

Let the conditional density of Y given X depend on regression parameters $\boldsymbol{\beta}$ and nuisance parameter σ. Assume for example that Y follows the parametric weibull AFT model with regression parameters $\boldsymbol{\beta}$ and nuisance parameter σ as shown in section 2.3:

$$f_{Y|X}(y_i|x_i,\beta,\sigma) = \gamma\rho^{-1}(y_i\rho^{-1})^{\gamma-1}\exp[-(y_i\rho^{-1})^\gamma] \text{ with } \rho = \exp(\beta_0 + x_i\beta_1) \text{ and } \gamma = \frac{1}{\sigma}$$

The maximum likelihood estimate for β_i and σ can be reached by solving the following estimating equations:

$$\begin{aligned}
\frac{\partial \ell}{\partial \beta_i} &= \sum_{i=1}^{n} \delta_{ix}\delta_{iy}\frac{\partial}{\partial \beta_i}\log(f_{Y|X}(y_i|x_i,\boldsymbol{\beta},\sigma)) + (1-\delta_{ix})(1-\delta_{iy})\frac{\partial}{\partial \beta_i}\log\left(\sum_{x_j\leq x_i} F_{Y|X}(y_i|x_j,\boldsymbol{\beta},\sigma)w_j\right) \\
&\quad + (1-\delta_{ix})\delta_{iy}\frac{\partial}{\partial \beta_i}\log\left(\sum_{x_j\leq x_i} f_{Y|X}(y_i|x_j,\boldsymbol{\beta},\sigma)w_j\right) + \delta_{ix}(1-\delta_{iy})\frac{\partial}{\partial \beta_i}\log(F_{Y|X}(y_i|x_i,\boldsymbol{\beta},\sigma)) \\
\frac{\partial \ell}{\partial \sigma} &= \sum_{i=1}^{n} \delta_{ix}\delta_{iy}\frac{\partial}{\partial \sigma}\log(f_{Y|X}(y_i|x_i,\boldsymbol{\beta},\sigma)) + (1-\delta_{ix})(1-\delta_{iy})\frac{\partial}{\partial \sigma}\log\left(\sum_{x_j\leq x_i} F_{Y|X}(y_i|x_j,\boldsymbol{\beta},\sigma)w_j\right) \\
&\quad + (1-\delta_{ix})\delta_{iy}\frac{\partial}{\partial \sigma}\log\left(\sum_{x_j\leq x_i} f_{Y|X}(y_i|x_j,\boldsymbol{\beta},\sigma)w_j\right) + \delta_{ix}(1-\delta_{iy})\frac{\partial}{\partial \sigma}\log(F_{Y|X}(y_i|x_i,\boldsymbol{\beta},\sigma))
\end{aligned}$$

Where: $w_j = F(x_j^+) - F(x_j^-) \overset{(1)}{=} \hat{S}(M - x_j^+) - \hat{S}(M - x_j^-)$

These estimating equations have no simple analytical solution. They have to be solved by the use of numerical optimization. In this paper it was chosen to implement the method with the Nelder–Mead optimization method.

The above estimating equations do not make any parametric assumptions on the distribution of the covariate X. Indeed, the cumulative density function of X is estimated with the non-parametric Kaplan-Meier estimate. The maximum likelihood estimates for the parameters in the AFT model can thus be reached without making any assumptions on the distribution of the covariate X, which was the purpose of this section.

2.5 Possible drawbacks of method

The first drawback of the method is the amount of censoring allowed for both X and Y. A large amount of censoring in X, in the case of a rare event, may result in a bad fit for the Kaplan-Meier estimator. Since the KM estimator is used to estimate the cumulative density function of X this may results in biased and inefficient parameter estimates. A large amount of censoring in Y may result in a bad fit for the accelerated failure time model, which in turn again results in biased and inefficient parameter estimates. These problems that are encountered for large amounts of censoring may also occur if a small sample size is considered.

The next problem that may occur is the non-convergence of the numerical methods used to maximize the log-likelihood. This may be avoided by choosing good initial values. For simulation studies this is not a problem, but in real data examples this forms a challenge. Good initial values for the parameters may be deduced from a complete case analysis or a substitution method.

The last problem concerns the parametric assumptions on the conditional distribution of Y given X. The choice of the right distribution for Y is an important aspect for the method to give valid results. For simulations finding the right distribution is no problem, while for real data this does impose a challenge. The goodness of fit of the parametric distribution must be checked by use of for example residual plots.

3 Simulation Study

3.1 Methods

For the simulation of the data a uniform distribution for the covariate X and a Weibull distribution for the response Y are assumed. Take $X \sim U(0, \lambda)$ and $Y|X \sim Weibull(\rho, \gamma)$. The parameter for the distribution of X, namely λ, is arbitrarily chosen as 5. The parameters for the distribution of Y, namely ρ and γ, follow from the assumed AFT model: $\log(Y_i) = \beta_0 + \beta_1 X_i + \sigma\epsilon_i$ where β_0, β_1 and σ are arbitrarily chosen. γ and ρ are then defined as $\gamma = 1/\sigma$ and $\rho = \exp(\beta_0 + \beta_1 X)$. Both parameters ρ and γ are restricted to be non-negative. The parameters which need to be optimized, namely β_0, β_1 and σ cannot impose any restriction. To ensure that the non-negative restriction on γ is satisfied, an exponential transformation is used for σ: $\sigma = \exp(\eta)$ where η can take on any real value. The restrictions on ρ is ensured by the definition of this value. For the simulations in this paper β_0 and β_1 are chose as 1 and σ is chosen as 0.5.

The censoring for X and Y is simulated by assuming uniform censoring distributions: $C_x \sim U(0, \lambda_x)$ and $C_y \sim U(0, \lambda_y)$ where the parameters λ_x and λ_y are defined to achieve a certain percentage of censoring in respectively X and Y. The censoring is assumed to be independent from both X and Y. The data for the censored covariate X is simulated as follows: Let X be a value sampled from $U(0, \lambda)$ and C_x a value sampled from $U(0, \lambda_x)$, the so called LOD. The data point that will be used is equal to the maximum of these two values, $\max(C_x, X)$. This results in a censored value if X is smaller than the LOD C_x and an observed value if X is larger than C_x. As mentioned before, the choice of λ_X is motivated by a desired probability π_x of censoring. The probability of censoring in X can be calculated as follows: (Ramos et al., 2020)

$$\pi_x = P(X < C_x) = \int_0^\infty \int_0^c f_{X,C_x}(x,c)dxdc = \int_0^\infty \int_0^c f_X(x) f_{C_x}(c)dxdc = \int_0^\infty F_X(c) f_{C_x}(c)dc$$

$$= \int_0^\lambda \frac{c}{\lambda} f_{C_x}(c)dc + \int_\lambda^\infty f_{C_x}(c)dc = \int_0^\lambda \frac{c}{\lambda}\frac{1}{\lambda_x}dc = \frac{1}{\lambda_x \lambda}\int_0^\lambda cdc = \frac{\lambda_x}{2\lambda}$$

The probability of censoring is thus dependent on the parameter values from both the distribution behind the censoring mechanism and the distribution of X. From the integral above it follows that if a probability π_x of censoring is requested, the parameter λ_x of the uniform censoring distribution should satisfy: $\lambda_x = 2\pi_x\lambda$.

The censoring for Y can be simulated analogously to the censoring for X. The derivation of the value for λ_y to reach a certain percentage of censoring is not as straightforward as the derivation for λ_x. The first complexity that emerges is the fact that the distribution of Y is conditional on X. This in turn results in the dependence of the probability of censoring on X. Conditional on X the probability of censoring in Y can be calculated as follows:

$$\pi_y = P(Y < C_y|X) = \int_0^\infty \int_0^c f_{Y,C_x|X}(y,c|x)dydc = \int_0^\infty \int_0^c f_{Y|X}(y|x) f_{C_x}(c)dxdc$$

$$= \int_0^\infty F_{Y|X}(c|x) f_{C_x}(c)dc = \frac{1}{\lambda_y}\int_0^{\lambda_y} F_{Y|X}(c|x)dc$$

The next complication that emerges is the fact that the integral above has no simple analytical solution. The integral can be solved with the help of numerical integration techniques. The function `integrate` in R (R Core Team, 2023), which makes use of an adaptive enrichment method, can be used. The conditionality of the probability on X makes it difficult to determine an exact value for λ_y to reach

a certain probability of censoring in Y. For the simulations in this paper the values of λ_y are chosen by simulating data sets with different values for this parameter. For each dataset the proportion of censoring can be estimated by dividing the total number of censored observations in Y by the total number of observations. In table 1 the λ_y's corresponding to approximately 10%, 25%, 50% and 75% of censoring are given. In figure 1 the probability of censoring in function of X is presented for the values of λ_y in table 1. In this figure it can be seen that, for all λ_y, larger values of X result in a lower probability of censoring in Y. It can also be seen that for a fixed value of X a larger value of λ_y results in a larger probability of censoring in Y.

$\boldsymbol{\lambda_y}$	$\boldsymbol{E[\pi_y]}$
6	0.1055
15	0.242
62	0.502
230	0.7485

Table 1: Values of λ_y corresponding to 10%, 25%, 50% and 75% mean probability of censoring in Y

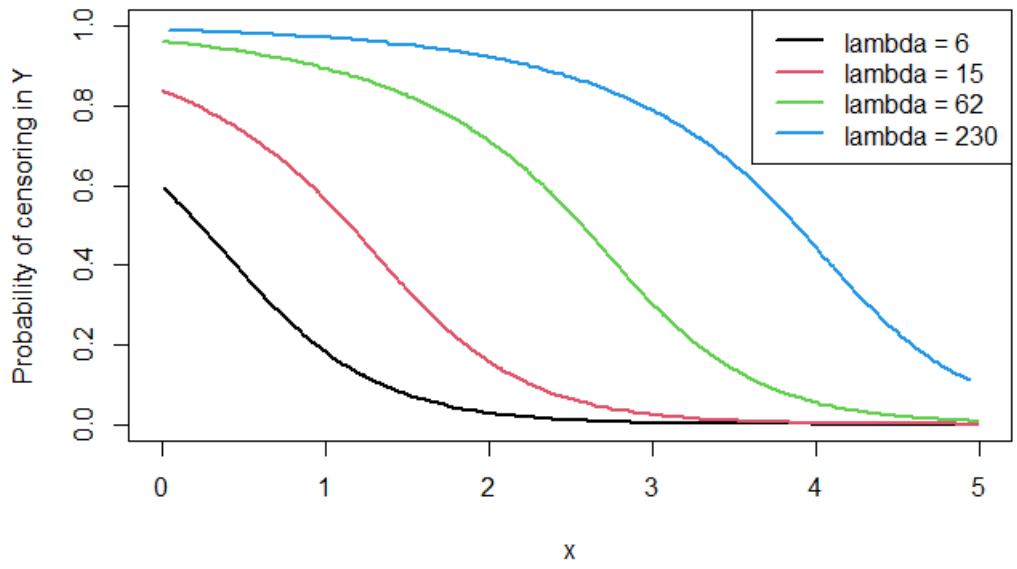

Figure 1: Probability of censoring in function of x for the values of λ_y presented in table 1.

Multiple situations will be simulated to test the performance of the method. Both the probability of censoring in X and Y will be varied from small to large. The number of observations (n) and simulations ($nsim$) will also be varied. The number of observations will be taken equal to 100, 200 and 500 to check the performance on small and large sample sizes. The standard number of simulations is taken to be 100 or 200. Since the code runs sufficiently fast, 500 simulations will also be considered. For each simulation the mean, bias, mean square error (MSE) and mean empirical standard error (se) of each parameter will be calculated. To calculate the se of σ the transformation of this parameter has to be taken into account via the delta method. For each parameter the theoretical 95% confidence intervals (CI) are calculated. For this the following normality assumptions are made:

- $\hat{\beta}_i \xrightarrow{D} N(\beta_i, se_{\beta_i})$ for $n \to \infty$, $i = 1, 2$
- $\hat{\sigma} \xrightarrow{D} N(\sigma, se_\sigma)$ for $n \to \infty$

These normality assumptions will be checked with normal Q-Q plots. For each of the confidence intervals the coverage probabilities will also be calculated. Lastly the results from the proposed method in this paper will be compared with the results from a complete case analysis and the substitution method. For the complete case analysis each observation with a censored value for X will be ignored. On the remaining observations a Weibull AFT model is performed. For the substitution method the censored values of X will be imputed by the LOD. A Weibull AFT model is then performed on the imputed dataset. In subsection 3.2 the results for each simulation are compared and discussed.

The simulations were conducted using R (R Core Team, 2023), R studio (Posit team, 2023), the survival package (Therneau, 2023) and the DescTools package (Signorell, 2023). For purposes of reproducibility a seed with value `2023` is used.

3.2 Results and discussion

In figure 2 the Q-Q plots for the three parameters are given for the situation where the percentage of censoring for both X and Y are equal to 10%. In these plots it can be seen that the normality assumption is valid for all the parameters. Increasing the percentage of censoring in X to 50% and 75%, while keeping the percentage of censoring in Y equal to 10% produces respectively the Q-Q plots in figure 3 and 4. The normality assumption for σ is again valid here. For β_0 and β_1 the Q-Q plot suggests a deviation from the normality assumption. The plots show overdispersion, the standard deviation from the normal distribution is higher than the standard error of the parameters. When increasing the censoring in Y to 75%, while keeping the percentage of censoring in X equal to 10% the Q-Q plots in figure 5 are produced. For all parameters the normality assumption is valid again, although there seems to be slight skewness for the β parameters. If both the censoring in X and Y is increased to 75% the Q-Q plots in 6 are produced. The overdispersion that was seen in figure 3 and 4 is not present anymore for the β parameters. In appendix A the Q-Q plots for all other combinations of the probabilities of censoring are given. In general the normality assumptions are valid for all parameters when there is 10% and 25% censoring in X, even when the probability of censoring in Y is increased. Increasing the probability in X to 50% and 75% causes a deviation form the normality assumption for the β parameters. In these cases overdispersion is observed. The overdispersion disappears when the censoring in Y is also increased. An explanation for this may be the large increase of the standard errors due to large amounts of censoring.

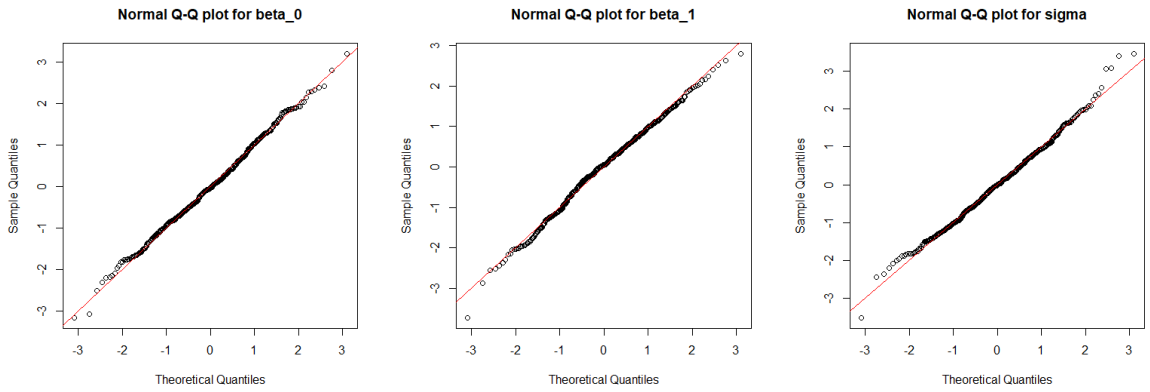

Figure 2: Normal Q-Q plots for β_0, β_1 and σ with the red line representing the line $y = x$ with $n = 500$, $nsim = 500$, probability of censoring in $X = 10\%$ and probability of censoring in $Y = 10\%$

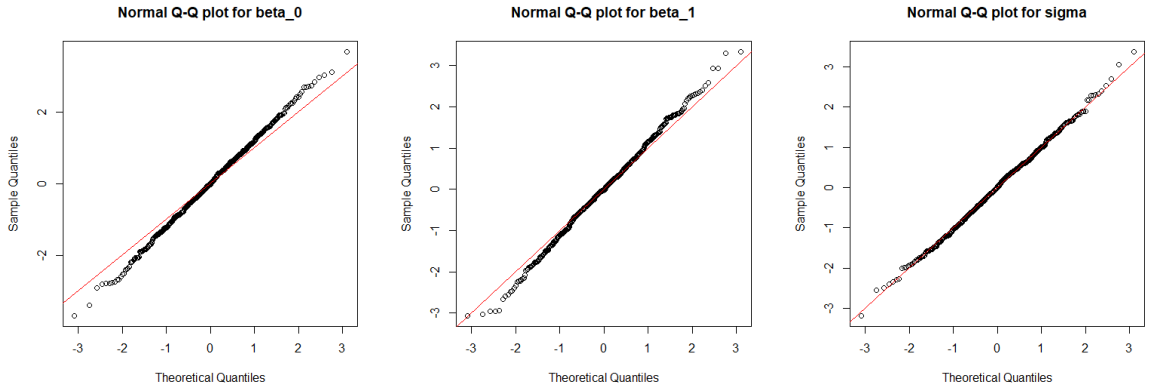

Figure 3: Normal Q-Q plots for β_0, β_1 and σ with the red line representing the line $y = x$ with $n = 500$, $nsim = 500$, probability of censoring in $X = 50\%$ and probability of censoring in $Y = 10\%$

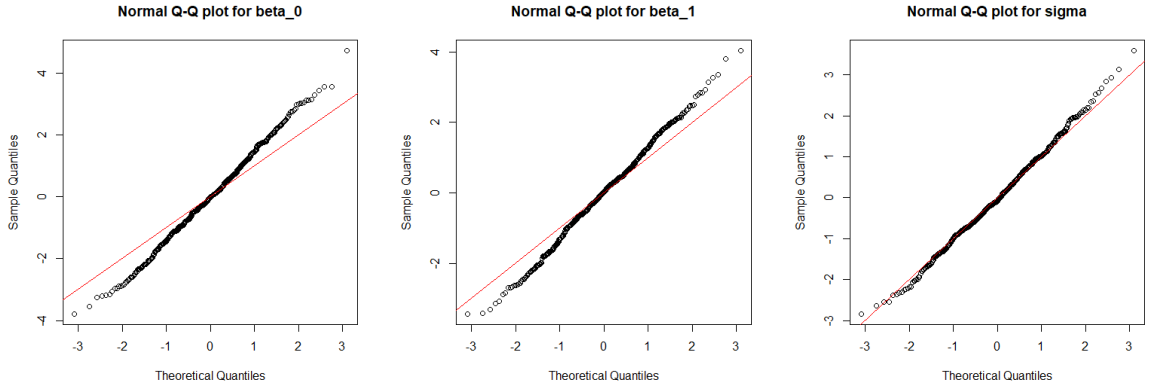

Figure 4: Normal Q-Q plots for β_0, β_1 and σ with the red line representing the line $y = x$ with $n = 500$, $nsim = 500$, probability of censoring in $X = 75\%$ and probability of censoring in $Y = 10\%$

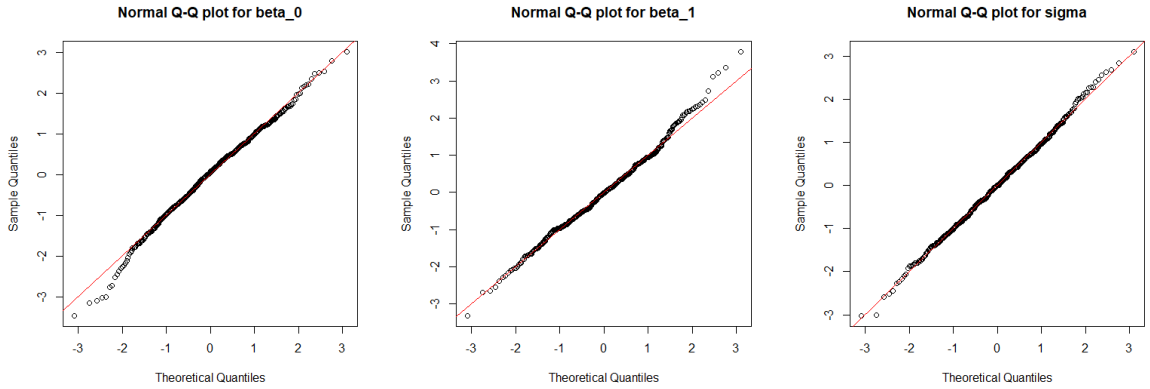

Figure 5: Normal Q-Q plots for β_0, β_1 and σ with the red line representing the line $y = x$ with $n = 500$, $nsim = 500$, probability of censoring in $X = 10\%$ and probability of censoring in $Y = 75\%$

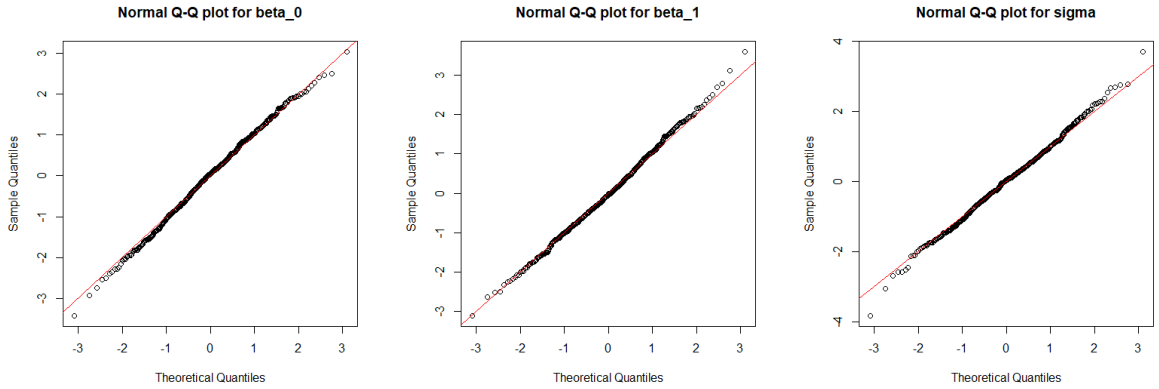

Figure 6: Normal Q-Q plots for β_0, β_1 and σ with the red line representing the line $y = x$ with $n = 500$, $nsim = 500$, probability of censoring in $X = 75\%$ and probability of censoring in $Y = 75\%$

In table 2 the mean parameter estimates along with their bias, mean square error and mean standard error are given for the different percentages of censoring in X and Y. Each value in this table was produced under 500 observations and 500 simulations. Looking at the bias for each parameter, the method performs considerable well in all scenarios with differing percentage of censoring. It can be seen that an increase in the probability of censoring in both X and Y results in an increase in the bias, MSE and se. What can be noticed is that the effect of increasing the censoring in Y is larger than the effect of increasing the censoring in X. In general the bias, MSE and se for β_0 are larger than those for β_1 and σ. In table 5 and 7 in appendix B the results for the number of observations equal to 100 and 200 are given where $nsim = 500$. Decreasing the number of observations results in an increased bias, MSE and se. Although there is an increase, it is not drastic. The method still produces sufficiently unbiased estimates for a smaller number of observations. In tables 9 and 11 in appendix B the results for a smaller number of simulations are given. The results for a smaller number of simulations show the same trend as the results for $nsim = 500$. Not all tables for each combination of $nsim$ and n are presented in this paper due to similar trends in the results.

In table 3 the confidence intervals and coverage probabilities for each parameter in the different scenarios are given where the number of observations and simulations were both equal to 500. It can be seen that the width of the confidence interval increases with increasing probability of censoring in both X and Y. The coverage probability for σ lies approximately always around 0.95. For β_0 and β_1 this is not always the case, especially for larger amounts of censoring in X. As shown before the normality assumption for β_0 and β_1 is not valid for larger censoring probabilities in X, which also reflects in the coverage probabilities. For 50% and 75% censoring in X the coverage probability again increases to 0.95% for increasing censoring in Y. This may be due to the large uncertainty around the parameter, which results in very wide confidence intervals. In table 6 and 8 in appendix B the confidence intervals for smaller number of observations are shown. Smaller number of observations result in wider confidence intervals due to the larger standard errors. When also large censoring probabilities are observed this may become a problem. As can be seen in table 6 the confidence interval for a censoring probability of 75% in both X and Y includes zero. The confidence interval became so wide it suggests no relationship between X and Y while there is a relationship. For the larger censoring probability in X the deviation from the coverage probability of 95% is larger for smaller number of observations. An explanation may be the larger deviation from the asymptotic normality due to smaller number of observations.

In table 4 a comparison of the proposed method in this paper with the complete case analysis and substitution method is given. The results in this table are produced with 500 observations and simulations. It can be seen that the bias and the standard error for the method in this paper are in general smaller than those for the complete case analysis and substitution method, especially for large amounts of censoring in X. The amount of censoring in Y does not seem to have a large influence on the differences in the bias and standard error between the methods. The substitution method shows highly biased estimates when the censoring in X increases. The complete case analysis still produces sufficiently unbiased, but less efficient, estimates compared to the estimates from the method in this paper. In table 13 in appendix B the results for the comparison of the methods in the case of a smaller number of observations is given ($n = 100$). For a smaller number of observations the difference in performance of our method and the others becomes even more clear, especially in the situation of a large percentage of censoring in X.

		$\beta_0 = 1$				$\beta_1 = 1$				$\sigma = 0.5$			
Censoring in X	**Censoring in Y**	**Estimate**	**Bias**	**MSE**	**SE**	**Estimate**	**Bias**	**MSE**	**SE**	**Estimate**	**Bias**	**MSE**	**SE**
10%	10%	0.99941	-0.00059	0.00223	0.04789	0.99953	-0.00047	0.00026	0.01610	0.49861	-0.00139	0.00033	0.01835
	25%	0.99660	-0.00340	0.00298	0.05577	1.00027	0.00027	0.00031	0.01789	0.49868	-0.00132	0.00038	0.01967
	50%	0.99328	-0.00672	0.00816	0.08652	1.00103	0.00103	0.00064	0.02463	0.49851	-0.00149	0.00055	0.02385
	75%	0.98385	-0.01615	0.02246	0.14569	1.00322	0.00322	0.00146	0.03688	0.49656	-0.00344	0.00108	0.03249
25%	10%	0.99872	-0.00128	0.00350	0.05494	0.99966	-0.00034	0.00035	0.01782	0.49892	-0.00108	0.00036	0.01941
	25%	0.99564	-0.00436	0.00413	0.06270	1.00051	0.00051	0.00039	0.01960	0.49879	-0.00121	0.00039	0.02049
	50%	0.99150	-0.00850	0.00933	0.09332	1.00145	0.00145	0.00071	0.02633	0.49873	-0.00127	0.00056	0.02422
	75%	0.97940	-0.02060	0.02579	0.15482	1.00425	0.00425	0.00166	0.03906	0.49668	-0.00332	0.00110	0.03272
50%	10%	0.99807	-0.00193	0.00639	0.06593	0.99966	-0.00034	0.00052	0.02020	0.50100	0.00100	0.00055	0.02303
	25%	0.99659	-0.00341	0.00708	0.07551	1.00012	0.00012	0.00055	0.02233	0.50017	0.00017	0.00057	0.02415
	50%	0.99342	-0.00658	0.01326	0.11099	1.00100	0.00100	0.00092	0.03016	0.49887	-0.00113	0.00073	0.02750
	75%	0.98049	-0.01951	0.03369	0.18305	1.00381	0.00381	0.00205	0.04522	0.49706	-0.00294	0.00130	0.03574
75%	10%	0.99673	-0.00327	0.01094	0.07322	0.99986	-0.00014	0.00084	0.02230	0.50425	0.00425	0.00087	0.02740
	25%	0.99556	-0.00444	0.01192	0.08424	1.00035	0.00035	0.00088	0.02466	0.50231	0.00231	0.00091	0.02885
	50%	0.98984	-0.01016	0.01940	0.12415	1.00204	0.00204	0.00129	0.03332	0.49893	-0.00107	0.00108	0.03267
	75%	0.97255	-0.02745	0.04708	0.20590	1.00591	0.00591	0.00279	0.05040	0.49534	-0.00466	0.00196	0.04186

Table 2: Parameter estimates and goodness of fit measures for $n = 500$ and $nsim = 500$

		$\beta_0 = 1$				$\beta_1 = 1$				$\sigma = 0.5$			
Censoring in X	**Censoring in Y**	**Estimate**	**95% CI**		**CP**	**Estimate**	**95% CI**		**CP**	**Estimate**	**95% CI**		**CP**
			Lower	**Upper**			**Lower**	**Upper**			**Lower**	**Upper**	
10%	10%	0.99941	0.90555	1.09327	0.964	0.99953	0.96797	1.03108	0.946	0.49861	0.46265	0.53457	0.960
	25%	0.99660	0.88729	1.10591	0.956	1.00027	0.96521	1.03534	0.952	0.49868	0.46013	0.53723	0.962
	50%	0.99328	0.82371	1.16285	0.948	1.00103	0.95276	1.04931	0.942	0.49851	0.45177	0.54525	0.956
	75%	0.98385	0.69831	1.26939	0.944	1.00322	0.93094	1.07550	0.934	0.49656	0.43288	0.56024	0.944
25%	10%	0.99872	0.89103	1.10641	0.944	0.99966	0.96474	1.03458	0.940	0.49892	0.46087	0.53697	0.962
	25%	0.99564	0.87274	1.11854	0.954	1.00051	0.96209	1.03892	0.950	0.49879	0.45863	0.53896	0.962
	50%	0.99150	0.80859	1.17441	0.948	1.00145	0.94984	1.05306	0.950	0.49873	0.45126	0.54620	0.962
	75%	0.97940	0.67594	1.28285	0.940	1.00425	0.92770	1.08081	0.932	0.49668	0.43253	0.56082	0.946
50%	10%	0.99807	0.86884	1.12730	0.898	0.99966	0.96006	1.03926	0.926	0.50100	0.45587	0.54613	0.958
	25%	0.99659	0.84859	1.14459	0.924	1.00012	0.95635	1.04389	0.936	0.50017	0.45284	0.54749	0.956
	50%	0.99342	0.77587	1.21097	0.940	1.00100	0.94188	1.06011	0.944	0.49887	0.44496	0.55278	0.964
	75%	0.98049	0.62171	1.33928	0.954	1.00381	0.91518	1.09244	0.954	0.49706	0.42701	0.56710	0.958
75%	10%	0.99673	0.85323	1.14023	0.814	0.99986	0.95616	1.04357	0.848	0.50425	0.45055	0.55796	0.924
	25%	0.99556	0.83045	1.16067	0.868	1.00035	0.95202	1.04869	0.890	0.50231	0.44577	0.55885	0.936
	50%	0.98984	0.74651	1.23318	0.916	1.00204	0.93673	1.06735	0.908	0.49893	0.43490	0.56295	0.948
	75%	0.97255	0.56898	1.37611	0.946	1.00591	0.90713	1.10469	0.942	0.49534	0.41330	0.57738	0.940

Table 3: Confidence intervals and Coverage probabilities for $n = 500$ and $nsim = 500$

			$\beta_0 = 1$				$\beta_1 = 1$				σ			
Censoring in X	**Censoring in Y**	**Method**	**Estimate**	**Bias**	**MSE**	**SE**	**Estimate**	**Bias**	**MSE**	**SE**	**Estimate**	**Bias**	**MSE**	**SE**
10%	10%	Our	0.99941	-0.00059	0.00223	0.04789	0.99953	-0.00047	0.00026	0.01610	0.49861	0.00139	0.00033	0.01835
		CC	0.99824	-0.00176	0.00290	0.05541	0.99984	-0.00016	0.00031	0.01797	0.49859	-0.00141	0.00034	0.06208
		LOD	0.91911	-0.08089	0.00897	0.04932	1.02084	0.02084	0.00071	0.01673	0.50755	0.00755	0.00039	0.06050
	25%	Our	0.99660	-0.00340	0.00298	0.05577	1.00027	0.00027	0.00031	0.01789	0.49868	-0.00132	0.00038	0.01967
		CC	0.99526	-0.00474	0.00352	0.06109	1.00063	0.00063	0.00035	0.01925	0.49872	-0.00128	0.00038	0.06581
		LOD	*	*	*	*	*	*	*	*	*	*	*	*
	50%	Our	0.99328	-0.00672	0.00816	0.08652	1.00103	0.00103	0.00064	0.02463	0.49851	-0.00149	0.00055	0.02385
		CC	0.99145	-0.00855	0.00883	0.09122	1.00150	0.00150	0.00069	0.02582	0.49854	-0.00146	0.00055	0.07916
		LOD	*	*	*	*	*	*	*	*	*	*	*	*
	75%	Our	0.98385	-0.01615	0.02246	0.14569	1.00322	0.00322	0.00146	0.03688	0.49656	-0.00344	0.00108	0.03249
		CC	0.98109	-0.01891	0.02407	0.15145	1.00388	0.00388	0.00157	0.03825	0.49657	-0.00343	0.00108	0.10780
		LOD	0.95718	-0.04282	0.02384	0.14586	1.00964	0.00964	0.00153	0.03695	0.49775	-0.00225	0.00108	0.10761
25%	10%	Our	0.99872	-0.00128	0.00350	0.05494	0.99966	-0.00034	0.00035	0.01782	0.49892	-0.00108	0.00036	0.01941
		CC	0.99578	-0.00422	0.00486	0.07185	1.00053	0.00053	0.00044	0.02170	0.49816	-0.00184	0.00040	0.06728
		LOD	0.66260	-0.33740	0.11885	0.06626	1.06900	0.06900	0.00520	0.02173	0.60031	0.10031	0.01055	0.06733
	25%	Our	0.99564	-0.00436	0.00413	0.06270	1.00051	0.00051	0.00039	0.01960	0.49879	-0.00121	0.00039	0.02049
		CC	0.99275	-0.00725	0.00577	0.07663	1.00132	0.00132	0.00050	0.02281	0.49830	-0.00170	0.00041	0.06977
		LOD	0.65327	-0.34673	0.12623	0.07216	1.07388	0.07388	0.00597	0.02300	0.58593	0.08593	0.00791	0.07117
	50%	Our	0.99150	-0.00850	0.00933	0.09332	1.00145	0.00145	0.00071	0.02633	0.49873	0.00127	0.00056	0.02422
		CC	0.98816	-0.01184	0.01136	0.10544	1.00233	0.00233	0.00084	0.02920	0.49859	0.00141	0.00057	0.08113
		LOD	0.72058	0.27942	0.08844	0.09816	1.06345	0.06345	0.00482	0.02813	0.53886	-0.03886	0.00215	0.08209
	75%	Our	0.97940	-0.02060	0.02579	0.15482	1.00425	0.00425	0.00166	0.03906	0.49668	-0.00332	0.00110	0.03272
		CC	0.98102	-0.01898	0.03091	0.16923	1.00385	0.00385	0.00197	0.04235	0.49657	-0.00343	0.00112	0.10911
		LOD	0.76869	-0.23131	0.08255	0.15707	1.05137	0.05137	0.00448	0.03988	0.51962	0.01962	0.00164	0.11045

Table 4: Parameter estimates and goodness of fit measures for $n = 500$ and $nsim = 500$ for the method introduced in this paper (Our), complete case analysis (CC) and substitution method (LOD). $*$ indicates that the AFT model did not converge to a value.

			$\beta_0 = 1$				$\beta_1 = 1$				σ			
Censoring in X	**Censoring in Y**	**Method**	**Estimate**	**Bias**	**MSE**	**SE**	**Estimate**	**Bias**	**MSE**	**SE**	**Estimate**	**Bias**	**MSE**	**SE**
50%	10%	Our	0.99807	-0.00193	0.00639	0.06593	0.99966	-0.00034	0.00052	0.02020	0.50100	0.00100	0.00055	0.02303
		CC	0.99796	-0.00204	0.00939	0.09642	1.00008	0.00008	0.00071	0.02705	0.49757	-0.00243	0.00055	0.08216
		LOD	0.66986	-0.33014	0.11987	0.13513	0.93448	-0.06552	0.00529	0.03812	1.00044	0.50044	0.25219	0.10487
	25%	Our	0.99659	-0.00341	0.00708	0.07551	1.00012	0.00012	0.00055	0.02233	0.50017	0.00017	0.00057	0.02415
		CC	0.99589	-0.00411	0.01066	0.10206	1.00059	0.00059	0.00080	0.02833	0.49754	-0.00246	0.00057	0.08438
		LOD	0.58692	-0.41308	0.18395	0.14799	0.95275	-0.04725	0.00335	0.04112	1.02493	0.52493	0.27771	0.11586
	50%	Our	0.99342	-0.00658	0.01326	0.11099	1.00100	0.00100	0.00092	0.03016	0.49887	-0.00113	0.00073	0.02750
		CC	0.98915	-0.01085	0.01910	0.13718	1.00220	0.00220	0.00128	0.03607	0.49709	-0.00291	0.00076	0.09455
		LOD	0.48872	-0.51128	0.28830	0.19445	0.98134	-0.01866	0.00223	0.05066	0.99540	0.49540	0.24791	0.13570
	75%	Our	0.98049	-0.01951	0.03369	0.18305	1.00381	0.00381	0.00205	0.04522	0.49706	-0.00294	0.00130	0.03574
		CC	0.96924	-0.03076	0.04741	0.21840	1.00649	0.00649	0.00280	0.05295	0.49569	-0.00431	0.00133	0.12210
		LOD	0.45704	-0.54296	0.35997	0.28362	1.00208	0.00208	0.00371	0.06779	0.90426	0.40426	0.16760	0.16766
75%	10%	Our	0.99673	-0.00327	0.01094	0.07322	0.99986	-0.00014	0.00084	0.02230	0.50425	0.00425	0.00087	0.02740
		CC	0.99388	-0.00612	0.01328	0.11812	1.00111	0.00111	0.00104	0.03312	0.49544	-0.00456	0.00093	0.10062
		LOD	2.35946	1.35946	1.88051	0.21124	0.36454	-0.63546	0.40582	0.04702	1.39481	0.89481	0.80264	0.14710
	25%	Our	0.99556	-0.00444	0.01192	0.08424	1.00035	0.00035	0.00088	0.02466	0.50231	0.00231	0.00091	0.02885
		CC	0.98969	-0.01031	0.01556	0.12520	1.00216	0.00216	0.00118	0.03474	0.49539	-0.00461	0.00099	0.10336
		LOD	2.30668	1.30668	1.74253	0.22142	0.36852	-0.63148	0.40088	0.04902	1.45468	0.95468	0.91401	0.16948
	50%	Our	0.98984	-0.01016	0.01940	0.12415	1.00204	0.00204	0.00129	0.03332	0.49893	-0.00107	0.00108	0.03267
		CC	0.98165	-0.01835	0.02639	0.16846	1.00410	0.00410	0.00179	0.04427	0.49441	-0.00559	0.00126	0.11580
		LOD	2.34832	1.34832	1.85286	0.23971	0.35676	-0.64324	0.41568	0.05134	1.48308	0.98308	0.97023	0.21423
	75%	Our	0.97255	-0.02745	0.04708	0.20590	1.00591	0.00591	0.00279	0.05040	0.49534	-0.00466	0.00196	0.04186
		CC	0.95495	-0.04505	0.07417	0.26939	1.00994	0.00994	0.00433	0.06527	0.49186	-0.00814	0.00226	0.14965
		LOD	2.68415	1.68415	2.87753	0.28082	0.29718	-0.70282	0.49548	0.05497	1.42716	0.92716	0.86709	0.27739

Table 4: (Continued)

4 Ethical thinking, societal relevance, and stakeholder awareness

Left-censoring occurs more often than acknowledged by many companies and institutes. Firstly it is important that all companies are made aware of the concepts and consequences of left-censored data. Because this type of data is often neglected. Secondly all the companies who encounter such left-censored data and want to use it in a regression context should be made aware of the proposed method in this paper. Until now left censored covariates are often ignored or dealt with by using methods like a complete case analysis or imputation. These methods are often used because of their simplicity. What is often disregarded is that these methods either produce biased or highly inefficient estimates. These estimation issues regarding the effect of a covariate on a health related response can all cause ethical concerns. An example where these ethical issues may occur concerns the assessment of the impact of the concentration of heavy metals or chemical compounds in certain areas on human health. Take for example the health concerns regarding the increased concentration of perfluorooctane sulfonate (PFOS) in the area of the 3M fluorochemical plant in Zwijndrecht, Belgium. (Groffen et al., 2021) The measurements of the concentrations are often left-censored. If the left-censored concentrations are ignored or imputed with larger values, the areas with a low concentration are not considered in the estimation of the effect of the compound on the health related response. This causes an overestimation of the risk which may result in unnecessary financial investments and inconvenience for the population. Individuals or organizations who may gain political or economical profit may also abuse this overestimation of the risk. Lastly, it has to be noted that it is not ethical to base healthcare decisions on estimates with high uncertainty. Being aware of the left-censoring in covariates and deciding to use these methods due to their simplicity while more efficient and unbiased methods are available is therefore unethical. Although the method proposed in this paper is more complex to apply, it does produce unbiased and efficient estimates in most situations. While this method is better than using a complete case analysis or the substitution method, it still has its own drawbacks that have to be acknowledged. It should not be thought of as a magical tool which removes all negative consequences of censoring. There will always be some uncertainty left in the results, especially with large amounts of censoring. As mentioned in the introduction, this method is a non-parametric version of an already developed parametric method. When detailed knowledge about the distribution of X is available it is important to acknowledge that the method in this paper will probably produce less efficient estimates than the already existing parametric method.

5 Conclusion

In this paper a non-parametric maximum likelihood based method was developed to asses the effect of one non-negative left-censored covariate on a non-negative left-censored response. The parametric assumption on the covariate in the MLE method introduced by Tran et al. is in this paper lifted by estimating the cumulative density function of X in the likelihood by use of the non-parametric Kaplan-Meier estimator. In this paper the regression relation between Y and X was described by a parametric Weibull accelerated failure time model. The parameters from this regression model can be estimated by the proposed model in this paper.

With the help of simulations studies it was shown that the proposed method delivered parameter estimates with a sufficiently low bias and standard error in all scenarios with probability of censoring in both X and Y varying form low to high. It was also seen that the method performed well for both small (100) and large (500) sample sizes. The proposed method was compared to a complete case analysis and a single value substitution method. From this comparison it was concluded that the method proposed in this paper outperformed the other methods. Lastly, it was shown that the parameter estimates of the accelerated failure time model seem to follow a normal distribution with mean equal to the real parameter value and standard deviation equal to the standard error of the parameter. When increasing the censoring probability in X there seems to be signs of overdispersion for the regression coefficients. The standard deviation from the parameters are underestimated in these cases.

6 Ideas for future research

The method that was described in this paper provides a strong basis to handle both a left-censored covariate and response in a regression model. There are still some aspects of the method that can be expanded.

In this paper the covariate and the response are restricted to be non-negative. The theory can thus be extended to variables on the whole real line, which is a fairly easy thing to do. The method is also restricted to only one left-censored covariate. Because it is often of interest to asses the effect of multiple covariates on a response an expansion to multiple covariates, censored or not, may be of interest for future research. For this purpose matrix and vector notation can be introduced. To consider both non-censored and left censored covariates, the model can be split into 2 parts, a part for the fully observed and a part for the censored covariates.

What can also be of interest for future research is to investigate the asymptotic properties of the maximum likelihood estimator. As was shown in the simulation study the parameters seem to follow a normal distribution. This can be theoretically proven. Once these properties are known and proven they can be used to form exact confidence intervals and perform hypothesis tests for the parameters.

In the proposed methodology the conditional distribution of Y given X is described by a Weibull accelerated failure time model. An expansion of the methodology may be to allow other parametric distributions such as the log-normal. Another expansion related to this may be to consider a semi-parametric model in stead of the parametric AFT model.

In this paper the proposed model was compared to the complete case analysis and substitution method. Here it was shown that the proposed method outperformed the others. Several other methods to handle the situation of a left-censored covariate and response are published in the literature. It can thus be of interest to investigate all available methods and compare those to the proposed method.

A last part to considered for future research is to improve the efficiency of the R-code. Due to time constraints the main focus was to implement the method in R such that it produced correct results. For future research it is also important to ensure that the code works as fast and efficient as possible.

References

Apostol, T. M. (1974). *Mathematical analysis* (2nd ed.). Reading, Mass: Addison-Wesley.

Groffen, T., Bervoets, L., Jeong, Y., Willems, T., Eens, M., & Prinsen, E. (2021). A rapid method for the detection and quantification of legacy and emerging per- and polyfluoroalkyl substances (pfas) in bird feathers using uplc-ms/ms. *Journal of Chromatography B*, *1172*, 122653. Retrieved from https://www.sciencedirect.com/science/article/pii/S1570023221001331 doi: https://doi.org/10.1016/j.jchromb.2021.122653

Helsel, D. R. (2011). *Statistics for censored environmental data using minitab and r* (Vol. 77). John Wiley & Sons.

Huynh, T., Ramachandran, G., Banerjee, S., Monteiro, J., Stenzel, M., Sandler, D. P., ... Stewart, P. A. (2014, 09). Comparison of Methods for Analyzing Left-Censored Occupational Exposure Data. *The Annals of Occupational Hygiene*, *58*(9), 1126-1142. Retrieved from https://doi.org/10.1093/annhyg/meu067 doi: 10.1093/annhyg/meu067

Liu, E. (2018). Using weibull accelerated failure time regression model to predict survival time and life expectancy. *bioRxiv*. doi: 10.1101/362186

Posit team. (2023). Rstudio: Integrated development environment for r [Computer software manual]. Boston, MA. Retrieved from http://www.posit.co/

R Core Team. (2023). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from https://www.R-project.org/

Ramos, P. L., Guzman, D. C., Mota, A. L., Rodrigues, F. A., & Louzada, F. (2020). Sampling with censored data: a practical guide. *arXiv preprint arXiv:2011.08417*.

Sattar, A., Sinha, S. K., & Morris, N. J. (2012). A parametric survival model when a covariate is subject to left-censoring. *Journal of biometrics & biostatistics*(2). doi: 10.4172/2155-6180.S3-002

She, N. (1997). Analyzing censored water quality data using a non-parametric approach1. *JAWRA Journal of the American Water Resources Association*, *33*(3), 615-624. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1752-1688.1997.tb03536.x doi: https://doi.org/10.1111/j.1752-1688.1997.tb03536.x

Signorell, A. (2023). Desctools: Tools for descriptive statistics [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=DescTools (R package version 0.99.49)

Therneau, T. M. (2023). A package for survival analysis in r [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=survival (R package version 3.5-5)

Tran, T. M. P., Abrams, S., Aerts, M., Maertens, K., & Hens, N. (2021). Measuring association among censored antibody titer data. *Statistics in Medicine*, *40*(16), 3740-3761. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.8995 doi: https://doi.org/10.1002/sim.8995

Zaffora, B., Magistris, M., Chevalier, J.-P., Luccioni, C., Saporta, G., & Ulrici, L. (2017). A new approach to characterize very-low-level radioactive waste produced at hadron accelerators. *Applied Radiation and Isotopes*, *122*, 141-147. Retrieved from https://www.sciencedirect.com/science/article/pii/S096980431630906X doi: https://doi.org/10.1016/j.apradiso.2017.01.019

A Figures

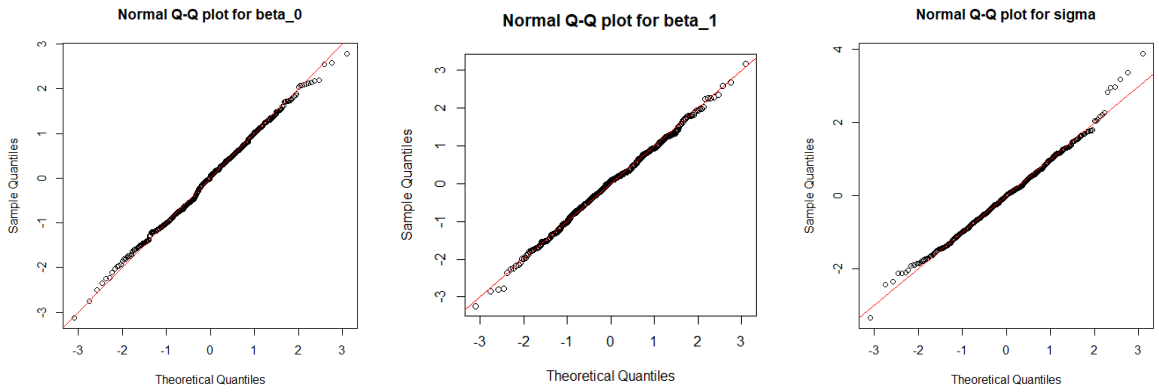

Figure 7: Normal Q-Q plots for β_0, β_1 and σ with the red line representing the line $y = x$ with $n = 500$, $nsim = 500$, probability of censoring in $X = 10\%$ and probability of censoring in $Y = 25\%$

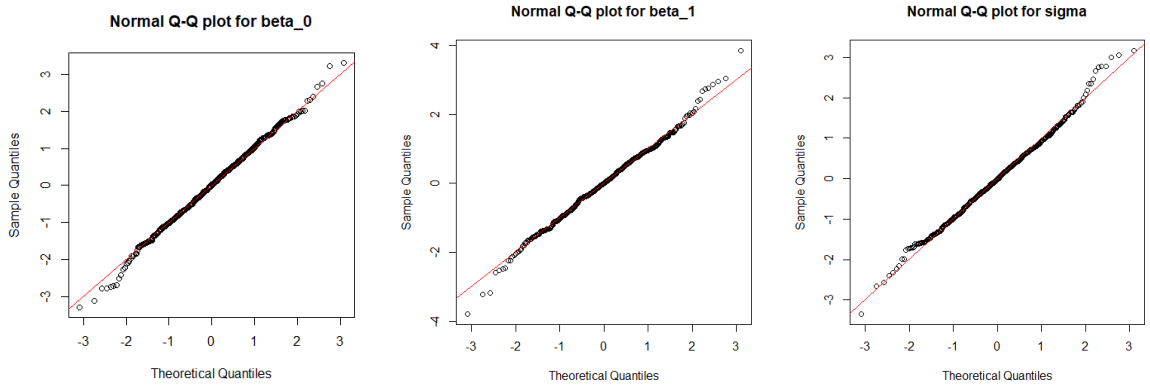

Figure 8: Normal Q-Q plots for β_0, β_1 and σ with the red line representing the line $y = x$ with $n = 500$, $nsim = 500$, probability of censoring in $X = 10\%$ and probability of censoring in $Y = 50\%$

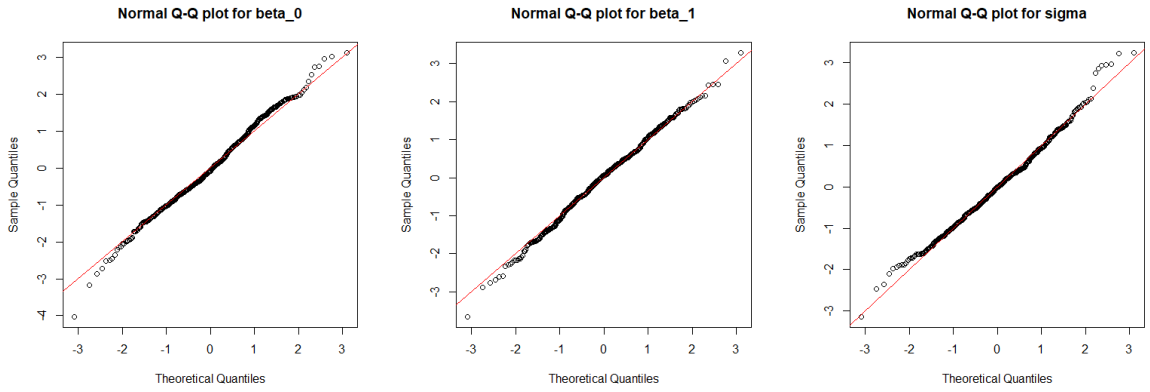

Figure 9: Normal Q-Q plots for β_0, β_1 and σ with the red line representing the line $y = x$ with $n = 500$, $nsim = 500$, probability of censoring in $X = 25\%$ and probability of censoring in $Y = 10\%$

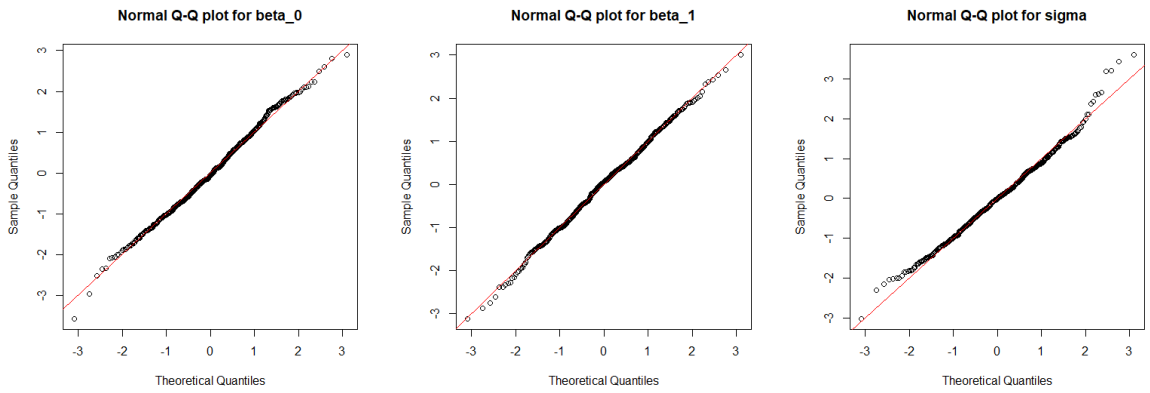

Figure 10: Normal Q-Q plots for β_0, β_1 and σ with the red line representing the line $y = x$ with $n = 500$, $nsim = 500$, probability of censoring in $X = 25\%$ and probability of censoring in $Y = 25\%$

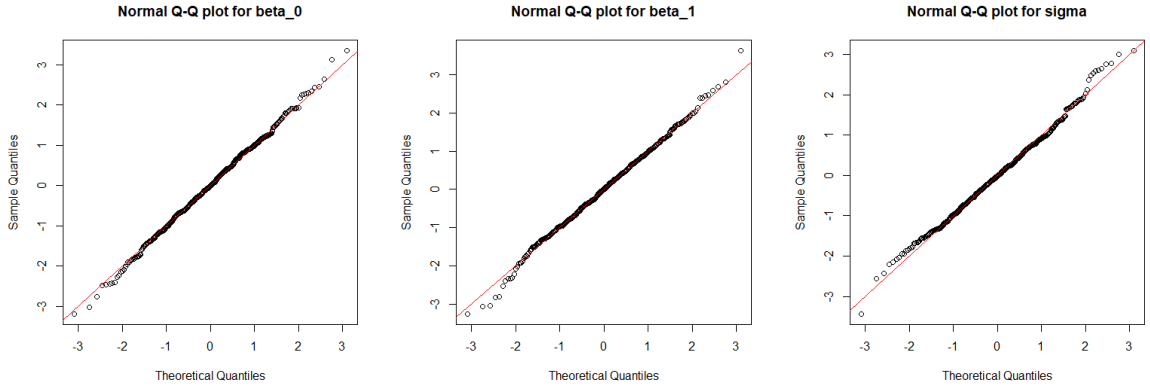

Figure 11: Normal Q-Q plots for β_0, β_1 and σ with the red line representing the line $y = x$ with $n = 500$, $nsim = 500$, probability of censoring in $X = 25\%$ and probability of censoring in $Y = 50\%$

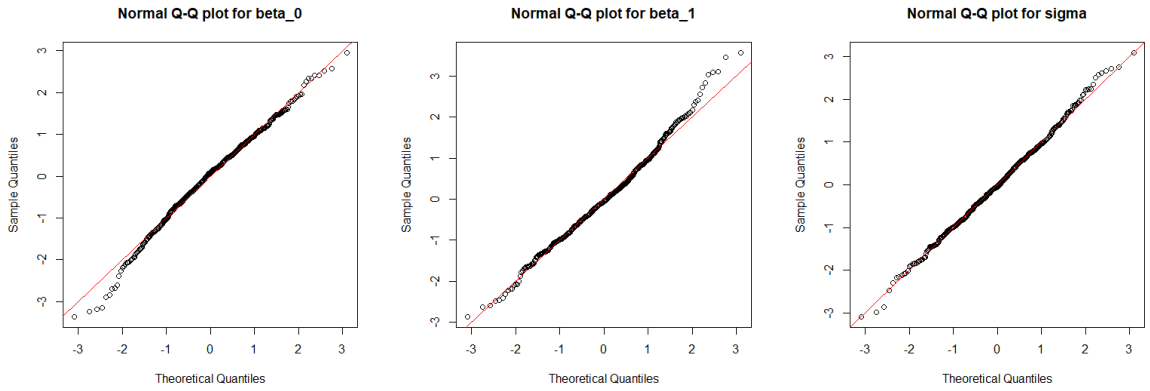

Figure 12: Normal Q-Q plots for β_0, β_1 and σ with the red line representing the line $y = x$ with $n = 500$, $nsim = 500$, probability of censoring in $X = 25\%$ and probability of censoring in $Y = 75\%$

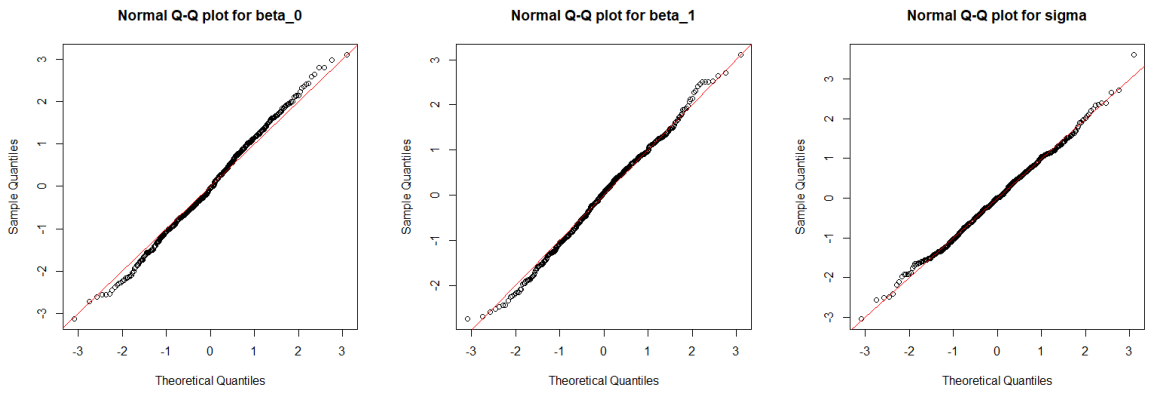

Figure 13: Normal Q-Q plots for β_0, β_1 and σ with the red line representing the line $y = x$ with $n = 500$, $nsim = 500$, probability of censoring in $X = 50\%$ and probability of censoring in $Y = 25\%$

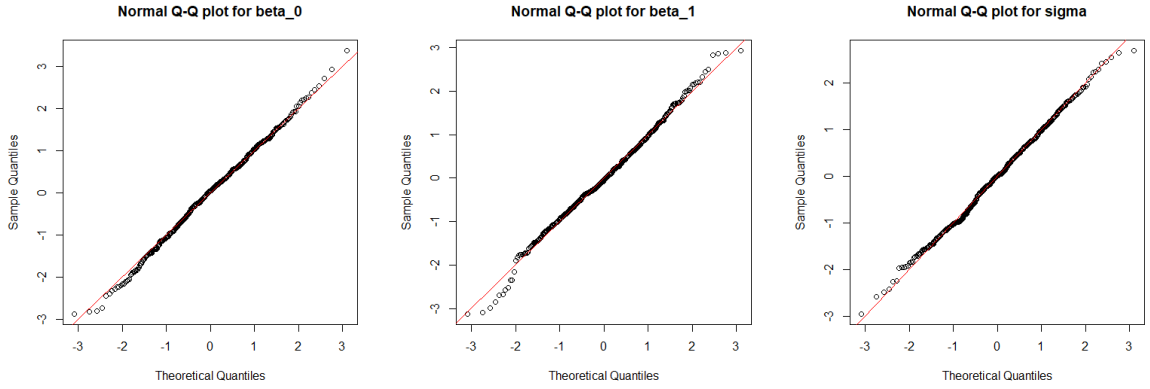

Figure 14: Normal Q-Q plots for β_0, β_1 and σ with the red line representing the line $y = x$ with $n = 500$, $nsim = 500$, probability of censoring in $X = 50\%$ and probability of censoring in $Y = 50\%$

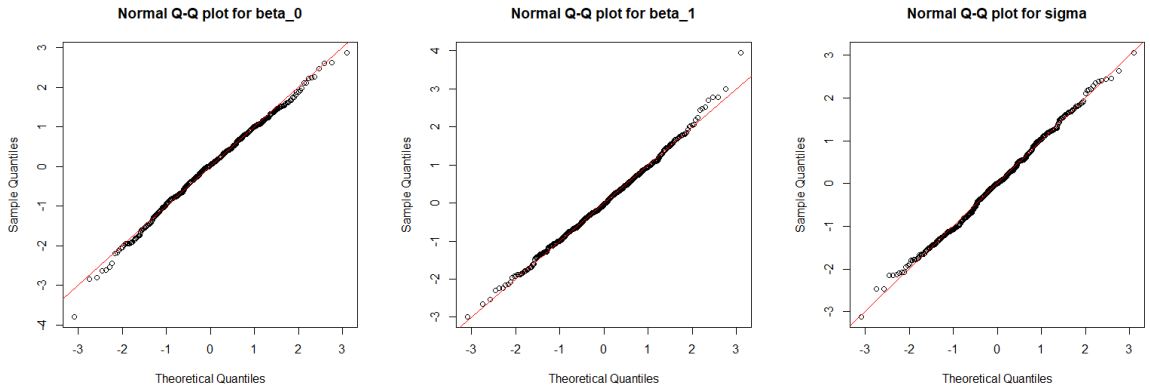

Figure 15: Normal Q-Q plots for β_0, β_1 and σ with the red line representing the line $y = x$ with $n = 500$, $nsim = 500$, probability of censoring in $X = 50\%$ and probability of censoring in $Y = 75\%$

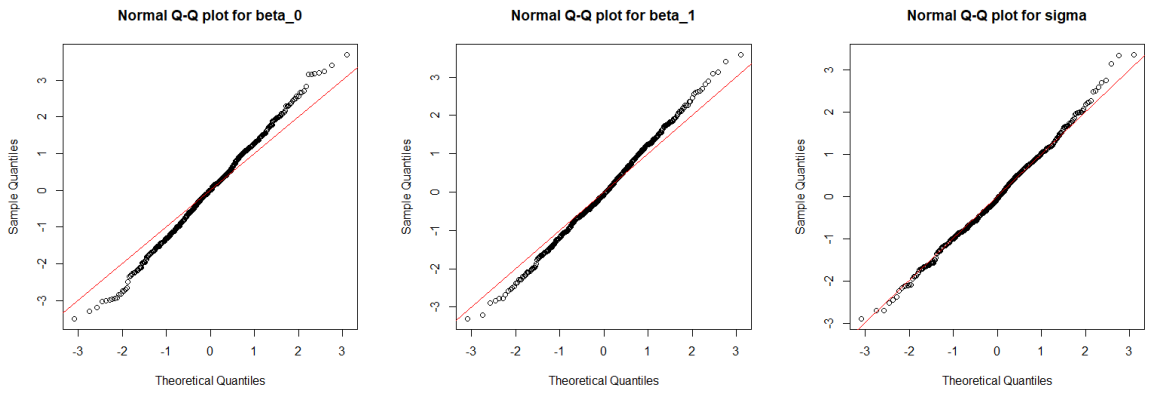

Figure 16: Normal Q-Q plots for β_0, β_1 and σ with the red line representing the line $y = x$ with $n = 500$, $nsim = 500$, probability of censoring in $X = 75\%$ and probability of censoring in $Y = 25\%$

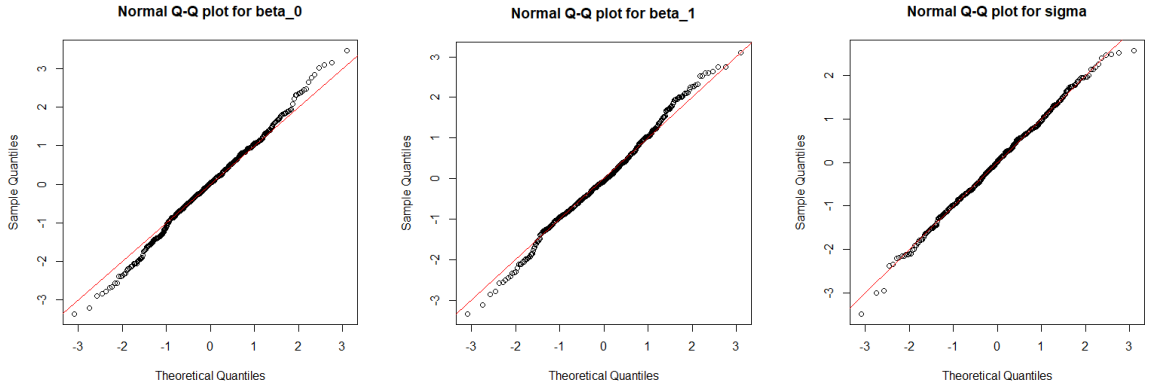

Figure 17: Normal Q-Q plots for β_0, β_1 and σ with the red line representing the line $y = x$ with $n = 500$, $nsim = 500$, probability of censoring in $X = 75\%$ and probability of censoring in $Y = 50\%$

B Tables

		$\beta_0 = 1$				$\beta_1 = 1$				σ			
Censoring in X	**Censoring in Y**	**Estimate**	**Bias**	**MSE**	**SE**	**Estimate**	**Bias**	**MSE**	**SE**	**Estimate**	**Bias**	**MSE**	**SE**
10%	10%	0.99126	-0.00874	0.01345	0.10728	1.00194	0.00194	0.00143	0.03615	0.49426	-0.00574	0.00172	0.04076
	25%	0.99226	-0.00774	0.01787	0.12473	1.00189	0.00189	0.00172	0.04009	0.49170	-0.00830	0.00198	0.04349
	50%	0.98691	-0.01309	0.04097	0.19579	1.00295	0.00295	0.00328	0.05572	0.48921	-0.01079	0.00286	0.05283
	75%	0.94369	-0.05631	0.14561	0.34623	1.01294	0.01294	0.00929	0.08718	0.47926	-0.02074	0.00527	0.07205
25%	10%	0.98211	-0.01789	0.02230	0.12285	1.00399	0.00399	0.00212	0.04008	0.49779	-0.00221	0.00190	0.04331
	25%	0.99026	-0.00974	0.02506	0.13974	1.00199	0.00199	0.00225	0.04385	0.49406	-0.00594	0.00210	0.04556
	50%	0.99062	-0.00938	0.04914	0.21035	1.00162	0.00162	0.00385	0.05936	0.49069	-0.00931	0.00291	0.05384
	75%	0.95360	-0.04640	0.15590	0.36324	1.01034	0.01034	0.00997	0.09115	0.47998	-0.02002	0.00525	0.07272
50%	10%	0.97141	-0.02859	0.04514	0.14834	1.00568	0.00568	0.00365	0.04599	0.50769	0.00769	0.00293	0.05183
	25%	0.98736	-0.01264	0.04751	0.16835	1.00205	0.00205	0.00380	0.05021	0.50097	0.00097	0.00300	0.05418
	50%	0.99601	-0.00399	0.07834	0.25120	1.00016	0.00016	0.00576	0.06833	0.49298	-0.00702	0.00373	0.06154
	75%	0.95493	-0.04507	0.23296	0.42605	1.00965	0.00965	0.01401	0.10474	0.47931	-0.02069	0.00628	0.07958
75%	10%	0.95039	-0.04961	0.07506	0.16733	1.00936	0.00936	0.00539	0.05181	0.52438	0.02438	0.00548	0.06225
	25%	0.97046	-0.02954	0.07807	0.18801	1.00513	0.00513	0.00556	0.05585	0.51385	0.01385	0.00514	0.06556
	50%	0.97496	-0.02504	0.11916	0.28290	1.00454	0.00454	0.00793	0.07620	0.50000	0.00000	0.00581	0.07421
	75%	0.90420	-0.09580	0.35082	0.47447	1.02116	0.02116	0.02011	0.11594	0.47978	0.02022	0.00951	0.09454

Table 5: Parameter estimates and goodness of fit measures for $n = 100$ and $nsim = 500$

		β_0				β_1				σ			
Censoring in X	**Censoring in Y**	**Estimate**	**95% CI**		**CP**	**Estimate**	**95% CI**		**CP**	**Estimate**	**95% CI**		**CP**
			Lower	**Upper**			**Lower**	**Upper**			**Lower**	**Upper**	
10%	10%	0.99126	0.78100	1.20153	0.932	1.00194	0.93109	1.07279	0.948	0.49426	0.41436	0.57416	0.950
	25%	0.99226	0.74778	1.23674	0.934	1.00189	0.92331	1.08047	0.942	0.49170	0.40647	0.57694	0.946
	50%	0.98691	0.60316	1.37065	0.946	1.00295	0.89373	1.11217	0.950	0.48921	0.38567	0.59275	0.944
	75%	0.94369	0.26508	1.62230	0.936	1.01294	0.84207	1.18381	0.946	0.47926	0.33803	0.62049	0.952
25%	10%	0.98211	0.74133	1.22290	0.890	1.00399	0.92543	1.08254	0.906	0.49779	0.41291	0.58268	0.940
	25%	0.99026	0.71637	1.26415	0.920	1.00199	0.91605	1.08793	0.926	0.49406	0.40477	0.58335	0.944
	50%	0.99062	0.57834	1.40290	0.942	1.00162	0.88528	1.11795	0.948	0.49069	0.38516	0.59621	0.944
	75%	0.95360	0.24165	1.66554	0.940	1.01034	0.83169	1.18899	0.942	0.47998	0.33745	0.62250	0.956
50%	10%	0.97141	0.68067	1.26215	0.830	1.00568	0.91555	1.09581	0.864	0.50769	0.40609	0.60928	0.940
	25%	0.98736	0.65739	1.31733	0.870	1.00205	0.90365	1.10046	0.882	0.50097	0.39477	0.60717	0.944
	50%	0.99601	0.50367	1.48836	0.918	1.00016	0.86623	1.13409	0.922	0.49298	0.37237	0.61359	0.942
	75%	0.95493	0.11986	1.78999	0.930	1.00965	0.80436	1.21495	0.930	0.47931	0.32333	0.63528	0.952
75%	10%	0.95039	0.62243	1.27835	0.786	1.00936	0.90782	1.11091	0.822	0.52438	0.40236	0.64639	0.916
	25%	0.97046	0.60197	1.33896	0.816	1.00513	0.89567	1.11459	0.854	0.51385	0.38536	0.64234	0.926
	50%	0.97496	0.42048	1.52944	0.878	1.00454	0.85519	1.15389	0.908	0.50000	0.35455	0.64545	0.946
	75%	0.90420	-0.02576	1.83417	0.904	1.02116	0.79392	1.24840	0.906	0.47978	0.29448	0.66507	0.938

Table 6: Confidence intervals and Coverage probabilities for $n = 100$ and $nsim = 500$

		$\beta_0 = 1$				$\beta_1 = 1$				σ			
Censoring in X	**Censoring in Y**	**Estimate**	**Bias**	**MSE**	**SE**	**Estimate**	**Bias**	**MSE**	**SE**	**Estimate**	**Bias**	**MSE**	**SE**
10%	10%	0.99822	-0.00178	0.00570	0.07544	0.99951	-0.00049	0.00060	0.02536	0.49550	-0.00450	0.00092	0.02886
	25%	0.99738	-0.00262	0.00725	0.08790	0.99977	-0.00023	0.00071	0.02818	0.49464	-0.00536	0.00110	0.03089
	50%	0.99200	-0.00800	0.01907	0.13729	1.00113	0.00113	0.00147	0.03902	0.49251	-0.00749	0.00158	0.03731
	75%	0.95697	-0.04303	0.06452	0.23553	1.00895	0.00895	0.00399	0.05948	0.49023	-0.00977	0.00282	0.05109
25%	10%	0.99568	-0.00432	0.00861	0.08643	0.99995	-0.00005	0.00080	0.02807	0.49747	-0.00253	0.00100	0.03061
	25%	0.99592	-0.00408	0.01056	0.09850	1.00003	0.00003	0.00094	0.03081	0.49581	-0.00419	0.00115	0.03225
	50%	0.99078	-0.00922	0.02195	0.14753	1.00135	0.00135	0.00166	0.04159	0.49277	-0.00723	0.00158	0.03791
	75%	0.95780	-0.04220	0.07235	0.24984	1.00866	0.00866	0.00445	0.06288	0.49042	-0.00958	0.00283	0.05150
50%	10%	0.98555	-0.01445	0.01939	0.10356	1.00235	0.00235	0.00150	0.03189	0.50233	0.00233	0.00145	0.03643
	25%	0.99005	-0.00995	0.02277	0.11825	1.00146	0.00146	0.00172	0.03509	0.49898	-0.00102	0.00159	0.03816
	50%	0.98434	-0.01566	0.03683	0.17552	1.00306	0.00306	0.00254	0.04770	0.49344	-0.00656	0.00201	0.04322
	75%	0.94928	-0.05072	0.10512	0.29542	1.01105	0.01105	0.00616	0.07284	0.48885	-0.01115	0.00340	0.05613
75%	10%	0.98160	-0.01840	0.03489	0.11579	1.00226	0.00226	0.00257	0.03552	0.51253	0.01253	0.00235	0.04370
	25%	0.99035	-0.00965	0.03639	0.13249	1.00046	0.00046	0.00270	0.03899	0.50706	0.00706	0.00245	0.04608
	50%	0.98241	-0.01759	0.05564	0.19743	1.00322	0.00322	0.00376	0.05300	0.49540	-0.00460	0.00289	0.05168
	75%	0.94717	-0.05283	0.15318	0.33095	1.01154	0.01154	0.00880	0.08092	0.48705	-0.01295	0.00486	0.06607

Table 7: Parameter estimates and goodness of fit measures for $n = 200$ and $nsim = 500$

		β_0				β_1				σ			
Censoring in X	**Censoring in Y**	**Estimate**	**95% CI**		**CP**	**Estimate**	**95% CI**		**CP**	**Estimate**	**95% CI**		**CP**
			Lower	**Upper**			**Lower**	**Upper**			**Lower**	**Upper**	
10%	10%	0.99822	0.85035	1.14609	0.948	0.99951	0.94979	1.04922	0.960	0.49550	0.43893	0.55207	0.934
	25%	0.99738	0.82509	1.16967	0.962	0.9997713	0.94455	1.05500	0.958	0.49464	0.43409	0.55518	0.924
	50%	0.99200	0.72292	1.26108	0.946	1.0011296	0.92465	1.07761	0.944	0.49251	0.41938	0.56563	0.930
	75%	0.95697	0.49533	1.41861	0.950	1.0089541	0.89238	1.12553	0.944	0.49023	0.39009	0.59036	0.944
25%	10%	0.99568	0.82627	1.16509	0.942	0.9999529	0.94494	1.05496	0.942	0.49747	0.43748	0.55746	0.936
	25%	0.99592	0.80287	1.18898	0.936	1.0000330	0.93965	1.06042	0.940	0.49581	0.43259	0.55902	0.938
	50%	0.99078	0.70163	1.27993	0.936	1.0013527	0.91985	1.08286	0.948	0.49277	0.41846	0.56708	0.940
	75%	0.95780	0.46812	1.44749	0.934	1.0086606	0.88542	1.13190	0.932	0.49042	0.38948	0.59136	0.946
50%	10%	0.98555	0.78256	1.18854	0.866	1.0023500	0.93985	1.06485	0.894	0.50233	0.43092	0.57373	0.930
	25%	0.99005	0.75827	1.22182	0.892	1.0014642	0.93269	1.07024	0.908	0.49898	0.42419	0.57377	0.932
	50%	0.98434	0.64032	1.32836	0.926	1.0030619	0.90956	1.09656	0.932	0.49344	0.40872	0.57815	0.944
	75%	0.94928	0.37025	1.52831	0.934	1.0110512	0.86829	1.15382	0.946	0.48885	0.37883	0.59887	0.940
75%	10%	0.98160	0.75465	1.20855	0.774	1.0022637	0.93265	1.07188	0.840	0.51253	0.42688	0.59819	0.930
	25%	0.99035	0.73067	1.25004	0.850	1.0004580	0.92403	1.07688	0.862	0.50706	0.41673	0.59738	0.928
	50%	0.98241	0.59545	1.36936	0.896	1.0032157	0.89933	1.10710	0.912	0.49540	0.39410	0.59669	0.932
	75%	0.94717	0.29850	1.59583	0.918	1.0115402	0.85294	1.17014	0.920	0.48705	0.35755	0.61655	0.942

Table 8: Confidence intervals and Coverage probabilities for $n = 200$ and $nsim = 500$

		$\beta_0 = 1$				$\beta_1 = 1$				σ			
Censoring in X	**Censoring in Y**	**Estimate**	**Bias**	**MSE**	**SE**	**Estimate**	**Bias**	**MSE**	**SE**	**Estimate**	**Bias**	**MSE**	**SE**
10%	10%	1.00225	0.00225	0.00232	0.04796	0.99875	-0.00125	0.00026	0.01609	0.49860	-0.00140	0.00029	0.01836
	25%	0.99875	-0.00125	0.00283	0.05576	0.99966	-0.00034	0.00028	0.01787	0.49889	-0.00111	0.00035	0.01967
	50%	0.98453	-0.01547	0.00845	0.08716	1.00323	0.00323	0.00063	0.02477	0.49859	-0.00141	0.00051	0.02385
	75%	0.97256	0.02744	0.02241	0.14657	1.00550	0.00550	0.00143	0.03707	0.49663	-0.00337	0.00112	0.03246
25%	10%	1.00393	0.00393	0.00370	0.05501	0.99827	-0.00173	0.00035	0.01780	0.49873	-0.00127	0.00030	0.01941
	25%	1.00053	0.00053	0.00404	0.06259	0.99917	-0.00083	0.00036	0.01954	0.49886	-0.00114	0.00035	0.02049
	50%	0.98577	0.01423	0.01029	0.09375	1.00283	0.00283	0.00075	0.02641	0.49898	-0.00102	0.00051	0.02423
	75%	0.97138	0.02862	0.02560	0.15517	1.00570	0.00570	0.00164	0.03911	0.49684	-0.00316	0.00113	0.03271
50%	10%	1.00146	0.00146	0.00611	0.06611	0.99875	-0.00125	0.00052	0.02022	0.50073	0.00073	0.00052	0.02304
	25%	1.00018	0.00018	0.00659	0.07534	0.99920	-0.00080	0.00053	0.02226	0.49957	-0.00043	0.00056	0.02414
	50%	0.98796	-0.01204	0.01354	0.11136	1.00234	0.00234	0.00095	0.03021	0.49814	-0.00186	0.00070	0.02747
	75%	0.97346	-0.02654	0.03059	0.18303	1.00512	0.00512	0.00191	0.04518	0.49597	-0.00403	0.00131	0.03563
75%	10%	1.00232	0.00232	0.01053	0.07355	0.99818	-0.00182	0.00082	0.02233	0.50419	0.00419	0.00083	0.02746
	25%	1.00162	0.00162	0.01122	0.08430	0.99854	-0.00146	0.00083	0.02462	0.50224	0.00224	0.00089	0.02892
	50%	0.98577	-0.01423	0.02015	0.12468	1.00282	0.00282	0.00131	0.03338	0.49861	-0.00139	0.00103	0.03266
	75%	0.95893	-0.04107	0.04649	0.20719	1.00825	0.00825	0.00265	0.05064	0.49586	-0.00414	0.00198	0.04177

Table 9: Parameter estimates and goodness of fit measures for $n = 500$ and $nsim = 200$

		β_0				β_1				σ			
Censoring in X	**Censoring in Y**	**Estimate**	**95% CI**		**CP**	**Estimate**	**95% CI**		**CP**	**Estimate**	**95% CI**		**CP**
			Lower	**Upper**			**Lower**	**Upper**			**Lower**	**Upper**	
10%	10%	1.00225	0.90825	1.09625	0.960	0.99875	0.96721	1.03030	0.955	0.49860	0.46263	0.53458	0.970
	25%	0.99875	0.88946	1.10804	0.970	0.99966	0.96464	1.03467	0.970	0.49889	0.46033	0.53745	0.975
	50%	0.98453	0.81369	1.15537	0.945	1.00323	0.95467	1.05179	0.935	0.49859	0.45185	0.54532	0.965
	75%	0.97256	0.68528	1.25984	0.965	1.00550	0.93285	1.07816	0.945	0.49663	0.43301	0.56025	0.935
25%	10%	1.00393	0.89611	1.11174	0.940	0.99827	0.96338	1.03316	0.940	0.49873	0.46068	0.53678	0.975
	25%	1.00053	0.87785	1.12320	0.960	0.99917	0.96087	1.03747	0.960	0.49886	0.45869	0.53902	0.975
	50%	0.98577	0.80201	1.16952	0.935	1.00283	0.95106	1.05460	0.935	0.49898	0.45149	0.54647	0.975
	75%	0.97138	0.66724	1.27552	0.955	1.00570	0.92904	1.08236	0.950	0.49684	0.43273	0.56095	0.935
50%	10%	1.00146	0.87188	1.13105	0.895	0.99875	0.95912	1.03838	0.915	0.50073	0.45557	0.54590	0.965
	25%	1.00018	0.85251	1.14785	0.935	0.99920	0.95557	1.04282	0.935	0.49957	0.45224	0.54689	0.960
	50%	0.98796	0.76969	1.20624	0.930	1.00234	0.94312	1.06156	0.940	0.49814	0.44430	0.55197	0.980
	75%	0.97346	0.61472	1.33219	0.965	1.00512	0.91656	1.09367	0.955	0.49597	0.42614	0.56580	0.960
75%	10%	1.00232	0.85815	1.14649	0.835	0.99818	0.95442	1.04194	0.870	0.50419	0.45037	0.55801	0.925
	25%	1.00162	0.83639	1.16685	0.870	0.99854	0.95029	1.04680	0.885	0.50224	0.44556	0.55891	0.935
	50%	0.98577	0.74141	1.23014	0.895	1.00282	0.93739	1.06825	0.890	0.49861	0.43461	0.56262	0.940
	75%	0.95893	0.55283	1.36502	0.950	1.00825	0.90899	1.10751	0.960	0.49586	0.41398	0.57773	0.945

Table 10: Confidence intervals and Coverage probabilities for $n = 500$ and $nsim = 200$

		$\beta_0 = 1$				$\beta_1 = 1$				σ			
Censoring in X	**Censoring in Y**	**Estimate**	**Bias**	**MSE**	**SE**	**Estimate**	**Bias**	**MSE**	**SE**	**Estimate**	**Bias**	**MSE**	**SE**
10%	10%	1.00953	0.00953	0.00263	0.04789	0.99577	-0.00423	0.00030	0.01609	0.49820	-0.00180	0.00030	0.01832
	25%	1.00404	0.00404	0.00317	0.05574	0.99716	-0.00284	0.00031	0.01788	0.49920	-0.00080	0.00037	0.01969
	50%	0.99386	-0.00614	0.00958	0.08675	0.99980	-0.00020	0.00068	0.02466	0.49752	-0.00248	0.00046	0.02382
	75%	0.98814	-0.01186	0.02213	0.14545	1.00075	0.00075	0.00139	0.03679	0.49572	-0.00428	0.00102	0.03252
25%	10%	1.01276	0.01276	0.00372	0.05497	0.99478	-0.00522	0.00037	0.01779	0.49869	-0.00131	0.00029	0.01939
	25%	1.00620	0.00620	0.00432	0.06260	0.99652	-0.00348	0.00039	0.01955	0.49935	-0.00065	0.00035	0.02052
	50%	0.99156	-0.00844	0.01115	0.09342	1.00032	0.00032	0.00078	0.02632	0.49800	-0.00200	0.00047	0.02420
	75%	0.98493	-0.01507	0.02472	0.15385	1.00146	0.00146	0.00156	0.03878	0.49596	-0.00404	0.00104	0.03277
50%	10%	1.00956	0.00956	0.00502	0.06615	0.99565	-0.00435	0.00047	0.02025	0.50157	0.00157	0.00049	0.02309
	25%	1.00484	0.00484	0.00586	0.07545	0.99695	-0.00305	0.00048	0.02232	0.50124	0.00124	0.00056	0.02426
	50%	0.99443	-0.00557	0.01225	0.11109	0.99969	-0.00031	0.00084	0.03016	0.49878	-0.00122	0.00064	0.02757
	75%	0.99087	-0.00913	0.02699	0.18073	1.00005	0.00005	0.00169	0.04463	0.49581	-0.00419	0.00123	0.03580
75%	10%	1.01142	0.01142	0.01090	0.07352	0.99523	-0.00477	0.00087	0.02231	0.50284	0.00284	0.00062	0.02745
	25%	0.97985	-0.02015	0.04215	0.13166	1.00536	0.00536	0.00276	0.03890	0.50640	0.00640	0.00290	0.04622
	50%	0.99021	-0.00979	0.01874	0.12471	1.00096	0.00096	0.00121	0.03340	0.49809	-0.00191	0.00080	0.03275
	75%	0.97667	-0.02333	0.03949	0.20480	1.00357	0.00357	0.00235	0.05008	0.49504	-0.00496	0.00171	0.04196

Table 11: Parameter estimates and goodness of fit measures for $n = 500$ and $nsim = 100$

		β_0				β_1				σ			
Censoring in X	**Censoring in Y**	**Estimate**	**95% CI**		**CP**	**Estimate**	**95% CI**		**CP**	**Estimate**	**95% CI**		**CP**
			Lower	**Upper**			**Lower**	**Upper**			**Lower**	**Upper**	
10%	10%	1.00953	0.91567	1.10339	0.950	0.99577	0.96424	1.02730	0.950	0.49820	0.46229	0.53411	0.950
	25%	1.00404	0.89478	1.11330	0.970	0.99716	0.96211	1.03220	0.960	0.49920	0.46061	0.53780	0.960
	50%	0.99386	0.82382	1.16390	0.940	0.99980	0.95146	1.04813	0.920	0.49752	0.45084	0.54421	0.990
	75%	0.98814	0.70307	1.27322	0.980	1.00075	0.92864	1.07287	0.960	0.49572	0.43199	0.55946	0.950
25%	10%	1.01276	0.90503	1.12050	0.940	0.99478	0.95991	1.02965	0.940	0.49869	0.46068	0.53670	0.970
	25%	1.00620	0.88351	1.12890	0.970	0.99652	0.95819	1.03484	0.960	0.49935	0.45913	0.53957	0.970
	50%	0.99156	0.80846	1.17466	0.950	1.00032	0.94874	1.05191	0.940	0.49800	0.45057	0.54543	0.990
	75%	0.98493	0.68339	1.28648	0.950	1.00146	0.92546	1.07747	0.940	0.49596	0.43172	0.56019	0.960
50%	10%	1.00956	0.87990	1.13921	0.920	0.99565	0.95596	1.03535	0.890	0.50157	0.45632	0.54682	0.960
	25%	1.00484	0.85696	1.15272	0.960	0.99695	0.95320	1.04070	0.950	0.50124	0.45370	0.54879	0.960
	50%	0.99443	0.77669	1.21218	0.950	0.99969	0.94056	1.05881	0.940	0.49878	0.44475	0.55282	0.990
	75%	0.99087	0.63664	1.34510	0.960	1.00005	0.91257	1.08753	0.960	0.49581	0.42564	5.65986	0.980
75%	10%	1.01142	0.86732	1.15553	0.810	0.99523	0.95150	1.03896	0.860	0.50284	0.44904	0.55665	0.950
	25%	0.97985	0.72179	1.23791	0.840	1.00536	0.92911	1.08161	0.850	0.50640	0.41580	0.59699	0.890
	50%	0.99021	0.74579	1.23463	0.910	1.00096	0.93551	1.06641	0.880	0.49809	0.43391	0.56227	0.970
	75%	0.97667	0.57527	1.37807	0.970	1.00357	0.90541	1.10173	0.960	0.49504	0.41280	0.57728	0.980

Table 12: Confidence intervals and Coverage probabilities for $n = 500$ and $nsim = 100$

			$\beta_0 = 1$				$\beta_1 = 1$				σ			
Censoring in X	**Censoring in Y**	**Method**	**Estimate**	**Bias**	**MSE**	**SE**	**Estimate**	**Bias**	**MSE**	**SE**	**Estimate**	**Bias**	**MSE**	**SE**
10%	10%	Our	0.99126	0.00874	0.01345	0.10728	1.00194	0.00194	0.00143	0.03615	0.49426	0.00574	0.00172	0.04076
		CC	0.99585	0.00415	0.01817	0.12459	1.00077	-0.00077	0.00182	0.04044	0.49213	0.00787	0.00184	0.13865
		LOD	0.91494	0.08506	0.02184	0.11102	1.02224	-0.02224	0.00203	0.03770	0.50279	-0.00279	0.00170	0.13530
	25%	Our	0.99226	0.00774	0.01787	0.12473	1.00189	0.00189	0.00172	0.04009	0.49170	0.00830	0.00198	0.04349
		CC	0.99615	0.00385	0.02136	0.13685	1.00078	-0.00078	0.00201	0.04316	0.49058	0.00942	0.00203	0.14660
		LOD	*	*	*	*	*	*	*	*	*	*	*	*
	50%	Our	0.98691	0.01309	0.04097	0.19579	1.00295	0.00295	0.00328	0.05572	0.48921	0.01079	0.00286	0.05283
		CC	0.98941	0.01059	0.04676	0.20621	1.00228	-0.00228	0.00370	0.05836	0.48867	0.01133	0.00288	0.17697
		LOD	0.95470	0.04530	0.04362	0.19727	1.01131	-0.01131	0.00344	0.05626	0.49135	0.00865	0.00285	0.17639
	75%	Our	0.94369	0.05631	0.14561	0.34623	1.01294	0.01294	0.00929	0.08718	0.47926	0.02074	0.00527	0.07205
		CC	0.94770	0.05230	0.15139	0.35909	1.01191	-0.01191	0.00962	0.09023	0.47895	0.02105	0.00529	0.24349
		LOD	0.92044	0.07956	0.14854	0.34677	1.01852	-0.01852	0.00947	0.08738	0.48039	0.01961	0.00526	0.24313
25%	10%	Our	0.98211	0.01789	0.02230	0.12285	1.00399	0.00399	0.00212	0.04008	0.49779	0.00221	0.00190	0.04331
		CC	0.98741	0.01259	0.02735	0.16130	1.00296	-0.00296	0.00247	0.04881	0.49065	0.00935	0.00207	0.15008
		LOD	0.65156	0.34844	0.14782	0.14935	1.07277	-0.07277	0.00762	0.04906	0.59419	-0.09419	0.01130	0.15034
	25%	Our	0.99026	0.00974	0.02506	0.13974	1.00199	0.00199	0.00225	0.04385	0.49406	0.00594	0.00210	0.04556
		CC	0.98825	0.01175	0.03249	0.17227	1.00282	-0.00282	0.00281	0.05134	0.48930	0.01070	0.00225	0.15547
		LOD	0.64485	0.35515	0.15796	0.16267	1.07695	-0.07695	0.00859	0.05191	0.57821	-0.07821	0.00873	0.15860
	50%	Our	0.99062	0.00938	0.04914	0.21035	1.00162	0.00162	0.00385	0.05936	0.49069	0.00931	0.00291	0.05384
		CC	0.99204	-0.00796	0.05771	0.23841	1.00146	-0.00146	0.00438	0.06598	0.48866	0.01134	0.00299	0.18165
		LOD	0.71070	-0.28930	0.13984	0.22359	1.06626	-0.06626	0.00864	0.06408	0.53118	-0.03118	0.00430	0.18396
	75%	Our	0.95360	0.04640	0.15590	0.36324	1.01034	0.01034	0.00997	0.09115	0.47998	0.02002	0.00525	0.07272
		CC	0.95224	-0.04776	0.17330	0.40040	1.01071	-0.01071	0.01088	0.09971	0.47850	0.02150	0.00538	0.24667
		LOD	0.72379	-0.27621	0.24168	0.37168	1.06236	-0.06236	0.01423	0.09397	0.50246	-0.00246	0.00549	0.24934

Table 13: Parameter estimates and goodness of fit measures for $n = 100$ and $nsim = 500$ for the method introduced in this paper (Our), complete case analysis (CC) and substitution method (LOD). $*$ indicates that the AFT model did not converge to a value

			$\beta_0 = 1$				$\beta_1 = 1$				σ			
Censoring in X	**Censoring in Y**	**Method**	**Estimate**	**Bias**	**MSE**	**SE**	**Estimate**	**Bias**	**MSE**	**SE**	**Estimate**	**Bias**	**MSE**	**SE**
50%	10%	Our	0.97141	0.02859	0.04514	0.14834	1.00568	0.00568	0.00365	0.04599	0.50769	0.00769	0.00293	0.05183
		CC	0.98186	-0.01814	0.05262	0.21775	1.00445	-0.00445	0.00415	0.06111	0.48532	0.01468	0.00312	0.18342
		LOD	0.66100	-0.33900	0.16696	0.30473	0.93687	0.06313	0.00893	0.08585	0.99246	-0.49246	0.25096	0.23386
	25%	Our	0.98736	0.01264	0.04751	0.16835	1.00205	0.00205	0.00380	0.05021	0.50097	0.00097	0.00300	0.05418
		CC	0.98114	-0.01886	0.05934	0.23160	1.00466	-0.00466	0.00455	0.06426	0.48441	0.01559	0.00324	0.18836
		LOD	0.57938	-0.42062	0.24412	0.33413	0.95493	0.04507	0.00782	0.09269	1.01482	-0.51482	0.27536	0.25789
	50%	Our	0.99601	0.00399	0.07834	0.25120	1.00016	0.00016	0.00576	0.06833	0.49298	0.00702	0.00373	0.06154
		CC	0.98355	-0.01645	0.10733	0.31362	1.00362	-0.00362	0.00740	0.08232	0.48342	0.01658	0.00392	0.21207
		LOD	0.47136	-0.52864	0.41973	0.44523	0.98462	0.01538	0.00981	0.11568	0.99085	-0.49085	0.25405	0.30582
	75%	Our	0.95493	0.04507	0.23296	0.42605	1.00965	0.00965	0.01401	0.10474	0.47931	0.02069	0.00628	0.07958
		CC	0.91832	-0.08168	0.33390	0.52568	1.01849	-0.01849	0.01882	0.12669	0.47206	0.02794	0.00674	0.27638
		LOD	0.36599	-0.63401	0.81180	0.68485	1.02045	-0.02045	0.02297	0.16276	0.90161	-0.40161	0.18380	0.38580
75%	10%	Our	0.95039	0.04961	0.07506	0.16733	1.00936	0.00936	0.00539	0.05181	0.52438	0.02438	0.00548	0.06225
		CC	0.98090	0.01910	0.07997	0.26606	1.00416	-0.00416	0.00651	0.07471	0.47747	0.02253	0.00530	0.22460
		LOD	2.38250	-1.38250	2.08367	0.47517	0.36147	0.63853	0.41791	0.10566	1.38413	-0.88413	0.79276	0.32775
	25%	Our	0.97046	0.02954	0.07807	0.18801	1.00513	0.00513	0.00556	0.05585	0.51385	0.01385	0.00514	0.06556
		CC	0.98420	0.01580	0.08626	0.28322	1.00335	-0.00335	0.00684	0.07858	0.47602	0.02398	0.00558	0.23049
		LOD	2.33331	-1.33331	1.96007	0.49731	0.36502	0.63498	0.41377	0.10998	1.44073	-0.94073	0.89951	0.37667
	50%	Our	0.97496	0.02504	0.11916	0.2829	1.00454	0.00454	0.00793	0.0762	0.50000	0.00000	0.00581	0.07421
		CC	0.98146	-0.01854	0.16263	0.38626	1.00325	-0.00325	0.01133	0.10134	0.47476	0.02524	0.00662	0.26004
		LOD	2.37346	1.37346	2.08006	0.54147	0.35224	0.64776	0.42959	0.11582	1.47558	-0.97558	0.97399	0.48389
	75%	Our	0.90420	0.09580	0.35082	0.47447	1.02116	0.02116	0.02011	0.11594	0.47978	0.02022	0.00951	0.09454
		CC	0.86292	-0.13708	0.63490	0.65843	1.03000	-0.03000	0.03564	0.15844	0.45916	0.04084	0.01193	0.34007
		LOD	2.69899	1.69899	3.11370	0.63955	0.29258	0.70742	0.50914	0.12502	1.42585	-0.92585	0.89902	0.64581

Table 13: (Continued)

C R-code

```
### loading libraries
library(survival)
library(stats4)
library(DescTools)

##############################################################################
#####################  Masterthesis: simulation of data ######################
##############################################################################

### function for creating data frame in right format
df <- function(x,y,censx,censy){
  case<-c(1:length(x))
  for (i in 1:length(x)){
    if (censx[i] == 1){ #x observed
      if (censy[i] == 0){ #y censored
        case[i] <- 3
      }else{ #y observed
        case[i] <- 4
      }
    }else{ #x censored
      if (censy[i] == 0){ #y censored
        case[i] <- 1
      }else{ #y observed
        case[i] <- 2
      }
    }
  }
  return(data.frame('x'=x,'y'=y,'censx'=censx,'censy'=censy,'case'=case))
}

### function for simulation censoring
censoring<-function(lambda,variable){
  cens <- c(1:n)
  u <- runif(n,0,lambda)
  for (i in 1:n){ #value=max(variable,u)
    if (variable[i]<u[i]){
      cens[i] <- 0
      variable[i]<-u[i]
    }
    else {
      cens[i] <- 1
    }
  }
  return(list('cens'=cens,'variable'=variable))
}
```

```r
### function for simulating data
data_simulation <- function(n,beta0,beta1,sigma,lambda,perc_censx,lambda_cy){
  #simulating covariate data assuming uniform distribution
  x<-runif(n,0,lambda)
  lambda_cx <- 2*perc_censx*lambda
  results_x<-censoring(lambda_cx,x)

  #simulating response data assuming Weibull AFT model
  rho <- exp(beta0+beta1*x)
  gamma <- 1/exp(sigma)
  y <- rweibull(n,shape=gamma,scale=rho)
  results_y<-censoring(lambda_cy,y)

  #creating data frame
  data <- df(results_x$variable,results_y$variable,results_x$cens,results_y$cens)
  return(data)
}

### Weibull AFT model (1 covariate)
weibull_density <- function(y,x,beta0,beta1,sigma){
  rho = exp(beta0 + beta1*x)
  gamma = 1/exp(sigma)
  return( (gamma/rho)*(y/rho)^(gamma-1)*exp(-(y/rho)^(gamma)))
} #checked vs dweibull(y_i,shape=gamma,scale=rho)
weibull_cumdensity <- function(y,x,beta0,beta1,sigma){
  rho = exp(beta0 + beta1*x)
  gamma = 1/exp(sigma)
  return(1-exp(-(y/rho)^(gamma)))
} #checked vs pweibull(y_i,shape=gamma,scale=rho,lower.tail=T)

### Determining lambda_cy to reach certain amount of censoring
set.seed(2023)
lambda_x<-5 #arbitrary choice
beta_0 = 1
beta_1 = 1
sigma_0 = log(0.5)
#numerical integration to determine probability of censoring in Y dependent on X
censy_given_lambda <- function(lambda_censy){
  x<-sort(runif(200,0,lambda_x))
  perc_censy<-c(1:length(x))
  for (j in 1:length(x)){
    f<-function(y){
      (1/lambda_censy)*weibull_cumdensity(y,x[j],beta_0,beta_1,sigma_0)
    }
    perc<-integrate(f,lower=0,upper=lambda_censy)
    perc_censy[j]<- perc$value
  }
  df<-data.frame('x'=x,'prob'=perc_censy)
  return(df)
}
```

```r
set.seed(2023)
n <- 2000 #nr of datapoints
beta_0 = 1
beta_1 = 1
sigma_0 = log(0.5)
lambda_x <- 5 #X ~ U(0,lambda)
perc_censx <- 0.25 #probability of censoring in x
lambda_y10<-6
lambda_y25<-15
lambda_y50<-62
lambda_y75<-230
chosen_lambdas<-c(lambda_y10,lambda_y25,lambda_y50,lambda_y75)

data<-data_simulation(n,beta_0,beta_1,sigma_0,lambda_x,perc_censx,chosen_lambdas[4])
censy<-1-sum(data$censy)/n

#plotting probability of censoring versus X for chosen lambdas
results <- censy_given_lambda(chosen_lambdas[1])
plot(results$x,results$prob,type='l',lwd=2,ylab='Probability of censoring in Y',
     xlab='x',ylim=c(0,1),xlim=c(0,5))
legend('topright',
       legend=c(
         "lambda = 6",
         "lambda = 15",
         "lambda = 62",
         "lambda = 230"),
       lty=1,col=seq(1:length(chosen_lambdas)),cex=1,lwd=2)
for (i in 2:length(chosen_lambdas)){
  results <- censy_given_lambda(chosen_lambdas[i])
  lines(results$x,results$prob,col=i,lwd=2)
}

################################################################################
######################### Masterthesis: R-code for method ######################
################################################################################

### Simulate data to test method
set.seed(2023)
n <- 500 #nr of datapoints
beta_0 = 1
beta_1 = 1
sigma_0 = log(0.5)
lambda_x <- 5 #X ~ U(0,lambda)
perc_censx <- 0.50 #probability of censoring in x
lambda_cy<-chosen_lambdas[2]
data<-data_simulation(n,beta_0,beta_1,sigma_0,lambda_x,perc_censx,lambda_cy)
```

```
### Function for KM for left censored data
#input:
  #x=left-censored data,
  #censx = censoring indicator (0=censored, 1=observed)
left_km<-function(x,censx){
  #transforming left-censored data into right-censored data
  M = max(x)+1
  x_tilde = M-x

  #estimating survival function (KM)
  fit <- survfit(Surv(x_tilde,censx)~1)
  s <- summary(fit)$surv #survival prob
  values <- M-summary(fit)$time #x values belonging to survival prob

  #'jumps/weights' of survival function
  l <- length(s)
  if (s[l]!=0){ #smallest value is censored
    weights <- c(1:(l+1))
    weights[1]<-1-s[1]
    for (i in 1:(l-1)){
      weights[i+1]<-s[i]-s[i+1]
    }
    #below and equal to smallest censored value: cum density = 0
    lowest_cens_val <- min(x[censx==0])
    weights[l+1] <- s[l]
    values<-c(values, lowest_cens_val)
  }else{ #smallest value is observed
    weights <- c(1:l)
    weights[1]<-1-s[1]
    for (i in 1:(l-1)){
      weights[i+1]<-s[i]-s[i+1]
    }
  }
  return(list('weights'= rev(weights),'x' = rev(values)))
}

### Functions defining sums needed for calculation of Likelihood
km <- left_km(data$x,data$censx)
sum_ycens<-function(x_i,y_i,beta0,beta1,sigma){
  sum <- 0
  j=1
  x<-km$x[j]
  while ( (x <= x_i) & (j<(length(km$x)+1)) ){
    sum <- sum + (weibull_cumdensity(y_i,x,beta0,beta1,sigma)*km$weights[j])
    j=j+1
    x<-km$x[j]
  }
  return(sum)
}
sum_yobs<-function(x_i,y_i,beta0,beta1,sigma){
  sum <- 0
```

```
  j=1
  x<-km$x[j]
  while ( (x <= x_i) & (j<(length(km$x)+1)) ){
    sum <- sum + (weibull_density(y_i,x,beta0,beta1,sigma)*km$weights[j])
    j=j+1
    x<-km$x[j]
  }
  return(sum)
}

### Minus log likelihood function
minusloglik <- function(beta0,beta1,sigma){
  loglik <- 0
  for (i in 1:length(data$x)){
    y_i<-data$y[i]
    x_i<-data$x[i]
    result<- switch(data$case[i],
                    sum_ycens(x_i,y_i,beta0,beta1,sigma),
                    sum_yobs(x_i,y_i,beta0,beta1,sigma),
                    weibull_cumdensity(y_i,x_i,beta0,beta1,sigma),
                    weibull_density(y_i,x_i,beta0,beta1,sigma)
                    )
    loglik <- loglik + log(result)
  }
  return(-loglik)
}

### Maximum likelihood estimate
mle(minusloglik,start=list(beta0=beta_0,beta1=beta_1,sigma=sigma_0))

##############################################################################
########################### Masterthesis: simulations ########################
##############################################################################

simulation<-function(n,nsim,perc_censx,lambda_cy,beta_0,beta_1,sigma_0){
  lambda <- 5 #X ~ U(0,lambda)

  beta0_vec<-c(1:nsim)
  beta1_vec<-c(1:nsim)
  sigma_vec<-c(1:nsim)
  se_beta0_vec <-c(1:nsim)
  se_beta1_vec <-c(1:nsim)
  se_sigma_vec <-c(1:nsim)
  prob_censy_vec<-c(1:nsim)
  prob_censx_vec<-c(1:nsim)

  for (sim in 1:nsim){

    data<-data_simulation(n,beta_0,beta_1,sigma_0,lambda,perc_censx,lambda_cy)
    prob_censy_vec[sim]<-1-sum(data$censy)/n
    prob_censx_vec[sim]<-1-sum(data$censx)/n
```

```
km <- left_km(data$x,data$censx) #'jumps' for cumulative density function of x

sum_ycens<-function(x_i,y_i,beta0,beta1,sigma){
  sum <- 0
  j=1
  x<-km$x[j]
  while ( (x <= x_i) & (j<(length(km$x)+1)) ){
    sum <- sum + (weibull_cumdensity(y_i,x,beta0,beta1,sigma)*km$weights[j])
    j=j+1
    x<-km$x[j]
  }
  return(sum)
}

sum_yobs<-function(x_i,y_i,beta0,beta1,sigma){
  sum <- 0
  j=1
  x<-km$x[j]
  while ( (x <= x_i) & (j<(length(km$x)+1)) ){
    sum <- sum + (weibull_density(y_i,x,beta0,beta1,sigma)*km$weights[j])
    j=j+1
    x<-km$x[j]
  }
  return(sum)
}

#minus log likelihood function
minusloglik <- function(beta0,beta1,sigma){
  loglik <- 0
  for (i in 1:length(data$x)){
    y_i<-data$y[i]
    x_i<-data$x[i]
    result<- switch(data$case[i],
                    sum_ycens(x_i,y_i,beta0,beta1,sigma),
                    sum_yobs(x_i,y_i,beta0,beta1,sigma),
                    weibull_cumdensity(y_i,x_i,beta0,beta1,sigma),
                    weibull_density(y_i,x_i,beta0,beta1,sigma)
    )
    loglik <- loglik + log(result)
  }
  return(-loglik)
}

#mle estimates
results<-mle(minusloglik,start=list(beta0=beta_0,beta1=beta_1,sigma=sigma_0))

beta0_vec[sim] <- coef(results)[1]
se_beta0_vec[sim] <- coef(summary(results))[1,2]
beta1_vec[sim] <- coef(results)[2]
se_beta1_vec[sim] <- coef(summary(results))[2,2]
```

```
  sigma_vec[sim] <- exp(coef(results)[3])
  se_sigma_vec[sim] <- coef(summary(results))[3,2]*exp(coef(results)[3])
}
#mean value / estimate for parameters
mean_beta0 = mean(beta0_vec)
mean_beta1 = mean(beta1_vec)
mean_sigma = mean(sigma_vec)

#bias for parameters
bias_beta0 = mean_beta0 - beta_0
bias_beta1 = mean_beta1 - beta_1
bias_sigma = mean_sigma - exp(sigma_0)

#standard error for parameters
se_beta0 = mean(se_beta0_vec)
se_beta1 = mean(se_beta1_vec)
se_sigma = mean(se_sigma_vec)

#QQ plots
qqnorm((beta0_vec-mean_beta0)/se_beta0,main='Normal Q-Q plot for beta_0')
abline(a=0,b=1,col='red')
qqnorm((beta1_vec-mean_beta1)/se_beta1,main='Normal Q-Q plot for beta_1')
abline(a=0,b=1,col='red')
qqnorm((sigma_vec-mean_sigma)/se_sigma,main='Normal Q-Q plot for sigma')
abline(a=0,b=1,col='red')

#confidence interval + coverage probability
beta0_ci_l<-mean_beta0-1.96*se_beta0
beta0_ci_u<-mean_beta0+1.96*se_beta0
beta0_cp<-sum(ifelse(beta0_vec<beta0_ci_u & beta0_vec>beta0_ci_l,1,0))/nsim

beta1_ci_l<-mean_beta1-1.96*se_beta1
beta1_ci_u<-mean_beta1+1.96*se_beta1
beta1_cp<-sum(ifelse(beta1_vec<beta1_ci_u & beta1_vec>beta1_ci_l,1,0))/nsim

sigma_ci_l<-mean_sigma-1.96*se_sigma
sigma_ci_u<-mean_sigma+1.96*se_sigma
sigma_cp<-sum(ifelse(sigma_vec<sigma_ci_u & sigma_vec>sigma_ci_l,1,0))/nsim

#MSE
mse_beta0<-sum((beta0_vec-beta_0)^2)/nsim
mse_beta1<-sum((beta1_vec-beta_1)^2)/nsim
mse_sigma<-sum((sigma_vec-exp(sigma_0))^2)/nsim

return(list('mean_beta0'=mean_beta0,
            'bias_beta0'=bias_beta0,
            'mse_beta0'=mse_beta0,
            'se_beta0'=se_beta0,
            'mean_beta1'=mean_beta1,
            'bias_beta1'=bias_beta1,
            'mse_beta1'=mse_beta1,
```

```
              'se_beta1'= se_beta1,
              'mean_sigma'= mean_sigma,
              'bias_sigma'=bias_sigma,
              'mse_sigma'=mse_sigma,
              'se_sigma'=se_sigma,
              'ci_beta0'=c(beta0_ci_l,beta0_ci_u),
              'cp_beta0'=beta0_cp,
              'ci_beta1'=c(beta1_ci_l,beta1_ci_u),
              'cp_beta1'=beta1_cp,
              'ci_sigma'=c(sigma_ci_l,sigma_ci_u),
              'cp_sigma'=sigma_cp,
              'mean_censy'=mean(prob_censy_vec),
              'mean_censx'=mean(prob_censx_vec)
              ))
}

simulation_otherMethods<-function(n,nsim,perc_censx,lambda_cy,beta_0,beta_1,sigma_0){
  lambda_x <- 5 #X ~ U(0,lambda)

  beta0_vec_cc<-c(1:nsim)
  beta1_vec_cc<-c(1:nsim)
  sigma_vec_cc<-c(1:nsim)
  se_beta0_vec_cc <-c(1:nsim)
  se_beta1_vec_cc <-c(1:nsim)
  se_sigma_vec_cc <-c(1:nsim)

  beta0_vec_sub<-c(1:nsim)
  beta1_vec_sub<-c(1:nsim)
  sigma_vec_sub<-c(1:nsim)
  se_beta0_vec_sub <-c(1:nsim)
  se_beta1_vec_sub <-c(1:nsim)
  se_sigma_vec_sub <-c(1:nsim)

  for (sim in 1:nsim){

    data<-data_simulation(n,beta_0,beta_1,sigma_0,lambda_x,perc_censx,lambda_cy)

    #complete case method
    x_cc <- data$x[data$censx==1]
    y_cc <- data$y[data$censx==1]
    censy_cc <- data$censy[data$censx==1]
    results_cc <- survreg(Surv(y_cc,censy_cc,type='left') ~ x_cc,dist="weibull")

    #substitution method
    results_sub <- survreg(Surv(y,censy,type='left') ~ x,data,dist="weibull")

    beta0_vec_cc[sim] <- coef(results_cc)[1]
    se_beta0_vec_cc[sim] <- summary(results_cc)$table[1,2]
    beta1_vec_cc[sim] <- coef(results_cc)[2]
    se_beta1_vec_cc[sim] <- summary(results_cc)$table[2,2]
    sigma_vec_cc[sim] <- results_cc$scale
```

```
  se_sigma_vec_cc[sim] <- summary(results_cc)$table[3,2]*exp(results_cc$scale)

  beta0_vec_sub[sim] <- coef(results_sub)[1]
  se_beta0_vec_sub[sim] <- summary(results_sub)$table[1,2]
  beta1_vec_sub[sim] <- coef(results_sub)[2]
  se_beta1_vec_sub[sim] <- summary(results_sub)$table[2,2]
  sigma_vec_sub[sim] <- results_sub$scale
  se_sigma_vec_sub[sim] <- summary(results_sub)$table[3,2]*exp(results_sub$scale)
}
#mean value / estimate for parameters
mean_beta0_cc = mean(beta0_vec_cc)
mean_beta1_cc = mean(beta1_vec_cc)
mean_sigma_cc = mean(sigma_vec_cc)

mean_beta0_sub = mean(beta0_vec_sub)
mean_beta1_sub = mean(beta1_vec_sub)
mean_sigma_sub = mean(sigma_vec_sub)

#bias for parameters
bias_beta0_cc= mean_beta0_cc - beta_0
bias_beta1_cc = mean_beta1_cc - beta_1
bias_sigma_cc = mean_sigma_cc - exp(sigma_0)

bias_beta0_sub = beta_0 - mean_beta0_sub
bias_beta1_sub = beta_1 - mean_beta1_sub
bias_sigma_sub = exp(sigma_0) - mean_sigma_sub

#standard error for parameters
se_beta0_cc = mean(se_beta0_vec_cc)
se_beta1_cc = mean(se_beta1_vec_cc)
se_sigma_cc = mean(se_sigma_vec_cc)

se_beta0_sub = mean(se_beta0_vec_sub)
se_beta1_sub = mean(se_beta1_vec_sub)
se_sigma_sub = mean(se_sigma_vec_sub)

#MSE
mse_beta0_cc<-sum((beta0_vec_cc-beta_0)^2)/nsim
mse_beta1_cc<-sum((beta1_vec_cc-beta_1)^2)/nsim
mse_sigma_cc<-sum((sigma_vec_cc-exp(sigma_0))^2)/nsim

mse_beta0_sub<-sum((beta0_vec_sub-beta_0)^2)/nsim
mse_beta1_sub<-sum((beta1_vec_sub-beta_1)^2)/nsim
mse_sigma_sub<-sum((sigma_vec_sub-exp(sigma_0))^2)/nsim

return(list('mean_beta0_cc'=mean_beta0_cc,
            'bias_beta0_cc'=bias_beta0_cc,
            'mse_beta0_cc'=mse_beta0_cc,
            'se_beta0_cc'=se_beta0_cc,
            'mean_beta1_cc'=mean_beta1_cc,
            'bias_beta1_cc'=bias_beta1_cc,
```

```
              'mse_beta1_cc'=mse_beta1_cc,
              'se_beta1_cc'= se_beta1_cc,
              'mean_sigma_cc'= mean_sigma_cc,
              'bias_sigma_cc'=bias_sigma_cc,
              'mse_sigma_cc'=mse_sigma_cc,
              'se_sigma_cc'=se_sigma_cc,
              'mean_beta0_sub'=mean_beta0_sub,
              'bias_beta0_sub'=bias_beta0_sub,
              'mse_beta0_sub'=mse_beta0_sub,
              'se_beta0_sub'=se_beta0_sub,
              'mean_beta1_sub'=mean_beta1_sub,
              'bias_beta1_sub'=bias_beta1_sub,
              'mse_beta1_sub'=mse_beta1_sub,
              'se_beta1_sub'= se_beta1_sub,
              'mean_sigma_sub'= mean_sigma_sub,
              'bias_sigma_sub'=bias_sigma_sub,
              'mse_sigma_sub'=mse_sigma_sub,
              'se_sigma_sub'=se_sigma_sub
  ))
}

set.seed(2023)

n <- 100 #nr of datapoints
nsim <- 500 #nr of simulations
perc_censx<-0.75 #probability of censoring in x
lambda_cy <-chosen_lambdas[4]
beta_0 <- 1
beta_1 <- 1
sigma_0 <- log(0.5)

start <- Sys.time()
sim<-simulation_otherMethods(n,nsim,perc_censx,lambda_cy,beta_0,beta_1,sigma_0)
print( Sys.time() - start )
unlist(sim)
```