



SCHOOL of
GRADUATE STUDIES
EAST TENNESSEE STATE UNIVERSITY

East Tennessee State University
Digital Commons @ East Tennessee
State University

Electronic Theses and Dissertations


Student Works

8-2019

Robustness of Semi-Parametric Survival Model: Simulation Studies and Application to Clinical Data

Isaac Nwi-Mozu
East Tennessee State University

Follow this and additional works at: <https://dc.etsu.edu/etd>

 Part of the [Statistical Models Commons](#), and the [Survival Analysis Commons](#)

Recommended Citation

Nwi-Mozu, Isaac, "Robustness of Semi-Parametric Survival Model: Simulation Studies and Application to Clinical Data" (2019). *Electronic Theses and Dissertations*. Paper 3618. <https://dc.etsu.edu/etd/3618>

This Thesis - unrestricted is brought to you for free and open access by the Student Works at Digital Commons @ East Tennessee State University. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of Digital Commons @ East Tennessee State University. For more information, please contact digilib@etsu.edu.

Robustness of Semi-Parametric Survival Model: Simulation Studies and Application
to Clinical Data

A thesis

presented to

the faculty of the Department of Mathematics

East Tennessee State University

In partial fulfillment

of the requirements for the degree

Master of Science in Mathematical Sciences

by

Isaac Nwi - Mozu

August 2019

JeanMarie Hendrickson, Ph.D., Chair

Robert Price, Ph.D.

Nicole Lewis, Ph.D.

Keywords: parametric/ semi-parametric model, C.P.H, survival analysis

ABSTRACT

Robustness of Semi-Parametric Survival Model: Simulation Studies and Application
to Clinical Data

by

Isaac Nwi-Mozu

An efficient way of analyzing survival clinical data such as cancer data is a great concern to health experts. In this study, we investigate and propose an efficient way of handling survival clinical data. Simulation studies were conducted to compare performances of various forms of survival model techniques using an R package “survsim”. Models performance was conducted with varying sample sizes as small ($n < 30$), mild sample ($30 \leq n \leq 5000$) and large sample ($n > 5000$). For small and mild samples, the performance of the semi-parametric outperform or approximate the performance of the parametric model. However, for large samples, the parametric model outperforms the semi-parametric model. We compared the effectiveness and reliability of our proposed techniques using a real clinical data of mild sample size. Finally, systematic steps on how to model and explain the proposed techniques on real survival clinical data was provided.

Copyright by Isaac Nwi-Mozu, 2019

All Rights Reserved

DEDICATION

This thesis is solely dedicated to my lovely brother Francis Kaku of blessed memory. Rest in peace.

ACKNOWLEDGMENTS

My greatest word of thanks goes to the Almighty God for the guidance given to me for the whole duration of study. My sincere thanks goes to my adviser Dr. Jean Marie Hendrickson for her contribution in this thesis. Her patient, direction, supervision and guidance has made the completion of this study a reality. She saw the potential in me and agreed to work with me as my adviser on the proposed topic. I am grateful for her time and contribution in making this thesis a success.

I am very appreciative to my committee members, Dr. Robert Price and Dr. Nicole Lewis for sacrificing their precious time to serve as my committee members. I thank them for their your brilliant suggestions and comments that improved the work.

I am especially grateful to the Department of Mathematics and Statistics for the financial assistantship given to me throughout my studies. I must express my profound gratitude to Dr. Robert Bob Gardner, Dr. Teresa Haynes, Dr. Rodney Keaton and the entire lecturers in the department for the knowledge they have imparted in me.

I must express my very profound gratitude to my colleagues for the knowledge and fun we shared together throughout my years of study. I am fortunate to have an amazing colleagues.

My special thanks goes to my parents and siblings for their prayers and support. I would like to express my deepest appreciation to Mr/Mrs. Joseph Kusi for their support and encouragement. They made my living and movements during my studies easy. Thank you!

TABLE OF CONTENTS

ABSTRACT	2
DEDICATION	4
ACKNOWLEDGMENTS	5
LIST OF TABLES	11
LIST OF FIGURES	12
1 INTRODUCTION	13
1.1 Background of the Study	13
1.2 Problem Statement	14
1.3 Objectives of the Study	15
1.4 Scope of the Study	16
1.5 Significance/Justification of the Study	16
1.6 Materials and Method	16
1.7 Limitation of the Study	17
1.8 Summary/Organization of the Study	17
2 LITERATURE REVIEW	18
2.1 General Framework of Survival Analysis	18
2.2 Approaches to Survival Analysis	20
2.2.1 Non Parametric Survival Function	20
2.2.2 Semi-Parametric versus Parametric Survival Function	21
2.3 Application of Survival Analysis with Reference to Clinical Data	23
2.4 Simulation Study	24
3 METHODOLOGY AND MATHEMATICAL FORMULATION	26

3.1	Survival Function	26
3.2	Properties of Survival Function	26
3.2.1	Relationship between Survival Function and Cumulative Distribution Function	27
3.2.2	The Hazard Function	27
3.2.3	Discrete Random Variable of Survival Function	28
3.2.4	Continuous Random Variable of Survival Function	28
3.2.5	Estimation of Survival Function	29
3.3	The Non-Parametric Method (Kaplan-Meier Estimator)	30
3.3.1	Estimating Kaplan-Meier Estimator	30
3.4	Semi- Parametric Model	32
3.4.1	Cox Proportional Hazard Model (Cox PHM)	32
3.4.2	Method of Estimation	33
3.4.3	Assumption of Cox PH	34
3.4.4	Checking Cox PH Assumption	35
3.4.5	What to do when Cox PH assumption is rejected	35
3.5	Parametric Survival Model	36
3.5.1	Accelerated Failure - Time Models	36
3.5.2	The Exponential Distribution	38
3.5.3	The Weibull Distribution	39
3.5.4	Log-Logistic Distribution	41
3.5.5	Log-Normal Distribution	43
3.5.6	Problem Associated With AFT Model	44

	3.5.7	Models Selection	44
	3.6	Simulation Study	45
	3.7	Real Data	46
	3.8	Computer Software and Packages	46
4		ANALYSIS	47
	4.1	Simulation Studies	47
		4.1.1 Simulation Method and Sample Size	47
	4.2	Simulation of Parametric Results	50
		4.2.1 Simulation of Exponential Survival Data without Co- variate	50
		4.2.2 Simulation of Weibull Survival Data without Covariate	52
		4.2.3 Simulation of log-logistic Survival Data with Covariate	52
		4.2.4 Simulation of log-normal Survival Data without Co- variate	53
		4.2.5 Simulation of Cox Survival Data without Covariate .	55
		4.2.6 Simulated Data with Covariate	56
		4.2.7 Simulation of Exponential Survival Data with Covariate	57
		4.2.8 Simulation of Weibull Survival Data with Covariate .	60
		4.2.9 Simulation of log-logistic Survival Data with Covariate	62
		4.2.10 Simulation of log-normal Survival Data with Covariate	64
	4.3	Simulation of Semi-Parametric Results	66
		4.3.1 Simulation of Cox Proportional Hazard Survival Data with Covariate	68

4.4	Data Application	70
4.4.1	Introduction	70
4.4.2	Description of Data Set	70
4.5	Preliminary Results	71
4.6	Kaplan Meier Estimate	72
4.7	Comparing Survival Models	74
4.8	Some Steps to Model and Explain Survival Clinical Data . . .	75
4.8.1	Model Diagnostic/Adequacy	75
4.8.2	Assessing Proportional Hazard (PH) Assumption . .	76
4.8.3	Checking Influential/Outlier Observation	78
4.8.4	Checking the Assumption of Nonlinearity	79
4.8.5	Checking Overall Model Fit	81
4.9	Estimating the Semi-Parametric Model	82
4.10	Further Analysis of Model Development	84
5	DISCUSSION AND CONCLUSION	88
5.1	Future Work	90
	REFERENCES	91
	APPENDICES	96
1	Appendix	96
1.1	R Code	96
1.1.1	Simulation Codes	96
	VITA	235

LIST OF TABLES

1	Simulation of exponential Survival data (time to censorship follows exponential without covariate)	51
2	Simulation of Weibull Survival data (time to censorship follows Weibull without covariate)	53
3	Simulation of log-logistic Survival data (time to censorship follows log-logistic without covariate)	54
4	Simulation of log-normal Survival data (time to censorship follows log-normal without covariate)	55
5	Simulation of Cox survival data (time to censorship follows Cox without covariate)	56
6	Simulation of Exponential survival data (time to censorship with covariate)	59
7	Simulation of Weibull survival data (time to censorship with covariate)	61
8	Simulation of log-logistic survival data (time to censorship with covariate)	63
9	Simulation of log-normal survival data (time to censorship with covariate)	65
10	Simulation of Cox survival data (time to censorship with covariate) .	69
11	First four cases for each treatments group	71
12	Descriptive Statistics of Samples	72
13	Fitted model comparison	75
14	Summary of semi-parametric model	75
15	Test for Proportional Hazard Assumption	76
16	Summary of semi-parametric model	84

17	Summary of fitted semi-parametric model involving treatment only .	85
18	Summary of fitted semi-parametric model involving age only	85
19	Summary of fully semi-parametric model for two categorical covariate	86
20	Test for Proportional Hazard Assumption for two covariate	87
21	Summary of semi-parametric model for categorical age	87

LIST OF FIGURES

1	Illustration of types of censoring	20
2	Steps for analyzing survival clinical data	24
3	Kaplan Meier Graph for no covariate	73
4	Kaplan-Meier Hazard Estimates	74
5	Schoenfeld residuals for covariate against time.	77
6	Index plots of dfbeta for Cox regression of time to age and treatment.	78
7	Nonlinearity of functional form of covariate age	80
8	Cumulative hazard plot of the Cox-Snell residual for Cox PH model to check the overall fitted model.	81

1 INTRODUCTION

1.1 Background of the Study

Survival analysis is a collection of statistical methods that are used to describe, explain, or predict the occurrence and timing of events. The study of survival analysis became very famous in the early 1970's. The outcome variable of interest is time until an event occurs. The time may be measured in years, months, weeks, or days from the beginning of follow-up of an individual/object until an event occurs. The term event is used to mean death, disease incidence, relapse from remission, recovery (e.g., return to work) or any designated experience of interest that may happen to an individual/object. The event of interest is sometimes called failure. For illustration purposes, consider a study that follows lung cancer patients in remission over some months to see how long a patient stays in remission. Here, the event of interest/failure is "going out of remission", the outcome is "time until the patient goes out of remission and time is "months."

In some survival studies, more than one event of interest (multiple events) can occur or be considered. When more than one event occurs, the problem under study is called recurrent event or competing risk problem. For example, the study of time until an individual experiences a headache and stomach pain. The basic functions of survival analysis are the same in all fields, but the name usually depends on the field of study. For instance, in engineering it is called reliability analysis, in sociology it is known as event history analysis, in economics it is famous with the name of duration analysis and, as the name suggests, medical researchers give it the name of survival

analysis. In survival analysis, the time is called survival time.

The methods for handling survival analysis differ from other statistical methods with the reasons that: time is always the response variable, there are possibilities of staggered entries (the unit in the study have different time of entrance) which does not affect survival analysis method since the method deals with the length of time of observation and not based on the same entrance time and finally the assumptions of normality does not hold since survival data are generally skewed.

Three main methods, namely, non parametric, semi-parametric and parametric are usually used to analyze survival data. The methods give similar results but each has its own unique procedure usually under specific assumptions or no assumptions. In this study, we demonstrate that semi-parametric approach is robust among all the three main methods used in survival analysis data. We consequently examine the robustness under study through different sample sizes with varying amounts of censored observations.

1.2 Problem Statement

Survival analysts especially clinicians, are usually on the look out for the “easiest” way to analyze survival data. The study of survival analysis is moderately new with limited software or software packages to handle some survival analysis approach [28] particularly in the area of parametric survival studies. This problem tends to impede which specific procedure to be used in analyzing survival data in practice. When survival analysts are just interested in examining the trend of occurrences of diseases or to compare the efficacies of treatments, the non-parametric approach introduced by

Kaplan-Meier (KM) is usually used since no rigorous assumption is made. However, in most cases, analyst would not just be interested in comparing but also to make a mathematical prediction via modeling of survival data. Modeling, of course, tends to be one of the most intensive/difficult and time consuming approaches in statistical analysis. In the field of survival analysis, lack of statistical software/packages coupled with intensive assumptions to handle some aspect of parametric distributions impede its usage by most clinicians. In practice, few distributions such as Weibull, Exponential, Log-logistics and Lognormal are usually used in parametric survival model. Thus, this study tend to demonstrate/justify a way that can be used to handle survival data especially modeling clinical data through semi-parametric approach whiles relying on less rigorous assumptions.

1.3 Objectives of the Study

The main objectives of the study are as follows:

- To show the robustness of semi-parametric approach among all the survival analysis techniques.
- To compare the results of survival modeling techniques through model selection approach.
- To learn an effective method specific to survival model techniques for survival data.
- To provide some steps to analyze and model real survival clinical data using the semi-parametric approach.

1.4 Scope of the Study

The study will cover the few parametric survival models used in practice such as Weibull, exponential, log-logistics and lognormal distribution and semi-parametric survival models proposed by [6, 7]. The other approach such as non-parametric (KM) is of little concern and so may not be discuss in details.

1.5 Significance/Justification of the Study

Specific or alternative approach to survival modeling is mostly of concern. By the end of this study, clinicians will be able to find an efficient way to model survival data. We also hope this thesis will guide or raise the confidence of other survival analyst in different field a better method to be used in modeling survival data. Finally, the study also broadens the researcher knowledge in steps to handle survival data.

1.6 Materials and Method

We simulate various forms of survival data with varying sample size and censored observations that follows a particular parametric models. We fit each of the simulated data with semi-parametric and other forms of parametric models. We compare the results through model selection procedure using Akaike Information Criterion (AIC) and their standard errors. With similar approach, we fit a real data taken from clinical data. We compare the results consistencies with the simulation results. The results of the clinical data are further examine with graphical output. Conclusions and recommendations are then made based on the research objectives. Throughout the study, R - software version 3.5.1 were used.

1.7 Limitation of the Study

Sometimes in survival study, more than one event is considered usually called multiple event or recurrent event or competing risk problems. In this study, both simulations and real data are limited to single event problems. For example, the study of the time until a breast cancer patient die with the diseases. We however, refer any interested reader to the work of [11] on multiple event problems.

1.8 Summary/Organization of the Study

To achieve expediency when reading, the study was organized in five main chapters with sections and subsections. The first chapter covers the introduction of the study with the following sections: background of the study, problem statement, objectives, scope of the study, significant of the study, materials and methods, limitation and organization of the study. Chapter two provides the literature review on conceptual, empirical works and methods of previous researchers and limitation's, simulations and application to clinical data with pertinent references. Chapter three deals with the mathematical formulations and procedure's used in achieving the study. Chapter four provides a critical analysis, presentation and interpretation of simulation studies and real clinical data. Chapter five gives the conclusion part which involves discussion of the results, conclusion, recommendation of the study and future direction.

2 LITERATURE REVIEW

2.1 General Framework of Survival Analysis

The origins of survival analysis may be tracked from the study of mortality in the seventeenth century. It has become one of the most frequently used methods for analyzing data in many field from epidemiology, engineering, marketing and so on. Some of the objectives of survival analysis include estimation, interpretation and comparison of survival of groups. The primary variable in survival analysis is survival time. The term “survival time” is used generally to mean the time until an event of interest occurs. For example, the time until married couples divorce. One special features of survival time is that it is a nonnegative random variable. The second special feature is that the survival time always follows a skewed distribution. Though, transformation can be applied to make the distribution more symmetric, using a different model may be more satisfactory [16].

Survival analysis is special from any other statistical methods of analysis because of some missing information during the study. Some of the experimental units may not have experienced the event of interest before the study ends, study dropouts and lose to follow up thereby making the dataset incomplete. This phenomenon of incompleteness is termed as censoring. Censoring is group into right, left and random censoring. For illustration of this phenomenon, consider figure 1 that study cancer patients for eight weeks until patient’s die with the cancer. The event of interest is death. The patients have staggered entries into the study i.e, the patients have different entry time.

For right censoring, consider patients B, C, D, and E. The study ends (after 8 weeks) and B and D are still alive. Patients C did not show interest in the study and so dropped out. The investigator were not able to track patient E and so was lost to follow up.

Random censoring occurs when the death is not related to the study under consideration. Suppose that both A and F had fatal car accident and died. The death is not related to the study under consideration, i.e cancer.

Left censoring, the individual has already experience the event before enrolling into the study but the exact time of first exposure in not known. Suppose that a college counselor wants to study the time until newly admitted students smoke cigarette before graduation. If a student has smoked before being admitted then that student is left censored.

In all the illustrative examples, we do not know the individual/object exact survival time. The most widely used technique is right censoring.

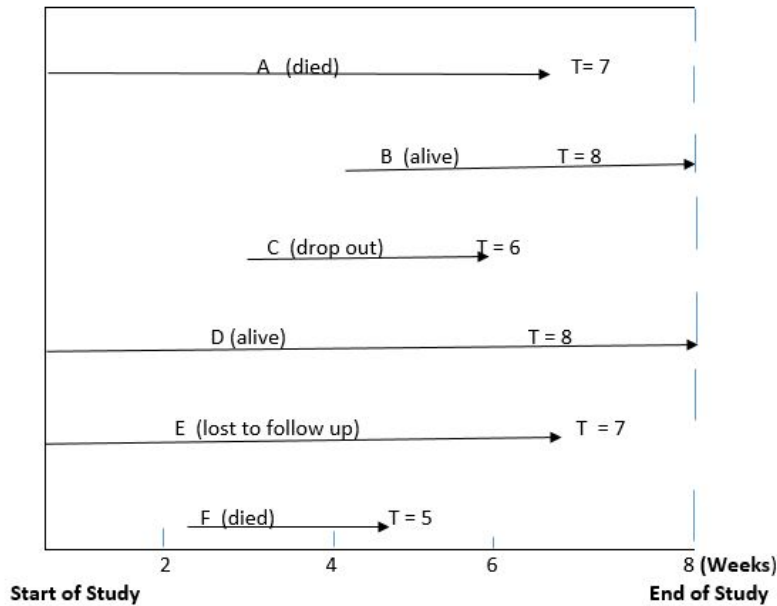


Figure 1: Illustration of types of censoring

2.2 Approaches to Survival Analysis

2.2.1 Non Parametric Survival Function

The commonly used nonparametric method for estimation of the survival function is the [10]. The nonparametric analyses is often used where there is doubt about the exact form of distribution. The advantages and popularity of Kaplan Meier(K-M) is its accessibility in most computer and statistical software. Also, the non-parametric method makes no assumption about the distribution of the failure time. However, using a non-parametric analysis typically generated much wider confidence bounds than those calculated via parametric analysis. The K-M survival curve is the plot of the Kaplan-Meier estimator of the survival function $S(t)$ against time t . The graph is a step function that decreases at every event time. In modeling of survival data the

K-M curve may give a hint about which model is more appropriate for a particular data set. If the empirical survival curve behaves similarly to a particular parametric distribution then this model may have a good fit to the data [4] and [14].

2.2.2 Semi-Parametric versus Parametric Survival Function

Which method is most accurate among the three methods of survival analysis is still under battle especially in the studies of clinical data. Many researchers have proposed which methods fits well in their research studies based on the underlying assumptions. Parametric method assume that the time to event variable comes from a specific distributional family. Based on asymptotic results, under certain circumstances, parametric models lead to more efficient parameter estimates. In the work of [13] they indicated that when the underlying assumptions of the distribution follows a known parametric distribution, the result has more power (little variance) than the semi parametric model and are easier to interpret.

Furthermore, parametric survival models estimate the survival function more precisely than the non-parametric method. Unfortunately, in real data, it is often hard for a data to precisely follow a particular parametric distribution. The claim works in simulation studies [21]. One main challenges in parametric approach is lack of availability of software [28] and in some cases, hard to be analyzed by many health experts, biostatistician and statistician alike. Again, fully parametric models involve stronger restrictive assumptions than semi-parametric. There are many forms of parametric distribution in statistics but few such as Weibull, Exponential, Log-logistic and lognormal are commonly used and tested for survival models in many books and lit-

erature. Researchers typically want to know how well their statistical models fit the data. Comparisons of model selection can be accomplished with Akaike's information criterion (AIC) or Schwarz's Bayesian Information Criterion (SBC or BIC). Usually, the AIC is used to select the best model among the few parametric model specified [25]. Following the concern of [28] it is clear that given availability of software to handle almost all the parametric models, the best model that is commonly selected by AIC as best among the few parametric models used in literature may not even worth selected as best. In most cases, the best model selected by AIC do not satisfy its underlying distribution assumption. These problems, commonly make parametric approach of little use in practice.

Again, though fully parametric survival models under certain circumstances may lead to more efficient parameter estimates than both nonparametric and semi-parametric models [20], they are not often used in the analysis of medical studies of survival as they involved stronger assumptions which may make them more prone to danger of model misspecification than the semi-parametric model.

The semi-parametric model is another approach in survival model. The well know semi-parametric survival analysis model is the [6, 7]. This approach is commonly used by survival analyst in analyzing clinical data [1]. The Cox model focuses directly on the hazard function with a mild assumptions about the baseline hazard function. That is, the semi-parametric nature of the Cox regression, although based on a parametric regression model, it does not make specific assumptions about the probability distribution of the event time.

The hazard and the probability of survival are interrelated; the relationship may

be complex, but sufficient to be able to determine a parameter with the other. Thus, given the survival function $S(t)$, we can obtain the corresponding hazard function $\lambda(t)$ and vice versa. The main assumption of semi-parametric model is that hazard is proportional. In the case where the hazard is not proportional, [7] proposed a method generally called extended Cox model, a means to handle non proportionality assumption by including time dependent covariate or using stratification. The Cox model tends to be “robust” so that the results from using the Cox model is closely approximate the results for the correct parametric model [13]. In a study by [20], they analyzed four different kinds of clinical data and the results shows that in most cases the semi-parametric and parametric results were similar. Many forms of robustness has been discussed and details can be found in the works of [17] and [15].

To avoid increased efforts with model checking and because dangers of misspecification are assumed to increase with parametric survival models, clinicians tend to prefer Cox’s model. The proportional hazards regression model is very popular due to the easy concept and accessibility of software [3].

2.3 Application of Survival Analysis with Reference to Clinical Data

The term survival analysis is predominant in the field of medicine. The main objective in the study of survival analysis in clinical trials is often to compare the performance of treatments or the survival differences of two or more groups, see for example the work of [16]. In order to investigate the comparative performance of Cox and parametric survival models under the typical conditions of clinical studies, either some samples of simulations or by means of real data sets is used. For instance, in the

study of the comparison of Cox and parametric models in clinical studies, [20] used real data where they studied four different clinical data on lung cancer, infections after heart transplantation, breast cancer and liver transplant with different sample sizes ranging from 217 and 526, in some cases the cox provides similar results to the parametric models. They also compared the results with parameter variability.

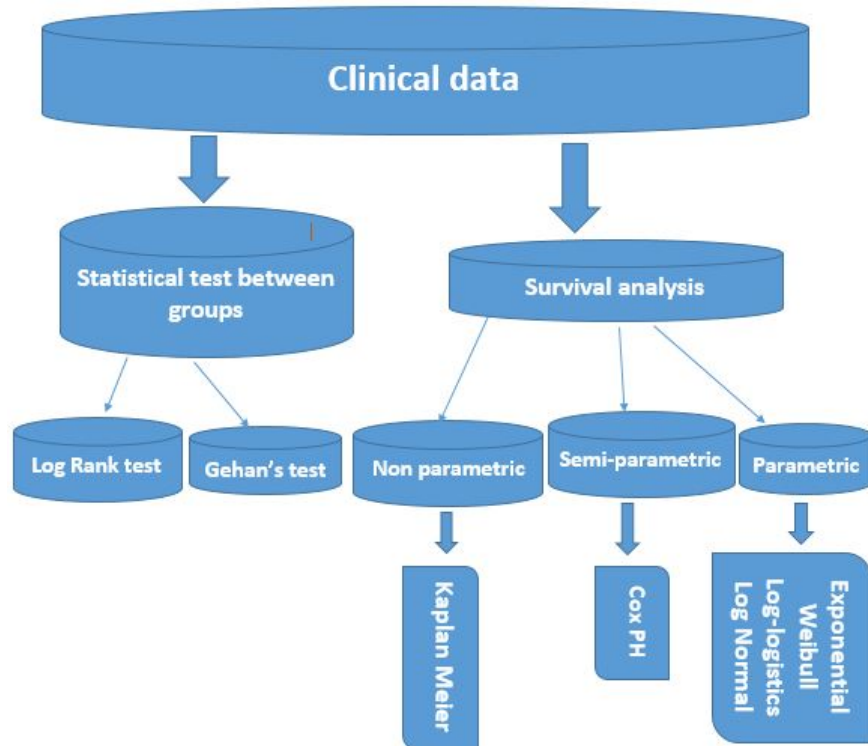


Figure 2: Steps for analyzing survival clinical data

2.4 Simulation Study

Simulating a data to check the performance of a particular techniques in survival analysis has gained root in recent studies because of the rapid growth in the study area as well as accessibility of software. Simulation studies, uses computer intensive

procedures to assess the performance of a variety of statistical methods and evaluating the properties of the model. The generation of survival data is the most fundamental and important component in the simulation of clinical study. However, few published articles are available in literature which make use of the simulation of complex survival data, one possible reason being a lack of easily used tools facilitating such simulations. One popularly used software for generating simulations of clinical data is R. In R, there are a lot of functions for survival analysis in the package “survival”. In recent study, [19] wrote some packages for R and outlined the procedure for generating clinical data. This thesis explore the usage of the packages and procedures suggested.

A comprehensive techniques for simulation of survival data is given by [18] and [24].

3 METHODOLOGY AND MATHEMATICAL FORMULATION

3.1 Survival Function

Let T be the random variable representing survival time of a subject, the survival function $S(t)$, is defined as the probability that an individual survive beyond time t .

It is given by:

$$S(t) = P(T > t), 0 \leq S(t) \leq 1 \quad (1)$$

3.2 Properties of Survival Function

1. It is decreasing function.
2. At time $t = 0$, $S(t)=1$
3. At time $t = \infty$, $S(t) = 0$. As time goes to infinity, the survival curve goes to 0.

If the failure time is a discrete random variable then:

$$\begin{aligned} S(t) &= \sum_{a_j \geq t} f(a_j) \\ &= \sum_{a_j \geq} f_j \end{aligned} \quad (2)$$

If the failure time is continuous random variable then:

$$S(t) = \int_t^{\infty} f(u)du, \forall t \in [0, \infty] \quad (3)$$

3.2.1 Relationship between Survival Function and Cumulative Distribution Function

Let $f(t), t \geq 0$, denote the probability density function (pdf) of T and $F(t) = P(T \leq t) = \int_0^t f(u)du, t \geq 0$ be the cumulative distribution function (cdf) of T , then

$$\begin{aligned} S(t) &= P(T > t) \\ &= 1 - P(T \leq t) \\ &= 1 - F(t) \end{aligned} \tag{4}$$

Taking the differential of both sides

$$\begin{aligned} S'(t) &= -F'(t) \\ &= -f(t) \end{aligned} \tag{5}$$

3.2.2 The Hazard Function

The hazard function $\lambda(t)$, is a measure of the probability of failure during a very small interval, assuming that the individual has survived at the beginning of the interval. it is defined mathematically as:

$$\lambda(t) = \lim_{\nabla t \rightarrow 0} \frac{P(\text{an individual who survive to time } t \text{ fail in } (t, t + \nabla t))}{\nabla t} \tag{6}$$

$\lambda(t)$ has two major properties: it is always nonnegative and has no upper bound. The cumulative hazard Λ , and the hazard function $\lambda(t)$, for continuous random variable are related by:

$$\Lambda = \int_0^t \lambda(u)du \tag{7}$$

The cumulative hazard $\Lambda(t)$, and the hazard function $\lambda(t)$, for discrete random variable are related by:

$$\lambda(t) = \sum_{j:a_j < t} \lambda(a_j) \quad (8)$$

3.2.3 Discrete Random Variable of Survival Function

If the failure time is discrete then:

$$\begin{aligned} \lambda(a_j) &= \lambda_j = P(T = a_j | T \geq a_j) \\ &= \frac{P(T = a_j)}{P(T \geq a_j)} \\ &= \frac{f(a_j)}{S(a_j)} \end{aligned} \quad (9)$$

3.2.4 Continuous Random Variable of Survival Function

If the failure time is continuous then:

$$\begin{aligned} \lambda(t) &= \lim_{\nabla t \rightarrow 0} \frac{1}{\nabla t} P(t \leq T < t + \nabla t | T \geq t) \\ &= \lim_{\nabla t \rightarrow 0} \frac{1}{\nabla t} \frac{P([t \leq T < t + \nabla t] \cap [T \geq t])}{P(T \geq t)} \\ &= \lim_{\nabla t \rightarrow 0} \frac{1}{\nabla t} \frac{P([t \leq T < t + \nabla t])}{P(T \geq t)} \\ &= \frac{f(t)}{S(t)} \end{aligned} \quad (10)$$

From equation (5)

$$\lambda(t) = \frac{-S'(t)}{S(t)} \quad (11)$$

$$= \frac{-d(\log S(t))}{dt} \quad (12)$$

Integrating both sides:

$$\Lambda(t) = -\log S(t) \quad (13)$$

$$S(t) = \exp^{-\Lambda(t)} \quad (14)$$

$$S(t) = \exp^{-\int_0^t \lambda(t)dt} \quad (15)$$

Equation (14) gives the relationship between cumulative hazard and the survival function while equation (15) gives the relationship between hazard function and the survival function. Thus given any one of three functions $S(t)$, $\Lambda(t)$ and $f(t)$, the others two can be derived.

3.2.5 Estimation of Survival Function

There are two ways of estimating survival function and the hazard function.

1. By specifying a parametric model for $S(t)$ based on a particular density function $f(t)$
2. By developing an empirical estimate of the survival function (i.e., non-parametric estimation)

If there is no censoring, the empirical estimate of the survival function $S(t)$, is proportion of individuals with event times greater than t .

$$S(t) = \frac{\text{number of individuals with failure time greater than } t}{\text{total sample size}} \quad (16)$$

When there is censoring, equation (16) loses its power to estimate $S(t)$ because the numerator can no longer be estimated.

There are three techniques in describing survival data that contain censoring. These are non parametric (Kaplan Meier), semi-parametric (usually called the Cox Proportional Hazard) and parametric.

3.3 The Non-Parametric Method (Kaplan-Meier Estimator)

The important assumption of the Kaplan-Meier (KM) survival function is that the distribution of censoring times is independent of the exact survival times. The estimated survival probabilities are computed using a product limit formula with the log-rank test used to compare two or more survival groups and the variance estimated using the Greenwood's formula. The test statistic for the log-rank test is a bit complicated mathematically especially comparing more than two groups so usually a computer software is used. We present a brief method of estimating KM survival function. We refer interested reader in log-rank test and variances to [14] and [27].

3.3.1 Estimating Kaplan-Meier Estimator

Consider the conditional probability:

$$\begin{aligned} P(A_1|A_2) &= \frac{P(A_1 \cap A_2)}{P(A_2)} \\ &= \frac{A_1 \cdot P(A_2|A_1)}{A(2)}, P(A_2) \geq 0 \end{aligned} \tag{17}$$

The general multiplication rule for event A_1 , A_2 and A_3 :

$$P(A_1 \cap A_2 \cap A_3) = P(A_1) \cdot P(A_2|A_3) \cdot P(A_3|A_1 \cap A_2) \tag{18}$$

But $P(T \geq a_1) = 1 - P(T < a_1)$

If $S(t)$ is the survival time such that $a_k \leq t \leq a_{k+1}, t \in (a_k, a_{k+1})$ then,

$$\begin{aligned}
S(t) &= P(T \geq a_{k+1}) \\
&= P(T \geq a_1, T \geq a_2, \dots, T \geq a_{k+1}) \\
&= P(T \geq a_1) \prod_{j=1}^k P(T \geq a_{j+1} | T > a_j) \\
&= P(T \geq a_1) [P(T \geq a_2 | T \geq a_1) \cdot P(T \geq a_3 | T \geq a_2), \dots, P(T \geq a_{k+1} | T \geq a_k)] \\
&= \prod_{j=1}^k P(1 - P(T = a_j | T \geq a_j)) \\
&= \prod_{j=1}^k P(1 - \lambda_j)
\end{aligned} \tag{19}$$

Kaplan Meier estimator of the survival function at time t , for $a_k \leq t \leq a_{k+1}$ is given by:

$$S(t) = \prod_{j=1}^k \left(1 - \frac{d_j}{r_j}\right) \tag{20}$$

Where

$$\begin{aligned}
\lambda_j &= P(T = a_j | T \geq a_j) \\
&= \frac{d_j}{r_j}
\end{aligned} \tag{21}$$

d_j is the number of individual who fail in the j th interval

r_j is the number at risk right before the j^{th} failure time and beyond (including censored observation)

3.4 Semi- Parametric Model

3.4.1 Cox Proportional Hazard Model (Cox PHM)

Cox emphasize more on the hazard function than the survival time.

Suppose that the data taken on n subject are denoted by (t_i, δ_i, Z_i) , where t_i is the failure time of the i th subject, δ_i is the failure indicator such that for the i th subject $\delta_i = 1$ if a subject is observed to failure and $\delta_i = 0$ otherwise and $Z_i = (Z_{i1}, \dots, Z_{ip})$ is p-dimensional vector of covariate. We can estimate the parameters β which reflect the effects of treatment and other covariates without having to make any assumptions about the form of $\lambda_o(t)$. In other words, we don't have to assume that $\lambda_o(t)$ follows an exponential model, or a Weibull model, or any other particular parametric model.

Let

$$\lambda(t, Z) = \lambda_o \psi(Z) \tag{22}$$

We can write the second term as $\psi(Z) = \exp(\beta Z)$

Suppose $Z = 1$ for treatment subject and $Z = 0$ for untreatment subject.

Then this model compares increase by a factor $\exp(\beta)$ for treatment subject verses untreatment subject. The Cox proportional hazard model assume that the hazard function $\lambda(t)$ for the i th subject with covariate value Z_i has the following form

$$\lambda(t, Z) = \lambda_0(t) \exp(\beta' Z) \tag{23}$$

Where $\lambda_0(t)$ is an unspecified baseline hazard function reflecting the underlying hazard for subjects with all covariate Z_1, Z_2, \dots, Z_i equal 0. (i.e reference group) and β is a p -vector of unknown regression coefficient. The model 3.23 is referred as a semi-parametric hazard function because the dependence function $\exp(\beta'Z)$ can be modeled explicitly but no specific probability distribution is assumed for the survival time.

For illustration purpose, consider a categorical variable with two levels (say $X= 1$, for treatment group and $X= 0$ for placebo), then the hazard ratio (HR), for th two groups is define as

$$HR = \frac{h(t|X = 1)}{h(t|X = 0)} = \exp(\beta). \quad (24)$$

When $HR = 1$, then the individuals in the two categories are at the same risk of getting the event, $HR > 1$, the individuals in the first group ($X=1$) are at higher risk of getting the event and if $HR < 1$, the individuals in the second group ($X= 0$) are at a high risk of getting the event.

3.4.2 Method of Estimation

Suppose X_1, \dots, X_n be n possible right censored failure times and Z_1, \dots, Z_n be the corresponding covariate vectors where Z_i is observed on $[0, X_i]$. Let the censoring be assumed to be non informative. Then under the Cox proportional hazard model (Cox PHM), the estimation of the regression coefficients β can be found by using the partial likelihood estimation of Cox model which is a product over the set of observed death times of the conditional probability of seeing the observing death, given the set of individual at risk at those times.

At each death time Γ_j , the contribution to the likelihood is

$$L_\beta = \frac{P(\text{individual } j \text{ fails} | \text{at risk at } \Gamma_j)}{\sum_{\rho \in R(\Gamma_j)} P(\text{individual } \rho \text{ fails} | \text{at risk at } \Gamma_j)}$$

$$= \frac{\lambda(\Gamma_j, Z_j)}{\sum_{\rho \in R(\Gamma_j)} \lambda(\Gamma_j), Z_j}$$

Under the PH assumption : (25)

$\lambda(t, z) = \lambda_o(t) \exp \beta Z$ so we get

$$L^{partial}(\beta) = \prod_{j=1}^k \frac{\lambda_o(\Gamma_j) \exp \beta Z}{\sum_{\rho \in R(\Gamma_j)} \lambda_o(\Gamma_j) \exp \beta Z_i}$$

$$= \prod_{j=1}^k \frac{\exp \beta Z}{\sum_{\rho \in R(\Gamma_j)} \exp \beta Z_i}$$

Therefore given a sets of covariates $Z_i = (Z_{1i}, Z_{2i}, \dots, Z_{pi})$ we can model their hazard rates as some multiple of the baseline hazard rate

$$\lambda_i(t, Z_i) = \lambda_o(t) \exp(\beta_1 Z_{1i}, \beta_1 Z_{2i}, \dots, \beta_1 Z_{pi})$$

We can write the hazard ratio for the i^{th} individual to the reference group as

$$\log\left(\frac{\lambda_i(t)}{\lambda_o(t)}\right) = \beta_1 Z_1 + \beta_2 Z_2 + \dots + \beta_p Z_p \tag{26}$$

The Cox PHM is a linear model for the log of the hazard ratio.

3.4.3 Assumption of Cox PH

- The baseline hazard depends on time t , but not on the covariates.
- The hazard ratio, i.e., $\exp(\beta Z)$, depends on the covariates but not on time t .

3.4.4 Checking Cox PH Assumption

There are several options for checking the assumption of proportional hazards:

1. Graphical Method

- Plot of the survival estimates for two subgroup
- Plots of $\log[-\log(S)]$ versus $\log(t)$ for two subgroups.
- Plots of observed survival probabilities versus expected under PH model

2. Use of goodness of fit test

We can construct a goodness-of-fit test based on comparing the observed survival probability with the expected under the assumption of proportional hazards.

3. Including Interaction terms between a covariate and time t

Time dependent covariates. If the covariate is continuous, break into categories.

3.4.5 What to do when Cox PH assumption is rejected

- Examine the structural form of the predictors in the model. Perhaps they have a more complicated relationship with the log hazard.
- Repeat the analysis by stratifying on the exposure variable. If there are no other predictors of interest then do not fit any model, just obtain Kaplan-Meier curves for each exposure group separately. If there are additional control variables fit a Cox model in which the levels of the exposure variable are treated as strata.
- Start the analysis at a time when the PH assumption appears to hold, and use a Cox PH model for only those individuals that survived that long.

- Fit a modified Cox model that includes a time-dependent variable that incorporates the interaction of exposure with time. Such a model is called an extended Cox model.
- In the worst scenario, switch to a parametric model.

3.5 Parametric Survival Model

3.5.1 Accelerated Failure - Time Models

The parametric models use the Accelerated Failure Time (AFT) model for estimating univariate survival censored data. In the AFT model more attention is given to the survival time than to hazard function. The AFT model means that the survival function of an individual with covariate X at time t is the same survival function of an individual with a baseline survival function at a time $\exp(\beta^t X)t$ where β^t is a vector of regression coefficients. It means

$$S(t|X) = S_0[\exp(\beta^t X)t] \text{ for all } t \quad (27)$$

where $\exp(\beta^t X)$ is called an acceleration factor. The acceleration factor allows to evaluate the effect of predictor variables on survival time. That is, how a change in covariate alters the time scale from the baseline time scale. It also implies that the median time to event with covariate X is the baseline median time to event divided by its acceleration factor. Similarly, the hazard function is also calculated. It is related to the baseline hazard rate.

$$\lambda(t|X) = \lambda_0[\exp(\beta^t X)t] \exp(\beta^t X), \text{ for all } t \quad (28)$$

In general, for linear regression model, the response variable is given by

$$T_i = \beta' X_i + \epsilon_i \quad (29)$$

If the response variable T_i in equation (29) is now replace by $\log T_i$, the resultant model is called the AFT model. That is:

$$\log T_i = \beta' X_i + \epsilon_i \quad (30)$$

Where X_i 's are the independent variable's (covariate's),

β' are constants to be determined and

ϵ_i is the error term.

The role of the covariates in the above equation is to accelerate (or decelerate) the time to failure. The error term ϵ_i , is assumed to be independent and identical distributed with mean zero. The various choice of ϵ_i distribution lead to the regression version of different parametric survival models. For the log transformation of T, its main purpose is to ensure that predicted values of survival time T, are positive, regardless of the values of the x's and the β 's. Equation (30) is referred to as the parametric model if the distribution of baseline hazards function is specified and if not; it is called a semi parametric model.

If a parametric model has just n scale parameter λ , that parameter is replaced by:

$$\begin{aligned} \lambda &= \exp(-\alpha_0 - \alpha_1 x_1 - \alpha_2 x_2 - \alpha_2 x_3 - \dots - \alpha_n x_n) \\ &= \exp(\alpha_o). \exp(\alpha_1 x_1). \exp\left(\frac{1}{p}\epsilon\right) \end{aligned} \quad (31)$$

The general form of the AFT model in terms of the survival time T is

$$T = \exp(-\alpha_o - \alpha_1 x - \dots, -\frac{1}{p}\epsilon) \quad (32)$$

AFT model is multiplicative in terms of T and additive in terms of $\log(T)$. For simplicity and interpretation, the time ratio (TR), which is the exponentiated regression coefficients ($\exp(\beta)$), is recommended to express in the same way hazard ratio (HR) is interpreted in the models of proportional hazards. When $TR > 1$ for a covariate indicates a slow down or prolongs the time to event and $TR < 1$ for a covariate suggests the occurrence of earlier event is more likely [12].

One assumption of the parametric model is that the survival time is assumed to follow a distribution with density function $f(t)$. Exponential, Weibull, Log normal, Log-logistic are the most familiar survival distributions. We discuss the various AFT models below.

3.5.2 The Exponential Distribution

The exponential distribution is a parametric distribution with only one parameter. In exponential distribution, hazard function is assumed to be constant over time. Consider the density function $f(t)$, given by:

$$f(t) = \lambda \exp(-\lambda t) \tag{33}$$

Then for eqn(3.3) above we have the survival function as:

$$\begin{aligned} S(t) &= \int_t^{\infty} f(u) du \\ &= \exp(-\lambda t) \end{aligned} \tag{34}$$

The above equation (34) gives the survival function for a survival data that follows an exponential equation.

To check if the exponential model is appropriate,

Take log of both side of equation (34):

$$\log S(t) = -\lambda t \quad (35)$$

A graph of Kaplan Meier survival estimate of $\log S(t)$ against time t , should produce approximately straight line with negative slope if the exponential assumption or model is valid.

The hazard function is calculated as:

$$\begin{aligned} \lambda(t) &= \frac{f(u)}{S(t)} \\ &= \lambda \end{aligned} \quad (36)$$

From equation(7), the cumulative hazard is estimated as:

$$\begin{aligned} \Lambda(t) &= -\log S(t) \\ &= \lambda t \end{aligned} \quad (37)$$

The exponential proportional hazard and acceleration failure time are the same model except that the parameterization is different.

3.5.3 The Weibull Distribution

The Weibull model is a parametric distribution with two parameters. The Weibull model is a slight modification of the exponential model. The model is obtain if the error term ϵ has an extreme value distribution with the following density: $f_\epsilon(x) = \exp(x - \exp^x)$, $-\infty < x < \infty$

Equivalently, T has the Weibull distribution with the following density:

$$f(t) = \lambda k (\lambda t)^{(k-1)} \exp(-(\lambda t)^k), t \geq 0.$$

Then from from equation (3) we have the survival function as:

$$\begin{aligned} S(t) &= \int_t^{\infty} f(u) du \\ &= \exp(-(\lambda t)^k) \end{aligned} \tag{38}$$

The above equation (38) gives the survival function for a survival data that follows a Weibull model. To check if the Weibull model is appropriate, take.

log of both side of equation (38):

$$\log S(t) = -(\lambda t)^k \tag{39}$$

$$\log(-\log S(t)) = k \log \lambda + k \log t \tag{40}$$

A straight line in the plot of $\log(-\log S(t))$ vs. $\log t$ indicates a Weibull model. We can use the above equation to check if the Weibull model is a reasonable choice for the survival time given a data set.

Alternatively, we can assume a Weibull model for the survival time and use the data to estimate $S(t)$ and plot this estimate against the Kaplan-Meier estimate as proposed for the exponential distribution. A (approximate) straight line indicates the Weibull model is a reasonable choice for the data.

The hazard function is given as:

$$\begin{aligned} \lambda(t) &= \frac{f(t)}{S(t)} \\ &= \lambda k (\lambda t)^{k-1} \end{aligned} \tag{41}$$

The cumulative hazard is estimated as:

$$\begin{aligned}\Lambda(t) &= -\log S(t) \\ &= (\lambda t)^k.\end{aligned}\tag{42}$$

where λ and k are the scale and shape parameter respectively. The Weibull distribution is convenient because of its simple form. it includes several hazard shapes:

$k = 1$ means constant hazard, $0 < k < 1$ implies decreasing hazard and $k > 1$ implies increasing hazard.

The Weibull distribution reduces to exponential distribution when $k = 1$.

The λ is a scale parameter in that the effect of different values of λ is just to change the scale on the horizontal (t) axis, not the basic shape of the graph. This model is very flexible and has been found to provide a good description of many types of time-to-event data.

We might expect an increasing Weibull hazard to be useful for modeling survival times of leukemia patients not responding to treatment, where the event of interest is death. As survival time increases for such a patient, and as the prognosis accordingly worsens, the patient's potential for dying of the disease also increases.

We might expect some decreasing Weibull hazard to well model the death times of patients recovering from surgery. The potential for dying after surgery usually decreases as the time after surgery increases, at least for a while.

3.5.4 Log-Logistic Distribution

The lifetime T is log-logistically distributed if $Y = \log(T)$ is logistically distributed with location parameter μ and scale parameter γ . Hence, Y is also of the form

$Y = \mu + \gamma Z$ where Z is a standard logistic random variable with density

$$\frac{\exp(z)}{(1 + \exp(z))^2}, -\infty < z < \infty$$

Equivalently, T has the log-logistic distribution with the following density:

$$f(t) = \lambda k (\lambda t)^{k-1} (1 + (\lambda t)^k)^{-2}, t \geq 0, k > 0, \lambda > 0 \quad (43)$$

We have the survival function as

$$\begin{aligned} S(t) &= \int_t^\infty f(u) du \\ &= \frac{1}{1 + (\lambda t)^k} \end{aligned} \quad (44)$$

Equation (3.44) can further be simplified as:

$$\frac{S(t)}{1 - S(t)} = (\lambda t)^{-k} \quad (45)$$

Taking log of both side of equation (45) gives:

$$-\log\left(\frac{S(t)}{1 - S(t)}\right) = k \log \lambda + k \log t \quad (46)$$

Similarly, a straight (approximately straight line) line in the plot of $\log t$ against $-\log\left(\frac{S(t)}{1 - S(t)}\right)$ indicates a log-logistics model. We can use the above equation to check if the log-logistics model is a reasonable choice for the survival time given a data set.

The hazard function is given as:

$$\begin{aligned}\lambda(t) &= \frac{f(t)}{S(t)} \\ &= \frac{\lambda k (\lambda t)^{k-1}}{1 + (\lambda t)^k}\end{aligned}\tag{47}$$

From equation (47) above, the cumulative hazard is estimated as.

$$\begin{aligned}\lambda &= -\log S(t) \\ &= -\log\left[\frac{1}{1 + (\lambda t)^k}\right]\end{aligned}\tag{48}$$

The log-logistic distribution is also characterized by two parameters. The shape parameter k and the scale parameter λ .

The hazard function is identical to the Weibull hazard aside from the denominator factor $1 + (\lambda t)^k$.

For $\alpha < 1$, it is monotone decreasing from ∞ and is monotone decreasing if $\lambda = 1$

If $\alpha > 1$, the hazard increases from zero to a maximum at $t = \frac{(k-1)^{\frac{1}{k}}}{\lambda}$ and decreases toward zero thereafter.

3.5.5 Log-Normal Distribution

The Log-normal has two parameter, scale parameter σ and shape parameter Γ_m has probability density function

$$f(t) = \frac{1}{\sigma t \sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2}[\ln t - \ln \Gamma_m]^2\right), 0 < t < \infty\tag{49}$$

The survival function is given as:

$$S(t) = 1 - F(t) = 1 - \Phi\left[\frac{\ln t - \ln \Gamma_m}{\sigma}\right]\tag{50}$$

Where $F(t) = \int_0^t \frac{1}{\sigma x \sqrt{2\pi}} \exp(-\frac{1}{2\sigma^2})([\ln x - \ln \Gamma_m]) dx = \Phi[\frac{\ln t - \ln \Gamma_m}{\sigma}]$ and $\Phi(z)$ denotes the standard normal cumulative distribution function.

The hazard function is calculated as

$$\lambda(t) = \frac{f(t)}{S(t)} = \frac{[\frac{1}{t\sigma\sqrt{2\pi}} \exp(-\frac{[\ln t - \ln \Gamma_m]^2}{2\sigma^2})]}{1 - \Phi[\frac{\ln t - \ln \Gamma_m}{\sigma}]} \quad (51)$$

3.5.6 Problem Associated With AFT Model

The main reason of the unpopularity of accelerated failure time model is:

- It is complicated estimation process, even if the data set consists of small number of covariates.
- AFT is based on a parametric model, which may be difficult sometime to fit.

3.5.7 Models Selection

We fit various survival distribution functions to the data and visually compare how similar the survival functions are to the Kaplan-Meier estimate of the survival function. The graph of the distribution that is nearest to that of the Kaplan-Meier estimate is the best model.

Another is to use either Bayesian information criterion (BIC) or Schwarz criterion (also SBC, SBIC) or Akaike Information Criterion (AIC) to choose the closest model.

For example, the AIC is give as, $-2\text{Log}(\text{maximum likelihood}) + k \times p$, where p is number of parameters in each model under consideration and k a predetermined constant. For this study, we use the AIC to select the best model. In some case, we use both method to select the model. Finally, we compare the standard variability of

the covariates for each model.

3.6 Simulation Study

First, we perform a simulation study where we generate a purely parametric survival model for each of the commonly used survival analysis such as Weibull, Exponential and so on. We fit the simulated data for all the parametric and semi-parametric model and compare the results by examining the average AIC (\overline{AIC}) and average BIC (\overline{BIC}). We compare the results to the results obtained by [2].

Next, we simulate a purely semi-parametric model and fit the data with all the well known parametric models. We expect that regardless of the best parametric model selected by \overline{AIC} and \overline{BIC} , its variance will be greater than that of the semi-parametric model. Finally we compare all the variances and \overline{AIC} and \overline{BIC} the simulated data and relevant conclusion made. For time convenience, we simulate randomly 1000 different data samples, 500 different data samples and 50 different data samples for each of small size, medium size and large size respectively and the averages of the outputs (coefficient, p-values, standard error, AIC, BIC, test statistic) calculated for comparison.

3.7 Real Data

We use any arbitrary clinical data and fit it with all the semi-parametric model and parametric and compare the performance. This is done to compare the results of the simulation study and real clinical data. Finally, we provide some steps to analyze real survival clinical data to fulfill research objective four.

3.8 Computer Software and Packages

Throughout this study, we use the R software which is an open source for the results analysis. [19] presented an appealing R-packages for both simulation studies of clinical data and its application to real clinical data. Thus, the analysis of the study is guide by their work.

4 ANALYSIS

4.1 Simulation Studies

We perform simulations to examine the performance of the techniques used in survival studies. Throughout this study, we simulate a right censored survival data which is commonly used in clinical studies. Many methods and algorithms have been proposed for simulating a right censored survival data, see for example [5], [26], [29], [9] and [19]. Though the works of [9] and [19] are available (with code in R), the former focuses more on generating right censored data under the Cox model but the latter focuses on generating parametric survival models for clinical data. Thus in this study, we simulate data according to the procedure in the work of the latter.

4.1.1 Simulation Method and Sample Size

Most statistical studies define small sample size as any sample less than 30 and any samples 30 and above as large. In the case of survival analysis, how large or small samples must be is not well investigated. However, in the master's thesis of [22], the following samples ($n=20, 100, 1000$) were defined as small, moderate and large for simulation studies to examine accuracies. In this study, we simulate any sample less than 30 as small samples, between 30 and 5000 as medium sample and above 5000 as large samples. That is, $n < 30$ as small samples, $30 \leq n \leq 5000$ as medium sample and $n \geq 5000$ as large samples.

In each of the models (parametric and semi-parametric), we investigate the performance for small sample data, medium sample data and large sample data. For

example, in exponential model, we examine performance by independently simulating small and large samples where the underlying distribution follows an exponential distribution. We highlight the R codes by [19] as follows: The function is `simple.surv.sim(n, foltime, dist.ev, anc.ev, beta0.ev, dist.cens = "weibull", anc.cens, beta0.cens, z = NA, beta = NA, x = NA)`

The description of the arguments are:

- `n`: Integer value indicating the desired size of the cohort to be simulated.
- `foltime`: Real number that indicates the maximum time of follow-up of the simulated cohort.
- `dist.ev`: Time to event distribution, with possible values “weibull”, for Weibull distribution, “lnorm” for log-normal distribution and “llogistic” for log-logistic distribution.
- `anc.ev`: Ancillary parameter for the time to event distribution.
- `beta0.ev`: β_0 parameter for the time to event distribution.
- `dist.cens`: String indicating the time to censoring distribution, with possible values “weibull” for Weibull distribution (this is the default value), “lnorm” for log-normal distribution and “llogistic” for log-logistic distribution. If no distribution is specified, the time to censoring is assumed to follow a Weibull distribution.
- `anc.cens`: Real number containing the ancillary parameter for the time to censoring distribution.

- **z:** Vector with three elements that contains information relative to a random effect used in order to introduce individual heterogeneity. The first element indicates the distribution, where “unif” stands for a uniform distribution, “gamma” stands for a gamma distribution, “exp” stands for an exponential distribution, “weibull” stands for a Weibull distribution and “invgauss” stands for an inverse Gaussian distribution. The second and third elements indicate the minimum and maximum in the case of a uniform distribution (both must be positive) and the parameters in the case of the rest of distributions. Note that just one parameter is needed in the case of the exponential distribution. Its default value is NULL, indicating that no individual heterogeneity is introduced.
- **beta:** List of vectors indicating the effect of the corresponding covariate. The number of vectors in beta must match the number of covariates, and the length of each vector must match the number of events considered. Its default value is NA, indicating that no covariates are included.
- **x:** List of vectors indicating the distribution and parameters of any covariate that the user needs to introduce in the simulated cohort. The possible distributions are normal distribution (“normal”), uniform distribution (“unif”) and Bernoulli distribution (“bern”). Its default value is NA, indicating that no covariates are included. The number of vectors in x must match the number of vectors in beta. Each vector in x must contain the name of the distribution and the parameter(s), which are: the probability of success in the case of a Bernoulli distribution, the mean and the variance in the case of a normal distribution;

and the minimum and maximum in the case of a uniform distribution.

4.2 Simulation of Parametric Results

To study the performance of different methods of survival analysis of clinical data, we conducted series of simulations to assess the performance of the methods used in survival studies with varying sample cohort using the techniques described above. The simulation study is divided into two cases. First, we consider the case where the time to survival or event of patients do not depend on any covariate. Second, we consider the case where time to survival or event of patients depend on covariate.

4.2.1 Simulation of Exponential Survival Data without Covariate

We simulated small sample of 12, medium samples of 250 and large samples of 30000 cohort where the time to censoring of an event is assumed to follow an exponential distribution. Each of the small samples, medium samples, and large samples were generated 1000 times, 500 times, and 50 times respectively. For each iteration of the generated data set, the data set is fitted for all the methods. Finally, the averages of the output of the results of the fitted model are calculated. That is, the average values of the intercept, standard error, test statistic (z), p-value, AIC, and BIC were calculated. The results of the fitted methods for each of the samples are summarized in Table 1. Since the semi-parametric given by the Cox model explains the ratio of covariates, the intercept value is not relevant. Thus, much of the emphasis is placed on comparison of all the methods using two different model selection procedure: average AIC (\overline{AIC}) and average BIC (\overline{BIC}). From table 1, we could see that

the semi-parametric tend to perform well than all the parametric model for small and medium sample by comparing the results of the average AIC and average BIC values, where smaller (\overline{AIC}) and (\overline{BIC}) value indicates better performance. Since the underlying survival distribution follows the exponential distribution, the exponential distribution outperforms all the parametric distribution in terms of the (\overline{AIC}) and (\overline{BIC}) values.

However, when the samples is “extremely large” the semi-parametric model performs poorly among all the methods whiles the exponential distribution outperform all the methods being studied.

Table 1: Simulation of exponential Survival data (time to censorship follows exponential without covariate)

Model	sample(n)	$\overline{intercept}$	$\overline{Std.Error}$	\bar{z}	\bar{p}	\overline{AIC}	\overline{BIC}
Cox	12	N/A	N/A	N/A	N/A	33.37969	33.37969
	250	N/A	N/A	N/A	N/A	1876.401	1876.401
	30000	N/A	N/A	N/A	N/A	461630.6	461630.6
Exponential	12	5.894048	0.3233470	18.27889	7.2e-46	135.39336	135.87827
	250	5.837965	0.06957457	83.92166	0	2828.680	2832.202
	30000	5.839947	0.006351270	919.4935	0	339129.5	339137.8
Weibull	12	5.802771	0.4444749	14.19697	1.97e-10	134.73313	135.70294
	250	5.836036	0.07109874	82.34210	0	2829.636	2836.679
	30000	5.838373	0.006497584	898.5631	0	339128.8	339145.4
loglogistic	12	5.1744	0.5202828	10.93681	1.83e-06	135.41144	136.38125
	250	5.377179	0.08237585	65.53340	0	2849.124	2856.167
	30000	5.378736	0.007526099	714.6950	0	341475.9	341492.5
lognormal	12	5.092055	0.5395235	10.54511	7.03e-06	135.38614	136.35596
	250	5.291689	0.08875915	59.95347	0	2862.027	2869.070
	30000	5.293383	0.008122511	651.7205	0	343138.6	343155.3

4.2.2 Simulation of Weibull Survival Data without Covariate

We simulated small sample ($n = 20$) 1000 times, medium samples ($n = 1000$) 500 times and large samples ($n = 200000$) 50 times cohort where the time to censoring of an event is assumed to follow a Weibull distribution. The averages of the fitted methods for each of the samples of the test statistic, p-values, AIC and BIC are calculated and are summarized in Table 2. From table 2, we could see that the semi-parametric tend to perform well than all the parametric model for small and medium sample by comparing the results of the average AIC and average BIC values (\overline{AIC} and \overline{BIC}). Also since the underlying survival distribution follows the Weibull distribution, the Weibull distribution outperforms all the parametric distribution in terms of (\overline{AIC} and \overline{BIC}). However, when the sample is “extremely large” the semi-parametric model performs poorly among all the methods whiles the Weibull distribution outperforms all the methods being studied when the sample is “extremely large”.

4.2.3 Simulation of log-logistic Survival Data with Covariate

Again, samples of 15, 250 and 55000 patients for small, medium and large cohort respectively were produce. Each of the small samples, medium samples, and large samples were generated 1000 times, 500 times, and 50 times respectively. The averages of the intercepts, standard errors, test statistics (z-value), p-values, AICs, and BICs were calculated for all the iterated generated data. The time to censoring of an event is assumed to follow log-logistic survival distribution. The details of the results of the fitted methods for each of the samples are also summarized in Table 3.

Table 2: Simulation of Weibull Survival data (time to censorship follows Weibull without covariate)

Model	sample(n)	$\overline{intercept}$	$\overline{Std.Error}$	\bar{z}	\bar{p}	\overline{AIC}	\overline{BIC}
Cox	20	N/A	N/A	N/A	N/A	73.1845	73.1845
	1000	N/A	N/A	N/A	N/A	10009.63	10009.63
	200000	N/A	N/A	N/A	N/A	3760872	3760872
Exponential	20	5.930245	0.2458455	24.14801	5.15e-86	232.444	233.4396
	1000	5.960220	0.03466166	171.95806	0	11589.67	11594.57
	200000	5.964046	0.002452332	2431.990	0	2315973	2315984
Weibull	20	5.815542	0.3483787	17.53748	1.70e-19	229.9415	231.933
	1000	5.832879	0.05023059	116.21886	0	11389.67	11399.48
	200000	5.837833	0.003550825	1644.086	0	2275994	2276014
loglogistic	20	5.172823	0.4064085	13.50898	5.99e-11	231.06151	233.05297
	1000	5.188445	0.05859378	88.64547	0	11456.01	11465.83
	200000	5.193596	0.004141231	1254.127	0	2289114	2289134
lognormal	20	5.077692	0.4281959	12.75151	4.35e-09	231.40035	233.39181
	1000	5.075810	0.06358719	79.94075	0	11512.79	11522.61
	200000	5.081902	0.004497175	1130.031	0	2300710	2300730

The result indicates that the semi-parametric performs well than all the parametric model for small and medium samples when their average AIC's and average BIC's are compared. Again, the underlying survival distribution follows the log-logistic distribution and so performs well than all the parametric distribution given their \overline{AIC} and \overline{BIC} values. But, for large samples, the semi-parametric tends to perform poorly. Finally, for large samples, the performance favors the model which underlying data were generated which is the log-logistic model.

4.2.4 Simulation of log-normal Survival Data without Covariate

We simulate series of different samples of 21, 850 and 90877 patients for small, medium and large cohort respectively where the underlying survival is assumed to

Table 3: Simulation of log-logistic Survival data (time to censorship follows log-logistic without covariate)

Model	sample(n)	$\overline{intercept}$	$\overline{Std.Error}$	\bar{z}	\bar{p}	\overline{AIC}	\overline{BIC}
Cox	15	N/A	N/A	N/A	N/A	41.15444	41.15444
	250	N/A	N/A	N/A	N/A	1684.184	1684.184
	55000	N/A	N/A	N/A	N/A	809659.1	809659.1
Exponential	15	6.386107	0.3026980	21.18674	8.77e-64	165.66675	166.37480
	250	6.421266	0.07352000	87.35931	0	2750.116	2753.638
	55000	6.422400	0.004953705	1296.485	0	604946.5	604955.4
Weibull	15	6.374691	0.3081751	22.18989	7.14e-20	166.42417	167.84027
	250	6.399375	0.08097077	79.25752	0	2748.371	2755.414
	55000	6.399589	0.005482754	1167.233	0	604276.2	604294.1
loglogistic	15	5.853048	0.3270505	19.15606	1.23e-16	165.65946	167.07556
	250	5.838385	0.08242404	71.06992	0	2726.360	2733.403
	55000	5.836116	0.005565791	1048.582	0	599314.4	599332.2
lognormal	15	5.850588	0.3328321	18.89259	3.95e-15	165.53615	166.95225
	250	5.833797	0.08570682	68.32991	0	2730.259	2737.302
	55000	5.831562	0.005794916	1006.337	0	600256.9	600274.8

follow log-normal distribution. Each samples were simulated 1000 times, 500 times, and 50 times for small, medium, and large samples respectively. Each generated set of sample is fitted for all the models and the averages of the results calculated. The result of the fitted models for each methods in Table 4 indicates that the semi-parametric again performs well than all the parametric model for small and medium sample when their $\overline{AIC}'s$ and $\overline{BIC}'s$ are considered. Also, the underlying survival distribution follows the log-normal distribution and so performs well than all the parametric distribution for small and medium samples given their \overline{AIC} and \overline{BIC} values. However, for large samples, the semi-parametric tends to perform poorly. Finally, for large samples, the performance favors the model which underling data were generated which is the log-normal model.

Table 4: Simulation of log-normal Survival data (time to censorship follows log-normal without covariate)

Model	sample(n)	$\overline{intercept}$	$\overline{Std.Error}$	\bar{z}	\bar{p}	\overline{AIC}	\overline{BIC}
Cox	21	N/A	N/A	N/A	N/A	70.62906	79.09117
	850	N/A	N/A	N/A	N/A	8523.776	8523.776
	90877	N/A	N/A	N/A	N/A	1651919	1651919
Exponential	21	6.105117	0.2345773	26.05894	3.07e-114	209.43587	262.47595
	850	6.118625	0.03675770	166.4634	0	10540.540	10545.285
	90877	6.116722	0.003552734	1721.695	0	1127680	1127689
Weibull	21	6.162951	0.1513406	42.38768	4.91e-93	205.25894	258.87848
	850	6.182724	0.02547889	242.9230	0	10361.740	10371.230
	90877	6.181315	0.002470627	2501.947	0	1108648	1108667
loglogistic	21	5.833951	0.1552328	38.89667	1.05e-108	205.48083	257.33907
	850	5.836889	0.02505478	233.1387	0	10264.828	10274.318
	90877	5.835077	0.002423438	2407.788	0	1098103	1098122
lognormal	21	5.833981	0.1547295	38.91167	9.25e-112	206.75782	256.92217
	850	5.837679	0.02508462	232.8850	0	10251.694	10261.184
	90877	5.835743	0.002427653	2403.881	0	1096746	1096765

4.2.5 Simulation of Cox Survival Data without Covariate

Simulations of a series of different samples of 10, 150 and 100000 patients for small, medium and large cohort respectively where the underlying survival is assumed to follow Cox model. Patient were followed up for 360 days for each sample and 35% of the sample were censored for each sample generated. The iteration for each sample size is similar to what was described in the previous simulations. Table 5 of the fitted results indicates that the semi-parametric again performs well than all the parametric model for small and medium sample when their \overline{AIC}' and $\overline{BIC}'s$ are considered. Even for large samples where the underlying generated survival follows Cox distribution, the Cox still performs poorly with very high \overline{AIC} and \overline{BIC} values for all the models. Thus for large samples the performance tend to favor parametric models.

Table 5: Simulation of Cox survival data (time to censorship follows Cox without covariate)

Model	sample(n)	$\overline{intercept}$	$\overline{Std.Error}$	\bar{z}	\bar{p}	\overline{AIC}	\overline{BIC}
Cox	10	N/A	N/A	N/A	N/A	19.85896	19.85896
	150	N/A	N/A	N/A	N/A	794.7005	794.7005
	100000	N/A	N/A	N/A	N/A	383901.1	383901.1
Exponential	10	5.634211	0.4031290	14.14542	6.6e-19	87.10755	97.31362
	150	5.621197	0.10140854	55.46912	0.0e+00	1292.8059	1303.0120
	100000	1.563474	0.001098	398.5763	0	239613.0	239612.4
Weibull	10	5.528904	0.2701993	42.56078	6.2e-06	83.66731	104.07946
	150	5.535819	0.06577372	107.09468	1.03e-183	1247.0264	1267.4385
	100000	1.543658	0.0007478	686.0517	0	232677.2	232676.1
log-logistic	10	5.268704	0.3302799	34.37519	8.2e-06	84.60531	105.01745
	150	5.296869	0.08276830	83.36597	2.07e-82	1265.7037	1286.1159
	100000	1.474360	0.000967	511.0536	0	236523.1	236522.0
log-normal	10	5.219795	0.3583867	31.58276	1.1e-05	84.76526	105.17741
	150	5.240754	0.09919346	65.60072	4.62e-80	1283.1576	1303.5698
	100000	1.452852	0.001088	437.9845	0	237919.4	237918.2

4.2.6 Simulated Data with Covariate

In the study of survival of clinical data, almost in all cases, certain factors (covariate) contribute to the survival of patients. Thus, in this simulation study, we include two covariates and compare the performance of the methods under consideration. For illustration purpose, we simulate one continuous covariate (say age of the patient, X_2) and a binary covariate (say treatments, X_1 : 0 for those receiving placebo, and 1 for those in control treatment). The binary covariate could even be gender (male and female) depending on the researcher's interest of covariate. Throughout the simulation, X_2 is a continuous covariate and X_1 is a binary covariate.

Also, for the case of the binary covariate, we simulate that equal number of patients must be in each group in all cases. For clarity, we simulate that 50% of the patient must be in group 0 and the remaining 50% in group 1 of the binary covariate.

Finally, distribution of covariate generated are as follows. The continuous covariate X_1 is generated from a uniform distribution as $X_1 \sim Unif(0.7, 1.3)$. The binary covariate X_1 is generated from Bernoulli distribution as $X_1 \sim Bern(0.5)$.

4.2.7 Simulation of Exponential Survival Data with Covariate

We generate samples of 10, 250 and 45000 for small, medium and large samples respectively under an exponential distribution. The cohort were followed up for 360 days, 360 days and 150 days for small, medium and large samples. The small samples, medium samples, and large samples were generated 1000 times, 500 times, and 50 times respectively for each of the sample type. Each of the generated samples were each fitted for all the methods in this study and the averages of the results summarized in Table 6. It can be observed that for small and medium samples, performance favors the semi-parametric model (small \overline{AIC} and \overline{BIC} values). For large samples the performance of the semi-parametric model is poor (high \overline{AIC} and \overline{BIC} values) among all the models.

We mush emphasis that since the samples were generated in a known distribution, it is obvious that it parameter estimates and performance will be better than other fully parametric models. In other words, one could simply fit the simulated samples with the semi-parametric and only one parametric (i.e, the parametric model under which the samples were simulated)and compare performance. Hence, it is more useful to compare the performance of parametric distribution model under which samples were generated with the performance of the semi-parametric model.

Thus, to avoid monotonous in our interpretation, much of the comparative analysis

will be made between the semi-parametric model and the parametric model which samples were generated in the rest of the analysis.

Table 6: Simulation of Exponential survival data (time to censorship with covariate)

Model	sample	covariate	\overline{Value}	$\overline{Std.Error}$	\bar{z}	\bar{p}	\overline{AIC}	\overline{BIC}
			Cox	10	X_1	0.059388	37.665206	0.01258957
		X_2	0.0080765	0.1169624	0.002991014	0.01701126		
	250	X_1	0.3925214	0.1575598	2.484398	0.07177409	1640.220	1647.263
		X_2	0.0254968	0.4516806	0.05670527	0.4973323		
	45000	X_1	0.401465	0.01506086	26.65516	3.451791e-132	376161.2	376178.6
		X_2	0.006848	0.04294151	0.1593177	0.4334652		
Exponential	10	b_0	0.237375	15.541808	0.06759053	0.007452858		
		X_1	-0.042988	15.460610	-0.01993349	0.01400709	3.221776	3.253548
		X_2	-0.0151809	0.1170535	-0.003926755	0.01618880		
	250	b_0	5.855663	0.4662019	12.572572	4.679135e-23		
		X_1	-0.39376	0.1570137	-2.502219	0.07070395	2196.986	2207.550
		X_2	-0.026629	0.4512109	-0.05945475	0.4972441		
	45000	b_0	5.83984	0.04443916	131.4124	0		
		X_1	-0.40168	0.01505380	-26.68209	1.902403e-132	238525.6	238551.7
		X_2	-0.0068767	0.04294144	-0.1599885	0.4333185		
Weibull	10	b_0	0.248799	16.263680	0.06409116	0.008345858		
		X_1	-0.043485	16.156253	-0.01936511	0.01497723	3.218572	3.260934
		X_2	-0.0255928	0.1500695	-0.002946753	0.01605309		
	250	b_0	5.856456	0.4673411	12.602111	1.442533e-18		
		X_1	-0.3936983	0.1582113	-2.484287	0.07106828	2197.981	2212.067
		X_2	-0.0268666	0.4519497	-0.05861969	0.4964511		
	45000	b_0	5.842988	0.04508371	129.6077	0		
		X_1	-0.40278	0.01529061	-26.34092	1.294019e-129	238526.4	238561.2
		X_2	-0.0069368	0.04308059	-0.1596136	0.4334480		
log-logistic	10	b_0	0.236438	8.207400	0.05927121	0.008422555		
		X_1	-0.039033	8.092092	-0.01681612	0.01449137	3.225656	3.268018
		X_2	-0.041383	0.1621016	-0.009605188	0.01505491		
	250	b_0	5.442316	0.5410118	10.118816	3.162461e-12		
		X_1	-0.419574	0.1818082	-2.311889	0.09050745	2203.030	2217.116
		X_2	-0.0157177	0.5245480	-0.03107060	0.5165724		
	45000	b_0	5.550196	0.04945685	112.2262	0		
		X_1	-0.434446	0.01667724	-26.05034	2.176905e-125	238650.9	238685.7
		X_2	-0.007031	0.04761656	-0.1463851	0.4368622		
log-normal	10	b_0	0.228019	2.485063	0.05798123	0.008495297		
		X_1	-0.036658	2.370384	-0.01663126	0.01414176	3.223758	3.266120
		X_2	-0.0376628	0.1631550	-0.008718365	0.01499554		
	250	b_0	5.419017	0.5928152	9.209212	8.334583e-10		
		X_1	-0.44644	0.1991860	-2.246301	0.10534848	2214.172	2228.258
		X_2	-0.0062	0.5745877	-0.01673131	0.5134795		
	45000	b_0	5.67346	0.05633588	100.7119	0		
		X_1	-0.47334	0.01890206	-25.04305	5.451927e-117	239658.7	239693.5
		X_2	-0.0091616	0.05415234	-0.1677195	0.4336860		

4.2.8 Simulation of Weibull Survival Data with Covariate

We simulate samples of 10, 1000 and 200000 for small, medium and samples under the distribution model. The cohorts of each samples were follow up for a period of 360 days and each sample type simulated 1000 times, 500 times and 50 times for small, medium, and large sample respectively. The simulated samples were each fitted for each model and the averages of the results are summarized in Table 7. Comparing the \overline{AIC} and \overline{BIC} values, the semi-parametric model tend to perform well for small and medium samples. However, for very large samples, the semi-parametric model performs poorly among all the models since it has both high \overline{AIC} and \overline{BIC} values.

Table 7: Simulation of Weibull survival data (time to censorship with covariate)

Model	sample	covariate	\overline{Value}	$\overline{Std.Error}$	\bar{z}	\bar{p}	\overline{AIC}	\overline{BIC}
Cox	10	X_1	0.0396	50.178426	0.03332747	0.04536250	2.509183	2.566675
		X_2	0.045357	0.3101453	0.006956623	0.04482158		
	1000	X_1	0.273027	0.07792900	3.501730	0.01189620	8508.132	8517.947
		X_2	0.014356	0.2243463	0.06390703	0.5035667		
	200000	X_1	0.279118	0.005507000	50.68400	0	3106802	3106822
		X_2	0.00411	0.01583917	0.2595638	0.5237324		
Exponential	10	b_0	0.599343	0.3055827	0.1896213	0.01936460		
		X_1	-0.02581	14.583251	-0.04890304	0.04025647	8.579595	8.66583
		X_2	-0.060689	0.2994519	-0.012230750	0.03794964	2	
	1000	b_0	5.748624	0.2316635	24.82072	1.125740e-102		
		X_1	-0.305745	0.07784194	-3.926465	0.00792505	8746.762	8761.485
		X_2	-0.016651	0.2243627	-0.07414446	0.4674620		
	200000	b_0	5.742962	0.01634429	351.3747	0		
		X_1	-0.3120770	0.005501265	-56.72800	0	1746638	1746668
		X_2	-0.004469	0.01583951	-0.2822335	0.4858628		
Weibull	10	b_0	0.611206	0.3871473	0.1763026	0.02207182		
		X_1	-0.011929	28.856142	-0.04461817	0.04174504	8.600580	8.715562
		X_2	-0.062399	0.3783291	-0.011493813	0.04117687		
	1000	b_0	5.848815	0.3316986	17.65415	2.528717e-46		
		X_1	-0.391027	0.11175300	-3.494777	0.01180615	8620.648	8640.279
		X_2	-0.020958	0.3209838	-0.06463003	0.5029684		
	200000	b_0	5.842158	0.02341978	249.4558	0		
		X_1	-0.399644	0.007903925	-50.56274	0	1721117	1721158
		X_2	-0.005882	0.02267720	-0.2595496	0.5237207		
log-logistic	10	b_0	0.565613	0.4027479	0.1602734	0.02374943		
		X_1	-0.017981	23.960318	-0.04111905	0.03892699	8.613693	8.728675
		X_2	-0.072547	0.3951261	-0.014286017	0.04167630		
	1000	b_0	5.276027	0.3843186	13.74636	2.407410e-26		
		X_1	-0.423621	0.12919552	-3.276892	0.01768908	8638.123	8657.754
		X_2	-0.014541	0.3727170	-0.03873524	0.5104250		
	200000	b_0	5.27796	0.02716112	194.3222	0		
		X_1	-0.4317097	0.009138496	-47.24100	0	1724586	1724627
		X_2	-0.008002	0.02635224	-0.3040300	0.5273711		
log-normal	10	b_0	0.555313	0.4130427	0.1579265	0.02375455		
		X_1	-0.038287	7.397695	-0.04029724	0.03865069	8.600742	8.715724
		X_2	-0.060386	0.4037084	-0.013968284	0.04025346		
	1000	b_0	5.276027	0.3843186	13.74636	2.407410e-26		
		X_1	-0.423621	0.12919552	-3.276892	0.01768908	8684.567	8704.198
		X_2	-0.014541	0.3727170	-0.03873524	0.5104250		
	200000	b_0	5.260330	0.02999507	175.3755	0		
		X_1	-0.458007	0.010084970	-45.41519	0	1733950	1733991
		X_2	-0.009537	0.02907691	-0.3288029	0.4965488		

4.2.9 Simulation of log-logistic Survival Data with Covariate

We simulate a series of different samples of 20, 550 and 20000 patients for small, medium and large cohort respectively where the underlying survival is assumed to follow log-logistic distribution. Each generated set of samples were generated 1000 times, 500 times, and 50 times for small, medium, and large samples respectively and was fitted for all the models. The averages of the fitted samples for each model is also summarize in Table 8. For small and medium samples, the results of the semi-parametric and parametric log-logistic model are similar in terms of their average p-value ($\overline{p-value}$) (none of the covariate were significant at 5% level of significant). However, the semi-parametric provides a better performance for small and medium samples in terms of their \overline{AIC} and \overline{BIC} values. The results also indicates that for very large samples, covariate X_1 is significant for all the method but X_2 is not significant for any of the methods. Thus, the results do not differ in terms of the number of significant predictors. Also by comparing the values of \overline{AIC} and \overline{BIC} values, semi-parametric tend to perform poorly among all the models for large samples.

Table 8: Simulation of log-logistic survival data (time to censorship with covariate)

Model	sample	covariate	\overline{Value}	$\overline{Std.Error}$	\bar{z}	\bar{p}	\overline{AIC}	\overline{BIC}
			Cox	20	X_1	0.138287	248.19757	0.2614369
		X_2	0.000368	2.395963	-0.018828988	0.49118		
	550	X_1	0.487035	0.1875895	2.585173	0.064878	1474.888	1483.508
		X_2	0.003297	0.5294499	0.003957619	0.480203		
	20000	X_1	0.405298	0.01981992	20.44806	3.211139e-75	194801.0	194816.8
		X_2	-0.0112	0.05671206	-0.1982523	0.5222800		
Exponential	20	b_0	5.909656	59.18038	2.373453	0.11475		
		X_1	-0.173364	171.72863	-0.3034061	0.43979	112.01460	115.00180
		X_2	-0.007068	2.422518	0.017113612	0.440556		
	550	b_0	6.409939	0.5491633	11.67888	1.298e-20		
		X_1	-0.478161	0.1875405	-2.538584	0.0673807	1720.794	1733.723
		X_2	-0.002751	0.5293852	-0.002925319	0.48600		
	20000	b_0	6.225267	0.05858806	106.25615	0		
		X_1	-0.394317	0.01979715	-19.91698	7.575187e-71	145378.5	145402.2
		X_2	0.011096	0.05671265	0.1963765	0.5301931		
Weibull	20	b_0	6.814543	94.22768	1.540211	0.24439		
		X_1	-0.385897	230.61375	NaN	0.480595	108.72271	112.70564
		X_2	-0.002161	4.513716	0.016710200	0.48357		
	550	b_0	5.98373	0.4262679	14.13279	9.4174e-27		
		X_1	-0.3645209	0.1437305	-2.517728	0.065862	1711.686	1728.926
		X_2	-0.001941	0.3974758	-0.004027942	0.48032		
	20000	b_0	6.1396	0.05011332	122.52371	0		
		X_1	-0.34997	0.01702828	-20.54778	4.788047e-76	145057.3	145088.9
		X_2	0.00973	0.04834651	0.2011205	0.5178028		
log-logistic	20	b_0	6.18586	109.86290	1.327063	0.2986678		
		X_1	-0.399873	270.53292	-Inf	0.48047	108.58176	112.5646
		X_2	-0.052902	4.759786	0.009236182	0.480496		
	550	b_0	5.833189	0.4458001	13.16703	1.4175e-23		
		X_1	-0.3834139	0.1498249	-2.548083	0.06448824	1711.415	1728.655
		X_2	-0.002794	0.4204924	-0.005860332	0.4791404		
	20000	b_0	5.828024	0.05604131	104.00241	0		
		X_1	-0.39766	0.01887624	-21.06410	3.334302e-79	144880.4	144912.0
		X_2	0.007845	0.05427417	0.1444703	0.5334322		
log-normal	20	b_0	6.211266	46.79530	1.303389	0.3052368		
		X_1	-0.419369	91.47112	-0.2903933	0.4780203	108.39740	112.38033
		X_2	-0.0883192	4.933734	0.002704278	0.4748853		
	550	b_0	6.068170	0.5066934	12.05262	4.4043e-20		
		X_1	-0.424021	0.1677893	-2.524677	0.069854	1714.329	1731.569
		X_2	-0.002391	0.47693	-0.002978085	0.4756976		
	20000	b_0	5.853185	0.06029883	97.07686	0		
		X_1	-0.414857	0.02027311	-20.46037	1.151491e-73	145176.4	145208.0
		X_2	0.006618	0.05829062	0.1140256	0.5675334		

4.2.10 Simulation of log-normal Survival Data with Covariate

A simulation study of series of samples of size 21, 850 and 100000 for small, medium and large respectively was conducted for comparative performance of survival models method. The time to censoring of the samples were generated under the log-normal distribution. The follow up time for the sample are 360 days, 360 days and 300 days for small, medium and large samples respectively. the type of sample size were generated 1000 times, 500 times and 50 times for small, medium and large samples respectively. For each of the generated sample data, the data is fitted for all the methods under study and averages of the results calculated and summarized in Table 9. The results of the semi-parametric is similar to the results of the parametric simulated log-normal under study in terms of significant number of covariates for each type of sample size. When performances are compared by using the \overline{AIC} and \overline{BIC} values , the semi-parametric tends to perform better since it has the minimum \overline{AIC} and \overline{BIC} values for small and medium samples.

However, when the samples is very large, the results differ when \overline{AIC} and \overline{BIC} are compared.

Table 9: Simulation of log-normal survival data (time to censorship with covariate)

Model	sample	covariate	\overline{Value}	$\overline{Std.Error}$	\bar{z}	\bar{p}	\overline{AIC}	\overline{BIC}
			Cox	21	X_1	0.733287	27.742208	1.02725
		X_2	0.016421	1.8690216	0.015102	0.4809397		
	850	X_1	0.62989	0.089842	7.006612	3.190324e-07	6448.789	6458.280
		X_2	0.001612	0.2546217	0.005982	0.5200298		
	100000	X_1	0.67117	0.008997392	74.59536	0	1151509.5	1151528.5
		X_2	0.004558	0.02534807	0.179672	0.5104825		
Exponential	21	b_0	6.4222	30.350625	3.488097	0.014983251		
		X_1	-0.61013	29.118089	-0.877467	0.3935501	183.10829	186.2419
		X_2	-0.027857	1.7999060	-0.020866	0.5385295		
	850	b_0	6.307793	0.2634493	23.9492	7.740125e-104		
		X_1	-0.535069	0.089535	-5.972013	3.554397e-06	7221.937	7236.173
		X_2	-0.001113	0.2545625	-0.003871	0.5712404		
	100000	b_0	6.407502	0.02628869	243.737	0		
		X_1	-0.585787	0.008977863	-65.2473	0	731677.6	731706.2
		X_2	-0.004146	0.02534705	-0.163452	0.5560081		
Weibull	21	b_0	6.071413	17.520783	6.82095	0.002879047		
		X_1	-0.3795067	16.889717	-1.07762	0.3254981	177.66508	181.8432
		X_2	-0.007822	0.9228086	-0.01787	0.4549552		
	850	b_0	6.032028	0.1382701	43.694	0.00e+00		
		X_1	-0.337235	0.047784	-7.054729	2.794884e-07	7001.219	7020.200
		X_2	-0.000932	0.1332372	-0.007425	0.5094513		
	100000	b_0	5.995640	0.01307157	458.685	0		
		X_1	-0.3372	0.004556228	-74.0077	0	706861.8	706899.8
		X_2	-0.002248	0.01252263	-0.17882	0.5010833		
log-logistic	21	b_0	5.865219	12.870907	6.246614	0.002946945		
		X_1	-0.409601	12.204112	-1.184249	0.3089563	177.01406	181.1922
		X_2	-0.018099	0.9738970	-0.034531	0.4563818		
	850	b_0	5.827407	0.1531682	38.1017	1.908882e-243		
		X_1	-0.391265	0.05173026	-7.570256	1.898828e-08	6974.019	6993
		X_2	0.000842	0.1485039	0.0063715	0.5130259		
	100000	b_0	5.82241	0.01437375	405.078	0		
		X_1	-0.384998	0.004872704	-79.01075	0	704699.7	704737.7
		X_2	-0.003252	0.01388699	-0.233606	0.5018062		
log-normal	21	b_0	5.868668	4.554796	6.1931	0.002754313		
		X_1	-0.406912	3.880951	-1.203706	0.3126003	176.70719	180.8853
		X_2	-0.024051	0.9788660	-0.034284	0.4626586		
	850	b_0	5.836572	0.1563366	37.386	1.115407e-235		
		X_1	-0.399398	0.05279084	-7.571258	2.040593e-08	6964.787	6983.768
		X_2	-0.0013	0.15127	-0.00759	0.5172663		
	100000	b_0	5.839931	0.01488516	392.338	0		
		X_1	-0.39963	0.005035024	-79.36992	0	703728.5	703766.5
		X_2	-0.003867	0.01434987	-0.269091	0.5036786		

4.3 Simulation of Semi-Parametric Results

We use the function `sim.survdata` which depends on the package `coxed` to Simulate duration data for the Cox proportional hazards (semi-parametric) model. A call to the function is

```
sim.survdata(N = 1000, T = 100, type = "none", hazard.fun = NULL, num.data.frames =  
1, fixed.hazard = FALSE, knots = 8, spline = TRUE, X = NULL, beta = NULL, xvars =  
3, mu = 0, sd = 0.5, covariate = 1, low = 0, high = 1, compare = median, censor =  
0.1, censor.cond = FALSE)
```

The description of the argument is

- `N` : Number of observations in each generated data frame. Ignored if `X` is not `NULL`.
- `T`: The latest time point during which an observation may fail. Failures can occur as early as 1 and as late as `T`.
- `Type`: If “non” (the default) data are generated with no time-varying covariates or coefficients. If “`tv`”, data are generated with time-varying covariates, and if “`tvbeta`” data are generated with time-varying coefficients (see details).
- `hazard.fun`: A user-specified R function with one argument, representing time, that outputs the baseline hazard function. If `NULL`, a baseline hazard function is generated using the flexible-hazard method as described in [8] (see details).
- `num.data.frames`: The number of data frames to be generated.

- `fixed.hazard`: If TRUE, the same hazard function is used to generate each data frame. If FALSE (the default), different drawn hazard functions are used to generate each data frame. Ignored if `hazard.fun` is not NULL or if `num.data.frames` is 1
- `knots`: The number of points to draw while using the flexible-hazard method to generate hazard functions (default is 8). Ignored if `hazard.fun` is not NULL.
- `X`: A user-specified data frame containing the covariates that condition duration. If NULL, covariates are generated from normal distributions with means given by the `mu` argument and standard deviations given by the `sd` argument.
- `beta`: Either a user-specified vector containing the coefficients that for the linear part of the duration model, or a user specified matrix with rows equal to T for pre-specified time-varying coefficients. If NULL, coefficients are generated from normal distributions with means of 0 and standard deviations of 0.1.
- `xvars`: The number of covariates to generate. Ignored if `X` is not NULL.
- `mu`: If scalar, all covariates are generated to have means equal to this scalar. If a vector, it specifies the mean of each covariate separately, and it must be equal in length to `xvars`. Ignored if `X` is not NULL.
- `sd`: If scalar, all covariates are generated to have standard deviations equal to this scalar. If a vector, it specifies the standard deviation of each covariate separately, and it must be equal in length to `xvars`. Ignored if `X` is not NULL.

- covariate: Specification of the column number of the covariate in the X matrix for which to generate a simulated marginal effect (default is 1). The marginal effect is the difference in expected duration when the covariate is fixed at a high value and the expected duration when the covariate is fixed at a low value.
- low: The low value of the covariate for which to calculate a marginal effect.
- high: The high value of the covariate for which to calculate a marginal effect.
- compare: The statistic to employ when examining the two new vectors of expected durations (see details). The default is median.
- censor: The proportion of observations to designate as being right-censored.
- censor.cond Whether to make right-censoring conditional on the covariates (default is FALSE, but see details)

4.3.1 Simulation of Cox Proportional Hazard Survival Data with Covariate

In the case of the semi-parametric, we generate cohorts of 8, 1000 and 100000 for small, medium and large sample respectively where the time to censoring follows the Cox Proportional Hazard model. Each samples were followed up a period of 365 days with 35% of samples experiencing the event (death). The results of the average of the fitted models for each sample are also summarize in Table 10. The results of the fitted models are similar for all the methods in terms of the number of significant covariate for all the samples. However, for small and medium samples, the semi-parametric model performs better when the values of the \overline{AIC} and \overline{BIC} are compared. But for large samples, semi-parametric performs poor when the \overline{AIC} and \overline{BIC} are compared.

Table 10: Simulation of Cox survival data (time to censorship with covariate)

Model	sample	covariate	\overline{Value}	$\overline{Std.Error}$	\bar{z}	\bar{p}	\overline{AIC}	\overline{BIC}
			Cox	8	X_1	0.00391	0.0046236	7.5308e-05
		X_2	0.002126	0.002235	0.0016047	0.00094063		
	1000	X_1	0.00843	0.039554	0.21936	0.062683	7647.643	7643.643
		X_2	0.00091	0.039593	0.03236787	0.069607		
	100000	X_1	0.08389	0.003939	21.29	2e-16	1376806	1376825
		X_2	0.07665	0.003962	19.35	2e-16		
Exponential	8	b_0	0.01399	0.00230828	0.012125	1.3521e-11		
		X_1	-0.00291	0.0033780	7.0543e-05	0.00035836	0.14272196	0.14869636
		X_2	-0.00184	0.00171767	-0.001780	0.00087225		
	1000	b_0	5.7148	0.042059	136.023	0		
		X_1	-0.00776	0.039321	-0.197	0.06321495	8541.164	8535.164
		X_2	0.000794	0.039389	0.0122915	0.06519974		
	100000	b_0	5.63067	0.00392	1436.8	2e-16		
		X_1	-0.04353	0.00392	-11.1	2e-16	863504.1	863532.6
		X_2	-0.04085	0.00395	-10.3	2e-16		
Weibull	8	b_0	0.01257	0.00096973	0.02646	1.4905e-33		
		X_1	-0.00213	0.0012810	-1.2571e-03	0.00029967	0.13486288	0.14282874
		X_2	-0.000944	0.00082513	-0.0019159	0.0007318		
	1000	b_0	5.6031	0.02823	251.5916	0		
		X_1	-0.00511	0.026881	-0.21958	0.06853866	8227.314	8219.314
		X_2	0.00492729	0.026936	-0.00920692	0.06718532		
	100000	b_0	5.57014	0.00265	2102.2	2e-16		
		X_1	-0.03887	0.00264	-14.7	2e-16	852045.5	852083.5
		X_2	-0.03604	0.00266	-13.6	2e-16		
log-logistic	8	b_0	0.01255	0.00096615	0.026977	1.8843e-31		
		X_1	-0.00215	0.0013577	-1.4099e-03	0.0002485	0.13670952	0.14467537
		X_2	-0.00124	0.00084896	-0.002609	0.00041610		
	1000	b_0	5.3600	0.034215	203.7830	0		
		X_1	-0.00615	0.034836	-0.22063	0.06965971	8359.350	8351.350
		X_2	0.006799	0.034953	0.00565096	0.07258594		
	100000	b_0	5.37816	0.00338	1591.2	2e-16		
		X_1	-0.04904	0.00337	-14.6	2e-16	870532.3	870570.3
		X_2	-0.04649	0.00339	-13.7	2e-16		
log-normal	8	b_0	0.012575	0.00101086	0.02546	2.673e-31		
		X_1	-0.00203	0.0014379	-9.6347e-04	0.0002094	0.13589413	0.14385999
		X_2	-0.001357	0.00076809	-0.002886	0.0004698		
	1000	b_0	5.30853	0.040553	162.6022	0		
		X_1	-0.00694	0.040013	-0.1917	0.068778	8472.269	8464.269
		X_2	0.004669	0.040158	-0.000221	0.071131		
	100000	b_0	5.29659	0.00468	1132.8	2e-16		
		X_1	-0.06204	0.00455	-13.6	2e-16	888976.6	889014.6
		X_2	-0.05986	0.00458	-13.1	2e-16		

4.4 Data Application

4.4.1 Introduction

We apply the result in the simulation studies to a real clinical data set. In this section, we also provide some steps that could be used to analyze survival clinical data set. We start the analysis by visualizing the behavior of the data set through graphical approach using the Kaplan Meier non-parametric approach.

4.4.2 Description of Data Set

The data consist of 136 cervical cancer patients. Patients were followed up for a period of one year (365 days). The event is patients who died due to cervical cancer while those who were alive or lost to follow up are considered censored at the time of collecting the data. The following censoring assumptions on patient were made.

- Patients who were alive disease-free were considered censored.
- Patients who died from other causes than cervical cancer were considered censored.
- Patients who were still alive with disease as at their last follow up date were considered to be censored.
- Patients who were lost to follow-up were considered to have been censored.

The dependent variable is the time (days) until death of a patient.

The covariates considered in this study are age of the patients and treatments given to patient. Age is a continues covariate but the treatment is a discrete covariate with three levels. Patients diagnosed with only teletherapy are considered as group one. Patients diagnosed with combination of teletherapy and brachytherapy are considered

as group two. Patients diagnosed with combinations of teletherapy, brachytherapy and chemotherapy were considered as group three. The categorical covariate (treatment) were coded using i-1 dummy variables to avoid over parametrization as follows.

$$x_{i-1} = \begin{cases} 1 & \text{if patient is in group } i, \text{ for } i = 1, 2, 3 \\ 0 & \text{otherwise} \end{cases}$$

In this study, treatment group 3 was used as the baseline for the treatment covariate.

The table below shows a summary of the first four cases.

Table 11: First four cases for each treatments group

follow up time (days)	event	age	treatment 1	treatment 2
189	0	45	1	0
148	1	66	1	0
100	0	52	1	0
221	0	67	1	0

Event: 1 = if patient died, 0 = if patient alive (censored)

4.5 Preliminary Results

The cancer patients were diagnosed with three kinds of treatments at age 27 years and older. The mean age of the sample is 59.45588 years with standard deviation 14.38202 years and ages range from 27 years (minimum age) and 93 years (maximum age) at the beginning of the study.

Table 12: Descriptive Statistics of Samples

Age (years)	patients who died	patients who survived	Total
mean	70.625	58.75781	59.45588
standard deviation	14.12129	14.16091	14.38202
range	50 - 90	27 - 93	27 - 93
total (n)	8	128	136

4.6 Kaplan Meier Estimate

First, we analyze the data by looking at the trend of the survival of patient using Kaplan Meier (K-M) estimate. Figure 3 shows K-M graph that provides a general survival curve of patients. The graph is a step function that decreases monotonically (changes) at any time a patient experience the event (death). The graph gives the probability that a patient will survive at any given time (day). For instance, the probability that a patient will survive till day 300 is estimated to be 0.9 (from the graph). The estimated K-M survival probability at the end of the year (365 days) is about 0.85 from the graph. The graph also indicates that there were some censored observations (patient still alive at the end of 365 days) since the graph does not goes zero. Figure 4 provides the graph of the hazard of patients. It can be observe that the hazard of a patient increase as the days increase.

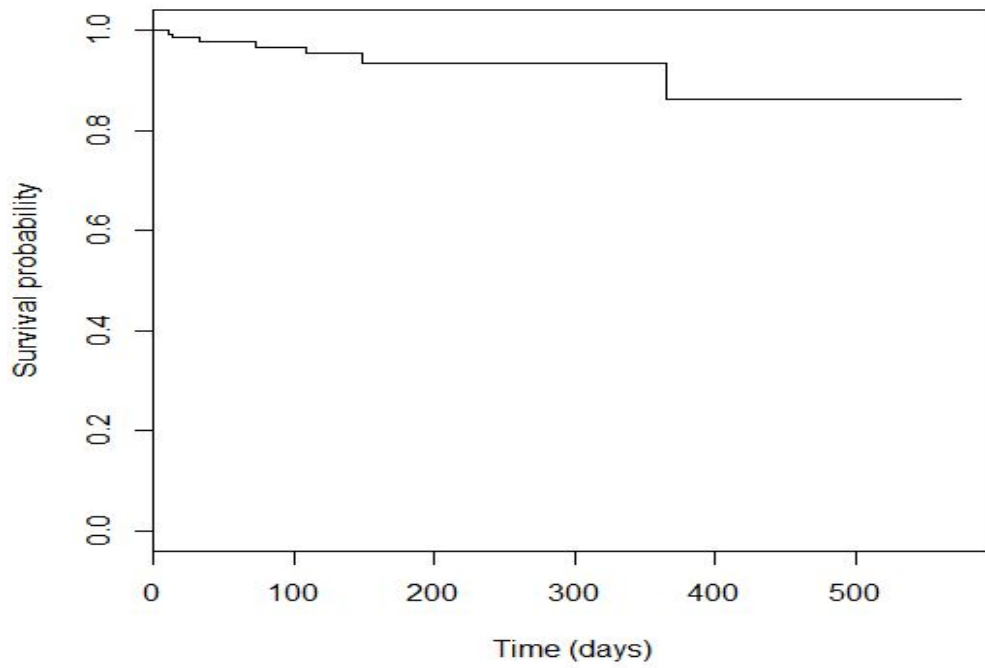


Figure 3: Kaplan Meier Graph for no covariate

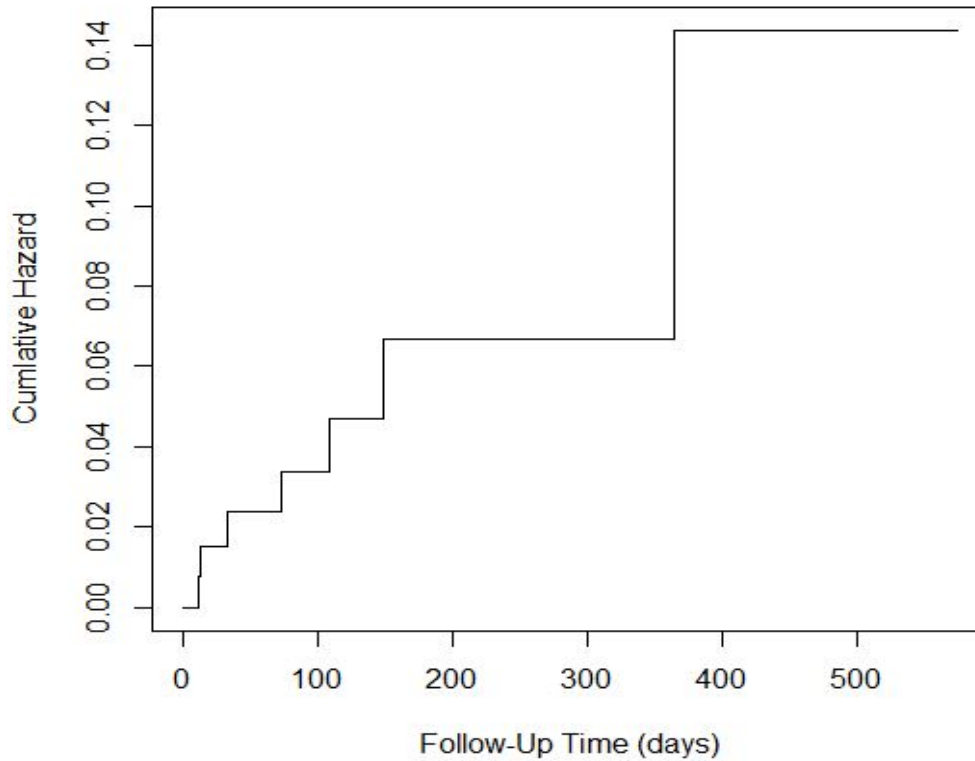


Figure 4: Kaplan-Meier Hazard Estimates

4.7 Comparing Survival Models

We fit a survival models to the data set for both semi-parametric model under the Cox Proportional Hazard Model (CPHM) assumption and the parametric survival model under the Accelerated Failure Time Model (AFTM). We compare the best model fit using the AIC and BIC. The results is summarized in Table 13. The result of the fitted model indicates that the semi-parametric model is a good fit since it has the smallest AIC (68.45273) and BIC(77.1907) values. Among the parametric case, exponential survival model is indicated a better fit under the AFTM. We should note that the CPHM is accurate and can be used for any inferences and prediction

provided the mild assumptions (assumption of proportionality hazard) is fully satisfied. Since the CPHM has been selected a better fit among all the model comparison, we now provide some steps to model and explain survival clinical data using the semi-parametric method. We start this by checking the selected model assumption.

Table 13: Fitted model comparison

Model	AIC	BIC
Cox	68.45273	77.1907
Exponential	145.1056	156.7562
Weibull	147.0642	161.6275
log-logistic	147.5194	162.0826
log-normal	148.1535	162.7168

Table 14: Summary of semi-parametric model

covariate	coefficient	exp(coef)	se(coef)	z	$Pr(> z)$	lower .95	upper .95
age	0.05962	1.06143	0.02720	2.192	0.0284	1.0063	1.120
treatment 1	0.44668	1.56311	0.92302	0.484	0.6284	0.2560	9.543
treatment 2	0.03533	1.03596	0.82639	0.043	0.9659	0.2051	5.233

4.8 Some Steps to Model and Explain Survival Clinical Data

The first step to model and explain survival clinical data under the semi-parametric model is to check model assumptions through model diagnostic.

4.8.1 Model Diagnostic/Adequacy

We assess the validity of the model to the data via Goodness-of-good fit. The diagnostic check is based on the residuals of the selected fitted model. Three kinds of model adequacy will be checked: assumptions of proportional hazard, influential/outlier observation and nonlinearity relationship between the log hazard and the

covariates.

4.8.2 Assessing Proportional Hazard (PH) Assumption

Two approaches will be used. First is the statistical approach (hypothesis testing) which tests the null hypothesis that proportional hazard assumption is met. This approach is done by estimating the test of the proportional hazards assumption for each of the covariate by correlation between the corresponding set of Schoenfeld’s residuals and survival time. Table 15 provides the results for the test. The values for the Rho shows the Pearson product moment correlation between the Schoenfeld’s residuals and lagged residuals for each covariate. The Chisq values are the test statistics. The “Global” gives the global test of probability for the model at once. The p-values give probability values under which the null hypothesis is tested. From the table, the p-values for each of the covariate suggests that PH assumption is met at $\alpha = 5\%$ significant level. Also the global test is also not statistically significant. Therefore, we can assume that proportional hazard is met.

Table 15: Test for Proportional Hazard Assumption

	rho	chisq	p
age	0.281	0.564	0.453
teletherapy	-0.183	0.231	0.631
telebrachy	-0.229	0.435	0.510
GLOBAL	NA	1.123	0.772

In Figure 5, we provide a graphical approach to check assumption of PH based on the Schoenfeld residual. The Schoenfeld residual are independent of time. A plot for each of the covariate that shows random pattern against time is evidence that

PH assumption is met. The solid line is a smoothing spline fit to the plot, with the dashed lines representing a ± 2 -standard-error band around the fit. A systematic departure from a horizontal line shows non-proportional hazards, since proportional hazards assumes that estimates of the covariates do not vary much over time. From the graphical inspection, each of the plotted shows a random scatter along with corresponding p-values for each covariate indicating that PH appears to be valid. Also, each p-value is not significant at 5% significant level showing that PH assumption is met. Finally, the p-value of the Global Schoenfeld test is 0.7715 indicating that overall, the PH assumption is reasonably satisfied.

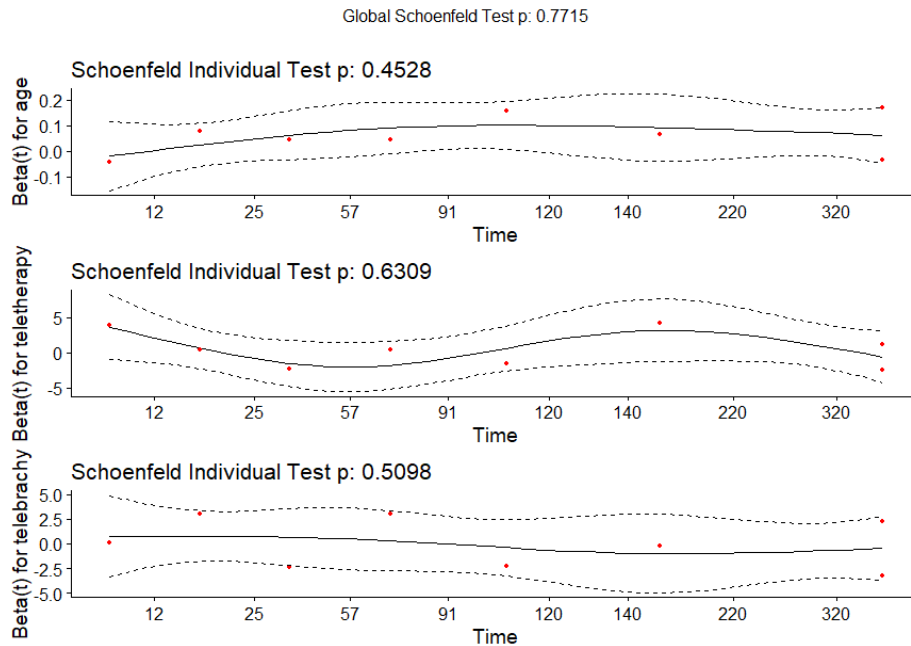


Figure 5: Schoenfeld residuals for covariate against time.

4.8.3 Checking Influential/Outlier Observation

Figure 6 shows a graphical display to check any influential/outlier observation. Influential observation is checked by comparing the magnitude of the largest $dfbeta$ values in the plotted graphs to each of the corresponding Cox regression coefficient. Any largest in magnitude of $dfbeta$ value greater than regression coefficient indicates influential observation. From the graph below, comparing the magnitude of the largest $dfbeta$ value to each of the corresponding regression coefficient indicates that none of the observations are influential.

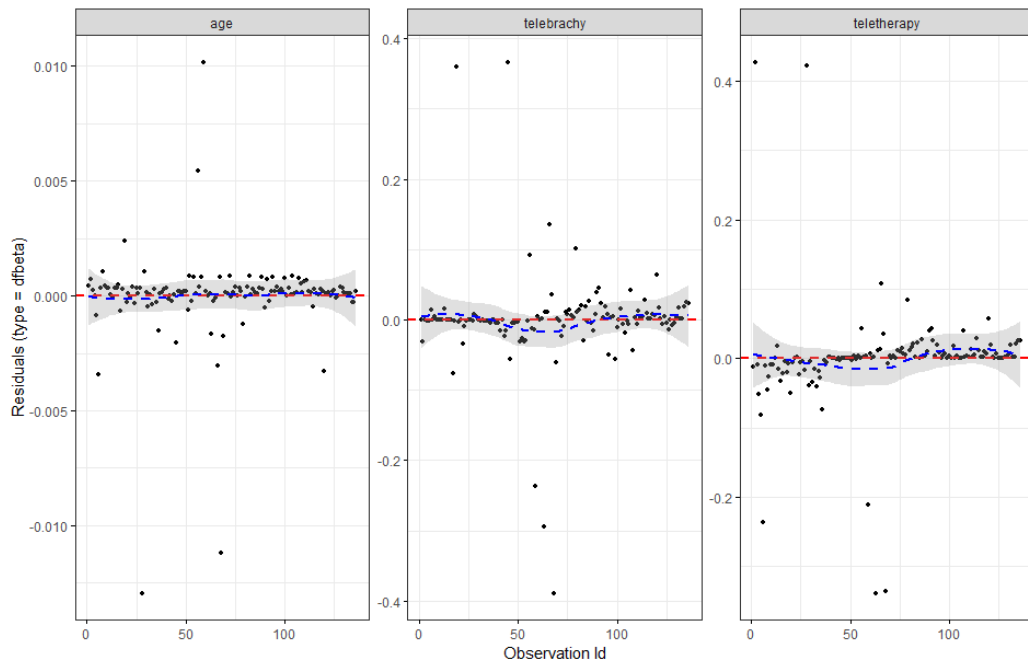


Figure 6: Index plots of $dfbeta$ for Cox regression of time to age and treatment.

4.8.4 Checking the Assumption of Nonlinearity

One assumption of the semi-parametric Cox model is the linearity of the covariate. The assumption of the nonlinearity only holds for continuous covariate. Since age is the only continuous covariate, we check for nonlinearity of its functional form. An easy way of checking this is by plotting the Martingale residuals against the continuous covariate. Any systematic pattern in the plot suggests that covariate is not fitted adequately and the assumption is violated. From figure 7, there is no systematic pattern in the plot of the graphs indicating that the assumption of nonlinearity of the continuous covariate age is moderately met.

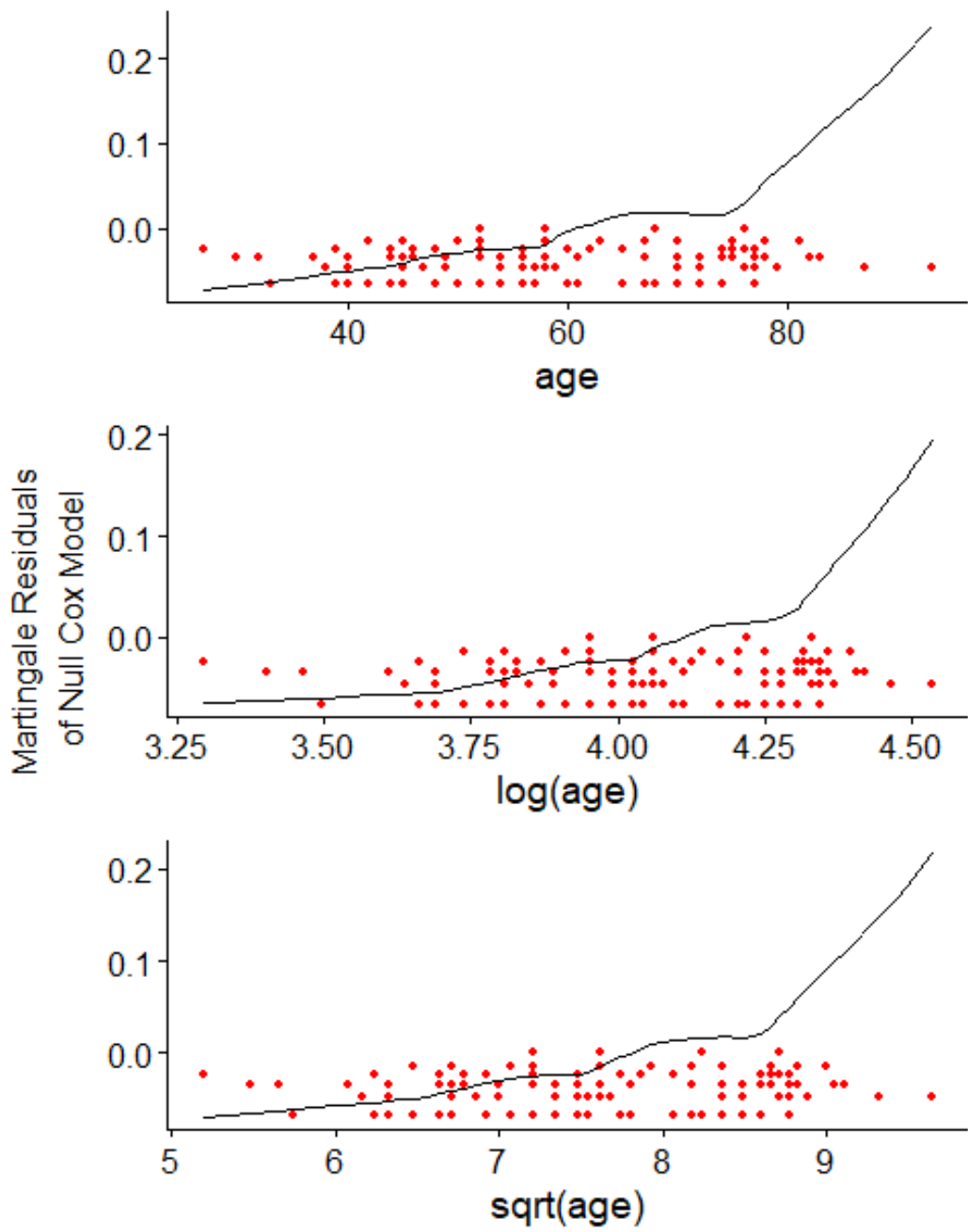


Figure 7: Nonlinearity of functional form of covariate age

4.8.5 Checking Overall Model Fit

We check the overall model fit using the Cox-Snell residuals. We plot the log cumulative hazard of Cox-Snell residual along with the best fitted straight line. If the straight line graph coincide with the step functions then the model is well fitted can be used for further analysis such as predictions and inferences. From figure 8, we can observe that both the straight and the step function graph reasonably move along together indicating that over all the model is well fitted.

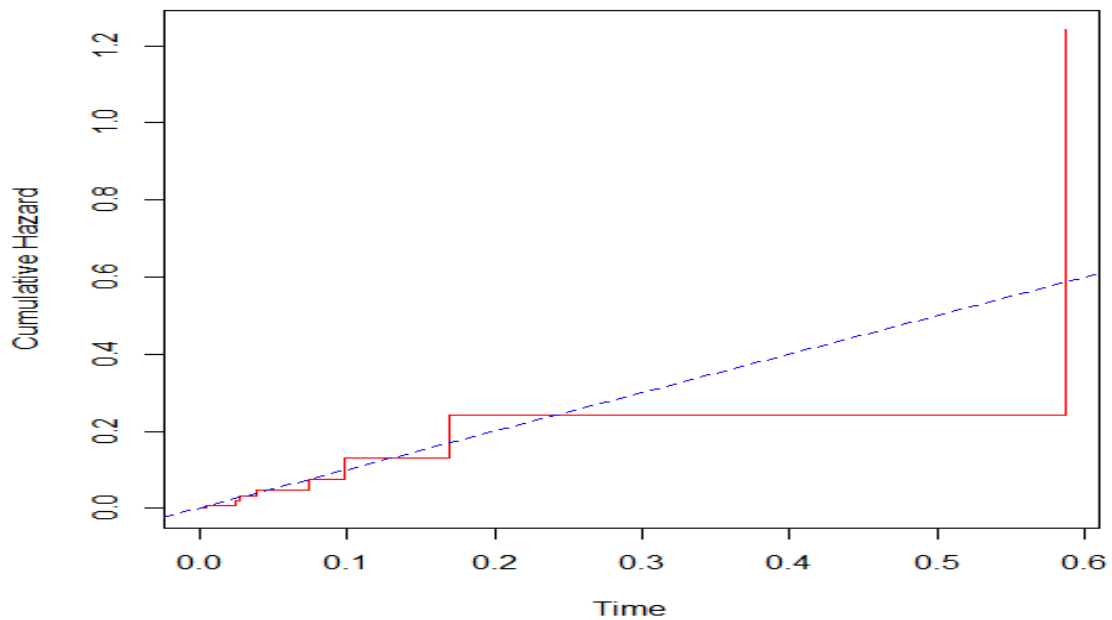


Figure 8: Cumulative hazard plot of the Cox-Snell residual for Cox PH model to check the overall fitted model.

4.9 Estimating the Semi-Parametric Model

Since the assumption of the selected model is fully satisfied through diagnostic check, the next step is to estimate the model and observe significant covariates and relevant inferences. Table 16 below provides the summary of fitted model with detailed explanation of the output as follows: The coefficient column provides the estimate of the coefficient of the regression parameters($X_{age}, X_{treatments}$) with interpretation that the estimated coefficient b_1 , gives the expected change of the log hazard ratio for every one unit increase in the corresponding covariate (X_1) when all other covariate are held constant (or are equal to zero). For instance, the estimated coefficient for age is 0.05962 indicating the expected change of the log hazard ratio for every one unit increase in age when treatment is held constant (when treatment is zero). Also, the sign of the positive coefficient indicates a positive relationship between the covariate and the event (death). For instance, the positive coefficient of age indicates that the older (higher) the patient the higher the risk of patient dieing with cancer.

The $\exp(\text{coef})$ gives the hazard ratio (HR) between two groups with $\text{HR} < 1$ indicating reduction in hazard (good prognostic factor in cancer studies), $\text{HR} > 1$ indicating increase in hazard (bad prognostic factor in cancer studies) and $\text{HR} = 1$ indicating no effect. Three groups of treatment were considered in this studies with patient receiving teletherapy as treatment 1, combination of teletherapy and chemotherapy as treatment 2 and combinations of teletherapy+chemotherapy + brachytherapy as treatment 3. Treatment 3 was used as the baseline to compare the other two treatments. From the table below, for teletherapy we have $\exp(0.44668)=1.56311$ indicating that patients receiving teletherapy only increase hazard by a factor of

1.56311(56.31 %) times higher than patients receiving treatment 3 when age is held constant. Also, for age we have $\exp(0.05962)=1.06143$ indicating that a unit increase in age increase hazard (death) by a factor of 1.06143 (6.1%) when treatment is held constant (bad prognostic in this case).

The $se(coef)$ column gives the standard error of the estimated coefficient.

The z column gives the value of the Wald statistic which is calculated as the ratio of the regression coefficient to its standard error (ie, $z = \frac{coef}{se(coefficient)}$). The Wald statistic test under the null hypothesis if the regression parameter is zero ($\beta = 0$). The Wald statistic z^* , is compared to the critical value $z_{1-\frac{\alpha}{2}}$ at $\alpha = 5\%$ significant level. If $|Z| > z_{1-\frac{\alpha}{2}}$ indicate statistically different from 0 ($z_{1-\frac{0.05}{2}}$). From the table, only age is statistically significant implying that age is the only factor in this study that contribute to the hazard of cancer patient.

The $Pr(> |z|)$ gives the corresponding p-value of the Wald statistic for large samples. It measure the statistical significant of the covariate. It gives the same conclusion as the Wald statistic. Any $Pr(> |z|)$ less than the $\alpha - level(5\%)$ of significant indicate as statistically significant different from 0. From the table, none of the treatments is statistically significant (p-value > 0.05) but age is statically significant (p-value = 0.0284).

The lower 0.95 and upper 0.95 give the confidence interval of the hazard ration ($\exp(coef)$). It also measure the statistical significant of the covariate. It also gives the same conclusion as the result of the Wald test and p-value approach. To measure the importance of the covariate, any covariate which confidence interval include one implies that the covariate is not statistically significant. From the table below, age

has a confidence interval of 1.0063 and 1.120 which does not include one. Hence age is statistically significant. But treatment has confident interval including one indicating none significant factor.

In summary, the estimated semi-parametric model are:

$$\lambda(t)_{death} = \lambda(t)_{alive} e^{0.05962X_{age}} \quad (52)$$

$$\lambda(t)_{death} = \lambda(t)_{alive} e^{0.05962X_{age} + 0.44668X_{teletherapy}} \quad (53)$$

$$\lambda(t)_{death} = \lambda(t)_{alive} e^{0.05962X_{age} + 0.03533X_{tele-brachy}} \quad (54)$$

The above three separate models can be put into a single model as:

$$\lambda(t)_{death} = \lambda(t)_{alive} e^{0.05962X_{age} + 0.44668X_{teletherapy} + 0.03533X_{tele-brachy}} \quad (55)$$

Table 16: Summary of semi-parametric model

covariate	coefficient	exp(coef)	se(coef)	z	$Pr(> z)$	lower .95	upper .95
age	0.05962	1.06143	0.02720	2.192	0.0284	1.0063	1.120
teletherapy	0.44668	1.56311	0.92302	0.484	0.6284	0.2560	9.543
tele-brachytherapy	0.03533	1.03596	0.82639	0.043	0.9659	0.2051	5.233

4.10 Further Analysis of Model Development

As in the case of statistical model, health analyst may want to fit model that involves only significant covariate by fitting each covariate separately (similar to step-wise regression). Table 17 shows the result of the fitted Cox model involving only treatment only. We can observe that gain none of the treatments has a significant influence (p-values > 0.05) on the hazard of cancer patient under study as in the case

of the fitted full model in Table 16. However, the sign of the coefficient of treatment II (teletherapy + brachytherapy) changes to negative. However, in Table 18 which provides a summary of fitted Cox model involving only covariate age shows significant (p-value = 0.0297) factor with a positive coefficient sign. This shows age is a key contributing covariate that significantly affect the hazard of the cervical cancer patient. Since age was only significant factor (in this stepwise approach) the interpretation of the coefficient is similar to those in fully fitted model discussed above. We provide write the model as:

$$\lambda(t)_{death} = \lambda(t)_{alive} e^{0.05785 X_{age}} \quad (56)$$

Table 17: Summary of fitted semi-parametric model involving treatment only

covariate	coefficient	exp(coef)	se(coef)	z	$Pr(> z)$	lower .95	upper .95
teletherapy	0.24236	1.27425	0.91587	0.265	0.791	0.2117	7.671
tele-brachytherapy	-0.05566	0.94586	0.82324	-0.068	0.946	0.1884	4.749

Table 18: Summary of fitted semi-parametric model involving age only

covariate	coefficient	exp(coef)	se(coef)	z	$Pr(> z)$	lower .95	upper .95
age	0.05785	1.05955	0.02661	2.174	0.0297	1.006	1.116

Finally, differences in risk factor of age group are influence by some external factors such as regular exercise, diet, frequent medical check up, race and so forth making age at which patient is at a higher risk of cancer differ from geographical setting to the other. For instance, whiles age groups of patient above thirty years may be at higher

risk of cervical cancer in Asia, the result of the age group may be different in USA due to the some factors listed above. Thus, we or any health analyst may be interested to know which of the age category from this data (geographical setting) has a higher risk of dieing with the cancer under study. We investigate this by categorizing age into two groups. We compare patients over 64 years and those patient of 64 years and below (age ≤ 64 years) where patients below 65 years are use as the baseline. Table 20 shows the assumption of the newly fitted model where the assumptions of the proportional hazard is fully met. The assumption of linearity is omitted since the two covariate are now categorical. Table 19 provides a result of the fitted Cox model for the two categorical covariate. Again, none of the treatment is significant (p-value > 0.05). Age of the patient is significant (p-value = 0.0385) with patients over 65 years more at a higher risk of dieing with the cancer (base on positive coefficient value) than those patient below 65 years. That is patients over 65 years old are at a higher risk of dieing with the cancer by a factor of 5.4592 (45.9%) times higher than patients less than or equal to 65 years when treatment is held constant. The fully fitted model in this case is written in a single model as:

$$\lambda(t)_{death} = \lambda(t)_{alive} e^{1.6973X_{age>65yrs} + 0.2668X_{teletherapy} - 0.1966X_{tele-brachy}} \quad (57)$$

Table 19: Summary of fully semi-parametric model for two categorical covariate

covariate	coefficient	exp(coef)	se(coef)	z	$Pr(> z)$	lower .95	upper .95
age > 65 years	1.6973	5.4592	0.8194	2.071	0.0383	1.0955	27.205
teletherapy	0.2668	1.3057	0.9146	0.292	0.7705	0.2175	7.840
tele-brachytherapy	-0.1966	0.8215	0.8328	-0.236	0.8133	0.1606	4.202

Table 20: Test for Proportional Hazard Assumption for two covariate

	rho	chisq	p
age > 65 years	0.0301	0.00672	0.935
teletherapy	-0.2418	0.43035	0.512
telebrachy	-0.2713	0.59551	0.440
GLOBAL	NA	0.72411	0.868

Again, if we are interested in only the covariate (age category) that has a major significant effect on the hazard of patient then Table 21 provides the results in this case. The interpretation of the coefficient and other important results is similar to those explain earlier. The fitted model is written as:

$$\lambda(t)_{death} = \lambda(t)_{alive} e^{1.6695 X_{age>65yrs}} \quad (58)$$

Table 21: Summary of semi-parametric model for categorical age

covariate	coefficient	exp(coef)	se(coef)	z	$Pr(> z)$	lower .95	upper .95
age > 65 years	1.6695	5.3093	0.8169	2.044	0.041	1.071	26.33

5 DISCUSSION AND CONCLUSION

Simulation studies are mostly used to demonstrate, justify and evaluate properties of statistical method when theoretical properties are difficult to handle. They are also useful for assessing the behavior of analytic techniques when various conditions impose difficulties in practice. One main benefit of this technique is that when the properties are well characterized, they can be used with high amount of confidence. One main purpose of this study is to propose a best way of analyzing survival clinical data which results can be estimated and trusted with confidence. One of the main challenges of health analyst in the study of survival analysis is which method to be used to analyze their data. We have demonstrated how the semi-parametric survival model tends to be robust and how suitable it can approximate the parametric survival model in the study of health data using R - package proposed by [19]. We simulated a right censored clinical data that follows various forms of parametric distributions and semi-parametric distribution. The simulation study was conducted taken into consideration the various forms of sample sizes. We considered three forms of sample sizes: small samples ($n < 30$), mild sample ($30 \leq n \leq 5000$) and large samples ($n > 5000$). From the study, when the sample size of the simulation study is small or mild, the semi-parametric model tends to perform well than the parametric model based on model comparison techniques. However, for large samples, the parametric method outperforms the semi-parametric model.

To assess the effectiveness of our proposed technique, we compared the reliability and effectiveness of the proposed technique using a real clinical data of mild samples. Based on the results of various model comparison, the semi-parametric under the

Cox Proportional Hazard emerged best fit conforming to the results of the simulation study supported with concise steps to model and explain survival clinical data.

In practice, researchers in the study of survival clinical studies (cancer studies for instance) do not have control over the samples to be used to test the efficacy of certain treatments as in the case of survival analysis in the application of business and so sample size is mostly small or mild. For instance, a health analyst can not go around and be choosing kidney cancer patients randomly to be in his/her study to test efficacy of certain treatments. Only the few patients who truly have kidney cancer and avail themselves to be treated are used to test treatment efficacy. Thus, in most of survival analysis of cancer study, the sample size is either small or mild making this proposed study efficient.

Nevertheless, we believe that it may not always be the case in the study of survival clinical data that the semi-parametric will outperform the parametric model even for small and mild clinical data especially when the assumptions of the parametric model underlying the data is well satisfied. When the assumptions of the parametric methods fits the data well, the results of the estimates are always better and reliable than the semi-parametric model. Thus, even when clinical data satisfied both the semi-parametric and parametric model, we recommend that semi-parametric approach may be used due to its flexibility, simplicity and availability in many software and the two results may be approximated unless the researcher is an expert in handling parametric techniques.

We hope that the proposed techniques will allows researchers to relax extreme parametric assumptions on the survival model imposed by existing methods. Thus,

unless one is more convinced and confidence of the kind of parametric model that fits well of the survival clinical data, we propose that the semi-parametric model must be used.

5.1 Future Work

We proposed a moderate efficient method for analyzing survival clinical data through simulation studies with a more reliable steps by steps approach to model and explain survival clinical data. However, more can be done to validate the results of this study. For instance, to compare the reliability of our proposed methods using real clinical data, certain covariate such as time varying covariate (covariate that changes over time during follow up periods) such as smoking status, income and so forth and other time independent covariate (covariate that do not change over the study period) such as family history for large and small samples of real clinical data were not used to compare the results of the simulation study. We recommend that future studies that compare the reliability of our propose method be conducted to include more covariates that are time dependent and independent be used for each of small samples and large samples. Furthermore, the research could be extended to situation where subjects experience multiple events of interest.

BIBLIOGRAPHY

- [1] A. Abadi, P . Yavari, M. Dehghani-Arani, H. Alavi-Majd, E. Ghasemi, F. Amanpour, and C. Bajdik. Cox models survival analysis based on breast cancer treatments. *Iranian Journal of Cancer Prevention*, 2014.
- [2] A. A. Abiodun. Analyzing competing risk survival time data using cox and parametric proportional hazards models. *ResearchGate*, 2018.
- [3] G.D. Altman and B. L de Stavola. Practical problems in fitting a proportional hazards model to data with updated measurements of the covariates. *Statistics in Medicine*, 13:301–341, 1994.
- [4] S. K Asiedu-Addo, I. Nwi-Mozu, and M. Ali. Modeling the survival of retrospective clinical data from prostate cancer patients in komfo anokye teaching hospital, ghana. *International Journal of Applied Science and Technology*, Vol. 5, No. 6, 2015.
- [5] R. Benderl, T. Augustin, and M. Blettner. Generating survival times to simulate cox proportional hazards models. *Statistics in Medicine*, 24(11):1713–1723. doi: 10.1002/sim.2059, 2005.
- [6] D. R. Cox. Regression models and life tables (with discussion). *Journal of the Royal Statistical Society, Series B* 34, 187-220., 1972.
- [7] D. R Cox. Partial likelihood. *Biometrika*, 62, 269 - 276, 1975.

- [8] J. J. Harden and J. Kropko. Simulating duration data for the cox model. *Political Science Research and Methods* <https://doi.org/10.1017/psrm.2018.19>, 2018.
- [9] D. J. Hendry. Data generation for the cox proportional hazards model with time-dependent covariates: a method for medical researchers. *Statistics in Medicine*, 33(3):436–454. doi: 10.1002/sim.5945, 2014.
- [10] E. L Kaplan and L. P. Meier. Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 53:457–481, 1958.
- [11] P. J. Kelly and L. L . Lim. Survival analysis for recurrent event data: An application to childhood infectious diseases. *Statistics in Medicine*, 19(1):13–33, 2000.
- [12] S. P. Khanal, V. Sreenivas, and S. K . Acharya³. Accelerated failure time models: An application in the survival of acute liver failure patients in india. *International Journal of Science and Research (IJSR)*, Vol 3, 2014.
- [13] David G. Kleinbaum and Mitchel Klein. *Survival Analysis: A Self-Learning Text*, Second Edition. Springer, New York, 1996.
- [14] O. Korosteleva. *Clinical Statistics: Introducing Clinical Trials, Survival Analysis, and Longitudinal Data Analysis*. Jones and Bartlett, 2009.
- [15] M. R. Kosorok, B. L. Lee, and J. P. Fine. Robust inference for univariate proportional hazards frailty regression models. *The Annals of Statistics*, Vol. 32, No. 4:pp. 1448–1491, 2004.

- [16] E. T. Lee and O. T. Go. Survival analysis in public health research. *Annu. Rev. Public Health*, 18:105–34, 1997.
- [17] D. Y. Lin and L. J. Wei. The robust inference for the cox proportional hazards model. *Journal of the American Statistical Association*, Vol. 84, No. 408:pp.1074–1078, 1989.
- [18] E. M. Montez-Rath, K. Kapphahn, M. B. Mathur, A. A. Mitani, D. J. Hendry, and M. Desai. Guidelines for generating rightcensored outcomes from a cox model extended to accommodate time-varying covariates. *Journal of Modern Applied Statistical Methods*, 16(1):86–106. doi: 10.22237/jmasm/1493597100, 2017.
- [19] D. Morina and A. Navarro. The r package survsim for the simulation of simple and complex survival data. *Journal of Statistical Software*, Volume 59 (2), 2014.
- [20] A. Nardi and M. Schemper. Comparing cox and parametric models in clinical studies. *Statistics in Medicine*, 2003:3597–3610, 2003.
- [21] N. S. Ngwa, J Howard, J. H. Cabral, M. D Cheng, M. J. Pencina, D. R. Gagnon, M. P. LaValley, and L. A. Cupples. Comparison of time dependent cox regression, pooled logistic regression and cross sectional pooling with simulations and an application to the framingham heart study. *BMC Medical Research Methodology*, 16:148 DOI 10.1186/s12874-016-0248-6, 2016.
- [22] Mark A. Nielsen. Parameter estimation for the two-parameter weibull distribution. Master’s thesis, Brigham Young University, 2011.

- [23] A. M. Pourhoseingholi, E. Hajizadeh, M. B. Dehkordi, A. Safaee, A. Abadi, and R. M. Zali Reza Mohammad. Comparing cox regression and parametric models for survival of patients with gastric carcinoma. *Asian Pacific J Cancer Prev*, Vol 8, 412-416, 2007.
- [24] B. L. J. Qian and P y. Chen. Generating survival data in the simulation studies of cox model. In *Third International Conference on Information and Computing*, 2010.
- [25] R. Saikia and M. P Barman. Comparing accelerated failure time models with its specific distributions in the analysis of esophagus cancer patients data. *International Journal of Computational and Applied Mathematics*, ISSN 1819-4966 Volume 12, Number 2 pp. 411-424, 2017.
- [26] P. M. Sylvestre and M. Abrahamowicz. Comparison of algorithms to generate event times conditional on time-dependent covariates. *Statistics in Medicine*, 27(14):2618–2634. doi: 10.1002/sim.3092, 2008.
- [27] Mara Tableman and Jong Sung Kim. *Survival Analysis Using S: Analysis of Time - to -Event Data*. Chapman and Hall/CRC, 2003.
- [28] Q . Zaman and K. P. Pfeiffer. Survival analysis in medical research. <http://interstat.statjournals.net/YEAR/2011/articles/1105005.pdf>, 17(4):1(36), 2011.

- [29] M. Zhou. Understanding the cox regression models with timechange covariates. *The American Statistician*, 55(2):153–155, doi:10.1198/000313001750358491, 2001.

1 Appendix

1.1 R Code

1.1.1 Simulation Codes

```
library(survival)

library(rms)

library(Hmisc)

library(lattice)

library(Formula)

library(ggplot2)

library(mgcv)

library(coxed)

library(eha)

library(statmod)

library(survsim)

##simulation of Weibull censored time to event data for small sample size

sim.size<-1000

mat<-matrix(0,ncol=30,nrow=sim.size)

for(i in 1:sim.size){

sim.data.nocov.w.s=simple.surv.sim(n=20,foltime=3600,dist.ev=c('weibull'))
```

```

,anc.ev=c(0.69978200185280),beta0.ev=c(5.84298525742252),
anc.cens=1.17783687569519,beta0.cens=7.39773677281100,
z=list(c("unif", 0.8, 1.2)), dist.cens="weibull", beta=NA, x=NA)

data.frame(sim.data.nocov.w.s)

attach(sim.data.nocov.w.s)

cox.model.nocov.w.s=coxph(Surv(stop, status)~1,data=sim.data.nocov.w.s,
method="breslow")

weib.model.nocov.w.s= survreg(Surv(stop, status) ~ 1, dist="weibull",
data=sim.data.nocov.w.s)

mat[i,7]<-summary(weib.model.nocov.w.s)$coefficients
mat[i,8]<-summary(weib.model.nocov.w.s)$table[1,2]
mat[i,9]<-summary(weib.model.nocov.w.s)$table[1,3]
mat[i,10]<-summary(weib.model.nocov.w.s)$table[1,4]

exp.model.nocov.w.s=survreg(Surv(stop, status) ~ 1,dist="exp",
data=sim.data.nocov.w.s)

mat[i,13]<-summary(exp.model.nocov.w.s)$coefficients
mat[i,14]<-summary(exp.model.nocov.w.s)$table[1,2]
mat[i,15]<-summary(exp.model.nocov.w.s)$table[1,3]

```

```

mat[i,16]<-summary(exp.model.nocov.w.s)$table[1,4]

logl.model.nocov.w.s= survreg(Surv(stop, status) ~ 1,
dist="loglogistic",data=sim.data.nocov.w.s)

mat[i,19]<-summary(logl.model.nocov.w.s)$coefficients
mat[i,20]<-summary(logl.model.nocov.w.s)$table[1,2]
mat[i,21]<-summary(logl.model.nocov.w.s)$table[1,3]
mat[i,22]<-summary(logl.model.nocov.w.s)$table[1,4]

logNorm.model.nocov.w.s= survreg(Surv(stop, status) ~ 1,
dist="lognormal",data=sim.data.nocov.w.s)

summary(logNorm.model.nocov.w.s)

mat[i,25]<-summary(logNorm.model.nocov.w.s)$coefficients
mat[i,26]<-summary(logNorm.model.nocov.w.s)$table[1,2]
mat[i,27]<-summary(logNorm.model.nocov.w.s)$table[1,3]
mat[i,28]<-summary(logNorm.model.nocov.w.s)$table[1,4]

#####AIC

mat[i,5]<-extractAIC(cox.model.nocov.w.s)[2]
mat[i,11]<-extractAIC(weib.model.nocov.w.s)[2]
mat[i,17]<-extractAIC(exp.model.nocov.w.s)[2]

```

```

mat[i,23]<-extractAIC(logl.model.nocov.w.s)[2]
mat[i,29]<-extractAIC(logNorm.model.nocov.w.s)[2]

mat[i,6]<-AIC(cox.model.nocov.w.s,k = log(length(start)))
mat[i,12]<-AIC(weib.model.nocov.w.s,k = log(length(start)))
mat[i,18]<-AIC(exp.model.nocov.w.s,k = log(length(start)))
mat[i,24]<-AIC(logl.model.nocov.w.s,k = log(length(start)))
mat[i,30]<-AIC(logNorm.model.nocov.w.s,k = log(length(start)))

}

colMeans(mat)
results<-data.frame(matrix(colMeans(mat),ncol=6,byrow=T))
rownames(results)<-c("cox","weib","exp","logl","logNorm")
colnames(results)<-c("intercept","stderr","statistic",
"p-value","AIC","BIC")
results

##simulation of Weibull censord time to event data for medium
#sample size

```

```

sim.size<-500

mat.m<-matrix(0,ncol=30,nrow=sim.size)

for(i in 1:sim.size){

sim.data.nocov.w.m=simple.surv.sim(n=1000,foltime=3600,

dist.ev=c('weibull')

,anc.ev=c(0.69978200185280),beta0.ev=c(5.84298525742252),

anc.cens=1.17783687569519,beta0.cens=7.39773677281100,

z=list(c("unif", 0.8, 1.2)), dist.cens="weibull", beta=NA, x=NA)

data.frame(sim.data.nocov.w.m)

names(sim.data.nocov.w.m)

attach(sim.data.nocov.w.m)

summary(sim.data.nocov.w.m)

cox.model.nocov.w.m =coxph(Surv(stop, status) ~ 1,

data=sim.data.nocov.w.m,method="breslow")

summary(cox.model.nocov.w.m )

weib.model.nocov.w.m= survreg(Surv(stop, status) ~ 1,

dist="weibull",data=sim.data.nocov.w.m)

summary(weib.model.nocov.w.m)

mat.m[i,7]<-summary(weib.model.nocov.w.m)$coefficients

```

```

mat.m[i,8]<-summary(weib.model.nocov.w.m)$table[1,2]
mat.m[i,9]<-summary(weib.model.nocov.w.m)$table[1,3]
mat.m[i,10]<-summary(weib.model.nocov.w.m)$table[1,4]

exp.model.nocov.w.m=survreg(Surv(stop, status) ~ 1,
dist="exp",data=sim.data.nocov.w.m)

summary(exp.model.nocov.w.m)

mat.m[i,13]<-summary(exp.model.nocov.w.m)$coefficients
mat.m[i,14]<-summary(exp.model.nocov.w.m)$table[1,2]
mat.m[i,15]<-summary(exp.model.nocov.w.m)$table[1,3]
mat.m[i,16]<-summary(exp.model.nocov.w.m)$table[1,4]

logl.model.nocov.w.m= survreg(Surv(stop, status) ~ 1,
dist="loglogistic",data=sim.data.nocov.w.m)

summary(logl.model.nocov.w.m)

mat.m[i,19]<-summary(logl.model.nocov.w.m)$coefficients
mat.m[i,20]<-summary(logl.model.nocov.w.m)$table[1,2]
mat.m[i,21]<-summary(logl.model.nocov.w.m)$table[1,3]
mat.m[i,22]<-summary(logl.model.nocov.w.m)$table[1,4]

logNorm.model.nocov.w.m= survreg(Surv(stop, status) ~ 1,

```

```

dist="lognormal",data=sim.data.nocov.w.m)

summary(logNorm.model.nocov.w.m)

mat.m[i,25]<-summary(logNorm.model.nocov.w.m)$coefficients

mat.m[i,26]<-summary(logNorm.model.nocov.w.m)$table[1,2]

mat.m[i,27]<-summary(logNorm.model.nocov.w.m)$table[1,3]

mat.m[i,28]<-summary(logNorm.model.nocov.w.m)$table[1,4]

#####AIC

mat.m[i,5]<-extractAIC(cox.model.nocov.w.m)[2]

mat.m[i,11]<-extractAIC(weib.model.nocov.w.m)[2]

mat.m[i,17]<-extractAIC(exp.model.nocov.w.m)[2]

mat.m[i,23]<-extractAIC(logl.model.nocov.w.m)[2]

mat.m[i,29]<-extractAIC(logNorm.model.nocov.w.m)[2]

mat.m[i,6]<-AIC(cox.model.nocov.w.m,k = log(length(start)))

mat.m[i,12]<-AIC(weib.model.nocov.w.m,k = log(length(start)))

mat.m[i,18]<-AIC(exp.model.nocov.w.m,k = log(length(start)))

mat.m[i,24]<-AIC(logl.model.nocov.w.m,k = log(length(start)))

mat.m[i,30]<-AIC(logNorm.model.nocov.w.m,k = log(length(start)))

}

```

```

colMeans(mat.m)

results.m<-data.frame(matrix(colMeans(mat.m),ncol=6,byrow=T))

rownames(results.m)<-c("cox","weib","exp","logl","logNorm")

colnames(results.m)<-c("intercept","stderr","statistic","p-value",
"AIC","BIC")

results.m

##simulation of Weibull censord time to event data (for large data)

sim.size<-50

mat.w.L<-matrix(0,ncol=30,nrow=sim.size)

for(i in 1:sim.size){

sim.data.nocov.w.L= simple.surv.sim(n=200000, foltime=3600,
dist.ev=c('weib'),anc.ev=c(0.69978200185280),
beta0.ev=c(5.84298525742252),anc.cens=1.17783687569519,
beta0.cens=7.39773677281100,z=list(c("unif", 0.8, 1.2)),
dist.cens="weibull", beta=NA, x=NA)

data.frame(sim.data.nocov.w.L)

attach(sim.data.nocov.w.L)

```



```

cox.model.nocov.w.L =coxph(Surv(stop, status) ~ 1,
data=sim.data.nocov.w.L,method="breslow")

weib.model.nocov.w.L= survreg(Surv(stop, status) ~ 1,
dist="weibull",data=sim.data.nocov.w.L)

mat.w.L[i,7]<-summary(weib.model.nocov.w.L)$coefficients
mat.w.L[i,8]<-summary(weib.model.nocov.w.L)$table[1,2]
mat.w.L[i,9]<-summary(weib.model.nocov.w.L)$table[1,3]
mat.w.L[i,10]<-summary(weib.model.nocov.w.L)$table[1,4]

exp.model.nocov.w.L=survreg(Surv(stop, status) ~ 1,
dist="exp",data=sim.data.nocov.w.L)

mat.w.L[i,13]<-summary(exp.model.nocov.w.L)$coefficients
mat.w.L[i,14]<-summary(exp.model.nocov.w.L)$table[1,2]
mat.w.L[i,15]<-summary(exp.model.nocov.w.L)$table[1,3]
mat.w.L[i,16]<-summary(exp.model.nocov.w.L)$table[1,4]

logl.model.nocov.w.L= survreg(Surv(stop, status) ~ 1,
dist="loglogistic",data=sim.data.nocov.w.L)

mat.w.L[i,19]<-summary(logl.model.nocov.w.L)$coefficients

```

```

mat.w.L[i,20]<-summary(logl.model.nocov.w.L)$table[1,2]
mat.w.L[i,21]<-summary(logl.model.nocov.w.L)$table[1,3]
mat.w.L[i,22]<-summary(logl.model.nocov.w.L)$table[1,4]

logNorm.model.nocov.w.L= survreg(Surv(stop, status) ~ 1,
  dist="lognormal",data=sim.data.nocov.w.L)

mat.w.L[i,25]<-summary(logNorm.model.nocov.w.L)$coefficients
mat.w.L[i,26]<-summary(logNorm.model.nocov.w.L)$table[1,2]
mat.w.L[i,27]<-summary(logNorm.model.nocov.w.L)$table[1,3]
mat.w.L[i,28]<-summary(logNorm.model.nocov.w.L)$table[1,4]

#####AIC

mat.w.L[i,5]<-extractAIC(cox.model.nocov.w.L)[2]
mat.w.L[i,11]<-extractAIC(weib.model.nocov.w.L)[2]
mat.w.L[i,17]<-extractAIC(exp.model.nocov.w.L)[2]
mat.w.L[i,23]<-extractAIC(logl.model.nocov.w.L)[2]
mat.w.L[i,29]<-extractAIC(logNorm.model.nocov.w.L)[2]

mat.w.L[i,6]<-AIC(cox.model.nocov.w.L,k = log(length(start)))
mat.w.L[i,12]<-AIC(weib.model.nocov.w.L,k = log(length(start)))
mat.w.L[i,18]<-AIC(exp.model.nocov.w.L,k = log(length(start)))

```

```

mat.w.L[i,24]<-AIC(logl.model.nocov.w.L,k = log(length(start)))
mat.w.L[i,30]<-AIC(logNorm.model.nocov.w.L,k = log(length(start)))

}

```

```

colMeans(mat.w.L)

results.w.L<-data.frame(matrix(colMeans(mat.w.L),ncol=6,byrow=T))

rownames(results.w.L)<-c("cox","weib","exp","logl","logNorm")
colnames(results.w.L)<-c("intercept","stderr","statistic",
"p-value","AIC","BIC")

results.w.L

```

```

#####

Simulation of Exponential with no covariate

#####

sim.size<-1000

mat.e.s<-matrix(0,ncol=30,nrow=sim.size)

for(i in 1:sim.size){

sim.data.nocov.e.s= simple.surv.sim(n=12, foltime=3600,

dist.ev=c('weib'),anc.ev=c(0.69978200185280),

beta0.ev=c(5.84298525742252)

```

```

,anc.cens=1,beta0.cens=7.39773677281100,z=list(c("unif", 0.8, 1.2)),
  dist.cens="weibull", beta=NA, x=NA)

data.frame(sim.data.nocov.e.s)

attach(sim.data.nocov.e.s)

cox.model.nocov.e.s =coxph(Surv(stop, status) ~ 1,
data=sim.data.nocov.e.s,method="breslow")

weib.model.nocov.e.s= survreg(Surv(stop, status) ~ 1,
dist="weibull",data=sim.data.nocov.e.s)

mat.e.s[i,7]<-summary(weib.model.nocov.e.s)$coefficients
mat.e.s[i,8]<-summary(weib.model.nocov.e.s)$table[1,2]
mat.e.s[i,9]<-summary(weib.model.nocov.e.s)$table[1,3]
mat.e.s[i,10]<-summary(weib.model.nocov.e.s)$table[1,4]

exp.model.nocov.e.s=survreg(Surv(stop, status) ~ 1,
dist="exp",data=sim.data.nocov.e.s)

mat.e.s[i,13]<-summary(exp.model.nocov.e.s)$coefficients
mat.e.s[i,14]<-summary(exp.model.nocov.e.s)$table[1,2]
mat.e.s[i,15]<-summary(exp.model.nocov.e.s)$table[1,3]
mat.e.s[i,16]<-summary(exp.model.nocov.e.s)$table[1,4]

```

```

logl.model.nocov.e.s= survreg(Surv(stop, status) ~ 1,
dist="loglogistic",data=sim.data.nocov.e.s)

mat.e.s[i,19]<-summary(logl.model.nocov.e.s)$coefficients
mat.e.s[i,20]<-summary(logl.model.nocov.e.s)$table[1,2]
mat.e.s[i,21]<-summary(logl.model.nocov.e.s)$table[1,3]
mat.e.s[i,22]<-summary(logl.model.nocov.e.s)$table[1,4]

logNorm.model.nocov.e.s= survreg(Surv(stop, status) ~ 1,
dist="lognormal",data=sim.data.nocov.e.s)

mat.e.s[i,25]<-summary(logNorm.model.nocov.e.s)$coefficients
mat.e.s[i,26]<-summary(logNorm.model.nocov.e.s)$table[1,2]
mat.e.s[i,27]<-summary(logNorm.model.nocov.e.s)$table[1,3]
mat.e.s[i,28]<-summary(logNorm.model.nocov.e.s)$table[1,4]

#####AIC

mat.e.s[i,5]<-extractAIC(cox.model.nocov.e.s)[2]
mat.e.s[i,11]<-extractAIC(weib.model.nocov.e.s)[2]
mat.e.s[i,17]<-extractAIC(exp.model.nocov.e.s)[2]
mat.e.s[i,23]<-extractAIC(logl.model.nocov.e.s)[2]

```

```

mat.e.s[i,29]<-extractAIC(logNorm.model.nocov.e.s)[2]

mat.e.s[i,6]<-AIC(cox.model.nocov.e.s,k = log(length(start)))
mat.e.s[i,12]<-AIC(weib.model.nocov.e.s,k = log(length(start)))
mat.e.s[i,18]<-AIC(exp.model.nocov.e.s,k = log(length(start)))
mat.e.s[i,24]<-AIC(logl.model.nocov.e.s,k = log(length(start)))
mat.e.s[i,30]<-AIC(logNorm.model.nocov.e.s,k = log(length(start)))

}

colMeans(mat.e.s)

results.e.s<-data.frame(matrix(colMeans(mat.e.s),ncol=6,byrow=T))
rownames(results.e.s)<-c("cox","weib","exp","logl","logNorm")
colnames(results.e.s)<-c("intercept","stderr","statistic",
"p-value","AIC","BIC")

results.e.s

###simulation of exponential censord time to event data (medium data)

```

```

sim.size<-500

mat.e.m<-matrix(0,ncol=30,nrow=sim.size)

for(i in 1:sim.size){

sim.data.nocov.e.m= simple.surv.sim(n=250, foltime=3600,

dist.ev=c('weibull'),anc.ev=c(0.99999),beta0.ev=c(5.84298525742252),

anc.cens=1,beta0.cens=7.39773677281100,z=list(c("unif", 0.8, 1.2)),

dist.cens="weibull", beta=NA, x=NA)

data.frame(sim.data.nocov.e.m)

attach(sim.data.nocov.e.m)

cox.model.nocov.e.m =coxph(Surv(stop, status) ~ 1,

data=sim.data.nocov.e.m,method="breslow")

weib.model.nocov.e.m= survreg(Surv(stop, status) ~ 1,

dist="weibull",data=sim.data.nocov.e.m)

mat.e.m[i,7]<-summary(weib.model.nocov.e.m)$coefficients

mat.e.m[i,8]<-summary(weib.model.nocov.e.m)$table[1,2]

mat.e.m[i,9]<-summary(weib.model.nocov.e.m)$table[1,3]

mat.e.m[i,10]<-summary(weib.model.nocov.e.m)$table[1,4]

```

```

exp.model.nocov.e.m=survreg(Surv(stop, status) ~ 1,
dist="exp",data=sim.data.nocov.e.m)

mat.e.m[i,13]<-summary(exp.model.nocov.e.m)$coefficients
mat.e.m[i,14]<-summary(exp.model.nocov.e.m)$table[1,2]
mat.e.m[i,15]<-summary(exp.model.nocov.e.m)$table[1,3]
mat.e.m[i,16]<-summary(exp.model.nocov.e.m)$table[1,4]

logl.model.nocov.e.m= survreg(Surv(stop, status) ~ 1,
dist="loglogistic",data=sim.data.nocov.e.m)

mat.e.m[i,19]<-summary(logl.model.nocov.e.m)$coefficients
mat.e.m[i,20]<-summary(logl.model.nocov.e.m)$table[1,2]
mat.e.m[i,21]<-summary(logl.model.nocov.e.m)$table[1,3]
mat.e.m[i,22]<-summary(logl.model.nocov.e.m)$table[1,4]

logNorm.model.nocov.e.m= survreg(Surv(stop, status) ~ 1,
dist="lognormal",data=sim.data.nocov.e.m)

mat.e.m[i,25]<-summary(logNorm.model.nocov.e.m)$coefficients
mat.e.m[i,26]<-summary(logNorm.model.nocov.e.m)$table[1,2]
mat.e.m[i,27]<-summary(logNorm.model.nocov.e.m)$table[1,3]
mat.e.m[i,28]<-summary(logNorm.model.nocov.e.m)$table[1,4]

```



```

#####AIC

mat.e.m[i,5]<-extractAIC(cox.model.nocov.e.m)[2]

mat.e.m[i,11]<-extractAIC(weib.model.nocov.e.m)[2]

mat.e.m[i,17]<-extractAIC(exp.model.nocov.e.m)[2]

mat.e.m[i,23]<-extractAIC(logl.model.nocov.e.m)[2]

mat.e.m[i,29]<-extractAIC(logNorm.model.nocov.e.m)[2]

mat.e.m[i,6]<-AIC(cox.model.nocov.e.m,k = log(length(start)))
mat.e.m[i,12]<-AIC(weib.model.nocov.e.m,k = log(length(start)))
mat.e.m[i,18]<-AIC(exp.model.nocov.e.m,k = log(length(start)))
mat.e.m[i,24]<-AIC(logl.model.nocov.e.m,k = log(length(start)))
mat.e.m[i,30]<-AIC(logNorm.model.nocov.e.m,k = log(length(start)))

}

colMeans(mat.e.m)

results.e.m<-data.frame(matrix(colMeans(mat.e.m),ncol=6,byrow=T))

rownames(results.e.m)<-c("cox","weib","exp","logl","logNorm")

colnames(results.e.m)<-c("intercept","stderr","statistic","p-value",
"AIC","BIC")

results.e.m

```

```

####simulation of exponential sensor time to event data (large data)

sim.size<-100

mat.e.L<-matrix(0,ncol=30,nrow=sim.size)

for(i in 1:sim.size){

sim.data.nocov.e.L= simple.surv.sim(n=30000, foltime=3600,

dist.ev=c('weibull'),anc.ev=c(1),beta0.ev=c(5.84298525742252),

anc.cens=1,beta0.cens=7.39773677281100,z=list(c("unif", 0.8, 1.2)),

dist.cens="weibull", beta=NA, x=NA)

data.frame(sim.data.nocov.e.L)

attach(sim.data.nocov.e.L)

cox.model.nocov.e.L =coxph(Surv(stop, status) ~ 1,

data=sim.data.nocov.e.L,method="breslow")

weib.model.nocov.e.L= survreg(Surv(stop, status) ~ 1,

dist="weibull",data=sim.data.nocov.e.L)

mat.e.L[i,7]<-summary(weib.model.nocov.e.L)$coefficients

```

```

mat.e.L[i,8]<-summary(weib.model.nocov.e.L)$table[1,2]
mat.e.L[i,9]<-summary(weib.model.nocov.e.L)$table[1,3]
mat.e.L[i,10]<-summary(weib.model.nocov.e.L)$table[1,4]

exp.model.nocov.e.L=survreg(Surv(stop, status) ~ 1,
dist="exp",data=sim.data.nocov.e.L)

mat.e.L[i,13]<-summary(exp.model.nocov.e.L)$coefficients
mat.e.L[i,14]<-summary(exp.model.nocov.e.L)$table[1,2]
mat.e.L[i,15]<-summary(exp.model.nocov.e.L)$table[1,3]
mat.e.L[i,16]<-summary(exp.model.nocov.e.L)$table[1,4]

logl.model.nocov.e.L= survreg(Surv(stop, status) ~ 1,
dist="loglogistic",data=sim.data.nocov.e.L)

mat.e.L[i,19]<-summary(logl.model.nocov.e.L)$coefficients
mat.e.L[i,20]<-summary(logl.model.nocov.e.L)$table[1,2]
mat.e.L[i,21]<-summary(logl.model.nocov.e.L)$table[1,3]
mat.e.L[i,22]<-summary(logl.model.nocov.e.L)$table[1,4]

logNorm.model.nocov.e.L= survreg(Surv(stop, status) ~ 1,
dist="lognormal",data=sim.data.nocov.e.L)

mat.e.L[i,25]<-summary(logNorm.model.nocov.e.L)$coefficients
mat.e.L[i,26]<-summary(logNorm.model.nocov.e.L)$table[1,2]

```

```

mat.e.L[i,27]<-summary(logNorm.model.nocov.e.L)$table[1,3]
mat.e.L[i,28]<-summary(logNorm.model.nocov.e.L)$table[1,4]

#####AIC

mat.e.L[i,5]<-extractAIC(cox.model.nocov.e.L)[2]
mat.e.L[i,11]<-extractAIC(weib.model.nocov.e.L)[2]
mat.e.L[i,17]<-extractAIC(exp.model.nocov.e.L)[2]
mat.e.L[i,23]<-extractAIC(logl.model.nocov.e.L)[2]
mat.e.L[i,29]<-extractAIC(logNorm.model.nocov.e.L)[2]

mat.e.L[i,6]<-AIC(cox.model.nocov.e.L,k = log(length(start)))
mat.e.L[i,12]<-AIC(weib.model.nocov.e.L,k = log(length(start)))
mat.e.L[i,18]<-AIC(exp.model.nocov.e.L,k = log(length(start)))
mat.e.L[i,24]<-AIC(logl.model.nocov.e.L,k = log(length(start)))
mat.e.L[i,30]<-AIC(logNorm.model.nocov.e.L,k = log(length(start)))
}

colMeans(mat.e.L)

results.e.L<-data.frame(matrix(colMeans(mat.e.L),ncol=6,byrow=T))

rownames(results.e.L)<-c("cox","weib","exp","logl","logNorm")

colnames(results.e.L)<-c("intercept","stderr","statistic","p-value",
"AIC","BIC")

results.e.L

```

```

#####
#####
Simulation of log-logistic with no covariate
#####
#####
##simulation of loglogistic censord time to event data (small data)

sim.size<-1000

mat.llg.s<-matrix(0,ncol=30,nrow=sim.size)

for(i in 1:sim.size){

sim.data.nocov.llg.s= simple.surv.sim(n=15, foltime=3600,
dist.ev=c('llogistic'),anc.ev=c(0.69978200185280),beta0.ev=c(5.842985
25742252),anc.cens=1.17783687569519,beta0.cens=7.39773677281100
,z=list(c("unif", 0.8, 1.2)), dist.cens="llogistic", beta=NA, x=NA)

data.frame(sim.data.nocov.llg.s)

attach(sim.data.nocov.llg.s)

cox.model.nocov.llg.s =coxph(Surv(stop, status) ~ 1,

```

```

data=sim.data.nocov.llg.s,method="breslow")

summary(cox.model.nocov.llg.s )

weib.model.nocov.llg.s= survreg(Surv(stop, status) ~ 1,
dist="weibull",data=sim.data.nocov.llg.s)

mat.llg.s[i,7]<-summary(weib.model.nocov.llg.s)$coefficients
mat.llg.s[i,8]<-summary(weib.model.nocov.llg.s)$table[1,2]
mat.llg.s[i,9]<-summary(weib.model.nocov.llg.s)$table[1,3]
mat.llg.s[i,10]<-summary(weib.model.nocov.llg.s)$table[1,4]

exp.model.nocov.llg.s=survreg(Surv(stop, status) ~ 1,
dist="exp",data=sim.data.nocov.llg.s)

mat.llg.s[i,13]<-summary(exp.model.nocov.llg.s)$coefficients
mat.llg.s[i,14]<-summary(exp.model.nocov.llg.s)$table[1,2]
mat.llg.s[i,15]<-summary(exp.model.nocov.llg.s)$table[1,3]
mat.llg.s[i,16]<-summary(exp.model.nocov.llg.s)$table[1,4]

logl.model.nocov.llg.s= survreg(Surv(stop, status) ~ 1,
dist="loglogistic",data=sim.data.nocov.llg.s)

mat.llg.s[i,19]<-summary(logl.model.nocov.llg.s)$coefficients
mat.llg.s[i,20]<-summary(logl.model.nocov.llg.s)$table[1,2]

```

```

mat.llg.s[i,21]<-summary(logl.model.nocov.llg.s)$table[1,3]
mat.llg.s[i,22]<-summary(logl.model.nocov.llg.s)$table[1,4]

logNorm.model.nocov.llg.s= survreg(Surv(stop, status) ~ 1,
dist="lognormal",data=sim.data.nocov.llg.s)

mat.llg.s[i,25]<-summary(logNorm.model.nocov.llg.s)$coefficients
mat.llg.s[i,26]<-summary(logNorm.model.nocov.llg.s)$table[1,2]
mat.llg.s[i,27]<-summary(logNorm.model.nocov.llg.s)$table[1,3]
mat.llg.s[i,28]<-summary(logNorm.model.nocov.llg.s)$table[1,4]

#####AIC

mat.llg.s[i,5]<-extractAIC(cox.model.nocov.llg.s)[2]
mat.llg.s[i,11]<-extractAIC(weib.model.nocov.llg.s)[2]
mat.llg.s[i,17]<-extractAIC(exp.model.nocov.llg.s)[2]
mat.llg.s[i,23]<-extractAIC(logl.model.nocov.llg.s)[2]
mat.llg.s[i,29]<-extractAIC(logNorm.model.nocov.llg.s)[2]

mat.llg.s[i,6]<-AIC(cox.model.nocov.llg.s,k = log(length(start)))
mat.llg.s[i,12]<-AIC(weib.model.nocov.llg.s,k = log(length(start)))
mat.llg.s[i,18]<-AIC(exp.model.nocov.llg.s,k = log(length(start)))

```

```

mat.llg.s[i,24]<-AIC(logl.model.nocov.llg.s,k = log(length(start)))
mat.llg.s[i,30]<-AIC(logNorm.model.nocov.llg.s,k = log(length(start)))

}

colMeans(mat.llg.s)

results.llg.s<-data.frame(matrix(colMeans(mat.llg.s),ncol=6,byrow=T))

rownames(results.llg.s)<-c("cox","weib","exp","logl","logNorm")

colnames(results.llg.s)<-c("intercept","stderr","statistic","p-value"
,"AIC","BIC")

results.llg.s

##simulation of loglogistic censord time to event data (medium data)

sim.size<-500

mat.llg.m<-matrix(0,ncol=30,nrow=sim.size)

for(i in 1:sim.size){

sim.data.nocov.llg.m= simple.surv.sim(n=250, foltime=3600,

dist.ev=c('llogistic'),anc.ev=c(0.69978200185280),beta0.ev=

c(5.84298525742252),anc.cens=1.17783687569519,

beta0.cens=7.39773677281100,z=list(c("unif", 0.8, 1.2)),

dist.cens="llogistic", beta=NA, x=NA)

```



```

data.frame(sim.data.nocov.llg.m)

attach(sim.data.nocov.llg.m)

cox.model.nocov.llg.m =coxph(Surv(stop, status) ~ 1,
data=sim.data.nocov.llg.m,method="breslow")

summary(cox.model.nocov.llg.m )

weib.model.nocov.llg.m= survreg(Surv(stop, status) ~ 1,
dist="weibull",data=sim.data.nocov.llg.m)

mat.llg.m[i,7]<-summary(weib.model.nocov.llg.m)$coefficients
mat.llg.m[i,8]<-summary(weib.model.nocov.llg.m)$table[1,2]
mat.llg.m[i,9]<-summary(weib.model.nocov.llg.m)$table[1,3]
mat.llg.m[i,10]<-summary(weib.model.nocov.llg.m)$table[1,4]

exp.model.nocov.llg.m=survreg(Surv(stop, status) ~ 1,
dist="exp",data=sim.data.nocov.llg.m)

mat.llg.m[i,13]<-summary(exp.model.nocov.llg.m)$coefficients
mat.llg.m[i,14]<-summary(exp.model.nocov.llg.m)$table[1,2]
mat.llg.m[i,15]<-summary(exp.model.nocov.llg.m)$table[1,3]
mat.llg.m[i,16]<-summary(exp.model.nocov.llg.m)$table[1,4]

```

```

logl.model.nocov.llg.m= survreg(Surv(stop, status) ~ 1,
dist="loglogistic",data=sim.data.nocov.llg.m)

mat.llg.m[i,19]<-summary(logl.model.nocov.llg.m)$coefficients
mat.llg.m[i,20]<-summary(logl.model.nocov.llg.m)$table[1,2]
mat.llg.m[i,21]<-summary(logl.model.nocov.llg.m)$table[1,3]
mat.llg.m[i,22]<-summary(logl.model.nocov.llg.m)$table[1,4]

logNorm.model.nocov.llg.m= survreg(Surv(stop, status) ~ 1,
dist="lognormal",data=sim.data.nocov.llg.m)

mat.llg.m[i,25]<-summary(logNorm.model.nocov.llg.m)$coefficients
mat.llg.m[i,26]<-summary(logNorm.model.nocov.llg.m)$table[1,2]
mat.llg.m[i,27]<-summary(logNorm.model.nocov.llg.m)$table[1,3]
mat.llg.m[i,28]<-summary(logNorm.model.nocov.llg.m)$table[1,4]

#####AIC

mat.llg.m[i,5]<-extractAIC(cox.model.nocov.llg.m)[2]
mat.llg.m[i,11]<-extractAIC(weib.model.nocov.llg.m)[2]
mat.llg.m[i,17]<-extractAIC(exp.model.nocov.llg.m)[2]
mat.llg.m[i,23]<-extractAIC(logl.model.nocov.llg.m)[2]
mat.llg.m[i,29]<-extractAIC(logNorm.model.nocov.llg.m)[2]

```

```

mat.llg.m[i,6]<-AIC(cox.model.nocov.llg.m,k = log(length(start)))
mat.llg.m[i,12]<-AIC(weib.model.nocov.llg.m,k = log(length(start)))
mat.llg.m[i,18]<-AIC(exp.model.nocov.llg.m,k = log(length(start)))
mat.llg.m[i,24]<-AIC(logl.model.nocov.llg.m,k = log(length(start)))
mat.llg.m[i,30]<-AIC(logNorm.model.nocov.llg.m,k = log(length(start)))

}

colMeans(mat.llg.m)

results.llg.m<-data.frame(matrix(colMeans(mat.llg.m),ncol=6,
byrow=T))

rownames(results.llg.m)<-c("cox","weib","exp","logl","logNorm")
colnames(results.llg.m)<-c("intercept","stderr","statistic",
"p-value","AIC","BIC")

results.llg.m

##simulation of loglogistic censord time to event data(large data)

sim.size<-1000

mat.llg.L<-matrix(0,ncol=30,nrow=sim.size)

for(i in 1:sim.size){

```

```

sim.data.nocov.llg.L= simple.surv.sim(n=55000, foltime=3600,
dist.ev=c('llogistic'), anc.ev=c(0.69978200185280),
beta0.ev=c(5.84298525742252), anc.cens=1.17783687569519,
beta0.cens=7.39773677281100, z=list(c("unif", 0.8, 1.2)),
dist.cens="llogistic", beta=NA, x=NA)

data.frame(sim.data.nocov.llg.L)

attach(sim.data.nocov.llg.L)

cox.model.nocov.llg.L =coxph(Surv(stop, status) ~ 1,
data=sim.data.nocov.llg.L, method="breslow")

weib.model.nocov.llg.L= survreg(Surv(stop, status) ~ 1,
dist="weibull", data=sim.data.nocov.llg.L)

mat.llg.L[i,7]<-summary(weib.model.nocov.llg.L)$coefficients
mat.llg.L[i,8]<-summary(weib.model.nocov.llg.L)$table[1,2]
mat.llg.L[i,9]<-summary(weib.model.nocov.llg.L)$table[1,3]
mat.llg.L[i,10]<-summary(weib.model.nocov.llg.L)$table[1,4]

exp.model.nocov.llg.L=survreg(Surv(stop, status) ~ 1,
dist="exp", data=sim.data.nocov.llg.L)

mat.llg.L[i,13]<-summary(exp.model.nocov.llg.L)$coefficients

```

```

mat.llg.L[i,14]<-summary(exp.model.nocov.llg.L)$table[1,2]
mat.llg.L[i,15]<-summary(exp.model.nocov.llg.L)$table[1,3]
mat.llg.L[i,16]<-summary(exp.model.nocov.llg.L)$table[1,4]

logl.model.nocov.llg.L= survreg(Surv(stop, status) ~ 1,
dist="loglogistic",data=sim.data.nocov.llg.L)
mat.llg.L[i,19]<-summary(logl.model.nocov.llg.L)$coefficients
mat.llg.L[i,20]<-summary(logl.model.nocov.llg.L)$table[1,2]
mat.llg.L[i,21]<-summary(logl.model.nocov.llg.L)$table[1,3]
mat.llg.L[i,22]<-summary(logl.model.nocov.llg.L)$table[1,4]

logNorm.model.nocov.llg.L= survreg(Surv(stop, status) ~ 1,
dist="lognormal",data=sim.data.nocov.llg.L)
mat.llg.L[i,25]<-summary(logNorm.model.nocov.llg.L)$coefficients
mat.llg.L[i,26]<-summary(logNorm.model.nocov.llg.L)$table[1,2]
mat.llg.L[i,27]<-summary(logNorm.model.nocov.llg.L)$table[1,3]
mat.llg.L[i,28]<-summary(logNorm.model.nocov.llg.L)$table[1,4]

#####AIC

mat.llg.L[i,5]<-extractAIC(cox.model.nocov.llg.L)[2]
mat.llg.L[i,11]<-extractAIC(weib.model.nocov.llg.L)[2]
mat.llg.L[i,17]<-extractAIC(exp.model.nocov.llg.L)[2]

```

```

mat.llg.L[i,23]<-extractAIC(logl.model.nocov.llg.L)[2]
mat.llg.L[i,29]<-extractAIC(logNorm.model.nocov.llg.L)[2]

mat.llg.L[i,6]<-AIC(cox.model.nocov.llg.L,k = log(length(start)))
mat.llg.L[i,12]<-AIC(weib.model.nocov.llg.L,k = log(length(start)))
mat.llg.L[i,18]<-AIC(exp.model.nocov.llg.L,k = log(length(start)))
mat.llg.L[i,24]<-AIC(logl.model.nocov.llg.L,k = log(length(start)))
mat.llg.L[i,30]<-AIC(logNorm.model.nocov.llg.L,k = log(length(start)))

}

colMeans(mat.llg.L)
results.llg.L<-data.frame(matrix(colMeans(mat.llg.L),ncol=6,byrow=T))
rownames(results.llg.L)<-c("cox","weib","exp","logl","logNorm")
colnames(results.llg.L)<-c("intercept","stderr","statistic","p-value",
"AIC","BIC")
results.llg.L

```

```
#####
```

```

#####

simulation lognormal with no covariate

#####

###simulation of lognormal censored time to event data (small data)

sim.size<-1000

mat.LN.s<-matrix(0,ncol=30,nrow=sim.size)

for(i in 1:sim.size){

sim.data.nocov.LN.s= simple.surv.sim(n=21, foltime=3600, dist.ev=
c('lnorm'),anc.ev=c(0.69978200185280),beta0.ev=c(5.84298525742252)
,anc.cens=1.17783687569519,beta0.cens=7.39773677281100,

z=list(c("unif", 0.8, 1.2)), dist.cens="lnorm", beta=NA, x=NA)

data.frame(sim.data.nocov.LN.s)

attach(sim.data.nocov.LN.s)

cox.model.nocov.LN.s =coxph(Surv(stop, status) ~ 1,
data=sim.data.nocov.LN.s,method="breslow")

weib.model.nocov.LN.s= survreg(Surv(stop, status) ~ 1,

```

```

dist="weibull",data=sim.data.nocov.LN.s)

mat.LN.s[i,7]<-summary(weib.model.nocov.LN.s)$coefficients

mat.LN.s[i,8]<-summary(weib.model.nocov.LN.s)$table[1,2]

mat.LN.s[i,9]<-summary(weib.model.nocov.LN.s)$table[1,3]

mat.LN.s[i,10]<-summary(weib.model.nocov.LN.s)$table[1,4]

exp.model.nocov.LN.s=survreg(Surv(stop, status) ~ 1,
dist="exp",data=sim.data.nocov.LN.s)

mat.LN.s[i,13]<-summary(exp.model.nocov.LN.s)$coefficients

mat.LN.s[i,14]<-summary(exp.model.nocov.LN.s)$table[1,2]

mat.LN.s[i,15]<-summary(exp.model.nocov.LN.s)$table[1,3]

mat.LN.s[i,16]<-summary(exp.model.nocov.LN.s)$table[1,4]

logl.model.nocov.LN.s= survreg(Surv(stop, status) ~ 1,
dist="loglogistic",data=sim.data.nocov.LN.s)

mat.LN.s[i,19]<-summary(logl.model.nocov.LN.s)$coefficients

mat.LN.s[i,20]<-summary(logl.model.nocov.LN.s)$table[1,2]

mat.LN.s[i,21]<-summary(logl.model.nocov.LN.s)$table[1,3]

mat.LN.s[i,22]<-summary(logl.model.nocov.LN.s)$table[1,4]

logNorm.model.nocov.LN.s= survreg(Surv(stop, status) ~ 1,
dist="lognormal",data=sim.data.nocov.LN.s)

```



```

mat.LN.s[i,25]<-summary(logNorm.model.nocov.LN.s)$coefficients
mat.LN.s[i,26]<-summary(logNorm.model.nocov.LN.s)$table[1,2]
mat.LN.s[i,27]<-summary(logNorm.model.nocov.LN.s)$table[1,3]
mat.LN.s[i,28]<-summary(logNorm.model.nocov.LN.s)$table[1,4]

#####AIC

mat.LN.s[i,5]<-extractAIC(cox.model.nocov.w.s)[2]
mat.LN.s[i,11]<-extractAIC(weib.model.nocov.w.s)[2]
mat.LN.s[i,17]<-extractAIC(exp.model.nocov.w.s)[2]
mat.LN.s[i,23]<-extractAIC(logl.model.nocov.w.s)[2]
mat.LN.s[i,29]<-extractAIC(logNorm.model.nocov.w.s)[2]

mat.LN.s[i,6]<-AIC(cox.model.nocov.LN.s,k = log(length(start)))
mat.LN.s[i,12]<-AIC(weib.model.nocov.LN.s,k = log(length(start)))
mat.LN.s[i,18]<-AIC(exp.model.nocov.LN.s,k = log(length(start)))
mat.LN.s[i,24]<-AIC(logl.model.nocov.LN.s,k = log(length(start)))
mat.LN.s[i,30]<-AIC(logNorm.model.nocov.LN.s,k = log(length(start)))

}

```

```

colMeans(mat.LN.s)

results.LN.s<-data.frame(matrix(colMeans(mat.LN.s),ncol=6,byrow=T))

rownames(results.LN.s)<-c("cox","weib","exp","logl","logNorm")

colnames(results.LN.s)<-c("intercept","stderr","statistic","p-value",
"AIC","BIC")

results.LN.s

###simulation of lognormal censord time to event data (medium data)

sim.size<-500

mat.nocov.LN.m<-matrix(0,ncol=30,nrow=sim.size)

for(i in 1:sim.size){

sim.data.nocov.LN.m= simple.surv.sim(n=850, foltime=3600,
dist.ev=c('lnorm'),anc.ev=c(0.69978200185280),beta0.ev=
c(5.84298525742252),anc.cens=1.17783687569519,
beta0.cens=7.39773677281100,z=list(c("unif", 0.8, 1.2)),
dist.cens="lnorm", beta=NA, x=NA)

data.frame(sim.data.nocov.LN.m)

attach(sim.data.nocov.LN.m)

```

```

cox.model.nocov.LN.m =coxph(Surv(stop, status) ~ 1,
data=sim.data.nocov.LN.m,method="breslow")

weib.model.nocov.LN.m= survreg(Surv(stop, status) ~ 1,
dist="weibull",data=sim.data.nocov.LN.m)

mat.nocov.LN.m[i,7]<-summary(weib.model.nocov.LN.m)$coefficients
mat.nocov.LN.m[i,8]<-summary(weib.model.nocov.LN.m)$table[1,2]
mat.nocov.LN.m[i,9]<-summary(weib.model.nocov.LN.m)$table[1,3]
mat.nocov.LN.m[i,10]<-summary(weib.model.nocov.LN.m)$table[1,4]

exp.model.nocov.LN.m=survreg(Surv(stop, status) ~ 1,
dist="exp",data=sim.data.nocov.LN.m)

mat.nocov.LN.m[i,13]<-summary(exp.model.nocov.LN.m)$coefficients
mat.nocov.LN.m[i,14]<-summary(exp.model.nocov.LN.m)$table[1,2]
mat.nocov.LN.m[i,15]<-summary(exp.model.nocov.LN.m)$table[1,3]
mat.nocov.LN.m[i,16]<-summary(exp.model.nocov.LN.m)$table[1,4]

logl.model.nocov.LN.m= survreg(Surv(stop, status) ~ 1,
dist="loglogistic",data=sim.data.nocov.LN.m)

mat.nocov.LN.m[i,19]<-summary(logl.model.nocov.LN.m)$coefficients

```

```

mat.nocov.LN.m[i,20]<-summary(logl.model.nocov.LN.m)$table[1,2]
mat.nocov.LN.m[i,21]<-summary(logl.model.nocov.LN.m)$table[1,3]
mat.nocov.LN.m[i,22]<-summary(logl.model.nocov.LN.m)$table[1,4]

logNorm.model.nocov.LN.m= survreg(Surv(stop, status) ~ 1,
dist="lognormal",data=sim.data.nocov.LN.m)

mat.nocov.LN.m[i,25]<-summary(logNorm.model.nocov.LN.m)$coefficients
mat.nocov.LN.m[i,26]<-summary(logNorm.model.nocov.LN.m)$table[1,2]
mat.nocov.LN.m[i,27]<-summary(logNorm.model.nocov.LN.m)$table[1,3]
mat.nocov.LN.m[i,28]<-summary(logNorm.model.nocov.LN.m)$table[1,4]

#####AIC

mat.nocov.LN.m[i,5]<-extractAIC(cox.model.nocov.LN.m)[2]
mat.nocov.LN.m[i,11]<-extractAIC(weib.model.nocov.LN.m)[2]
mat.nocov.LN.m[i,17]<-extractAIC(exp.model.nocov.LN.m)[2]
mat.nocov.LN.m[i,23]<-extractAIC(logl.model.nocov.LN.m)[2]
mat.nocov.LN.m[i,29]<-extractAIC(logNorm.model.nocov.LN.m)[2]

mat.nocov.LN.m[i,6]<-AIC(cox.model.nocov.LN.m,k=log(length(start)))
mat.nocov.LN.m[i,12]=AIC(weib.model.nocov.LN.m,k=log(length(start)))
mat.nocov.LN.m[i,18]=AIC(exp.model.nocov.LN.m,k=log(length(start)))
mat.nocov.LN.m[i,24]=AIC(logl.model.nocov.LN.m,k = log(length(start)))

```

```

mat.nocov.LN.m[i,30]=AIC(logNorm.model.nocov.LN.m,k=log(length(start)))
}

colMeans(mat.nocov.LN.m)

results.nocov.LN.m<-data.frame(matrix(colMeans(mat.nocov.LN.m),ncol=6,
byrow=T))

rownames(results.nocov.LN.m)<-c("cox","weib","exp","log1","logNorm")
colnames(results.nocov.LN.m)<-c("intercept","stderr","statistic",
"p-value","AIC","BIC")

results.nocov.LN.m

####simulation of lognormal sensord time to event data (large data)

sim.size<-1000

mat.nocov.LN.L<-matrix(0,ncol=30,nrow=sim.size)

for(i in 1:sim.size){

sim.data.nocov.LN.L= simple.surv.sim(n=90877, foltime=3600,
dist.ev=c('lnorm'),anc.ev=c(0.69978200185280),
beta0.ev=c(5.84298525742252),anc.cens=1.17783687569519,
beta0.cens=7.39773677281100,z=list(c("unif", 0.8, 1.2)),

```

```

dist.cens="lnorm", beta=NA, x=NA)

data.frame(sim.data.nocov.LN.L)

attach(sim.data.nocov.LN.L)

cox.model.nocov.LN.L =coxph(Surv(stop, status) ~ 1,
data=sim.data.nocov.LN.L,method="breslow")

weib.model.nocov.LN.L= survreg(Surv(stop, status) ~ 1,
dist="weibull",data=sim.data.nocov.LN.L)

mat.nocov.LN.L[i,7]<-summary(weib.model.nocov.LN.L)$coefficients
mat.nocov.LN.L[i,8]<-summary(weib.model.nocov.LN.L)$table[1,2]
mat.nocov.LN.L[i,9]<-summary(weib.model.nocov.LN.L)$table[1,3]
mat.nocov.LN.L[i,10]<-summary(weib.model.nocov.LN.L)$table[1,4]

exp.model.nocov.LN.L=survreg(Surv(stop, status) ~ 1,
dist="exp",data=sim.data.nocov.LN.L)

mat.nocov.LN.L[i,13]<-summary(exp.model.nocov.LN.L)$coefficients
mat.nocov.LN.L[i,14]<-summary(exp.model.nocov.LN.L)$table[1,2]
mat.nocov.LN.L[i,15]<-summary(exp.model.nocov.LN.L)$table[1,3]
mat.nocov.LN.L[i,16]<-summary(exp.model.nocov.LN.L)$table[1,4]

```

```

logl.model.nocov.LN.L= survreg(Surv(stop, status) ~ 1,
  dist="loglogistic",data=sim.data.nocov.LN.L)

mat.nocov.LN.L[i,19]<-summary(logl.model.nocov.LN.L)$coefficients
mat.nocov.LN.L[i,20]<-summary(logl.model.nocov.LN.L)$table[1,2]
mat.nocov.LN.L[i,21]<-summary(logl.model.nocov.LN.L)$table[1,3]
mat.nocov.LN.L[i,22]<-summary(logl.model.nocov.LN.L)$table[1,4]

logNorm.model.nocov.LN.L= survreg(Surv(stop, status) ~ 1,
  dist="lognormal",data=sim.data.nocov.LN.L)

mat.nocov.LN.L[i,25]<-summary(logNorm.model.nocov.LN.L)$coefficients
mat.nocov.LN.L[i,26]<-summary(logNorm.model.nocov.LN.L)$table[1,2]
mat.nocov.LN.L[i,27]<-summary(logNorm.model.nocov.LN.L)$table[1,3]
mat.nocov.LN.L[i,28]<-summary(logNorm.model.nocov.LN.L)$table[1,4]

#####AIC

mat.nocov.LN.L[i,5]<-extractAIC(cox.model.nocov.LN.L)[2]
mat.nocov.LN.L[i,11]<-extractAIC(weib.model.nocov.LN.L)[2]
mat.nocov.LN.L[i,17]<-extractAIC(exp.model.nocov.LN.L)[2]
mat.nocov.LN.L[i,23]<-extractAIC(logl.model.nocov.LN.L)[2]
mat.nocov.LN.L[i,29]<-extractAIC(logNorm.model.nocov.LN.L)[2]

```

```

mat.nocov.LN.L[i,6]=AIC(cox.model.nocov.LN.L,k=log(length(start)))
mat.nocov.LN.L[i,12]=AIC(weib.model.nocov.LN.L,k=log(length(start)))
mat.nocov.LN.L[i,18]=AIC(exp.model.nocov.LN.L,k=log(length(start)))
mat.nocov.LN.L[i,24]=AIC(logl.model.nocov.LN.L,k=log(length(start)))
mat.nocov.LN.L[i,30]=AIC(logNorm.model.nocov.LN.L,k=log(length(start)))

}

colMeans(mat.nocov.LN.L)

results.nocov.LN.L<-data.frame(matrix(colMeans(mat.nocov.LN.L),ncol=6,
byrow=T))

rownames(results.nocov.LN.L)<-c("cox","weib","exp","logl","logNorm")
colnames(results.nocov.LN.L)<-c("intercept","stderr","statistic",
"p-value","AIC","BIC")

results.nocov.LN.L

#####
#####

Simulation of Cox no covariate

#####
#####

#####simulation of Cox censord time to event data(small data)

```



```

sim.data.nocov.cox.s = sim.survdata(N=10, T=360, censor=.35,
num.data.frames = 1)

attach(sim.data.nocov.cox.s )

cox.model.nocov.cox.s =coxph(Surv(y, failed) ~ 1,
data=sim.data.nocov.cox.s$data,method="breslow")

weib.model.nocov.cox.s=survreg(Surv(y, failed) ~ 1,dist="weibull",
data=sim.data.nocov.cox.s$data)

mat.cox.s[i,7]<-summary(weib.model.nocov.cox.s)$coefficients
mat.cox.s[i,8]<-summary(weib.model.nocov.cox.s)$table[1,2]
mat.cox.s[i,9]<-summary(weib.model.nocov.cox.s)$table[1,3]
mat.cox.s[i,10]<-summary(weib.model.nocov.cox.s)$table[1,4]

exp.model.nocov.cox.s=survreg(Surv(y, failed) ~ 1,dist="exp",
data=sim.data.nocov.cox.s$data)

mat.cox.s[i,13]<-summary(exp.model.nocov.cox.s)$coefficients
mat.cox.s[i,14]<-summary(exp.model.nocov.cox.s)$table[1,2]
mat.cox.s[i,15]<-summary(exp.model.nocov.cox.s)$table[1,3]
mat.cox.s[i,16]<-summary(exp.model.nocov.cox.s)$table[1,4]

```

```

logl.model.nocov.cox.s=survreg(Surv(y, failed) ~ 1,dist="loglogistic",
  data=sim.data.nocov.cox.s$data)

mat.cox.s[i,19]<-summary(logl.model.nocov.cox.s)$coefficients
mat.cox.s[i,20]<-summary(logl.model.nocov.cox.s)$table[1,2]
mat.cox.s[i,21]<-summary(logl.model.nocov.cox.s)$table[1,3]
mat.cox.s[i,22]<-summary(logl.model.nocov.cox.s)$table[1,4]

logNorm.model.nocov.cox.s=survreg(Surv(y, failed) ~ 1,dist="lognormal",
  data=sim.data.nocov.cox.s$data)

mat.cox.s[i,25]<-summary(logNorm.model.nocov.cox.s)$coefficients
mat.cox.s[i,26]<-summary(logNorm.model.nocov.cox.s)$table[1,2]
mat.cox.s[i,27]<-summary(logNorm.model.nocov.cox.s)$table[1,3]
mat.cox.s[i,28]<-summary(logNorm.model.nocov.cox.s)$table[1,4]

#####AIC

mat.cox.s[i,5]<-extractAIC(cox.model.nocov.cox.s)[2]
mat.cox.s[i,11]<-extractAIC(weib.model.nocov.cox.s)[2]
mat.cox.s[i,17]<-extractAIC(exp.model.nocov.cox.s)[2]
mat.cox.s[i,23]<-extractAIC(logl.model.nocov.cox.s)[2]

```

```

mat.cox.s[i,29]<-extractAIC(logNorm.model.nocov.cox.s)[2]

mat.cox.s[i,6]=AIC(cox.model.nocov.cox.s,k=log(length(start)))
mat.cox.s[i,12]=AIC(weib.model.nocov.cox.s,k=log(length(start)))
mat.cox.s[i,18]=AIC(exp.model.nocov.cox.s,k=log(length(start)))
mat.cox.s[i,24]=AIC(logl.model.nocov.cox.s,k=log(length(start)))
mat.cox.s[i,30]=AIC(logNorm.model.nocov.cox.s,k=log(length(start)))

}

colMeans(mat.cox.s)

results.cox.s=data.frame(matrix(colMeans(mat.cox.s),ncol=6,byrow=T))
rownames(results.cox.s)<-c("cox","weib","exp","logl","logNorm")
colnames(results.cox.s)<-c("intercept","stderr","statistic","p-value",
"AIC","BIC")

results.cox.s

###simulation of Cox sensord time to event data (medium data)

sim.size<-500

mat.cox.m<-matrix(0,ncol=30,nrow=sim.size)

```

```

for(i in 1:sim.size){

sim.data.nocov.cox.m = sim.survdata(N=150, T=360, censor=.35,
  num.data.frames = 1)

attach(sim.data.nocov.cox.m )

cox.model.nocov.cox.m =coxph(Surv(y, failed) ~ 1,
  data=sim.data.nocov.cox.m$data,method="breslow")

weib.model.nocov.cox.m=survreg(Surv(y, failed) ~ 1,dist="weibull",
  data=sim.data.nocov.cox.m$data)

mat.cox.m[i,7]<-summary(weib.model.nocov.cox.m)$coefficients
mat.cox.m[i,8]<-summary(weib.model.nocov.cox.m)$table[1,2]
mat.cox.m[i,9]<-summary(weib.model.nocov.cox.m)$table[1,3]
mat.cox.m[i,10]<-summary(weib.model.nocov.cox.m)$table[1,4]

exp.model.nocov.cox.m=survreg(Surv(y, failed) ~ 1,dist="exp",
  data=sim.data.nocov.cox.m$data)

mat.cox.m[i,13]<-summary(exp.model.nocov.cox.m)$coefficients
mat.cox.m[i,14]<-summary(exp.model.nocov.cox.m)$table[1,2]
mat.cox.m[i,15]<-summary(exp.model.nocov.cox.m)$table[1,3]
mat.cox.m[i,16]<-summary(exp.model.nocov.cox.m)$table[1,4]

```

```

logl.model.nocov.cox.m=survreg(Surv(y, failed) ~ 1,dist="loglogistic",
  data=sim.data.nocov.cox.m$data)

mat.cox.m[i,19]<-summary(logl.model.nocov.cox.m)$coefficients
mat.cox.m[i,20]<-summary(logl.model.nocov.cox.m)$table[1,2]
mat.cox.m[i,21]<-summary(logl.model.nocov.cox.m)$table[1,3]
mat.cox.m[i,22]<-summary(logl.model.nocov.cox.m)$table[1,4]

logNorm.model.nocov.cox.m=survreg(Surv(y, failed) ~ 1,dist="lognormal",
  data=sim.data.nocov.cox.m$data)

mat.cox.m[i,25]<-summary(logNorm.model.nocov.cox.m)$coefficients
mat.cox.m[i,26]<-summary(logNorm.model.nocov.cox.m)$table[1,2]
mat.cox.m[i,27]<-summary(logNorm.model.nocov.cox.m)$table[1,3]
mat.cox.m[i,28]<-summary(logNorm.model.nocov.cox.m)$table[1,4]

#####AIC

mat.cox.m[i,5]<-extractAIC(cox.model.nocov.cox.m)[2]
mat.cox.m[i,11]<-extractAIC(weib.model.nocov.cox.m)[2]
mat.cox.m[i,17]<-extractAIC(exp.model.nocov.cox.m)[2]
mat.cox.m[i,23]<-extractAIC(logl.model.nocov.cox.m)[2]
mat.cox.m[i,29]<-extractAIC(logNorm.model.nocov.cox.m)[2]

```

```

mat.cox.m[i,6]<-AIC(cox.model.nocov.cox.m,k = log(length(start)))
mat.cox.m[i,12]<-AIC(weib.model.nocov.cox.m,k = log(length(start)))
mat.cox.m[i,18]<-AIC(exp.model.nocov.cox.m,k = log(length(start)))
mat.cox.m[i,24]<-AIC(logl.model.nocov.cox.m,k = log(length(start)))
mat.cox.m[i,30]<-AIC(logNorm.model.nocov.cox.m,k = log(length(start)))
}

```

```

colMeans(mat.cox.m)

results.cox.m<-data.frame(matrix(colMeans(mat.cox.m),ncol=6,byrow=T))

rownames(results.cox.m)<-c("cox","weib","exp","logl","logNorm")

colnames(results.cox.m)=c("intercept","stderr","statistic","p-value",
"AIC","BIC")

results.cox.m

```

```

#####simulation of Cox censord time to event data(large data)

```

```

sim.size<-50

mat.cox.L<-matrix(0,ncol=30,nrow=sim.size)

for(i in 1:sim.size){

sim.data.nocov.cox.L = sim.survdata(N=100000, T=360, censor=.35,

```

```

num.data.frames = 1)

attach(sim.data.nocov.cox.L )

cox.model.nocov.cox.L =coxph(Surv(y, failed) ~ 1,
data=sim.data.nocov.cox.L$data,method="breslow")

weib.model.nocov.cox.L=survreg(Surv(y, failed) ~ 1,dist="weibull",
data=sim.data.nocov.cox.L$data)

mat.cox.L[i,7]<-summary(weib.model.nocov.cox.L)$coefficients
mat.cox.L[i,8]<-summary(weib.model.nocov.cox.L)$table[1,2]
mat.cox.L[i,9]<-summary(weib.model.nocov.cox.L)$table[1,3]
mat.cox.L[i,10]<-summary(weib.model.nocov.cox.L)$table[1,4]

exp.model.nocov.cox.L=survreg(Surv(y, failed) ~ 1,dist="exp",
data=sim.data.nocov.cox.L$data)

mat.cox.L[i,13]<-summary(exp.model.nocov.cox.L)$coefficients
mat.cox.L[i,14]<-summary(exp.model.nocov.cox.L)$table[1,2]
mat.cox.L[i,15]<-summary(exp.model.nocov.cox.L)$table[1,3]
mat.cox.L[i,16]<-summary(exp.model.nocov.cox.L)$table[1,4]

logl.model.nocov.cox.L=survreg(Surv(y, failed) ~ 1,dist="loglogistic",
data=sim.data.nocov.cox.L$data)

```

```

mat.cox.L[i,19]<-summary(logl.model.nocov.cox.L)$coefficients
mat.cox.L[i,20]<-summary(logl.model.nocov.cox.L)$table[1,2]
mat.cox.L[i,21]<-summary(logl.model.nocov.cox.L)$table[1,3]
mat.cox.L[i,22]<-summary(logl.model.nocov.cox.L)$table[1,4]

logNorm.model.nocov.cox.L=survreg(Surv(y, failed) ~ 1,dist="lognormal",
  data=sim.data.nocov.cox.L$data)

mat.cox.L[i,25]<-summary(logNorm.model.nocov.cox.L)$coefficients
mat.cox.L[i,26]<-summary(logNorm.model.nocov.cox.L)$table[1,2]
mat.cox.L[i,27]<-summary(logNorm.model.nocov.cox.L)$table[1,3]
mat.cox.L[i,28]<-summary(logNorm.model.nocov.cox.L)$table[1,4]

#####AIC

mat.cox.L[i,5]<-extractAIC(cox.model.nocov.cox.L)[2]
mat.cox.L[i,11]<-extractAIC(weib.model.nocov.cox.L)[2]
mat.cox.L[i,17]<-extractAIC(exp.model.nocov.cox.L)[2]
mat.cox.L[i,23]<-extractAIC(logl.model.nocov.cox.L)[2]
mat.cox.L[i,29]<-extractAIC(logNorm.model.nocov.cox.L)[2]

```



```

mat.cox.L[i,6]=AIC(cox.model.nocov.cox.L,k=log(length(start)))
mat.cox.L[i,12]=AIC(weib.model.nocov.cox.L,k=log(length(start)))
mat.cox.L[i,18]=AIC(exp.model.nocov.cox.L,k=log(length(start)))
mat.cox.L[i,24]=AIC(logl.model.nocov.cox.L,k=log(length(start)))
mat.cox.L[i,30]=AIC(logNorm.model.nocov.cox.L,k=log(length(start)))
}

colMeans(mat.cox.L)
results.cox.L=data.frame(matrix(colMeans(mat.cox.L),ncol=6,byrow=T))
rownames(results.cox.L)=c("cox","weib","exp","logl","logNorm")
colnames(results.cox.L)=c("intercept","stderr","statistic","p-value",
"AIC","BIC")
results.cox.L

#####
#####
simulation with predictors
#####
#####

###simulation of Weibull censord time to event data (small data)

```

```

sim.size<-1000

mat1<-matrix(0,ncol=70,nrow=sim.size)

for(i in 1:sim.size){

sim.data.cov.w.s=simple.surv.sim(n=10,foltime=360,dist.ev=c('weibull'),
anc.ev=c(0.69978200185280),beta0.ev=c(5.84298525742252),
anc.cens=1.17783687569519, beta0.cens=7.39773677281100,
z=list(c("unif",0.8,1.2)),dist.cens="weibull",beta=list(c(-0.4),
c(0)), x=list(c("bern", 0.5), c("unif", 0.7, 1.3)))

data.frame(sim.data.cov.w.s)

attach(sim.data.cov.w.s)

cox.model.cov.w.s =coxph(Surv(stop,status) ~ as.factor(x) + x.1,
data=sim.data.cov.w.s,method="breslow")

mat1[i,5]<-summary(cox.model.cov.w.s )$coefficients[1,1]#coeff
mat1[i,6]<-summary(cox.model.cov.w.s )$coefficients[1,3]#std.err
mat1[i,7]<-summary(cox.model.cov.w.s )$coefficients[1,4]#statistic
mat1[i,8]<-summary(cox.model.cov.w.s )$coefficients[1,5]#p-val
mat1[i,9]<-summary(cox.model.cov.w.s )$coefficients[2,1]#coeff
mat1[i,10]<-summary(cox.model.cov.w.s )$coefficients[2,3]#std.err
mat1[i,11]<-summary(cox.model.cov.w.s )$coefficients[2,4]#statistic
mat1[i,12]<-summary(cox.model.cov.w.s )$coefficients[2,5]#p-val

```

```

weib.model.cov.w.s=survreg(Surv(stop,status)~as.factor(x)+x.1,
  dist="weibull",data=sim.data.cov.w.s)

mat1[i,15]<-summary(weib.model.cov.w.s)$table[1,1]#coeff
mat1[i,16]<-summary(weib.model.cov.w.s)$table[1,2]#std.err
mat1[i,17]<-summary(weib.model.cov.w.s)$table[1,3]#statistic
mat1[i,18]<-summary(weib.model.cov.w.s)$table[1,4]#p-val
mat1[i,19]<-summary(weib.model.cov.w.s)$table[2,1]#coeff
mat1[i,20]<-summary(weib.model.cov.w.s)$table[2,2]#std.err
mat1[i,21]<-summary(weib.model.cov.w.s)$table[2,3]#statistic
mat1[i,22]<-summary(weib.model.cov.w.s)$table[2,4]#p-val
mat1[i,23]<-summary(weib.model.cov.w.s)$table[3,1]#coeff
mat1[i,24]<-summary(weib.model.cov.w.s)$table[3,2]#std.err
mat1[i,25]<-summary(weib.model.cov.w.s)$table[3,3]#statistic
mat1[i,26]<-summary(weib.model.cov.w.s)$table[3,4]#p-val

exp.model.cov.w.s=survreg(Surv(stop,status)~as.factor(x)+x.1,
  dist="exp",data=sim.data.cov.w.s)

mat1[i,29]<-summary(exp.model.cov.w.s)$table[1,1]#coeff
mat1[i,30]<-summary(exp.model.cov.w.s)$table[1,2]#std.err
mat1[i,31]<-summary(exp.model.cov.w.s)$table[1,3]#statistic

```

```

mat1[i,32]<-summary(exp.model.cov.w.s)$table[1,4]#p-val
mat1[i,33]<-summary(exp.model.cov.w.s)$table[2,1]#coeff
mat1[i,34]<-summary(exp.model.cov.w.s)$table[2,2]#std.err
mat1[i,35]<-summary(exp.model.cov.w.s)$table[2,3]#statistic
mat1[i,36]<-summary(exp.model.cov.w.s)$table[2,4]#p-val
mat1[i,37]<-summary(exp.model.cov.w.s)$table[3,1]#coeff
mat1[i,38]<-summary(exp.model.cov.w.s)$table[3,2]#std.err
mat1[i,39]<-summary(exp.model.cov.w.s)$table[3,3]#statistic
mat1[i,40]<-summary(exp.model.cov.w.s)$table[3,4]#p-val

log1.model.cov.w.s=survreg(Surv(stop,status)~as.factor(x)+x.1,
  dist="loglogistic",data=sim.data.cov.w.s)

mat1[i,43]<-summary(log1.model.cov.w.s)$table[1,1]#coeff
mat1[i,44]<-summary(log1.model.cov.w.s)$table[1,2]#std.err
mat1[i,45]<-summary(log1.model.cov.w.s)$table[1,3]#statistic
mat1[i,46]<-summary(log1.model.cov.w.s)$table[1,4]#p-val
mat1[i,47]<-summary(log1.model.cov.w.s)$table[2,1]#coeff
mat1[i,48]<-summary(log1.model.cov.w.s)$table[2,2]#std.err
mat1[i,49]<-summary(log1.model.cov.w.s)$table[2,3]#statistic
mat1[i,50]<-summary(log1.model.cov.w.s)$table[2,4]#p-val
mat1[i,51]<-summary(log1.model.cov.w.s)$table[3,1]#coeff
mat1[i,52]<-summary(log1.model.cov.w.s)$table[3,2]#std.err

```

```

mat1[i,53]<-summary(logl.model.cov.w.s)$table[3,3]#statistic
mat1[i,54]<-summary(logl.model.cov.w.s)$table[3,4]#p-val

logNorm.model.cov.w.s=survreg(Surv(stop,status)~as.factor(x)+x.1,
  dist="lognormal",data=sim.data.cov.w.s)

mat1[i,57]<-summary(logNorm.model.cov.w.s)$table[1,1] #coeff
mat1[i,58]<-summary(logNorm.model.cov.w.s)$table[1,2] #std.err
mat1[i,59]<-summary(logNorm.model.cov.w.s)$table[1,3]#statistic
mat1[i,60]<-summary(logNorm.model.cov.w.s)$table[1,4] #p-val
mat1[i,61]<-summary(logNorm.model.cov.w.s)$table[2,1] #coeff
mat1[i,62]<-summary(logNorm.model.cov.w.s)$table[2,2] #std.err
mat1[i,63]<-summary(logNorm.model.cov.w.s)$table[2,3]#statistic
mat1[i,64]<-summary(logNorm.model.cov.w.s)$table[2,4] #p-val
mat1[i,65]<-summary(logNorm.model.cov.w.s)$table[3,1] #coeff
mat1[i,66]<-summary(logNorm.model.cov.w.s)$table[3,2] #std.err
mat1[i,67]<-summary(logNorm.model.cov.w.s)$table[3,3]#statistic
mat1[i,68]<-summary(logNorm.model.cov.w.s)$table[3,4] #p-val

#####AIC

mat1[i,13]<-extractAIC(cox.model.cov.w.s)[2];

```

```

mat1[i,14]<-AIC(cox.model.cov.w.s,k = log(length(start)))
mat1[i,27]<-extractAIC(weib.model.cov.w.s)[2];
mat1[i,28]<-AIC(weib.model.cov.w.s,k = log(length(start)))
mat1[i,41]<-extractAIC(exp.model.cov.w.s)[2];
mat1[i,42]<-AIC(exp.model.cov.w.s,k = log(length(start)))
mat1[i,55]<-extractAIC(logl.model.cov.w.s)[2];
mat1[i,56]<-AIC(logl.model.cov.w.s,k = log(length(start)))
mat1[i,69]<-extractAIC(logNorm.model.cov.w.s)[2];
mat1[i,70]<-AIC(logNorm.model.cov.w.s,k = log(length(start)))

}

```

```

colMeans(mat1)

result1<-data.frame(matrix(colMeans(mat1),ncol=14,byrow=T))
rownames(result1)<-c("cox","weib","exp","logl","logNorm")
colnames(result1)<-c("intercept","stderr","statistic",
"p-value","coeff1","stderr","statistic","p-value","coeff2",
"stderr","statistic","p-value","AIC","BIC")

result1

```

```

###simulation of Weibull censord time to event data

sim.size.cov.w.m<-500

mat1.cov.w.m<-matrix(0,ncol=70,nrow=sim.size.cov.w.m)

for(i in 1:sim.size){

sim.data.cov.w.m= simple.surv.sim(n=1000, foltime=360,

  dist.ev=c('weibull'),

  anc.ev=c(0.69978200185280),beta0.ev=c(5.84298525742252),

  anc.cens=1.17783687569519,beta0.cens=7.39773677281100,

  z=list(c("unif", 0.8, 1.2)), dist.cens="weibull", beta=list(c(-0.4),

  c(0)), x=list(c("bern", 0.5), c("unif", 0.7, 1.3)))

data.frame(sim.data.cov.w.m)

attach(sim.data.cov.w.m)

cox.model.cov.w.m =coxph(Surv(stop,status)~as.factor(x)+x.1,

  data=sim.data.cov.w.m,method="breslow")

mat1.cov.w.m[i,5]=summary(cox.model.cov.w.m)$coefficients[1,1]#coeff

mat1.cov.w.m[i,6]=summary(cox.model.cov.w.m)$coefficients[1,3]#std.err

mat1.cov.w.m[i,7]=summary(cox.model.cov.w.m)$coefficients[1,4]#statistic

mat1.cov.w.m[i,8]=summary(cox.model.cov.w.m )$coefficients[1,5]#p-val

```

```

mat1.cov.w.m[i,9]=summary(cox.model.cov.w.m)$coefficients[2,1]#coeff
mat1.cov.w.m[i,10]=summary(cox.model.cov.w.m)$coefficients[2,3]#std.err
mat1.cov.w.m[i,11]=summary(cox.model.cov.w.m)$coefficients[2,4]#statistic
mat1.cov.w.m[i,12]=summary(cox.model.cov.w.m)$coefficients[2,5]#p-val

weib.model.cov.w.m=survreg(Surv(stop, status)~as.factor(x)+x.1,
  dist="weibull",data=sim.data.cov.w.m)

mat1.cov.w.m[i,15]=summary(weib.model.cov.w.m)$table[1,1]#coeff
mat1.cov.w.m[i,16]=summary(weib.model.cov.w.m)$table[1,2]#std.err
mat1.cov.w.m[i,17]=summary(weib.model.cov.w.m)$table[1,3]#statistic
mat1.cov.w.m[i,18]=summary(weib.model.cov.w.m)$table[1,4]#p-val
mat1.cov.w.m[i,19]=summary(weib.model.cov.w.m)$table[2,1]#coeff
mat1.cov.w.m[i,20]=summary(weib.model.cov.w.m)$table[2,2]#std.err
mat1.cov.w.m[i,21]=summary(weib.model.cov.w.m)$table[2,3]#statistic
mat1.cov.w.m[i,22]=summary(weib.model.cov.w.m)$table[2,4]#p-val
mat1.cov.w.m[i,23]=summary(weib.model.cov.w.m)$table[3,1]#coeff
mat1.cov.w.m[i,24]=summary(weib.model.cov.w.m)$table[3,2]#std.err
mat1.cov.w.m[i,25]=summary(weib.model.cov.w.m)$table[3,3]#statistic
mat1.cov.w.m[i,26]=summary(weib.model.cov.w.m)$table[3,4]#p-val

exp.model.cov.w.m=survreg(Surv(stop, status) ~ as.factor(x) + x.1,
  dist="exp",data=sim.data.cov.w.m)

```



```

mat1.cov.w.m[i,29]=summary(exp.model.cov.w.m)$table[1,1]#coeff
mat1.cov.w.m[i,30]=summary(exp.model.cov.w.m)$table[1,2]#std.err
mat1.cov.w.m[i,31]=summary(exp.model.cov.w.m)$table[1,3]#statistic
mat1.cov.w.m[i,32]=summary(exp.model.cov.w.m)$table[1,4]#p-val
mat1.cov.w.m[i,33]=summary(exp.model.cov.w.m)$table[2,1]#coeff
mat1.cov.w.m[i,34]=summary(exp.model.cov.w.m)$table[2,2]#std.err
mat1.cov.w.m[i,35]=summary(exp.model.cov.w.m)$table[2,3]#statistic
mat1.cov.w.m[i,36]=summary(exp.model.cov.w.m)$table[2,4]#p-val
mat1.cov.w.m[i,37]=summary(exp.model.cov.w.m)$table[3,1]#coeff
mat1.cov.w.m[i,38]=summary(exp.model.cov.w.m)$table[3,2]#std.err
mat1.cov.w.m[i,39]=summary(exp.model.cov.w.m)$table[3,3]#statistic
mat1.cov.w.m[i,40]=summary(exp.model.cov.w.m)$table[3,4] #p-val

log1.model.cov.w.m= survreg(Surv(stop, status)~as.factor(x)+x.1,
  dist="loglogistic",data=sim.data.cov.w.m)

mat1.cov.w.m[i,43]=summary(log1.model.cov.w.m)$table[1,1]#coeff
mat1.cov.w.m[i,44]=summary(log1.model.cov.w.m)$table[1,2]#std.err
mat1.cov.w.m[i,45]=summary(log1.model.cov.w.m)$table[1,3]#statistic
mat1.cov.w.m[i,46]=summary(log1.model.cov.w.m)$table[1,4]#p-val
mat1.cov.w.m[i,47]=summary(log1.model.cov.w.m)$table[2,1]#coeff
mat1.cov.w.m[i,48]=summary(log1.model.cov.w.m)$table[2,2]#std.err

```

```

mat1.cov.w.m[i,49]=summary(logl.model.cov.w.m)$table[2,3]#statistic
mat1.cov.w.m[i,50]=summary(logl.model.cov.w.m)$table[2,4]#p-val
mat1.cov.w.m[i,51]=summary(logl.model.cov.w.m)$table[3,1]#coeff
mat1.cov.w.m[i,52]=summary(logl.model.cov.w.m)$table[3,2]#std.err
mat1.cov.w.m[i,53]=summary(logl.model.cov.w.m)$table[3,3]#statistic
mat1.cov.w.m[i,54]=summary(logl.model.cov.w.m)$table[3,4]#p-val

logNorm.model.cov.w.m=survreg(Surv(stop,status)~as.factor(x)+ x.1,
  dist="lognormal",data=sim.data.cov.w.m)

mat1.cov.w.m[i,57]=summary(logNorm.model.cov.w.m)$table[1,1]#coeff
mat1.cov.w.m[i,58]=summary(logNorm.model.cov.w.m)$table[1,2]#std.err
mat1.cov.w.m[i,59]=summary(logNorm.model.cov.w.m)$table[1,3]#statistic
mat1.cov.w.m[i,60]=summary(logNorm.model.cov.w.m)$table[1,4]#p-val
mat1.cov.w.m[i,61]=summary(logNorm.model.cov.w.m)$table[2,1]#coeff
mat1.cov.w.m[i,62]=summary(logNorm.model.cov.w.m)$table[2,2]#std.err
mat1.cov.w.m[i,63]=summary(logNorm.model.cov.w.m)$table[2,3]#statistic
mat1.cov.w.m[i,64]=summary(logNorm.model.cov.w.m)$table[2,4]#p-val
mat1.cov.w.m[i,65]=summary(logNorm.model.cov.w.m)$table[3,1]#coeff
mat1.cov.w.m[i,66]=summary(logNorm.model.cov.w.m)$table[3,2]#std.err
mat1.cov.w.m[i,67]=summary(logNorm.model.cov.w.m)$table[3,3]#statistic
mat1.cov.w.m[i,68]=summary(logNorm.model.cov.w.m)$table[3,4]#p-val

```

```
#####AIC
```

```
mat1.cov.w.m[i,13]<-extractAIC(cox.model.cov.w.m)[2];  
mat1.cov.w.m[i,14]<-AIC(cox.model.cov.w.m,k = log(length(start)))  
mat1.cov.w.m[i,27]<-extractAIC(weib.model.cov.w.m)[2];  
mat1.cov.w.m[i,28]<-AIC(weib.model.cov.w.m,k = log(length(start)))  
mat1.cov.w.m[i,41]<-extractAIC(exp.model.cov.w.m)[2];  
mat1.cov.w.m[i,42]<-AIC(exp.model.cov.w.m,k = log(length(start)))  
mat1.cov.w.m[i,55]<-extractAIC(logl.model.cov.w.m)[2];  
mat1.cov.w.m[i,56]<-AIC(logl.model.cov.w.m,k = log(length(start)))  
mat1.cov.w.m[i,69]<-extractAIC(logNorm.model.cov.w.m)[2];  
mat1.cov.w.m[i,70]<-AIC(logNorm.model.cov.w.m,k=log(length(start)))  
  
}
```

```
colMeans(mat1.cov.w.m)  
result1.cov.w.m=data.frame(matrix(colMeans(mat1.cov.w.m),ncol=14,byrow=T))  
rownames(result1.cov.w.m)<-c("cox","weib","exp","logl","logNorm")  
colnames(result1.cov.w.m)<-c("intercept","stderr","statistic","p-value",  
"coeff1","stderr","statistic","p-value","coeff2","stderr","statistic",
```

```

"p-value", "AIC", "BIC")

result1.cov.w.m

#####

#####

###simulation of Weibull censord time to event data(large data)

sim.size.cov.w.L<-50

mat1.cov.w.L<-matrix(0,ncol=70,nrow=sim.size.cov.w.L)

for(i in 1:sim.size.cov.w.L){

sim.data.cov.w.L= simple.surv.sim(n=200000, foltime=360,

dist.ev=c('weibull'), anc.ev=c(0.69978200185280),

beta0.ev=c(5.84298525742252)

, anc.cens=1.17783687569519, beta0.cens=7.39773677281100,

z=list(c("unif", 0.8, 1.2)), dist.cens="weibull", beta=list(c(-0.4),

c(0)), x=list(c("bern", 0.5), c("unif", 0.7, 1.3)))

data.frame(sim.data.cov.w.L)

attach(sim.data.cov.w.L)

```

```

cox.model.cov.w.L =coxph(Surv(stop, status) ~ as.factor(x) + x.1,
  data=sim.data.cov.w.L,method="breslow")

mat1.cov.w.L[i,5]=summary(cox.model.cov.w.L)$coefficients[1,1]#coeff
mat1.cov.w.L[i,6]=summary(cox.model.cov.w.L)$coefficients[1,3]#std.err
mat1.cov.w.L[i,7]=summary(cox.model.cov.w.L)$coefficients[1,4]#statistic
mat1.cov.w.L[i,8]=summary(cox.model.cov.w.L)$coefficients[1,5]#p-val
mat1.cov.w.L[i,9]=summary(cox.model.cov.w.L)$coefficients[2,1]#coeff
mat1.cov.w.L[i,10]=summary(cox.model.cov.w.L)$coefficients[2,3]#std.err
mat1.cov.w.L[i,11]=summary(cox.model.cov.w.L)$coefficients[2,4]#statistic
mat1.cov.w.L[i,12]=summary(cox.model.cov.w.L)$coefficients[2,5]#p-val

weib.model.cov.w.L= survreg(Surv(stop, status) ~ as.factor(x) + x.1,
  dist="weibull",data=sim.data.cov.w.L)

mat1.cov.w.L[i,15]=summary(weib.model.cov.w.L)$table[1,1]#coeff
mat1.cov.w.L[i,16]=summary(weib.model.cov.w.L)$table[1,2]#std.err
mat1.cov.w.L[i,17]=summary(weib.model.cov.w.L)$table[1,3]#statistic
mat1.cov.w.L[i,18]=summary(weib.model.cov.w.L)$table[1,4]#p-val
mat1.cov.w.L[i,19]=summary(weib.model.cov.w.L)$table[2,1]#coeff
mat1.cov.w.L[i,20]=summary(weib.model.cov.w.L)$table[2,2]#std.err
mat1.cov.w.L[i,21]=summary(weib.model.cov.w.L)$table[2,3]#statistic
mat1.cov.w.L[i,22]=summary(weib.model.cov.w.L)$table[2,4]#p-val

```

```
mat1.cov.w.L[i,23]=summary(weib.model.cov.w.L)$table[3,1]#coeff
mat1.cov.w.L[i,24]=summary(weib.model.cov.w.L)$table[3,2]#std.err
mat1.cov.w.L[i,25]=summary(weib.model.cov.w.L)$table[3,3]#statistic
mat1.cov.w.L[i,26]=summary(weib.model.cov.w.L)$table[3,4]#p-val
```

```
exp.model.cov.w.L=survreg(Surv(stop, status)~as.factor(x)+ x.1,
  dist="exp",data=sim.data.cov.w.L)
```

```
mat1.cov.w.L[i,29]=summary(exp.model.cov.w.L)$table[1,1]#coeff
mat1.cov.w.L[i,30]=summary(exp.model.cov.w.L)$table[1,2]#std.err
mat1.cov.w.L[i,31]=summary(exp.model.cov.w.L)$table[1,3]#statistic
mat1.cov.w.L[i,32]=summary(exp.model.cov.w.L)$table[1,4]#p-val
mat1.cov.w.L[i,33]=summary(exp.model.cov.w.L)$table[2,1]#coeff
mat1.cov.w.L[i,34]=summary(exp.model.cov.w.L)$table[2,2]#std.err
mat1.cov.w.L[i,35]=summary(exp.model.cov.w.L)$table[2,3]#statistic
mat1.cov.w.L[i,36]=summary(exp.model.cov.w.L)$table[2,4]#p-val
mat1.cov.w.L[i,37]=summary(exp.model.cov.w.L)$table[3,1]#coeff
mat1.cov.w.L[i,38]=summary(exp.model.cov.w.L)$table[3,2]#std.err
mat1.cov.w.L[i,39]=summary(exp.model.cov.w.L)$table[3,3]#statistic
mat1.cov.w.L[i,40]=summary(exp.model.cov.w.L)$table[3,4]#p-val
```

```
log1.model.cov.w.L= survreg(Surv(stop, status) ~ as.factor(x) + x.1,
```

```

dist="loglogistic",data=sim.data.cov.w.L)

mat1.cov.w.L[i,43]=summary(logl.model.cov.w.L)$table[1,1]#coeff
mat1.cov.w.L[i,44]=summary(logl.model.cov.w.L)$table[1,2]#std.err
mat1.cov.w.L[i,45]=summary(logl.model.cov.w.L)$table[1,3]#statistic
mat1.cov.w.L[i,46]=summary(logl.model.cov.w.L)$table[1,4]#p-val
mat1.cov.w.L[i,47]=summary(logl.model.cov.w.L)$table[2,1]#coeff
mat1.cov.w.L[i,48]=summary(logl.model.cov.w.L)$table[2,2]#std.err
mat1.cov.w.L[i,49]=summary(logl.model.cov.w.L)$table[2,3]#statistic
mat1.cov.w.L[i,50]=summary(logl.model.cov.w.L)$table[2,4]#p-val
mat1.cov.w.L[i,51]=summary(logl.model.cov.w.L)$table[3,1]#coeff
mat1.cov.w.L[i,52]=summary(logl.model.cov.w.L)$table[3,2]#std.err
mat1.cov.w.L[i,53]=summary(logl.model.cov.w.L)$table[3,3]#statistic
mat1.cov.w.L[i,54]=summary(logl.model.cov.w.L)$table[3,4]#p-val

logNorm.model.cov.w.L= survreg(Surv(stop, status)~as.factor(x)+x.1,
dist="lognormal",data=sim.data.cov.w.L)

mat1.cov.w.L[i,57]=summary(logNorm.model.cov.w.L)$table[1,1]#coeff
mat1.cov.w.L[i,58]=summary(logNorm.model.cov.w.L)$table[1,2]#std.err
mat1.cov.w.L[i,59]=summary(logNorm.model.cov.w.L)$table[1,3]#statistic
mat1.cov.w.L[i,60]=summary(logNorm.model.cov.w.L)$table[1,4]#p-val
mat1.cov.w.L[i,61]=summary(logNorm.model.cov.w.L)$table[2,1]#coeff
mat1.cov.w.L[i,62]=summary(logNorm.model.cov.w.L)$table[2,2]#std.err

```

```

mat1.cov.w.L[i,63]=summary(logNorm.model.cov.w.L)$table[2,3]#statistic
mat1.cov.w.L[i,64]=summary(logNorm.model.cov.w.L)$table[2,4]#p-val
mat1.cov.w.L[i,65]=summary(logNorm.model.cov.w.L)$table[3,1]#coeff
mat1.cov.w.L[i,66]=summary(logNorm.model.cov.w.L)$table[3,2]#std.err
mat1.cov.w.L[i,67]=summary(logNorm.model.cov.w.L)$table[3,3]#statistic
mat1.cov.w.L[i,68]=summary(logNorm.model.cov.w.L)$table[3,4]#p-val

#####AIC

mat1.cov.w.L[i,13]<-extractAIC(cox.model.cov.w.L)[2];
mat1.cov.w.L[i,14]<-AIC(cox.model.cov.w.L,k = log(length(start)))
mat1.cov.w.L[i,27]<-extractAIC(weib.model.cov.w.L)[2];
mat1.cov.w.L[i,28]<-AIC(weib.model.cov.w.L,k = log(length(start)))
mat1.cov.w.L[i,41]<-extractAIC(exp.model.cov.w.L)[2];
mat1.cov.w.L[i,42]<-AIC(exp.model.cov.w.L,k = log(length(start)))
mat1.cov.w.L[i,55]<-extractAIC(logl.model.cov.w.L)[2];
mat1.cov.w.L[i,56]<-AIC(logl.model.cov.w.L,k = log(length(start)))
mat1.cov.w.L[i,69]<-extractAIC(logNorm.model.cov.w.L)[2];
mat1.cov.w.L[i,70]<-AIC(logNorm.model.cov.w.L,k = log(length(start)))

}

```



```

colMeans(mat1.cov.w.L)

result1.cov.w.L=data.frame(matrix(colMeans(mat1.cov.w.L),ncol=14,
byrow=T))

rownames(result1.cov.w.L)=c("cox","weib","exp","log1","logNorm")

colnames(result1.cov.w.L)<-c("intercept","stderr","statistic","p-value",
"coeff1","stderr","statistic","p-value","coeff2","stderr","statistic",
"p-value","AIC","BIC")

result1.cov.w.L

#####

#####

Simulation of exponential with covariate

#####

#####

###simulation of exponential censord time to event data(small)

sim.size.cov.e.s<-1000

mat1.cov.e.s<-matrix(0,ncol=70,nrow=sim.size.cov.e.s)

```

```

for(i in 1:sim.size.cov.e.s){
sim.data.cov.e.s= simple.surv.sim(n=10, foltime=360,
dist.ev=c('weibull'),
anc.ev=c(0.69978200185280),beta0.ev=c(5.84298525742252),anc.cens=1,
beta0.cens=7.39773677281100,z=list(c("unif", 0.8, 1.2)),
dist.cens="weibull", beta=list(c(-0.4),
c(0)), x=list(c("bern", 0.5), c("unif", 0.7, 1.3)))

data.frame(sim.data.cov.e.s)

attach(sim.data.cov.e.s)

cox.model.cov.e.s =coxph(Surv(stop, status) ~ as.factor(x) + x.1,
data=sim.data.cov.e.s,method="breslow")

mat1.cov.e.s[i,5]<-summary(cox.model.cov.e.s)$coefficients[1,1]#coeff
mat1.cov.e.s[i,6]=summary(cox.model.cov.e.s)$coefficients[1,3]#std.err
mat1.cov.e.s[i,7]=summary(cox.model.cov.e.s)$coefficients[1,4]#statistic
mat1.cov.e.s[i,8]=summary(cox.model.cov.e.s)$coefficients[1,5]#p-val
mat1.cov.e.s[i,9]=summary(cox.model.cov.e.s)$coefficients[2,1]#coeff
mat1.cov.e.s[i,10]=summary(cox.model.cov.e.s)$coefficients[2,3]#std.err
mat1.cov.e.s[i,11]=summary(cox.model.cov.e.s)$coefficients[2,4]#statistic
mat1.cov.e.s[i,12]=summary(cox.model.cov.e.s)$coefficients[2,5]#p-val

```

```

weib.model.cov.e.s= survreg(Surv(stop, status) ~ as.factor(x) + x.1,
  dist="weibull",data=sim.data.cov.e.s)

mat1.cov.e.s[i,15]=summary(weib.model.cov.e.s)$table[1,1]#coeff
mat1.cov.e.s[i,16]=summary(weib.model.cov.e.s)$table[1,2]#std.err
mat1.cov.e.s[i,17]=summary(weib.model.cov.e.s)$table[1,3]#statistic
mat1.cov.e.s[i,18]=summary(weib.model.cov.e.s)$table[1,4]#p-val
mat1.cov.e.s[i,19]=summary(weib.model.cov.e.s)$table[2,1]#coeff
mat1.cov.e.s[i,20]=summary(weib.model.cov.e.s)$table[2,2]#std.err
mat1.cov.e.s[i,21]=summary(weib.model.cov.e.s)$table[2,3]#statistic
mat1.cov.e.s[i,22]=summary(weib.model.cov.e.s)$table[2,4]#p-val
mat1.cov.e.s[i,23]=summary(weib.model.cov.e.s)$table[3,1]#coeff
mat1.cov.e.s[i,24]=summary(weib.model.cov.e.s)$table[3,2]#std.err
mat1.cov.e.s[i,25]=summary(weib.model.cov.e.s)$table[3,3]#statistic
mat1.cov.e.s[i,26]=summary(weib.model.cov.e.s)$table[3,4]#p-val

exp.model.cov.e.s=survreg(Surv(stop, status)~as.factor(x) + x.1,
  dist="exp",data=sim.data.cov.e.s)

mat1.cov.e.s[i,29]=summary(exp.model.cov.e.s)$table[1,1]#coeff
mat1.cov.e.s[i,30]=summary(exp.model.cov.e.s)$table[1,2]#std.err
mat1.cov.e.s[i,31]=summary(exp.model.cov.e.s)$table[1,3]#statistic
mat1.cov.e.s[i,32]=summary(exp.model.cov.e.s)$table[1,4]#p-val

```

```

mat1.cov.e.s[i,33]=summary(exp.model.cov.e.s)$table[2,1]#coeff
mat1.cov.e.s[i,34]=summary(exp.model.cov.e.s)$table[2,2]#std.err
mat1.cov.e.s[i,35]=summary(exp.model.cov.e.s)$table[2,3]#statistic
mat1.cov.e.s[i,36]=summary(exp.model.cov.e.s)$table[2,4]#p-val
mat1.cov.e.s[i,37]=summary(exp.model.cov.e.s)$table[3,1]#coeff
mat1.cov.e.s[i,38]=summary(exp.model.cov.e.s)$table[3,2]#std.err
mat1.cov.e.s[i,39]=summary(exp.model.cov.e.s)$table[3,3]#statistic
mat1.cov.e.s[i,40]=summary(exp.model.cov.e.s)$table[3,4]#p-val

log1.model.cov.e.s=survreg(Surv(stop,status)~as.factor(x)+x.1,
  dist="loglogistic",data=sim.data.cov.e.s)

mat1.cov.e.s[i,43]=summary(log1.model.cov.e.s)$table[1,1]#coeff
mat1.cov.e.s[i,44]=summary(log1.model.cov.e.s)$table[1,2]#std.err
mat1.cov.e.s[i,45]=summary(log1.model.cov.e.s)$table[1,3]#statistic
mat1.cov.e.s[i,46]=summary(log1.model.cov.e.s)$table[1,4]#p-val
mat1.cov.e.s[i,47]=summary(log1.model.cov.e.s)$table[2,1]#coeff
mat1.cov.e.s[i,48]=summary(log1.model.cov.e.s)$table[2,2]#std.err
mat1.cov.e.s[i,49]=summary(log1.model.cov.e.s)$table[2,3]#statistic
mat1.cov.e.s[i,50]=summary(log1.model.cov.e.s)$table[2,4]#p-val
mat1.cov.e.s[i,51]=summary(log1.model.cov.e.s)$table[3,1]#coeff
mat1.cov.e.s[i,52]=summary(log1.model.cov.e.s)$table[3,2]#std.err
mat1.cov.e.s[i,53]=summary(log1.model.cov.e.s)$table[3,3]#statistic

```

```

mat1.cov.e.s[i,54]=summary(logl.model.cov.e.s)$table[3,4]#p-val

logNorm.model.cov.e.s=survreg(Surv(stop,status)~as.factor(x)+x.1,
  dist="lognormal",data=sim.data.cov.e.s)

mat1.cov.e.s[i,57]=summary(logNorm.model.cov.e.s)$table[1,1]#coeff
mat1.cov.e.s[i,58]=summary(logNorm.model.cov.e.s)$table[1,2]#std.err
mat1.cov.e.s[i,59]=summary(logNorm.model.cov.e.s)$table[1,3]#statistic
mat1.cov.e.s[i,60]=summary(logNorm.model.cov.e.s)$table[1,4]#p-val
mat1.cov.e.s[i,61]=summary(logNorm.model.cov.e.s)$table[2,1]#coeff
mat1.cov.e.s[i,62]=summary(logNorm.model.cov.e.s)$table[2,2]#std.err
mat1.cov.e.s[i,63]=summary(logNorm.model.cov.e.s)$table[2,3]#statistic
mat1.cov.e.s[i,64]=summary(logNorm.model.cov.e.s)$table[2,4]#p-val
mat1.cov.e.s[i,65]=summary(logNorm.model.cov.e.s)$table[3,1]#coeff
mat1.cov.e.s[i,66]=summary(logNorm.model.cov.e.s)$table[3,2]#std.err
mat1.cov.e.s[i,67]=summary(logNorm.model.cov.e.s)$table[3,3]#statistic
mat1.cov.e.s[i,68]=summary(logNorm.model.cov.e.s)$table[3,4]#p-val

#####AIC

mat1.cov.e.s[i,13]=extractAIC(cox.model.cov.e.s)[2];

```

```

mat1.cov.e.s[i,14]=AIC(cox.model.cov.e.s,k = log(length(start)))
mat1.cov.e.s[i,27]=extractAIC(weib.model.cov.e.s)[2];
mat1.cov.e.s[i,28]=AIC(weib.model.cov.e.s,k = log(length(start)))
mat1.cov.e.s[i,41]=extractAIC(exp.model.cov.e.s)[2];
mat1.cov.e.s[i,42]=AIC(exp.model.cov.e.s,k = log(length(start)))
mat1.cov.e.s[i,55]=extractAIC(logl.model.cov.e.s)[2];
mat1.cov.e.s[i,56]=AIC(logl.model.cov.e.s,k = log(length(start)))
mat1.cov.e.s[i,69]=extractAIC(logNorm.model.cov.e.s)[2];
mat1.cov.e.s[i,70]=AIC(logNorm.model.cov.e.s,k=log(length(start)))

}

colMeans(mat1.cov.e.s)

result1.cov.e.s=data.frame(matrix(colMeans(mat1.cov.e.s),ncol=14,
byrow=T))

rownames(result1.cov.e.s)<-c("cox","weib","exp","logl","logNorm")
colnames(result1.cov.e.s)<-c("intercept","stderr","statistic",
"p-value","coeff1","stderr","statistic","p-value","coeff2","stderr",
"statistic","p-value","AIC","BIC")

result1.cov.e.s

```

```

##simulation of exponential censord time to event data(medium data)

sim.size.cov.e.m<-1000

mat1.cov.e.m<-matrix(0,ncol=70,nrow=sim.size.cov.e.m)

for(i in 1:sim.size.cov.e.m){

sim.data.cov.e.m=simple.surv.sim(n=250,foltime=360,dist.ev=c('weibull')
,anc.ev=c(0.9999),beta0.ev=c(5.84298525742252),anc.cens=1,
beta0.cens=7.39773677281100,z=list(c("unif", 0.8, 1.2)),
dist.cens="weibull", beta=list(c(-0.4),
c(0)), x=list(c("bern", 0.5), c("unif", 0.7, 1.3)))

data.frame(sim.data.cov.e.m)

attach(sim.data.cov.e.m)

cox.model.cov.e.m =coxph(Surv(stop, status) ~ as.factor(x) + x.1,
data=sim.data.cov.e.m,method="breslow")

mat1.cov.e.m[i,5]=summary(cox.model.cov.e.m)$coefficients[1,1]#coeff
mat1.cov.e.m[i,6]=summary(cox.model.cov.e.m)$coefficients[1,3] #std.err
mat1.cov.e.m[i,7]=summary(cox.model.cov.e.m)$coefficients[1,4]#statistic
mat1.cov.e.m[i,8]=summary(cox.model.cov.e.m)$coefficients[1,5]#p-val

```

```

mat1.cov.e.m[i,9]=summary(cox.model.cov.e.m)$coefficients[2,1]#coeff
mat1.cov.e.m[i,10]=summary(cox.model.cov.e.m)$coefficients[2,3]#std.err
mat1.cov.e.m[i,11]=summary(cox.model.cov.e.m)$coefficients[2,4]#statistic
mat1.cov.e.m[i,12]=summary(cox.model.cov.e.m )$coefficients[2,5]#p-val

weib.model.cov.e.m= survreg(Surv(stop, status) ~ as.factor(x) + x.1,
  dist="weibull",data=sim.data.cov.e.m)

mat1.cov.e.m[i,15]<-summary(weib.model.cov.e.m)$table[1,1]#coeff
mat1.cov.e.m[i,16]<-summary(weib.model.cov.e.m)$table[1,2]#std.err
mat1.cov.e.m[i,17]<-summary(weib.model.cov.e.m)$table[1,3]#statistic
mat1.cov.e.m[i,18]<-summary(weib.model.cov.e.m)$table[1,4]#p-val
mat1.cov.e.m[i,19]<-summary(weib.model.cov.e.m)$table[2,1]#coeff
mat1.cov.e.m[i,20]<-summary(weib.model.cov.e.m)$table[2,2]#std.err
mat1.cov.e.m[i,21]<-summary(weib.model.cov.e.m)$table[2,3]#statistic
mat1.cov.e.m[i,22]<-summary(weib.model.cov.e.m)$table[2,4]#p-val
mat1.cov.e.m[i,23]<-summary(weib.model.cov.e.m)$table[3,1]#coeff
mat1.cov.e.m[i,24]<-summary(weib.model.cov.e.m)$table[3,2]#std.err
mat1.cov.e.m[i,25]<-summary(weib.model.cov.e.m)$table[3,3]#statistic
mat1.cov.e.m[i,26]<-summary(weib.model.cov.e.m)$table[3,4]#p-val

```



```

exp.model.cov.e.m=survreg(Surv(stop,status)~as.factor(x)+x.1,
  dist="exp",data=sim.data.cov.e.m)

mat1.cov.e.m[i,29]=summary(exp.model.cov.e.m)$table[1,1]#coeff
mat1.cov.e.m[i,30]=summary(exp.model.cov.e.m)$table[1,2]#std.err
mat1.cov.e.m[i,31]=summary(exp.model.cov.e.m)$table[1,3]#statistic
mat1.cov.e.m[i,32]=summary(exp.model.cov.e.m)$table[1,4]#p-val
mat1.cov.e.m[i,33]=summary(exp.model.cov.e.m)$table[2,1]#coeff
mat1.cov.e.m[i,34]=summary(exp.model.cov.e.m)$table[2,2]#std.err
mat1.cov.e.m[i,35]=summary(exp.model.cov.e.m)$table[2,3]#statistic
mat1.cov.e.m[i,36]=summary(exp.model.cov.e.m)$table[2,4]#p-val
mat1.cov.e.m[i,37]=summary(exp.model.cov.e.m)$table[3,1]#coeff
mat1.cov.e.m[i,38]=summary(exp.model.cov.e.m)$table[3,2]#std.err
mat1.cov.e.m[i,39]=summary(exp.model.cov.e.m)$table[3,3]#statistic
mat1.cov.e.m[i,40]=summary(exp.model.cov.e.m)$table[3,4]#p-val

log1.model.cov.e.m=survreg(Surv(stop,status)~as.factor(x)+x.1,
  dist="loglogistic",data=sim.data.cov.e.m)

mat1.cov.e.m[i,43]=summary(log1.model.cov.e.m)$table[1,1]#coeff
mat1.cov.e.m[i,44]=summary(log1.model.cov.e.m)$table[1,2]#std.err
mat1.cov.e.m[i,45]=summary(log1.model.cov.e.m)$table[1,3]#statistic
mat1.cov.e.m[i,46]=summary(log1.model.cov.e.m)$table[1,4]#p-val

```

```

mat1.cov.e.m[i,47]=summary(logl.model.cov.e.m)$table[2,1]#coeff
mat1.cov.e.m[i,48]=summary(logl.model.cov.e.m)$table[2,2]#std.err
mat1.cov.e.m[i,49]=summary(logl.model.cov.e.m)$table[2,3]#statistic
mat1.cov.e.m[i,50]=summary(logl.model.cov.e.m)$table[2,4]#p-val
mat1.cov.e.m[i,51]=summary(logl.model.cov.e.m)$table[3,1]#coeff
mat1.cov.e.m[i,52]=summary(logl.model.cov.e.m)$table[3,2]#std.err
mat1.cov.e.m[i,53]=summary(logl.model.cov.e.m)$table[3,3]#statistic
mat1.cov.e.m[i,54]=summary(logl.model.cov.e.m)$table[3,4]#p-val

logNorm.model.cov.e.m=survreg(Surv(stop,status)~as.factor(x)+x.1,
  dist="lognormal",data=sim.data.cov.e.m)
mat1.cov.e.m[i,57]=summary(logNorm.model.cov.e.m )$table[1,1]#coeff
mat1.cov.e.m[i,58]=summary(logNorm.model.cov.e.m )$table[1,2]#std.err
mat1.cov.e.m[i,59]=summary(logNorm.model.cov.e.m)$table[1,3]#statistic
mat1.cov.e.m[i,60]=summary(logNorm.model.cov.e.m)$table[1,4]#p-val
mat1.cov.e.m[i,61]=summary(logNorm.model.cov.e.m)$table[2,1]#coeff
mat1.cov.e.m[i,62]=summary(logNorm.model.cov.e.m)$table[2,2]#std.err
mat1.cov.e.m[i,63]=summary(logNorm.model.cov.e.m)$table[2,3]#statistic
mat1.cov.e.m[i,64]=summary(logNorm.model.cov.e.m)$table[2,4]#p-val
mat1.cov.e.m[i,65]=summary(logNorm.model.cov.e.m)$table[3,1]#coeff
mat1.cov.e.m[i,66]=summary(logNorm.model.cov.e.m)$table[3,2]#std.err

```

```

mat1.cov.e.m[i,67]=summary(logNorm.model.cov.e.m)$table[3,3]#statistic
mat1.cov.e.m[i,68]<-summary(logNorm.model.cov.e.m)$table[3,4]#p-val

#####AIC

mat1.cov.e.m[i,13]<-extractAIC(cox.model.cov.e.m)[2];
mat1.cov.e.m[i,14]<-AIC(cox.model.cov.e.m,k = log(length(start)))
mat1.cov.e.m[i,27]<-extractAIC(weib.model.cov.e.m)[2];
mat1.cov.e.m[i,28]<-AIC(weib.model.cov.e.m,k = log(length(start)))
mat1.cov.e.m[i,41]<-extractAIC(exp.model.cov.e.m)[2];
mat1.cov.e.m[i,42]<-AIC(exp.model.cov.e.m,k = log(length(start)))
mat1.cov.e.m[i,55]<-extractAIC(logl.model.cov.e.m)[2];
mat1.cov.e.m[i,56]<-AIC(logl.model.cov.e.m,k = log(length(start)))
mat1.cov.e.m[i,69]<-extractAIC(logNorm.model.cov.e.m)[2];
mat1.cov.e.m[i,70]=AIC(logNorm.model.cov.e.m,k =log(length(start)))

}

colMeans(mat1.cov.e.m)

```

```

result1.cov.e.m=data.frame(matrix(colMeans(mat1.cov.e.m),ncol=14,
byrow=T))

rownames(result1.cov.e.m)=c("cox","weib","exp","logl","logNorm")
colnames(result1.cov.e.m)=c("intercept","stderr","statistic","p-value",
"coeff1","stderr","statistic","p-value","coeff2","stderr","statistic",
"p-value","AIC","BIC")

result1.cov.e.m

```

```
#####
```

```
#####simulation of exponential censord time to event data
```

```

sim.size.cov.e.L<-1000

mat1.cov.e.L<-matrix(0,ncol=70,nrow=sim.size.cov.e.L)

for(i in 1:sim.size.cov.e.L){

sim.data.cov.e.L= simple.surv.sim(n=45000, foltime=150,

dist.ev=c('weibull'),

anc.ev=c(1),beta0.ev=c(5.84298525742252),anc.cens=1,

beta0.cens=7.39773677281100,z=list(c("unif", 0.8, 1.2)),

dist.cens="weibull", beta=list(c(-0.4),

```

```

c(0)), x=list(c("bern", 0.5), c("unif", 0.7, 1.3)))

data.frame(sim.data.cov.e.L)

attach(sim.data.cov.e.L)

cox.model.cov.e.L =coxph(Surv(stop, status) ~ as.factor(x) + x.1,
  data=sim.data.cov.e.L,method="breslow")

mat1.cov.e.L[i,5]=summary(cox.model.cov.e.L)$coefficients[1,1]#coeff
mat1.cov.e.L[i,6]=summary(cox.model.cov.e.L)$coefficients[1,3]#std.err
mat1.cov.e.L[i,7]=summary(cox.model.cov.e.L)$coefficients[1,4]#statistic
mat1.cov.e.L[i,8]=summary(cox.model.cov.e.L)$coefficients[1,5]#p-val
mat1.cov.e.L[i,9]=summary(cox.model.cov.e.L)$coefficients[2,1]#coeff
mat1.cov.e.L[i,10]=summary(cox.model.cov.e.L)$coefficients[2,3]#std.err
mat1.cov.e.L[i,11]=summary(cox.model.cov.e.L)$coefficients[2,4]#statistic
mat1.cov.e.L[i,12]=summary(cox.model.cov.e.L)$coefficients[2,5]#p-val

weib.model.cov.e.L= survreg(Surv(stop, status)~as.factor(x)+x.1,
  dist="weibull",data=sim.data.cov.e.L)

mat1.cov.e.L[i,15]=summary(weib.model.cov.e.L)$table[1,1]#coeff
mat1.cov.e.L[i,16]=summary(weib.model.cov.e.L)$table[1,2]#std.err
mat1.cov.e.L[i,17]=summary(weib.model.cov.e.L)$table[1,3]#statistic

```

```

mat1.cov.e.L[i,18]=summary(weib.model.cov.e.L)$table[1,4]#p-val
mat1.cov.e.L[i,19]=summary(weib.model.cov.e.L)$table[2,1]#coeff
mat1.cov.e.L[i,20]=summary(weib.model.cov.e.L)$table[2,2]#std.err
mat1.cov.e.L[i,21]=summary(weib.model.cov.e.L)$table[2,3]#statistic
mat1.cov.e.L[i,22]=summary(weib.model.cov.e.L)$table[2,4]#p-val
mat1.cov.e.L[i,23]=summary(weib.model.cov.e.L)$table[3,1]#coeff
mat1.cov.e.L[i,24]=summary(weib.model.cov.e.L)$table[3,2]#std.err
mat1.cov.e.L[i,25]=summary(weib.model.cov.e.L)$table[3,3]#statistic
mat1.cov.e.L[i,26]=summary(weib.model.cov.e.L)$table[3,4]#p-val

exp.model.cov.e.L=survreg(Surv(stop,status)~as.factor(x)+x.1,
  dist="exp",data=sim.data.cov.e.L)

mat1.cov.e.L[i,29]=summary(exp.model.cov.e.L)$table[1,1]#coeff
mat1.cov.e.L[i,30]=summary(exp.model.cov.e.L)$table[1,2]#std.err
mat1.cov.e.L[i,31]=summary(exp.model.cov.e.L)$table[1,3]#statistic
mat1.cov.e.L[i,32]=summary(exp.model.cov.e.L)$table[1,4]#p-val
mat1.cov.e.L[i,33]=summary(exp.model.cov.e.L)$table[2,1]#coeff
mat1.cov.e.L[i,34]=summary(exp.model.cov.e.L)$table[2,2]#std.err
mat1.cov.e.L[i,35]=summary(exp.model.cov.e.L)$table[2,3]#statistic
mat1.cov.e.L[i,36]=summary(exp.model.cov.e.L)$table[2,4]#p-val
mat1.cov.e.L[i,37]=summary(exp.model.cov.e.L)$table[3,1]#coeff
mat1.cov.e.L[i,38]=summary(exp.model.cov.e.L)$table[3,2]#std.err

```

```

mat1.cov.e.L[i,39]=summary(exp.model.cov.e.L)$table[3,3]#statistic
mat1.cov.e.L[i,40]=summary(exp.model.cov.e.L)$table[3,4]#p-val

logl.model.cov.e.L=survreg(Surv(stop,status)~as.factor(x)+x.1,
  dist="loglogistic",data=sim.data.cov.e.L)

mat1.cov.e.L[i,43]=summary(logl.model.cov.e.L)$table[1,1]#coeff
mat1.cov.e.L[i,44]=summary(logl.model.cov.e.L)$table[1,2]#std.err
mat1.cov.e.L[i,45]=summary(logl.model.cov.e.L)$table[1,3]#statistic
mat1.cov.e.L[i,46]=summary(logl.model.cov.e.L)$table[1,4]#p-val
mat1.cov.e.L[i,47]=summary(logl.model.cov.e.L)$table[2,1]#coeff
mat1.cov.e.L[i,48]=summary(logl.model.cov.e.L)$table[2,2]#std.err
mat1.cov.e.L[i,49]=summary(logl.model.cov.e.L)$table[2,3]#statistic
mat1.cov.e.L[i,50]=summary(logl.model.cov.e.L)$table[2,4]#p-val
mat1.cov.e.L[i,51]=summary(logl.model.cov.e.L)$table[3,1]#coeff
mat1.cov.e.L[i,52]=summary(logl.model.cov.e.L)$table[3,2]#std.err
mat1.cov.e.L[i,53]=summary(logl.model.cov.e.L)$table[3,3]#statistic
mat1.cov.e.L[i,54]=summary(logl.model.cov.e.L)$table[3,4]#p-val

logNorm.model.cov.e.L=survreg(Surv(stop,status)~as.factor(x)+x.1,

```

```

dist="lognormal",data=sim.data.cov.e.L)

mat1.cov.e.L[i,57]=summary(logNorm.model.cov.e.L)$table[1,1]#coeff
mat1.cov.e.L[i,58]=summary(logNorm.model.cov.e.L)$table[1,2]#std.err
mat1.cov.e.L[i,59]=summary(logNorm.model.cov.e.L)$table[1,3]#statistic
mat1.cov.e.L[i,60]=summary(logNorm.model.cov.e.L)$table[1,4]#p-val
mat1.cov.e.L[i,61]=summary(logNorm.model.cov.e.L)$table[2,1]#coeff
mat1.cov.e.L[i,62]=summary(logNorm.model.cov.e.L)$table[2,2]#std.err
mat1.cov.e.L[i,63]=summary(logNorm.model.cov.e.L)$table[2,3]#statistic
mat1.cov.e.L[i,64]=summary(logNorm.model.cov.e.L)$table[2,4]#p-val
mat1.cov.e.L[i,65]=summary(logNorm.model.cov.e.L)$table[3,1]#coeff
mat1.cov.e.L[i,66]=summary(logNorm.model.cov.e.L)$table[3,2]#std.err
mat1.cov.e.L[i,67]=summary(logNorm.model.cov.e.L)$table[3,3]#statistic
mat1.cov.e.L[i,68]=summary(logNorm.model.cov.e.L)$table[3,4]#p-val

#####AIC

mat1.cov.e.L[i,13]<-extractAIC(cox.model.cov.e.L)[2];
mat1.cov.e.L[i,14]<-AIC(cox.model.cov.e.L,k = log(length(start)))
mat1.cov.e.L[i,27]<-extractAIC(weib.model.cov.e.L)[2];
mat1.cov.e.L[i,28]<-AIC(weib.model.cov.e.L,k = log(length(start)))
mat1.cov.e.L[i,41]<-extractAIC(exp.model.cov.e.L)[2];

```



```

mat1.cov.e.L[i,42]<-AIC(exp.model.cov.e.L,k = log(length(start)))
mat1.cov.e.L[i,55]<-extractAIC(logl.model.cov.e.L)[2];
mat1.cov.e.L[i,56]<-AIC(logl.model.cov.e.L,k = log(length(start)))
mat1.cov.e.L[i,69]<-extractAIC(logNorm.model.cov.e.L)[2];
mat1.cov.e.L[i,70]<-AIC(logNorm.model.cov.e.L,k = log(length(start)))

}

```

```

colMeans(mat1.cov.e.L)

result1.cov.e.L<-data.frame(matrix(colMeans(mat1.cov.e.L),ncol=14,
byrow=T))

rownames(result1.cov.e.L)<-c("cox","weib","exp","logl","logNorm")
colnames(result1.cov.e.L)<-c("intercept","stderr","statistic",
"p-value","coeff1","stderr","statistic","p-value","coeff2","stderr",
"statistic","p-value","AIC","BIC")

result1.cov.e.L

```

```

#####
#####

```

Simulation of loglogistic with covariate

```
#####  
#####  
##simulation of loglogistic censord time to event data (small data)  
  
sim.size.cov.llg.s<-1000  
  
mat1.cov.llg.s<-matrix(0,ncol=70,nrow=sim.size.cov.llg.s)  
  
for(i in 1:sim.size.cov.llg.s){  
  
sim.data.cov.llg.s= simple.surv.sim(n=20, foltime=159,  
dist.ev=c('llogistic'),anc.ev=c(1.69978200185280),  
beta0.ev=c(5.84298525742252),anc.cens=.17783687569519,  
beta0.cens=7.39773677281100,z=list(c("unif", 0.8, 1.2)),  
dist.cens="llogistic", beta=list(c(-0.4),  
c(0)), x=list(c("bern", 0.5), c("unif", 0.7, 1.3)))  
  
data.frame(sim.data.cov.llg.s)  
  
attach(sim.data.cov.llg.s)  
  
cox.model.cov.llg.s =coxph(Surv(stop, status)~as.factor(x)+ x.1,  
data=sim.data.cov.llg.s,method="breslow")  
mat1.cov.llg.s[i,5]=summary(cox.model.cov.llg.s  
)$coefficients[1,1]#coeff
```

```

mat1.cov.llg.s[i,6]=summary(cox.model.cov.llg.s
)$coefficients[1,3]#std.err
mat1.cov.llg.s[i,7]<-summary(cox.model.cov.llg.s
)$coefficients[1,4]#statistic
mat1.cov.llg.s[i,8]<-summary(cox.model.cov.llg.s
)$coefficients[1,5]#p-val
mat1.cov.llg.s[i,9]<-summary(cox.model.cov.llg.s
)$coefficients[2,1]#coeff
mat1.cov.llg.s[i,10]<-summary(cox.model.cov.llg.s
)$coefficients[2,3]#std.err
mat1.cov.llg.s[i,11]<-summary(cox.model.cov.llg.s
)$coefficients[2,4]#statistic
mat1.cov.llg.s[i,12]<-summary(cox.model.cov.llg.s
)$coefficients[2,5] #p-val

weib.model.cov.llg.s=survreg(Surv(stop,status)~as.factor(x)+x.1,
dist="weibull",data=sim.data.cov.llg.s)
mat1.cov.llg.s[i,15]=summary(weib.model.cov.llg.s)$table[1,1]#coeff
mat1.cov.llg.s[i,16]=summary(weib.model.cov.llg.s)$table[1,2]#std.err
mat1.cov.llg.s[i,17]=summary(weib.model.cov.llg.s)$table[1,3]#statistic
mat1.cov.llg.s[i,18]=summary(weib.model.cov.llg.s)$table[1,4]#p-val
mat1.cov.llg.s[i,19]=summary(weib.model.cov.llg.s)$table[2,1]#coeff

```

```

mat1.cov.llg.s[i,20]=summary(weib.model.cov.llg.s)$table[2,2]#std.err
mat1.cov.llg.s[i,21]=summary(weib.model.cov.llg.s)$table[2,3]#statistic
mat1.cov.llg.s[i,22]=summary(weib.model.cov.llg.s)$table[2,4]#p-val
mat1.cov.llg.s[i,23]=summary(weib.model.cov.llg.s)$table[3,1]#coeff
mat1.cov.llg.s[i,24]=summary(weib.model.cov.llg.s)$table[3,2]#std.err
mat1.cov.llg.s[i,25]=summary(weib.model.cov.llg.s)$table[3,3]#statistic
mat1.cov.llg.s[i,26]<-summary(weib.model.cov.llg.s )$table[3,4]#p-val

exp.model.cov.llg.s=survreg(Surv(stop,status)~ as.factor(x)+ x.1,
  dist="exp",data=sim.data.cov.llg.s)

mat1.cov.llg.s[i,29]=summary(exp.model.cov.llg.s)$table[1,1]#coeff
mat1.cov.llg.s[i,30]=summary(exp.model.cov.llg.s)$table[1,2]#std.err
mat1.cov.llg.s[i,31]=summary(exp.model.cov.llg.s)$table[1,3]#statistic
mat1.cov.llg.s[i,32]=summary(exp.model.cov.llg.s)$table[1,4]#p-val
mat1.cov.llg.s[i,33]=summary(exp.model.cov.llg.s)$table[2,1]#coeff
mat1.cov.llg.s[i,34]=summary(exp.model.cov.llg.s)$table[2,2]#std.err
mat1.cov.llg.s[i,35]=summary(exp.model.cov.llg.s)$table[2,3]#statistic
mat1.cov.llg.s[i,36]=summary(exp.model.cov.llg.s)$table[2,4]#p-val
mat1.cov.llg.s[i,37]=summary(exp.model.cov.llg.s)$table[3,1]#coeff
mat1.cov.llg.s[i,38]=summary(exp.model.cov.llg.s)$table[3,2]#std.err

```

```

mat1.cov.llg.s[i,39]=summary(exp.model.cov.llg.s)$table[3,3]#statistic
mat1.cov.llg.s[i,40]=summary(exp.model.cov.llg.s)$table[3,4]#p-val

logl.model.cov.llg.s= survreg(Surv(stop, status)~as.factor(x)+ x.1,
  dist="loglogistic",data=sim.data.cov.llg.s)

mat1.cov.llg.s[i,43]=summary(logl.model.cov.llg.s)$table[1,1]#coeff
mat1.cov.llg.s[i,44]=summary(logl.model.cov.llg.s)$table[1,2]#std.err
mat1.cov.llg.s[i,45]=summary(logl.model.cov.llg.s)$table[1,3]#statistic
mat1.cov.llg.s[i,46]=summary(logl.model.cov.llg.s)$table[1,4]#p-val
mat1.cov.llg.s[i,47]=summary(logl.model.cov.llg.s)$table[2,1]#coeff
mat1.cov.llg.s[i,48]=summary(logl.model.cov.llg.s)$table[2,2]#std.err
mat1.cov.llg.s[i,49]=summary(logl.model.cov.llg.s)$table[2,3]#statistic
mat1.cov.llg.s[i,50]=summary(logl.model.cov.llg.s)$table[2,4]#p-val
mat1.cov.llg.s[i,51]=summary(logl.model.cov.llg.s)$table[3,1]#coeff
mat1.cov.llg.s[i,52]=summary(logl.model.cov.llg.s)$table[3,2]#std.err
mat1.cov.llg.s[i,53]=summary(logl.model.cov.llg.s)$table[3,3]#statistic
mat1.cov.llg.s[i,54]=summary(logl.model.cov.llg.s)$table[3,4]#p-val

logNorm.model.cov.llg.s=survreg(Surv(stop, status)~as.factor(x)+x.1,
  dist="lognormal",data=sim.data.cov.llg.s)

mat1.cov.llg.s[i,57]=summary(logNorm.model.cov.llg.s)$table[1,1]#coeff

```

```

mat1.cov.llg.s[i,58]=summary(logNorm.model.cov.llg.s)$table[1,2]#std.err
mat1.cov.llg.s[i,59]=summary(logNorm.model.cov.llg.s)$table[1,3]#statistic
mat1.cov.llg.s[i,60]=summary(logNorm.model.cov.llg.s )$table[1,4]#p-val
mat1.cov.llg.s[i,61]=summary(logNorm.model.cov.llg.s)$table[2,1] #coeff
mat1.cov.llg.s[i,62]=summary(logNorm.model.cov.llg.s)$table[2,2]#std.err
mat1.cov.llg.s[i,63]=summary(logNorm.model.cov.llg.s)$table[2,3]#statistic
mat1.cov.llg.s[i,64]=summary(logNorm.model.cov.llg.s)$table[2,4]#p-val
mat1.cov.llg.s[i,65]=summary(logNorm.model.cov.llg.s)$table[3,1]#coeff
mat1.cov.llg.s[i,66]=summary(logNorm.model.cov.llg.s)$table[3,2] #std.err
mat1.cov.llg.s[i,67]=summary(logNorm.model.cov.llg.s )$table[3,3]#statistic
mat1.cov.llg.s[i,68]=summary(logNorm.model.cov.llg.s)$table[3,4]#p-val

mat1.cov.llg.s[i,13]<-extractAIC(cox.model.cov.llg.s)[2];
mat1.cov.llg.s[i,14]<-AIC(cox.model.cov.llg.s,k = log(length(start)))
mat1.cov.llg.s[i,27]<-extractAIC(weib.model.cov.llg.s)[2];
mat1.cov.llg.s[i,28]<-AIC(weib.model.cov.llg.s,k = log(length(start)))
mat1.cov.llg.s[i,41]<-extractAIC(exp.model.cov.llg.s)[2];
mat1.cov.llg.s[i,42]<-AIC(exp.model.cov.llg.s,k = log(length(start)))
mat1.cov.llg.s[i,55]<-extractAIC(logl.model.cov.llg.s)[2];
mat1.cov.llg.s[i,56]<-AIC(logl.model.cov.llg.s,k = log(length(start)))
mat1.cov.llg.s[i,69]<-extractAIC(logNorm.model.cov.llg.s)[2];

```

```

mat1.cov.llg.s[i,70]<-AIC(logNorm.model.cov.llg.s,k = log(length(start)))

}

colMeans(mat1.cov.llg.s)

result1.cov.llg.s<-data.frame(matrix(colMeans(mat1.cov.llg.s),
ncol=14,byrow=T))

rownames(result1.cov.llg.s)<-c("cox","weib","exp","logl","logNorm")
colnames(result1.cov.llg.s)<-c("intercept","stderr","statistic",
"p-value","coeff1","stderr","statistic","p-value","coeff2","stderr",
"statistic","p-value","AIC","BIC")

result1.cov.llg.s

#####simulation of loglogistic censord time to event data(medium data)

sim.size.cov.llg.m<-500

mat1.cov.llg.m<-matrix(0,ncol=70,nrow=sim.size.cov.llg.m)

for(i in 1:sim.size.cov.llg.m){

sim.data.cov.llg.m= simple.surv.sim(n=550, foltime=120,

```

```

dist.ev=c('llogistic'), anc.ev=c(0.69978200185280),
beta0.ev=c(5.84298525742252), anc.cens=1.17783687569519,
beta0.cens=7.39773677281100, z=list(c("unif", 0.8, 1.2)),
  dist.cens="llogistic", beta=list(c(-0.4),
c(0)), x=list(c("bern", 0.5), c("unif", 0.7, 1.3)))

data.frame(sim.data.cov.llg.m)

attach(sim.data.cov.llg.m)

cox.model.cov.llg.m =coxph(Surv(stop, status) ~ as.factor(x) + x.1,
  data=sim.data.cov.llg.m, method="breslow")

mat1.cov.llg.m[i,5]<-summary(cox.model.cov.llg.m
)$coefficients[1,1] #coeff

mat1.cov.llg.m[i,6]<-summary(cox.model.cov.llg.m
)$coefficients[1,3] #std.err

mat1.cov.llg.m[i,7]<-summary(cox.model.cov.llg.m
)$coefficients[1,4] #statistic

mat1.cov.llg.m[i,8]<-summary(cox.model.cov.llg.m
)$coefficients[1,5] #p-val

mat1.cov.llg.m[i,9]<-summary(cox.model.cov.llg.m
)$coefficients[2,1] #coeff

mat1.cov.llg.m[i,10]<-summary(cox.model.cov.llg.m

```



```

)$.coefficients[2,3]#std.err

mat1.cov.llg.m[i,11]<-summary(cox.model.cov.llg.m

)$.coefficients[2,4]#statistic

mat1.cov.llg.m[i,12]<-summary(cox.model.cov.llg.m

)$.coefficients[2,5] #p-val

weib.model.cov.llg.m=survreg(Surv(stop,status)~as.factor(x)+x.1,

dist="weibull",data=sim.data.cov.llg.m)

mat1.cov.llg.m[i,15]=summary(weib.model.cov.llg.m)$table[1,1]#coeff
mat1.cov.llg.m[i,16]=summary(weib.model.cov.llg.m)$table[1,2]#std.err
mat1.cov.llg.m[i,17]=summary(weib.model.cov.llg.m)$table[1,3]#statistic
mat1.cov.llg.m[i,18]=summary(weib.model.cov.llg.m)$table[1,4]#p-val
mat1.cov.llg.m[i,19]=summary(weib.model.cov.llg.m)$table[2,1]#coeff
mat1.cov.llg.m[i,20]=summary(weib.model.cov.llg.m)$table[2,2]#std.err
mat1.cov.llg.m[i,21]=summary(weib.model.cov.llg.m)$table[2,3]#statistic
mat1.cov.llg.m[i,22]=summary(weib.model.cov.llg.m)$table[2,4]#p-val
mat1.cov.llg.m[i,23]=summary(weib.model.cov.llg.m)$table[3,1]#coeff
mat1.cov.llg.m[i,24]=summary(weib.model.cov.llg.m)$table[3,2]#std.err
mat1.cov.llg.m[i,25]=summary(weib.model.cov.llg.m)$table[3,3]#statistic
mat1.cov.llg.m[i,26]=summary(weib.model.cov.llg.m)$table[3,4]#p-val

exp.model.cov.llg.m=survreg(Surv(stop,status)~as.factor(x)+ x.1,

```

```

dist="exp",data=sim.data.cov.llg.m)

mat1.cov.llg.m[i,29]=summary(exp.model.cov.llg.m)$table[1,1]#coeff
mat1.cov.llg.m[i,30]=summary(exp.model.cov.llg.m)$table[1,2]#std.err
mat1.cov.llg.m[i,31]=summary(exp.model.cov.llg.m)$table[1,3]#statistic
mat1.cov.llg.m[i,32]=summary(exp.model.cov.llg.m)$table[1,4]#p-val
mat1.cov.llg.m[i,33]=summary(exp.model.cov.llg.m)$table[2,1]#coeff
mat1.cov.llg.m[i,34]=summary(exp.model.cov.llg.m)$table[2,2]#std.err
mat1.cov.llg.m[i,35]=summary(exp.model.cov.llg.m)$table[2,3]#statistic
mat1.cov.llg.m[i,36]=summary(exp.model.cov.llg.m)$table[2,4]#p-val
mat1.cov.llg.m[i,37]=summary(exp.model.cov.llg.m)$table[3,1]#coeff
mat1.cov.llg.m[i,38]=summary(exp.model.cov.llg.m)$table[3,2]#std.err
mat1.cov.llg.m[i,39]=summary(exp.model.cov.llg.m)$table[3,3]#statistic
mat1.cov.llg.m[i,40]=summary(exp.model.cov.llg.m)$table[3,4]#p-val

log1.model.cov.llg.m=survreg(Surv(stop,status)~as.factor(x)+x.1,
dist="loglogistic",data=sim.data.cov.llg.m)

mat1.cov.llg.m[i,43]=summary(log1.model.cov.llg.m)$table[1,1]#coeff
mat1.cov.llg.m[i,44]=summary(log1.model.cov.llg.m)$table[1,2]#std.err
mat1.cov.llg.m[i,45]=summary(log1.model.cov.llg.m)$table[1,3]#statistic
mat1.cov.llg.m[i,46]=summary(log1.model.cov.llg.m)$table[1,4]#p-val
mat1.cov.llg.m[i,47]=summary(log1.model.cov.llg.m)$table[2,1]#coeff
mat1.cov.llg.m[i,48]=summary(log1.model.cov.llg.m)$table[2,2]#std.err

```

```

mat1.cov.llg.m[i,49]=summary(logl.model.cov.llg.m)$table[2,3]#statistic
mat1.cov.llg.m[i,50]=summary(logl.model.cov.llg.m)$table[2,4]#p-val
mat1.cov.llg.m[i,51]=summary(logl.model.cov.llg.m)$table[3,1]#coeff
mat1.cov.llg.m[i,52]=summary(logl.model.cov.llg.m)$table[3,2]#std.err
mat1.cov.llg.m[i,53]=summary(logl.model.cov.llg.m )$table[3,3]#statistic
mat1.cov.llg.m[i,54]=summary(logl.model.cov.llg.m )$table[3,4]#p-val

logNorm.model.cov.llg.m=survreg(Surv(stop, status)~as.factor(x)+x.1,
  dist="lognormal",data=sim.data.cov.llg.m)

mat1.cov.llg.m[i,57]=summary(logNorm.model.cov.llg.m)$table[1,1]#coeff
mat1.cov.llg.m[i,58]=summary(logNorm.model.cov.llg.m)$table[1,2]#std.err
mat1.cov.llg.m[i,59]=summary(logNorm.model.cov.llg.m)$table[1,3]#statistic
mat1.cov.llg.m[i,60]=summary(logNorm.model.cov.llg.m)$table[1,4]#p-val
mat1.cov.llg.m[i,61]=summary(logNorm.model.cov.llg.m)$table[2,1]#coeff
mat1.cov.llg.m[i,62]=summary(logNorm.model.cov.llg.m )$table[2,2]#std.err
mat1.cov.llg.m[i,63]=summary(logNorm.model.cov.llg.m )$table[2,3]#statistic
mat1.cov.llg.m[i,64]=summary(logNorm.model.cov.llg.m)$table[2,4]#p-val
mat1.cov.llg.m[i,65]=summary(logNorm.model.cov.llg.m)$table[3,1]#coeff
mat1.cov.llg.m[i,66]=summary(logNorm.model.cov.llg.m)$table[3,2]#std.err
mat1.cov.llg.m[i,67]=summary(logNorm.model.cov.llg.m)$table[3,3] #statistic
mat1.cov.llg.m[i,68]=summary(logNorm.model.cov.llg.m)$table[3,4]#p-val

```

```

mat1.cov.llg.m[i,13]<-extractAIC(cox.model.cov.llg.m)[2];
mat1.cov.llg.m[i,14]<-AIC(cox.model.cov.llg.m,k = log(length(start)))
mat1.cov.llg.m[i,27]<-extractAIC(weib.model.cov.llg.m)[2];
mat1.cov.llg.m[i,28]<-AIC(weib.model.cov.llg.m,k = log(length(start)))
mat1.cov.llg.m[i,41]<-extractAIC(exp.model.cov.llg.m)[2];
mat1.cov.llg.m[i,42]<-AIC(exp.model.cov.llg.m,k = log(length(start)))
mat1.cov.llg.m[i,55]<-extractAIC(logl.model.cov.llg.m)[2];
mat1.cov.llg.m[i,56]<-AIC(logl.model.cov.llg.m,k = log(length(start)))
mat1.cov.llg.m[i,69]<-extractAIC(logNorm.model.cov.llg.m)[2];
mat1.cov.llg.m[i,70]<-AIC(logNorm.model.cov.llg.m,k = log(length(start)))
}

colMeans(mat1.cov.llg.m)

result1.cov.llg.m<-data.frame(matrix(colMeans(mat1.cov.llg.m),ncol=14,
byrow=T))

rownames(result1.cov.llg.m)<-c("cox","weib","exp","logl","logNorm")
colnames(result1.cov.llg.m)<-c("intercept","stderr","statistic",
"p-value","coeff1","stderr","statistic","p-value","coeff2","stderr",
"statistic","p-value","AIC","BIC")

result1.cov.llg.m

```

```

##simulation of loglogistic censord time to event data(large data)

sim.size.cov.llg.L<-1000

mat1.cov.llg.L<-matrix(0,ncol=70,nrow=sim.size.cov.llg.L)

for(i in 1:sim.size.cov.llg.L){

sim.data.cov.llg.L= simple.surv.sim(n=20000, foltime=360,
dist.ev=c('llogistic'),anc.ev=c(0.6997820018528),beta0.ev=
c(5.84298525742252),anc.cens=1.17783687569519,
beta0.cens=7.39773677281100,

z=list(c("unif", 0.8, 1.2)), dist.cens="llogistic",

beta=list(c(-0.4),c(0)),x=list(c("bern", 0.5),c("unif",0.7,1.3)))

data.frame(sim.data.cov.llg.L)

attach(sim.data.cov.llg.L)

cox.model.cov.llg.L =coxph(Surv(stop,status)~as.factor(x)+x.1,
data=sim.data.cov.llg.L,method="breslow")

mat1.cov.llg.L[i,5]<-summary(cox.model.cov.llg.L
)$coefficients[1,1] #coeff

mat1.cov.llg.L[i,6]<-summary(cox.model.cov.llg.L
)$coefficients[1,3]#std.err

```

```

mat1.cov.llg.L[i,7]<-summary(cox.model.cov.llg.L
)$.coefficients[1,4]#statistic
mat1.cov.llg.L[i,8]<-summary(cox.model.cov.llg.L
)$.coefficients[1,5] #p-val
mat1.cov.llg.L[i,9]<-summary(cox.model.cov.llg.L
)$.coefficients[2,1] #coeff
mat1.cov.llg.L[i,10]<-summary(cox.model.cov.llg.L
)$.coefficients[2,3]#std.err
mat1.cov.llg.L[i,11]<-summary(cox.model.cov.llg.L
)$.coefficients[2,4]#statistic
mat1.cov.llg.L[i,12]<-summary(cox.model.cov.llg.L
)$.coefficients[2,5]#p-val

weib.model.cov.llg.L= survreg(Surv(stop,status)~as.factor(x)+x.1,
dist="weibull",data=sim.data.cov.llg.L)
mat1.cov.llg.L[i,15]=summary(weib.model.cov.llg.L)$table[1,1]#coeff
mat1.cov.llg.L[i,16]=summary(weib.model.cov.llg.L)$table[1,2]#std.err
mat1.cov.llg.L[i,17]=summary(weib.model.cov.llg.L)$table[1,3]#statistic
mat1.cov.llg.L[i,18]=summary(weib.model.cov.llg.L)$table[1,4]#p-val
mat1.cov.llg.L[i,19]=summary(weib.model.cov.llg.L)$table[2,1]#coeff
mat1.cov.llg.L[i,20]=summary(weib.model.cov.llg.L)$table[2,2]#std.err

```

```

mat1.cov.llg.L[i,21]=summary(weib.model.cov.llg.L)$table[2,3]#statistic
mat1.cov.llg.L[i,22]=summary(weib.model.cov.llg.L)$table[2,4]#p-val
mat1.cov.llg.L[i,23]=summary(weib.model.cov.llg.L)$table[3,1]#coeff
mat1.cov.llg.L[i,24]=summary(weib.model.cov.llg.L)$table[3,2]#std.err
mat1.cov.llg.L[i,25]=summary(weib.model.cov.llg.L)$table[3,3]#statistic
mat1.cov.llg.L[i,26]=summary(weib.model.cov.llg.L)$table[3,4]#p-val

exp.model.cov.llg.L=survreg(Surv(stop,status)~as.factor(x)+x.1,
  dist="exp",data=sim.data.cov.llg.L)

mat1.cov.llg.L[i,29]=summary(exp.model.cov.llg.L)$table[1,1]#coeff
mat1.cov.llg.L[i,30]=summary(exp.model.cov.llg.L)$table[1,2]#std.err
mat1.cov.llg.L[i,31]=summary(exp.model.cov.llg.L)$table[1,3]#statistic
mat1.cov.llg.L[i,32]=summary(exp.model.cov.llg.L)$table[1,4]#p-val
mat1.cov.llg.L[i,33]=summary(exp.model.cov.llg.L)$table[2,1]#coeff
mat1.cov.llg.L[i,34]=summary(exp.model.cov.llg.L)$table[2,2]#std.err
mat1.cov.llg.L[i,35]=summary(exp.model.cov.llg.L)$table[2,3]#statistic
mat1.cov.llg.L[i,36]=summary(exp.model.cov.llg.L)$table[2,4]#p-val
mat1.cov.llg.L[i,37]=summary(exp.model.cov.llg.L)$table[3,1]#coeff
mat1.cov.llg.L[i,38]=summary(exp.model.cov.llg.L)$table[3,2]#std.err
mat1.cov.llg.L[i,39]=summary(exp.model.cov.llg.L)$table[3,3]#statistic
mat1.cov.llg.L[i,40]=summary(exp.model.cov.llg.L)$table[3,4]#p-val

```

```

logl.model.cov.llg.L= survreg(Surv(stop, status)~as.factor(x)+x.1,
  dist="loglogistic",data=sim.data.cov.llg.L)

mat1.cov.llg.L[i,43]=summary(logl.model.cov.llg.L )$table[1,1]#coeff
mat1.cov.llg.L[i,44]=summary(logl.model.cov.llg.L )$table[1,2] #std.err
mat1.cov.llg.L[i,45]=summary(logl.model.cov.llg.L )$table[1,3] #statistic
mat1.cov.llg.L[i,46]=summary(logl.model.cov.llg.L )$table[1,4] #p-val
mat1.cov.llg.L[i,47]=summary(logl.model.cov.llg.L )$table[2,1] #coeff
mat1.cov.llg.L[i,48]=summary(logl.model.cov.llg.L )$table[2,2] #std.err
mat1.cov.llg.L[i,49]=summary(logl.model.cov.llg.L )$table[2,3] #statistic
mat1.cov.llg.L[i,50]=summary(logl.model.cov.llg.L )$table[2,4] #p-val
mat1.cov.llg.L[i,51]=summary(logl.model.cov.llg.L )$table[3,1] #coeff
mat1.cov.llg.L[i,52]=summary(logl.model.cov.llg.L )$table[3,2] #std.err
mat1.cov.llg.L[i,53]=summary(logl.model.cov.llg.L )$table[3,3] #statistic
mat1.cov.llg.L[i,54]=summary(logl.model.cov.llg.L )$table[3,4] #p-val

logNorm.model.cov.llg.L= survreg(Surv(stop, status)~ as.factor(x) + x.1,
  dist="lognormal",data=sim.data.cov.llg.L)

mat1.cov.llg.L[i,57]=summary(logNorm.model.cov.llg.L)$table[1,1]#coeff
mat1.cov.llg.L[i,58]=summary(logNorm.model.cov.llg.L)$table[1,2]#std.err
mat1.cov.llg.L[i,59]=summary(logNorm.model.cov.llg.L)$table[1,3] #statistic
mat1.cov.llg.L[i,60]=summary(logNorm.model.cov.llg.L)$table[1,4]#p-val
mat1.cov.llg.L[i,61]=summary(logNorm.model.cov.llg.L)$table[2,1]#coeff

```



```

mat1.cov.llg.L[i,62]=summary(logNorm.model.cov.llg.L)$table[2,2]#std.err
mat1.cov.llg.L[i,63]=summary(logNorm.model.cov.llg.L)$table[2,3]#statistic
mat1.cov.llg.L[i,64]=summary(logNorm.model.cov.llg.L)$table[2,4]#p-val
mat1.cov.llg.L[i,65]=summary(logNorm.model.cov.llg.L)$table[3,1]#coeff
mat1.cov.llg.L[i,66]=summary(logNorm.model.cov.llg.L)$table[3,2]#std.err
mat1.cov.llg.L[i,67]=summary(logNorm.model.cov.llg.L)$table[3,3]#statistic
mat1.cov.llg.L[i,68]=summary(logNorm.model.cov.llg.L)$table[3,4]#p-val

#####AIC

mat1.cov.llg.L[i,13]<-extractAIC(cox.model.cov.llg.L)[2];
mat1.cov.llg.L[i,14]<-AIC(cox.model.cov.llg.L,k=log(length(start)))
mat1.cov.llg.L[i,27]<-extractAIC(weib.model.cov.llg.L)[2];
mat1.cov.llg.L[i,28]<-AIC(weib.model.cov.llg.L,k = log(length(start)))
mat1.cov.llg.L[i,41]<-extractAIC(exp.model.cov.llg.L)[2];
mat1.cov.llg.L[i,42]<-AIC(exp.model.cov.llg.L,k=log(length(start)))
mat1.cov.llg.L[i,55]<-extractAIC(logl.model.cov.llg.L)[2];
mat1.cov.llg.L[i,56]<-AIC(logl.model.cov.llg.L,k=log(length(start)))
mat1.cov.llg.L[i,69]<-extractAIC(logNorm.model.cov.llg.L)[2];
mat1.cov.llg.L[i,70]=AIC(logNorm.model.cov.llg.L,k=log(length(start)))

}

```

```

colMeans(mat1.cov.llg.L)
result1.cov.llg.L<-data.frame(matrix(colMeans(mat1.cov.llg.L),ncol=14,
byrow=T))
rownames(result1.cov.llg.L)<-c("cox","weib","exp","logl","logNorm")
colnames(result1.cov.llg.L)<-c("intercept","stderr","statistic",
"p-value","coeff1","stderr","statistic","p-value","coeff2","stderr",
"statistic","p-value","AIC","BIC")
result1.cov.llg.L

```

```
#####
```

```
#####
```

```
Simulation of lognormal with covariate
```

```
#####simulation of logno
```

```
sim.size.cov.LN.s<-1000
```

```
mat1.cov.LN.s<-matrix(0,ncol=70,nrow=sim.size.cov.LN.s)
```

```
for(i in 1:sim.size.cov.LN.s){
```

```
sim.data.cov.LN.s= simple.surv.sim(n=21, foltime=360,
```

```
dist.ev=c('lnorm'),
```

```

anc.ev=c(0.69978200185280),beta0.ev=c(5.84298525742252),
anc.cens=1.17783687569519,
beta0.cens=7.39773677281100,z=list(c("unif", 0.8, 1.2)),
  dist.cens="lnorm", beta=list(c(-0.4),
c(0)), x=list(c("bern", 0.5), c("unif", 0.7, 1.3)))

data.frame(sim.data.cov.LN.s)

attach(sim.data.cov.LN.s)

cox.model.cov.LN.s =coxph(Surv(stop, status) ~ as.factor(x) + x.1,
  data=sim.data.cov.LN.s,method="breslow")

mat1.cov.LN.s[i,5]<-summary(cox.model.cov.LN.s
)$coefficients[1,1]#coeff

mat1.cov.LN.s[i,6]<-summary(cox.model.cov.LN.s
)$coefficients[1,3]#std.err

mat1.cov.LN.s[i,7]<-summary(cox.model.cov.LN.s
)$coefficients[1,4]#statistic

mat1.cov.LN.s[i,8]<-summary(cox.model.cov.LN.s
)$coefficients[1,5]#p-val

mat1.cov.LN.s[i,9]<-summary(cox.model.cov.LN.s
)$coefficients[2,1] #coeff

mat1.cov.LN.s[i,10]<-summary(cox.model.cov.LN.s
)$coefficients[2,3]#std.err

```

```

mat1.cov.LN.s[i,11]<-summary(cox.model.cov.LN.s
)$.coefficients[2,4]#statistic

mat1.cov.LN.s[i,12]<-summary(cox.model.cov.LN.s
)$.coefficients[2,5] #p-val

weib.model.cov.LN.s=survreg(Surv(stop,status)~as.factor(x)+x.1,
dist="weibull",data=sim.data.cov.LN.s)

mat1.cov.LN.s[i,15]=summary(weib.model.cov.LN.s)$table[1,1]#coeff
mat1.cov.LN.s[i,16]=summary(weib.model.cov.LN.s)$table[1,2]#std.err
mat1.cov.LN.s[i,17]=summary(weib.model.cov.LN.s)$table[1,3]#statistic
mat1.cov.LN.s[i,18]=summary(weib.model.cov.LN.s)$table[1,4]#p-val
mat1.cov.LN.s[i,19]=summary(weib.model.cov.LN.s)$table[2,1]#coeff
mat1.cov.LN.s[i,20]=summary(weib.model.cov.LN.s)$table[2,2]#std.err
mat1.cov.LN.s[i,21]=summary(weib.model.cov.LN.s)$table[2,3]#statistic
mat1.cov.LN.s[i,22]=summary(weib.model.cov.LN.s)$table[2,4]#p-val
mat1.cov.LN.s[i,23]=summary(weib.model.cov.LN.s)$table[3,1]#coeff
mat1.cov.LN.s[i,24]=summary(weib.model.cov.LN.s)$table[3,2]#std.err
mat1.cov.LN.s[i,25]=summary(weib.model.cov.LN.s)$table[3,3]#statistic
mat1.cov.LN.s[i,26]=summary(weib.model.cov.LN.s)$table[3,4]#p-val

```

```

exp.model.cov.LN.s=survreg(Surv(stop, status)~as.factor(x)+x.1,
  dist="exp",data=sim.data.cov.LN.s)

mat1.cov.LN.s[i,29]=summary(exp.model.cov.LN.s)$table[1,1]#coeff
mat1.cov.LN.s[i,30]=summary(exp.model.cov.LN.s)$table[1,2]#std.err
mat1.cov.LN.s[i,31]=summary(exp.model.cov.LN.s)$table[1,3]#statistic
mat1.cov.LN.s[i,32]=summary(exp.model.cov.LN.s)$table[1,4]#p-val
mat1.cov.LN.s[i,33]=summary(exp.model.cov.LN.s)$table[2,1]#coeff
mat1.cov.LN.s[i,34]=summary(exp.model.cov.LN.s)$table[2,2]#std.err
mat1.cov.LN.s[i,35]=summary(exp.model.cov.LN.s)$table[2,3]#statistic
mat1.cov.LN.s[i,36]=summary(exp.model.cov.LN.s)$table[2,4]#p-val
mat1.cov.LN.s[i,37]=summary(exp.model.cov.LN.s)$table[3,1]#coeff
mat1.cov.LN.s[i,38]=summary(exp.model.cov.LN.s)$table[3,2]#std.err
mat1.cov.LN.s[i,39]=summary(exp.model.cov.LN.s)$table[3,3]#statistic
mat1.cov.LN.s[i,40]=summary(exp.model.cov.LN.s)$table[3,4]#p-val

```

```

log1.model.cov.LN.s= survreg(Surv(stop, status)~as.factor(x)+ x.1,
  dist="loglogistic",data=sim.data.cov.LN.s)

mat1.cov.LN.s[i,43]=summary(log1.model.cov.LN.s)$table[1,1]#coeff
mat1.cov.LN.s[i,44]=summary(log1.model.cov.LN.s)$table[1,2]#std.err

```

```

mat1.cov.LN.s[i,45]=summary(logl.model.cov.LN.s)$table[1,3]#statistic
mat1.cov.LN.s[i,46]=summary(logl.model.cov.LN.s)$table[1,4]#p-val
mat1.cov.LN.s[i,47]=summary(logl.model.cov.LN.s)$table[2,1]#coeff
mat1.cov.LN.s[i,48]=summary(logl.model.cov.LN.s)$table[2,2]#std.err
mat1.cov.LN.s[i,49]=summary(logl.model.cov.LN.s)$table[2,3]#statistic
mat1.cov.LN.s[i,50]=summary(logl.model.cov.LN.s)$table[2,4]#p-val
mat1.cov.LN.s[i,51]=summary(logl.model.cov.LN.s)$table[3,1]#coeff
mat1.cov.LN.s[i,52]=summary(logl.model.cov.LN.s)$table[3,2]#std.err
mat1.cov.LN.s[i,53]=summary(logl.model.cov.LN.s)$table[3,3]#statistic
mat1.cov.LN.s[i,54]=summary(logl.model.cov.LN.s)$table[3,4]#p-val

logNorm.model.cov.LN.s=survreg(Surv(stop,status)~as.factor(x)+x.1,
  dist="lognormal",data=sim.data.cov.LN.s)

mat1.cov.LN.s[i,57]=summary(logNorm.model.cov.LN.s)$table[1,1]#coeff
mat1.cov.LN.s[i,58]=summary(logNorm.model.cov.LN.s)$table[1,2]#std.err
mat1.cov.LN.s[i,59]=summary(logNorm.model.cov.LN.s)$table[1,3]#statistic
mat1.cov.LN.s[i,60]=summary(logNorm.model.cov.LN.s)$table[1,4]#p-val
mat1.cov.LN.s[i,61]=summary(logNorm.model.cov.LN.s)$table[2,1]#coeff
mat1.cov.LN.s[i,62]=summary(logNorm.model.cov.LN.s)$table[2,2]#std.err
mat1.cov.LN.s[i,63]=summary(logNorm.model.cov.LN.s)$table[2,3]#statistic
mat1.cov.LN.s[i,64]=summary(logNorm.model.cov.LN.s)$table[2,4]#p-val
mat1.cov.LN.s[i,65]=summary(logNorm.model.cov.LN.s)$table[3,1]#coeff

```

```

mat1.cov.LN.s[i,66]=summary(logNorm.model.cov.LN.s)$table[3,2]#std.err
mat1.cov.LN.s[i,67]=summary(logNorm.model.cov.LN.s)$table[3,3]#statistic
mat1.cov.LN.s[i,68]<-summary(logNorm.model.cov.LN.s )$table[3,4] #p-val

#####AIC

mat1.cov.LN.s[i,13]<-extractAIC(cox.model.cov.LN.s)[2];
mat1.cov.LN.s[i,14]<-AIC(cox.model.cov.LN.s,k = log(length(start)))
mat1.cov.LN.s[i,27]<-extractAIC(weib.model.cov.LN.s)[2];
mat1.cov.LN.s[i,28]<-AIC(weib.model.cov.LN.s,k = log(length(start)))
mat1.cov.LN.s[i,41]<-extractAIC(exp.model.cov.LN.s)[2];
mat1.cov.LN.s[i,42]<-AIC(exp.model.cov.LN.s,k = log(length(start)))
mat1.cov.LN.s[i,55]<-extractAIC(logl.model.cov.LN.s)[2];
mat1.cov.LN.s[i,56]<-AIC(logl.model.cov.LN.s,k = log(length(start)))
mat1.cov.LN.s[i,69]<-extractAIC(logNorm.model.cov.LN.s)[2];
mat1.cov.LN.s[i,70]<-AIC(logNorm.model.cov.LN.s,k = log(length(start)))

}

colMeans(mat1.cov.LN.s)

```

```

result1.cov.LN.s<-data.frame(matrix(colMeans(mat1.cov.LN.s),
ncol=14,byrow=T))

rownames(result1.cov.LN.s)<-c("cox","weib","exp","logl","logNorm")
colnames(result1.cov.LN.s)<-c("intercept","stderr","statistic",
"p-value","coeff1","stderr","statistic","p-value","coeff2","stderr",
"statistic","p-value","AIC","BIC")

result1.cov.LN.s

#####
##simulation of lognormal censord time to event data (medium data)

sim.size.cov.LN.m<-500

mat1.cov.LN.m<-matrix(0,ncol=70,nrow=sim.size.cov.LN.m)

for(i in 1:sim.size.cov.LN.m){

sim.data.cov.LN.m=simple.surv.sim(n=850,foltime=360,dist.ev=c('lnorm'),
anc.ev=c(0.69978200185280),beta0.ev=c(5.84298525742252),
anc.cens=1.17783687569519,beta0.cens=7.39773677281100,
z=list(c("unif", 0.8, 1.2)), dist.cens="lnorm", beta=list(c(-0.4),
c(0)), x=list(c("bern", 0.5), c("unif", 0.7, 1.3)))

data.frame(sim.data.cov.LN.m)

attach(sim.data.cov.LN.m)

```



```

cox.model.cov.LN.m =coxph(Surv(stop, status)~as.factor(x)+x.1,
  data=sim.data.cov.LN.m,method="breslow")
mat1.cov.LN.m[i,5]<-summary(cox.model.cov.LN.m
  )$coefficients[1,1]#coeff
mat1.cov.LN.m[i,6]<-summary(cox.model.cov.LN.m
  )$coefficients[1,3]#std.err
mat1.cov.LN.m[i,7]<-summary(cox.model.cov.LN.m
  )$coefficients[1,4]#statistic
mat1.cov.LN.m[i,8]<-summary(cox.model.cov.LN.m
  )$coefficients[1,5]#p-val
mat1.cov.LN.m[i,9]<-summary(cox.model.cov.LN.m
  )$coefficients[2,1]#coeff
mat1.cov.LN.m[i,10]<-summary(cox.model.cov.LN.m
  )$coefficients[2,3]#std.err
mat1.cov.LN.m[i,11]<-summary(cox.model.cov.LN.m
  )$coefficients[2,4]#statistic
mat1.cov.LN.m[i,12]=summary(cox.model.cov.LN.m)$coefficients[2,5]#p-val

weib.model.cov.LN.m= survreg(Surv(stop, status) ~ as.factor(x) + x.1,
  dist="weibull",data=sim.data.cov.LN.m)
mat1.cov.LN.m[i,15]=summary(weib.model.cov.LN.m)$table[1,1]#coeff
mat1.cov.LN.m[i,16]=summary(weib.model.cov.LN.m)$table[1,2]#std.err

```

```

mat1.cov.LN.m[i,17]=summary(weib.model.cov.LN.m)$table[1,3]#statistic
mat1.cov.LN.m[i,18]=summary(weib.model.cov.LN.m)$table[1,4]#p-val
mat1.cov.LN.m[i,19]=summary(weib.model.cov.LN.m)$table[2,1]#coeff
mat1.cov.LN.m[i,20]=summary(weib.model.cov.LN.m)$table[2,2]#std.err
mat1.cov.LN.m[i,21]=summary(weib.model.cov.LN.m)$table[2,3]#statistic
mat1.cov.LN.m[i,22]=summary(weib.model.cov.LN.m)$table[2,4]#p-val
mat1.cov.LN.m[i,23]=summary(weib.model.cov.LN.m)$table[3,1]#coeff
mat1.cov.LN.m[i,24]=summary(weib.model.cov.LN.m)$table[3,2]#std.err
mat1.cov.LN.m[i,25]=summary(weib.model.cov.LN.m)$table[3,3]#statistic
mat1.cov.LN.m[i,26]=summary(weib.model.cov.LN.m)$table[3,4]#p-val

exp.model.cov.LN.m=survreg(Surv(stop,status)~as.factor(x)+x.1,
  dist="exp",data=sim.data.cov.LN.m)

mat1.cov.LN.m[i,29]=summary(exp.model.cov.LN.m)$table[1,1]#coeff
mat1.cov.LN.m[i,30]=summary(exp.model.cov.LN.m)$table[1,2]#std.err
mat1.cov.LN.m[i,31]=summary(exp.model.cov.LN.m)$table[1,3]#statistic
mat1.cov.LN.m[i,32]=summary(exp.model.cov.LN.m)$table[1,4]#p-val
mat1.cov.LN.m[i,33]=summary(exp.model.cov.LN.m)$table[2,1]#coeff
mat1.cov.LN.m[i,34]=summary(exp.model.cov.LN.m)$table[2,2]#std.err
mat1.cov.LN.m[i,35]=summary(exp.model.cov.LN.m)$table[2,3]#statistic
mat1.cov.LN.m[i,36]=summary(exp.model.cov.LN.m)$table[2,4]#p-val
mat1.cov.LN.m[i,37]=summary(exp.model.cov.LN.m)$table[3,1]#coeff

```

```

mat1.cov.LN.m[i,38]=summary(exp.model.cov.LN.m)$table[3,2]#std.err
mat1.cov.LN.m[i,39]=summary(exp.model.cov.LN.m)$table[3,3]#statistic
mat1.cov.LN.m[i,40]=summary(exp.model.cov.LN.m)$table[3,4]#p-val

logl.model.cov.LN.m=survreg(Surv(stop,status)~as.factor(x)+x.1,
  dist="loglogistic",data=sim.data.cov.LN.m)

mat1.cov.LN.m[i,43]=summary(logl.model.cov.LN.m)$table[1,1]#coeff
mat1.cov.LN.m[i,44]=summary(logl.model.cov.LN.m)$table[1,2]#std.err
mat1.cov.LN.m[i,45]=summary(logl.model.cov.LN.m)$table[1,3]#statistic
mat1.cov.LN.m[i,46]=summary(logl.model.cov.LN.m)$table[1,4]#p-val
mat1.cov.LN.m[i,47]=summary(logl.model.cov.LN.m)$table[2,1]#coeff
mat1.cov.LN.m[i,48]=summary(logl.model.cov.LN.m)$table[2,2]#std.err
mat1.cov.LN.m[i,49]=summary(logl.model.cov.LN.m)$table[2,3]#statistic
mat1.cov.LN.m[i,50]=summary(logl.model.cov.LN.m)$table[2,4]#p-val
mat1.cov.LN.m[i,51]=summary(logl.model.cov.LN.m)$table[3,1]#coeff
mat1.cov.LN.m[i,52]=summary(logl.model.cov.LN.m)$table[3,2]#std.err
mat1.cov.LN.m[i,53]=summary(logl.model.cov.LN.m)$table[3,3]#statistic
mat1.cov.LN.m[i,54]=summary(logl.model.cov.LN.m)$table[3,4]#p-val

logNorm.model.cov.LN.m=survreg(Surv(stop,status)~as.factor(x)+ x.1,
  dist="lognormal",data=sim.data.cov.LN.m)

```

```

mat1.cov.LN.m[i,57]=summary(logNorm.model.cov.LN.m)$table[1,1]#coeff
mat1.cov.LN.m[i,58]=summary(logNorm.model.cov.LN.m)$table[1,2]#std.err
mat1.cov.LN.m[i,59]=summary(logNorm.model.cov.LN.m)$table[1,3]#statistic
mat1.cov.LN.m[i,60]=summary(logNorm.model.cov.LN.m)$table[1,4]#p-val
mat1.cov.LN.m[i,61]=summary(logNorm.model.cov.LN.m)$table[2,1]#coeff
mat1.cov.LN.m[i,62]=summary(logNorm.model.cov.LN.m)$table[2,2]#std.err
mat1.cov.LN.m[i,63]=summary(logNorm.model.cov.LN.m)$table[2,3]#statistic
mat1.cov.LN.m[i,64]=summary(logNorm.model.cov.LN.m)$table[2,4]#p-val
mat1.cov.LN.m[i,65]=summary(logNorm.model.cov.LN.m)$table[3,1]#coeff
mat1.cov.LN.m[i,66]=summary(logNorm.model.cov.LN.m)$table[3,2]#std.err
mat1.cov.LN.m[i,67]=summary(logNorm.model.cov.LN.m)$table[3,3]#statistic
mat1.cov.LN.m[i,68]=summary(logNorm.model.cov.LN.m)$table[3,4]#p-val

```

```
#####AIC
```

```

mat1.cov.LN.m[i,13]<-extractAIC(cox.model.cov.LN.m)[2];
mat1.cov.LN.m[i,14]<-AIC(cox.model.cov.LN.m,k = log(length(start)))
mat1.cov.LN.m[i,27]<-extractAIC(weib.model.cov.LN.m)[2];
mat1.cov.LN.m[i,28]<-AIC(weib.model.cov.LN.m,k = log(length(start)))
mat1.cov.LN.m[i,41]<-extractAIC(exp.model.cov.LN.m)[2];

```

```

mat1.cov.LN.m[i,42]<-AIC(exp.model.cov.LN.m,k = log(length(start)))
mat1.cov.LN.m[i,55]<-extractAIC(logl.model.cov.LN.m)[2];
mat1.cov.LN.m[i,56]<-AIC(logl.model.cov.LN.m,k = log(length(start)))
mat1.cov.LN.m[i,69]<-extractAIC(logNorm.model.cov.LN.m)[2];
mat1.cov.LN.m[i,70]<-AIC(logNorm.model.cov.LN.m,k = log(length(start)))

}

colMeans(mat1.cov.LN.m)

result1.cov.LN.m=data.frame(matrix(colMeans(mat1.cov.LN.m),
ncol=14,byrow=T))

rownames(result1.cov.LN.m)<-c("cox","weib","exp","logl","logNorm")
colnames(result1.cov.LN.m)<-c("intercept","stderr","statistic",
"p-value","coeff1","stderr","statistic","p-value","coeff2",
"stderr","statistic","p-value","AIC","BIC")

result1.cov.LN.m

###simulation of lognormal censord time to event data(large data)

sim.size.cov.LN.L<-50

mat1.cov.LN.L<-matrix(0,ncol=70,nrow=sim.size.cov.LN.L)

```

```

for(i in 1:sim.size.cov.LN.L){
sim.data.cov.LN.L= simple.surv.sim(n=100000, foltime=300,
dist.ev=c('lnorm'), anc.ev=c(0.69978200185280), beta0.ev=
c(5.84298525742252), anc.cens=1.17783687569519,
beta0.cens=7.39773677281100, z=list(c("unif", 0.8, 1.2)), dist.cens="lnorm",
beta=list(c(-0.4), c(0)), x=list(c("bern", 0.5), c("unif", 0.7, 1.3)))

data.frame(sim.data.cov.LN.L)

attach(sim.data.cov.LN.L)

cox.model.cov.LN.L =coxph(Surv(stop, status) ~ as.factor(x) + x.1,
data=sim.data.cov.LN.L, method="breslow")

mat1.cov.LN.L[i,5]=summary(cox.model.cov.LN.L)$coefficients[1,1]#coeff
mat1.cov.LN.L[i,6]=summary(cox.model.cov.LN.L)$coefficients[1,3]#std.err
mat1.cov.LN.L[i,7]=summary(cox.model.cov.LN.L)$coefficients[1,4]#statistic
mat1.cov.LN.L[i,8]=summary(cox.model.cov.LN.L)$coefficients[1,5]#p-val
mat1.cov.LN.L[i,9]=summary(cox.model.cov.LN.L)$coefficients[2,1]#coeff
mat1.cov.LN.L[i,10]=summary(cox.model.cov.LN.L)$coefficients[2,3]#std.err
mat1.cov.LN.L[i,11]=summary(cox.model.cov.LN.L)$coefficients[2,4]#statistic
mat1.cov.LN.L[i,12]=summary(cox.model.cov.LN.L)$coefficients[2,5]#p-val

```

```

weib.model.cov.LN.L= survreg(Surv(stop, status)~as.factor(x)+x.1,
  dist="weibull",data=sim.data.cov.LN.L)

mat1.cov.LN.L[i,15]=summary(weib.model.cov.LN.L)$table[1,1]#coeff
mat1.cov.LN.L[i,16]=summary(weib.model.cov.LN.L)$table[1,2]#std.err
mat1.cov.LN.L[i,17]=summary(weib.model.cov.LN.L)$table[1,3]#statistic
mat1.cov.LN.L[i,18]=summary(weib.model.cov.LN.L)$table[1,4]#p-val
mat1.cov.LN.L[i,19]=summary(weib.model.cov.LN.L)$table[2,1]#coeff
mat1.cov.LN.L[i,20]=summary(weib.model.cov.LN.L)$table[2,2]#std.err
mat1.cov.LN.L[i,21]=summary(weib.model.cov.LN.L)$table[2,3]#statistic
mat1.cov.LN.L[i,22]=summary(weib.model.cov.LN.L)$table[2,4]#p-val
mat1.cov.LN.L[i,23]=summary(weib.model.cov.LN.L)$table[3,1]#coeff
mat1.cov.LN.L[i,24]=summary(weib.model.cov.LN.L)$table[3,2]#std.err
mat1.cov.LN.L[i,25]=summary(weib.model.cov.LN.L)$table[3,3]#statistic
mat1.cov.LN.L[i,26]=summary(weib.model.cov.LN.L)$table[3,4]#p-val

exp.model.cov.LN.L=survreg(Surv(stop, status)~as.factor(x)+x.1,
  dist="exp",data=sim.data.cov.LN.L)

mat1.cov.LN.L[i,29]=summary(exp.model.cov.LN.L)$table[1,1]#coeff
mat1.cov.LN.L[i,30]=summary(exp.model.cov.LN.L)$table[1,2]#std.err
mat1.cov.LN.L[i,31]=summary(exp.model.cov.LN.L)$table[1,3]#statistic
mat1.cov.LN.L[i,32]=summary(exp.model.cov.LN.L)$table[1,4]#p-val
mat1.cov.LN.L[i,33]=summary(exp.model.cov.LN.L)$table[2,1]#coeff

```

```

mat1.cov.LN.L[i,34]=summary(exp.model.cov.LN.L)$table[2,2]#std.err
mat1.cov.LN.L[i,35]=summary(exp.model.cov.LN.L)$table[2,3]#statistic
mat1.cov.LN.L[i,36]=summary(exp.model.cov.LN.L)$table[2,4]#p-val
mat1.cov.LN.L[i,37]=summary(exp.model.cov.LN.L)$table[3,1]#coeff
mat1.cov.LN.L[i,38]=summary(exp.model.cov.LN.L)$table[3,2]#std.err
mat1.cov.LN.L[i,39]=summary(exp.model.cov.LN.L)$table[3,3]#statistic
mat1.cov.LN.L[i,40]=summary(exp.model.cov.LN.L)$table[3,4]#p-val

log1.model.cov.LN.L=survreg(Surv(stop,status)~as.factor(x)+x.1,
  dist="loglogistic",data=sim.data.cov.LN.L)

mat1.cov.LN.L[i,43]=summary(log1.model.cov.LN.L)$table[1,1]#coeff
mat1.cov.LN.L[i,44]=summary(log1.model.cov.LN.L)$table[1,2]#std.err
mat1.cov.LN.L[i,45]=summary(log1.model.cov.LN.L)$table[1,3]#statistic
mat1.cov.LN.L[i,46]=summary(log1.model.cov.LN.L)$table[1,4]#p-val
mat1.cov.LN.L[i,47]=summary(log1.model.cov.LN.L)$table[2,1]#coeff
mat1.cov.LN.L[i,48]=summary(log1.model.cov.LN.L)$table[2,2]#std.err
mat1.cov.LN.L[i,49]=summary(log1.model.cov.LN.L)$table[2,3]#statistic
mat1.cov.LN.L[i,50]=summary(log1.model.cov.LN.L)$table[2,4]#p-val
mat1.cov.LN.L[i,51]=summary(log1.model.cov.LN.L)$table[3,1]#coeff
mat1.cov.LN.L[i,52]=summary(log1.model.cov.LN.L)$table[3,2]#std.err
mat1.cov.LN.L[i,53]=summary(log1.model.cov.LN.L)$table[3,3]#statistic
mat1.cov.LN.L[i,54]=summary(log1.model.cov.LN.L)$table[3,4]#p-val

```



```

logNorm.model.cov.LN.L=survreg(Surv(stop,status)~as.factor(x)+ x.1,
  dist="lognormal",data=sim.data.cov.LN.L)

mat1.cov.LN.L[i,57]=summary(logNorm.model.cov.LN.L)$table[1,1]#coeff
mat1.cov.LN.L[i,58]=summary(logNorm.model.cov.LN.L)$table[1,2]#std.err
mat1.cov.LN.L[i,59]=summary(logNorm.model.cov.LN.L)$table[1,3]#statistic
mat1.cov.LN.L[i,60]=summary(logNorm.model.cov.LN.L)$table[1,4]#p-val
mat1.cov.LN.L[i,61]=summary(logNorm.model.cov.LN.L)$table[2,1]#coeff
mat1.cov.LN.L[i,62]=summary(logNorm.model.cov.LN.L)$table[2,2]#std.err
mat1.cov.LN.L[i,63]=summary(logNorm.model.cov.LN.L)$table[2,3]#statistic
mat1.cov.LN.L[i,64]=summary(logNorm.model.cov.LN.L)$table[2,4]#p-val
mat1.cov.LN.L[i,65]=summary(logNorm.model.cov.LN.L)$table[3,1]#coeff
mat1.cov.LN.L[i,66]=summary(logNorm.model.cov.LN.L)$table[3,2]#std.err
mat1.cov.LN.L[i,67]=summary(logNorm.model.cov.LN.L)$table[3,3]#statistic
mat1.cov.LN.L[i,68]=summary(logNorm.model.cov.LN.L)$table[3,4]#p-val

#####AIC

mat1.cov.LN.L[i,13]<-extractAIC(cox.model.cov.LN.L)[2];
mat1.cov.LN.L[i,14]<-AIC(cox.model.cov.LN.L,k = log(length(start)))

```

```

mat1.cov.LN.L[i,27]<-extractAIC(weib.model.cov.LN.L)[2];
mat1.cov.LN.L[i,28]<-AIC(weib.model.cov.LN.L,k = log(length(start)))
mat1.cov.LN.L[i,41]<-extractAIC(exp.model.cov.LN.L)[2];
mat1.cov.LN.L[i,42]<-AIC(exp.model.cov.LN.L,k = log(length(start)))
mat1.cov.LN.L[i,55]<-extractAIC(logl.model.cov.LN.L)[2];
mat1.cov.LN.L[i,56]<-AIC(logl.model.cov.LN.L,k = log(length(start)))
mat1.cov.LN.L[i,69]<-extractAIC(logNorm.model.cov.LN.L)[2];
mat1.cov.LN.L[i,70]<-AIC(logNorm.model.cov.LN.L,k = log(length(start)))

}

colMeans(mat1.cov.LN.L)

result1.cov.LN.L<-data.frame(matrix(colMeans(mat1.cov.LN.L),ncol=14,
byrow=T))

rownames(result1.cov.LN.L)<-c("cox","weib","exp","logl","logNorm")
colnames(result1.cov.LN.L)<-c("intercept","stderr","statistic",
"p-value","coeff1","stderr","statistic","p-value","coeff2","stderr",
"statistic","p-value","AIC","BIC")

result1.cov.LN.L

#####

```

```

Simulation of Cox with covariate

#####

###simulation of Cox censord time to event data (small data)

sim.size.cov.cox.s<-1000

mat1.cov.cox.s<-matrix(0,ncol=70,nrow=sim.size.cov.cox.s)

for(i in 1:sim.size.cov.cox.s){

sim.data.cov.cox.s = sim.survdata(N=8, T=365, censor.cond=TRUE,

censor=.35, num.data.frames = 1)

attach(sim.data.cov.cox.s )

cox.model.cov.cox.s =coxph(Surv(y, failed)~X1+X2,

data=sim.data.cov.cox.s$data,method="breslow")

mat1.cov.cox.s[i,5]=summary(cox.model.cov.cox.s

)$coefficients[1,1]#coeff

mat1.cov.cox.s[i,6]=summary(cox.model.cov.cox.s

)$coefficients[1,3]#std.err

mat1.cov.cox.s[i,7]=summary(cox.model.cov.cox.s

)$coefficients[1,4]#statistic

mat1.cov.cox.s[i,8]=summary(cox.model.cov.cox.s

```

```

)$coefficients[1,5]#p-val

mat1.cov.cox.s[i,9]=summary(cox.model.cov.cox.s

)$coefficients[2,1]#coeff

mat1.cov.cox.s[i,10]=summary(cox.model.cov.cox.s

)$coefficients[2,3]#std.err

mat1.cov.cox.s[i,11]=summary(cox.model.cov.cox.s

)$coefficients[2,4]#statistic

mat1.cov.cox.s[i,12]=summary(cox.model.cov.cox.s

)$coefficients[2,5]#p-val

weib.model.cov.cox.s=survreg(Surv(y,failed)~X1+X2,dist="weibull",

data=sim.data.cov.cox.s$data)

mat1.cov.cox.s[i,15]=summary(weib.model.cov.cox.s)$table[1,1]#coeff

mat1.cov.cox.s[i,16]=summary(weib.model.cov.cox.s)$table[1,2]#std.err

mat1.cov.cox.s[i,17]=summary(weib.model.cov.cox.s)$table[1,3]#statistic

mat1.cov.cox.s[i,18]=summary(weib.model.cov.cox.s)$table[1,4]#p-val

mat1.cov.cox.s[i,19]=summary(weib.model.cov.cox.s)$table[2,1]#coeff

mat1.cov.cox.s[i,20]=summary(weib.model.cov.cox.s)$table[2,2]#std.err

mat1.cov.cox.s[i,21]=summary(weib.model.cov.cox.s)$table[2,3]#statistic

mat1.cov.cox.s[i,22]=summary(weib.model.cov.cox.s)$table[2,4]#p-val

mat1.cov.cox.s[i,23]=summary(weib.model.cov.cox.s)$table[3,1]#coeff

```

```

mat1.cov.cox.s[i,24]=summary(weib.model.cov.cox.s)$table[3,2]#std.err
mat1.cov.cox.s[i,25]=summary(weib.model.cov.cox.s)$table[3,3]#statistic
mat1.cov.cox.s[i,26]=summary(weib.model.cov.cox.s)$table[3,4]#p-val

exp.model.cov.cox.s=survreg(Surv(y, failed)~X1+X2,dist="exp",
data=sim.data.cov.cox.s$data)

mat1.cov.cox.s[i,29]=summary(exp.model.cov.cox.s)$table[1,1]#coeff
mat1.cov.cox.s[i,30]=summary(exp.model.cov.cox.s)$table[1,2]#std.err
mat1.cov.cox.s[i,31]=summary(exp.model.cov.cox.s)$table[1,3]#statistic
mat1.cov.cox.s[i,32]=summary(exp.model.cov.cox.s)$table[1,4]#p-val
mat1.cov.cox.s[i,33]=summary(exp.model.cov.cox.s)$table[2,1]#coeff
mat1.cov.cox.s[i,34]=summary(exp.model.cov.cox.s)$table[2,2]#std.err
mat1.cov.cox.s[i,35]=summary(exp.model.cov.cox.s)$table[2,3]#statistic
mat1.cov.cox.s[i,36]=summary(exp.model.cov.cox.s)$table[2,4]#p-val
mat1.cov.cox.s[i,37]=summary(exp.model.cov.cox.s)$table[3,1]#coeff
mat1.cov.cox.s[i,38]=summary(exp.model.cov.cox.s)$table[3,2]#std.err
mat1.cov.cox.s[i,39]=summary(exp.model.cov.cox.s)$table[3,3]#statistic
mat1.cov.cox.s[i,40]=summary(exp.model.cov.cox.s )$table[3,4]#p-val

```

```

logl.model.cov.cox.s=survreg(Surv(y,failed)~X1+X2 ,
dist="loglogistic", data=sim.data.cov.cox.s$data)

mat1.cov.cox.s[i,43]=summary(logl.model.cov.cox.s)$table[1,1]#coeff
mat1.cov.cox.s[i,44]=summary(logl.model.cov.cox.s)$table[1,2]#std.err
mat1.cov.cox.s[i,45]=summary(logl.model.cov.cox.s)$table[1,3]#statistic
mat1.cov.cox.s[i,46]=summary(logl.model.cov.cox.s)$table[1,4]#p-val
mat1.cov.cox.s[i,47]=summary(logl.model.cov.cox.s)$table[2,1]#coeff
mat1.cov.cox.s[i,48]=summary(logl.model.cov.cox.s)$table[2,2]#std.err
mat1.cov.cox.s[i,49]=summary(logl.model.cov.cox.s)$table[2,3]#statistic
mat1.cov.cox.s[i,50]=summary(logl.model.cov.cox.s)$table[2,4]#p-val
mat1.cov.cox.s[i,51]=summary(logl.model.cov.cox.s)$table[3,1]#coeff
mat1.cov.cox.s[i,52]=summary(logl.model.cov.cox.s)$table[3,2]#std.err
mat1.cov.cox.s[i,53]=summary(logl.model.cov.cox.s)$table[3,3]#statistic
mat1.cov.cox.s[i,54]=summary(logl.model.cov.cox.s)$table[3,4]#p-val

logNorm.model.cov.cox.s=survreg(Surv(y, failed) ~ X1+X2
, dist="lognormal", data=sim.data.cov.cox.s$data)

mat1.cov.cox.s[i,57]=summary(logNorm.model.cov.cox.s)$table[1,1]#coeff
mat1.cov.cox.s[i,58]=summary(logNorm.model.cov.cox.s)$table[1,2]#std.err
mat1.cov.cox.s[i,59]=summary(logNorm.model.cov.cox.s)$table[1,3]#statistic
mat1.cov.cox.s[i,60]=summary(logNorm.model.cov.cox.s)$table[1,4]#p-val

```

```

mat1.cov.cox.s[i,61]=summary(logNorm.model.cov.cox.s)$table[2,1]#coeff
mat1.cov.cox.s[i,62]=summary(logNorm.model.cov.cox.s)$table[2,2]#std.err
mat1.cov.cox.s[i,63]=summary(logNorm.model.cov.cox.s)$table[2,3]#statistic
mat1.cov.cox.s[i,64]=summary(logNorm.model.cov.cox.s)$table[2,4]#p-val
mat1.cov.cox.s[i,65]=summary(logNorm.model.cov.cox.s)$table[3,1]#coeff
mat1.cov.cox.s[i,66]=summary(logNorm.model.cov.cox.s)$table[3,2] #std.err
mat1.cov.cox.s[i,67]=summary(logNorm.model.cov.cox.s)$table[3,3]#statistic
mat1.cov.cox.s[i,68]=summary(logNorm.model.cov.cox.s )$table[3,4]#p-val

#####AIC

mat1.cov.cox.s[i,13]<-extractAIC(cox.model.cov.cox.s)[2];
mat1.cov.cox.s[i,14]<-AIC(cox.model.cov.cox.s,k = log(length(start)))
mat1.cov.cox.s[i,27]<-extractAIC(weib.model.cov.cox.s)[2];
mat1.cov.cox.s[i,28]<-AIC(weib.model.cov.cox.s,k = log(length(start)))
mat1.cov.cox.s[i,41]<-extractAIC(exp.model.cov.cox.s)[2];
mat1.cov.cox.s[i,42]<-AIC(exp.model.cov.cox.s,k = log(length(start)))
mat1.cov.cox.s[i,55]<-extractAIC(logl.model.cov.cox.s)[2];
mat1.cov.cox.s[i,56]<-AIC(logl.model.cov.cox.s,k = log(length(start)))
mat1.cov.cox.s[i,69]<-extractAIC(logNorm.model.cov.cox.s)[2];
mat1.cov.cox.s[i,70]<-AIC(logNorm.model.cov.cox.s,k = log(length(start)))

```

```
}
```

```
colMeans(mat1.cov.cox.s)  
result1.cov.cox.s<-data.frame(matrix(colMeans(mat1.cov.cox.s),ncol=14,  
byrow=T))  
rownames(result1.cov.cox.s)<-c("cox","weib","exp","logl","logNorm")  
colnames(result1.cov.cox.s)<-c("intercept","stderr","statistic",  
"p-value","coeff1","stderr","statistic","p-value","coeff2","stderr",  
"statistic","p-value","AIC","BIC")  
result1.cov.cox.s
```

```
#####
```

```
###simulation of Cox censord time to event data(medium data)
```

```
sim.size.cov.cox.m<-500
```

```
mat1.cov.cox.m<-matrix(0,ncol=70,nrow=sim.size.cov.cox.m)
```

```
for(i in 1:sim.size.cov.cox.m){
```

```
sim.data.cov.cox.m = sim.survdata(N=1000, T=365, ,censor.cond=TRUE,
```



```

censor=.35, num.data.frames = 1)

attach(sim.data.cov.cox.m )

cox.model.cov.cox.m =coxph(Surv(y, failed) ~ X1+X2,
  data=sim.data.cov.cox.m$data,method="breslow")

mat1.cov.cox.m[i,5]=summary(cox.model.cov.cox.m
)$coefficients[1,1]#coeff

mat1.cov.cox.m[i,6]=summary(cox.model.cov.cox.m
)$coefficients[1,3]#std.err

mat1.cov.cox.m[i,7]=summary(cox.model.cov.cox.m
)$coefficients[1,4]#statistic

mat1.cov.cox.m[i,8]=summary(cox.model.cov.cox.m
)$coefficients[1,5]#p-val

mat1.cov.cox.m[i,9]=summary(cox.model.cov.cox.m
)$coefficients[2,1]#coeff

mat1.cov.cox.m[i,10]=summary(cox.model.cov.cox.m
)$coefficients[2,3]#std.err

mat1.cov.cox.m[i,11]=summary(cox.model.cov.cox.m
)$coefficients[2,4]#statistic

mat1.cov.cox.m[i,12]=summary(cox.model.cov.cox.m
)$coefficients[2,5]#p-val

```

```

weib.model.cov.cox.m=survreg(Surv(y, failed)~X1+X2,dist="weibull",
  data=sim.data.cov.cox.m$data)

mat1.cov.cox.m[i,15]=summary(weib.model.cov.cox.m)$table[1,1]#coeff
mat1.cov.cox.m[i,16]=summary(weib.model.cov.cox.m)$table[1,2]#std.err
mat1.cov.cox.m[i,17]=summary(weib.model.cov.cox.m)$table[1,3]#statistic
mat1.cov.cox.m[i,18]=summary(weib.model.cov.cox.m)$table[1,4]#p-val
mat1.cov.cox.m[i,19]=summary(weib.model.cov.cox.m)$table[2,1]#coeff
mat1.cov.cox.m[i,20]=summary(weib.model.cov.cox.m)$table[2,2]#std.err
mat1.cov.cox.m[i,21]=summary(weib.model.cov.cox.m)$table[2,3]#statistic
mat1.cov.cox.m[i,22]=summary(weib.model.cov.cox.m)$table[2,4]#p-val
mat1.cov.cox.m[i,23]=summary(weib.model.cov.cox.m)$table[3,1]#coeff
mat1.cov.cox.m[i,24]=summary(weib.model.cov.cox.m)$table[3,2]#std.err
mat1.cov.cox.m[i,25]=summary(weib.model.cov.cox.m)$table[3,3]#statistic
mat1.cov.cox.m[i,26]=summary(weib.model.cov.cox.m)$table[3,4]#p-val

exp.model.cov.cox.m=survreg(Surv(y, failed)~X1+X2,dist="exp",
  data=sim.data.cov.cox.m$data)

mat1.cov.cox.m[i,29]=summary(exp.model.cov.cox.m)$table[1,1]#coeff
mat1.cov.cox.m[i,30]=summary(exp.model.cov.cox.m)$table[1,2]#std.err
mat1.cov.cox.m[i,31]=summary(exp.model.cov.cox.m)$table[1,3]#statistic

```

```

mat1.cov.cox.m[i,32]=summary(exp.model.cov.cox.m)$table[1,4]#p-val
mat1.cov.cox.m[i,33]=summary(exp.model.cov.cox.m)$table[2,1]#coeff
mat1.cov.cox.m[i,34]=summary(exp.model.cov.cox.m)$table[2,2]#std.err
mat1.cov.cox.m[i,35]=summary(exp.model.cov.cox.m)$table[2,3]#statistic
mat1.cov.cox.m[i,36]=summary(exp.model.cov.cox.m)$table[2,4]#p-val
mat1.cov.cox.m[i,37]=summary(exp.model.cov.cox.m)$table[3,1]#coeff
mat1.cov.cox.m[i,38]=summary(exp.model.cov.cox.m)$table[3,2]#std.err
mat1.cov.cox.m[i,39]=summary(exp.model.cov.cox.m)$table[3,3]#statistic
mat1.cov.cox.m[i,40]=summary(exp.model.cov.cox.m)$table[3,4]#p-val

log1.model.cov.cox.m=survreg(Surv(y, failed) ~ X1+ X2,
dist="loglogistic",data=sim.data.cov.cox.m$data)

mat1.cov.cox.m[i,43]=summary(log1.model.cov.cox.m)$table[1,1]#coeff
mat1.cov.cox.m[i,44]=summary(log1.model.cov.cox.m)$table[1,2]#std.err
mat1.cov.cox.m[i,45]=summary(log1.model.cov.cox.m)$table[1,3]#statistic
mat1.cov.cox.m[i,46]=summary(log1.model.cov.cox.m)$table[1,4]#p-val
mat1.cov.cox.m[i,47]=summary(log1.model.cov.cox.m)$table[2,1]#coeff
mat1.cov.cox.m[i,48]=summary(log1.model.cov.cox.m)$table[2,2]#std.err
mat1.cov.cox.m[i,49]=summary(log1.model.cov.cox.m)$table[2,3]#statistic
mat1.cov.cox.m[i,50]=summary(log1.model.cov.cox.m)$table[2,4]#p-val
mat1.cov.cox.m[i,51]=summary(log1.model.cov.cox.m)$table[3,1]#coeff

```

```
mat1.cov.cox.m[i,52]=summary(logl.model.cov.cox.m)$table[3,2]#std.err
mat1.cov.cox.m[i,53]=summary(logl.model.cov.cox.m)$table[3,3]#statistic
mat1.cov.cox.m[i,54]=summary(logl.model.cov.cox.m)$table[3,4]#p-val
```

```
logNorm.model.cov.cox.m=survreg(Surv(y, failed) ~ X1 + X2
, dist="lognormal", data=sim.data.cov.cox.m$data)
mat1.cov.cox.m[i,57]<-summary(logNorm.model.cov.cox.m
)$table[1,1] #coeff
mat1.cov.cox.m[i,58]<-summary(logNorm.model.cov.cox.m
)$table[1,2] #std.err
mat1.cov.cox.m[i,59]<-summary(logNorm.model.cov.cox.m
)$table[1,3] #statistic
mat1.cov.cox.m[i,60]<-summary(logNorm.model.cov.cox.m
)$table[1,4] #p-val
mat1.cov.cox.m[i,61]<-summary(logNorm.model.cov.cox.m
)$table[2,1] #coeff
mat1.cov.cox.m[i,62]<-summary(logNorm.model.cov.cox.m
)$table[2,2] #std.err
mat1.cov.cox.m[i,63]<-summary(logNorm.model.cov.cox.m
)$table[2,3] #statistic
```

```

mat1.cov.cox.m[i,64]<-summary(logNorm.model.cov.cox.m
)$table[2,4] #p-val

mat1.cov.cox.m[i,65]<-summary(logNorm.model.cov.cox.m
)$table[3,1] #coeff

mat1.cov.cox.m[i,66]<-summary(logNorm.model.cov.cox.m
)$table[3,2] #std.err

mat1.cov.cox.m[i,67]<-summary(logNorm.model.cov.cox.m
)$table[3,3] #statistic

mat1.cov.cox.m[i,68]<-summary(logNorm.model.cov.cox.m
)$table[3,4] #p-val

#####AIC

mat1.cov.cox.m[i,13]<-extractAIC(cox.model.cov.cox.m)[2];
mat1.cov.cox.m[i,14]<-AIC(cox.model.cov.cox.m,k = log(length(start)))
mat1.cov.cox.m[i,27]<-extractAIC(weib.model.cov.cox.m)[2];
mat1.cov.cox.m[i,28]<-AIC(weib.model.cov.cox.m,k = log(length(start)))
mat1.cov.cox.m[i,41]<-extractAIC(exp.model.cov.cox.m)[2];
mat1.cov.cox.m[i,42]<-AIC(exp.model.cov.cox.m,k = log(length(start)))
mat1.cov.cox.m[i,55]<-extractAIC(logl.model.cov.cox.m)[2];
mat1.cov.cox.m[i,56]<-AIC(logl.model.cov.cox.m,k = log(length(start)))
mat1.cov.cox.m[i,69]<-extractAIC(logNorm.model.cov.cox.m)[2];

```

```

mat1.cov.cox.m[i,70]<-AIC(logNorm.model.cov.cox.m,k =
  log(length(start)))

}

colMeans(mat1.cov.cox.m)

result1.cov.cox.m<-data.frame(matrix(colMeans(mat1.cov.cox.m),
ncol=14,byrow=T))

rownames(result1.cov.cox.m)<-c("cox","weib","exp","logl","logNorm")
colnames(result1.cov.cox.m)<-c("intercept","stderr","statistic",
"p-value","coeff1","stderr","statistic","p-value","coeff2","stderr",
"statistic","p-value","AIC","BIC")

result1.cov.cox.m

#####

##simulation of Cox Censord time to event data(large data)

sim.size.cov.cox.L<-50

mat1.cov.cox.L<-matrix(0,ncol=70,nrow=sim.size.cov.cox.L)

```

```

for(i in 1:sim.size.cov.cox.L){

sim.data.cov.cox.L = sim.survdata(N=100000, T=360, censor=.35,

  num.data.frames = 1)

attach(sim.data.cov.cox.L )

cox.model.cov.cox.L =coxph(Surv(y, failed) ~ X1 + X2 ,

  data=sim.data.cov.cox.L$data,method="breslow")

mat1.cov.cox.L[i,5]<-summary(cox.model.cov.cox.L

)$coefficients[1,1] #coeff

mat1.cov.cox.L[i,6]=summary(cox.model.cov.cox.L

)$coefficients[1,3] #std.err

mat1.cov.cox.L[i,7]=summary(cox.model.cov.cox.L

)$coefficients[1,4] #statistic

mat1.cov.cox.L[i,8]<-summary(cox.model.cov.cox.L

)$coefficients[1,5] #p-val

mat1.cov.cox.L[i,9]=summary(cox.model.cov.cox.L

)$coefficients[2,1] #coeff

mat1.cov.cox.L[i,10]=summary(cox.model.cov.cox.L

)$coefficients[2,3] #std.err

mat1.cov.cox.L[i,11]=summary(cox.model.cov.cox.L

)$coefficients[2,4] #statistic

```

```

mat1.cov.cox.L[i,12]<-summary(cox.model.cov.cox.L
)$coefficients[2,5] #p-val

weib.model.cov.cox.L=survreg(Surv(y, failed) ~ X1 + X2
, dist="weibull", data=sim.data.cov.cox.L$data)

mat1.cov.cox.L[i,15]=summary(weib.model.cov.cox.L)$table[1,1]#coeff
mat1.cov.cox.L[i,16]=summary(weib.model.cov.cox.L)$table[1,2]#std.err
mat1.cov.cox.L[i,17]=summary(weib.model.cov.cox.L)$table[1,3]#statistic
mat1.cov.cox.L[i,18]=summary(weib.model.cov.cox.L)$table[1,4]#p-val
mat1.cov.cox.L[i,19]=summary(weib.model.cov.cox.L)$table[2,1]#coeff
mat1.cov.cox.L[i,20]=summary(weib.model.cov.cox.L)$table[2,2]#std.err
mat1.cov.cox.L[i,21]=summary(weib.model.cov.cox.L)$table[2,3]#statistic
mat1.cov.cox.L[i,22]=summary(weib.model.cov.cox.L)$table[2,4]#p-val
mat1.cov.cox.L[i,23]=summary(weib.model.cov.cox.L)$table[3,1]#coeff
mat1.cov.cox.L[i,24]=summary(weib.model.cov.cox.L)$table[3,2]#std.err
mat1.cov.cox.L[i,25]=summary(weib.model.cov.cox.L)$table[3,3]#statistic
mat1.cov.cox.L[i,26]=summary(weib.model.cov.cox.L)$table[3,4]#p-val

exp.model.cov.cox.L=survreg(Surv(y, failed)~X1+X2, dist="exp",
data=sim.data.cov.cox.L$data)

```



```

mat1.cov.cox.L[i,29]=summary(exp.model.cov.cox.L)$table[1,1]#coeff
mat1.cov.cox.L[i,30]=summary(exp.model.cov.cox.L)$table[1,2]#std.err
mat1.cov.cox.L[i,31]=summary(exp.model.cov.cox.L)$table[1,3]#statistic
mat1.cov.cox.L[i,32]=summary(exp.model.cov.cox.L)$table[1,4]#p-val
mat1.cov.cox.L[i,33]=summary(exp.model.cov.cox.L)$table[2,1]#coeff
mat1.cov.cox.L[i,34]=summary(exp.model.cov.cox.L)$table[2,2]#std.err
mat1.cov.cox.L[i,35]=summary(exp.model.cov.cox.L)$table[2,3]#statistic
mat1.cov.cox.L[i,36]=summary(exp.model.cov.cox.L)$table[2,4]#p-val
mat1.cov.cox.L[i,37]=summary(exp.model.cov.cox.L)$table[3,1]#coeff
mat1.cov.cox.L[i,38]=summary(exp.model.cov.cox.L)$table[3,2]#std.err
mat1.cov.cox.L[i,39]=summary(exp.model.cov.cox.L)$table[3,3]#statistic
mat1.cov.cox.L[i,40]=summary(exp.model.cov.cox.L)$table[3,4]#p-val

```

```

log1.model.cov.cox.L=survreg(Surv(y, failed)~X1+X2

```

```

, dist="loglogistic", data=sim.data.cov.cox.L$data)

```

```

mat1.cov.cox.L[i,43]=summary(log1.model.cov.cox.L)$table[1,1]#coeff
mat1.cov.cox.L[i,44]=summary(log1.model.cov.cox.L)$table[1,2]#std.err
mat1.cov.cox.L[i,45]=summary(log1.model.cov.cox.L)$table[1,3]#statistic
mat1.cov.cox.L[i,46]=summary(log1.model.cov.cox.L)$table[1,4]#p-val
mat1.cov.cox.L[i,47]=summary(log1.model.cov.cox.L)$table[2,1]#coeff

```

```

mat1.cov.cox.L[i,48]=summary(logl.model.cov.cox.L)$table[2,2]#std.err
mat1.cov.cox.L[i,49]=summary(logl.model.cov.cox.L)$table[2,3]#statistic
mat1.cov.cox.L[i,50]=summary(logl.model.cov.cox.L)$table[2,4]#p-val
mat1.cov.cox.L[i,51]=summary(logl.model.cov.cox.L)$table[3,1]#coeff
mat1.cov.cox.L[i,52]=summary(logl.model.cov.cox.L)$table[3,2]#std.err
mat1.cov.cox.L[i,53]=summary(logl.model.cov.cox.L)$table[3,3]#statistic
mat1.cov.cox.L[i,54]=summary(logl.model.cov.cox.L)$table[3,4]#p-val

```

```

logNorm.model.cov.cox.L=survreg(Surv(y, failed) ~ X1 +
  X2,dist="lognormal", data=sim.data.cov.cox.L$data)
mat1.cov.cox.L[i,57]<-summary(logNorm.model.cov.cox.L
)$table[1,1] #coeff
mat1.cov.cox.L[i,58]<-summary(logNorm.model.cov.cox.L
)$table[1,2] #std.err
mat1.cov.cox.L[i,59]<-summary(logNorm.model.cov.cox.L
)$table[1,3] #statistic
mat1.cov.cox.L[i,60]<-summary(logNorm.model.cov.cox.L
)$table[1,4] #p-val
mat1.cov.cox.L[i,61]<-summary(logNorm.model.cov.cox.L
)$table[2,1] #coeff

```

```

mat1.cov.cox.L[i,62]<-summary(logNorm.model.cov.cox.L
)$table[2,2] #std.err

mat1.cov.cox.L[i,63]<-summary(logNorm.model.cov.cox.L
)$table[2,3] #statistic

mat1.cov.cox.L[i,64]<-summary(logNorm.model.cov.cox.L
)$table[2,4] #p-val

mat1.cov.cox.L[i,65]<-summary(logNorm.model.cov.cox.L
)$table[3,1] #coeff

mat1.cov.cox.L[i,66]<-summary(logNorm.model.cov.cox.L
)$table[3,2] #std.err

mat1.cov.cox.L[i,67]=summary(logNorm.model.cov.cox.L
)$table[3,3] #statistic

mat1.cov.cox.L[i,68]<-summary(logNorm.model.cov.cox.L
)$table[3,4] #p-val

#####AIC

extractAIC(cox.model.cov.cox.L)[2];AIC(cox.model.cov.cox.L,
k = log(length(start)))

extractAIC(weib.model.cov.cox.L)[2];AIC(weib.model.cov.cox.L,
k = log(length(start)))

```

```

extractAIC(exp.model.cov.cox.L)[2];AIC(exp.model.cov.cox.L,
k = log(length(start)))

extractAIC(logl.model.cov.cox.L)[2];AIC(logl.model.cov.cox.L,
k = log(length(start)))

extractAIC(logNorm.model.cov.cox.L)[2];AIC(logNorm.model.cov.cox.L,
k = log(length(start)))

}

```

```

colMeans(mat1.cov.cox.L)

result1.cov.cox.L<-data.frame(matrix(colMeans(mat1.cov.cox.L),
ncol=14,byrow=T))

rownames(result1.cov.cox.L)<-c("cox","weib","exp","logl","logNorm")

colnames(result1.cov.cox.L)<-c("intercept","stderr","statistic",
"p-value","coeff1","stderr","statistic","p-value","coeff2",
"stderr","statistic","p-value","AIC","BIC")

result1.cov.cox.L

```

```
#####
```

```
THE END OF SIMULATION CODES
```

```
#####
```

```
#####
```

```
R Code for Real Data ( Cirvical Cancer Data)
```

```
#####3
```

```
cancer=read.csv("C:/Users/INM/etsudat.csv",header=T)
```

```
data.frame(cancer)
```

```

attach(cancer)

names(cancer)

library(survival)

library(ggplot2)

library(survminer)

mean(age) ##over all mean sample

sd(age) ###over all standard deviation sample

range(age) ##over all range sample

length(age[event==1])###number of patients who died

length(age[event==0])###number of patients who survived

mean(age[event==1])###number of patients who died

sd(age[event==1])###number of patients who died

range(age[event==1])###number of patients who died

mean(age[event==0])###number of patients who survived

sd(age[event==0])###number of patients who survived

range(age[event==0])###number of patients who survived

```

```

#####Kaplan Meier

kp=survfit(Surv(foltime, even)~1,data=cancer)

plot(kp,xlab="Time(days)",ylab="Survival▯probability",
conf.int=F) #Kaplan Meier Graph for no covariate

cumfit = survfit(Surv(foltime,even) ~ 1, data=cancer, type='fleming')
plot(cumfit,fun="cumhaz", xlab="Follow-Up▯Time▯(days)",conf.int=F,
ylab="Cumlative▯Hazard")##Kaplan-Meier Hazard Estimates

#####Fitting models and investigating best model

cox.real =coxph(Surv(foltime, even) ~ age+teletherapy+telebrachy,
data=cancer,method="breslow")

summary(cox.real )

weib.real= survreg(Surv(foltime, even) ~ age+teletherapy+telebrachy,
data=cancer, dist="weibull")

summary(weib.real)

exp.real=survreg(Surv(foltime, even) ~ age+teletherapy+telebrachy,
ata=cancer, dist="exp")

summary(exp.real)

```

```
logl.real= survreg(Surv(foltime, even)~ age+teletherapy+telebrachy,  
data=cancer, dist="loglogistic")  
summary(logl.real)
```

```
logNorm.real= survreg(Surv(foltime, even) ~ age+teletherapy+telebrachy,  
data=cancer, dist="lognormal")  
summary(logNorm.real)
```

```
extractAIC(cox.real)[2]
```

```
extractAIC(weib.real)[2]
```

```
extractAIC(exp.real)[2]
```

```
extractAIC(logl.real)[2]
```

```
extractAIC(logNorm.real)[2]
```

```
AIC(cox.real,k = log(length(even)))
```

```
AIC(weib.real,k = log(length(even)))
```

```
AIC(exp.real,k = log(length(even)))
```

```
AIC(logl.real,k = log(length(even)))
```

```
AIC(logNorm.real,k = log(length(even)))
```



```

#####Fitting cox model and checking assumptions

cox.real =coxph(Surv(foltime, even) ~ age+teletherapy+telebrachy,
data=cancer,method="breslow") fitting the full model

summary(cox.real )

test.ph = cox.zph(cox.real)####testing proportional hazard assumption
test.ph

par(mfrow=c(1,1))

ggcoxzph(test.ph)##producing graphical for CPH assumption

ggcoxdiagnostics(cox.real,type= "dfbeta",linear.predictions= FALSE,
ggtheme = theme_bw()) ##chcking influential observation

ggcoxfunctional(Surv(foltime,even)~age+log(age)+sqrt(age),
data=cancer)#####Checking nonlinearity

#####assessing overall model fit

```

```

cox.snell =even - resid(cox.real,type = "martingale")

coxph.res2 = survfit(Surv(cox.snell, even) ~ 1)

summary(coxph.res2)

Htilde = cumsum(coxph.res2$n.event / coxph.res2$n.risk)

plot((coxph.res2$time), (Htilde), type = 's', col = 'red',xlab="Time"
,ylab="Cumulative Hazard")

abline(0, 1, col = 'blue', lty = 2)

par(mfrow=c(1,1))

#####Fitting individual covariate separately

cox.drug =coxph(Surv(foltime, even) ~ teletherapy+telebrachy,
data=cancer,method="breslow")

summary(cox.drug )

cox.age=coxph(Surv(foltime,even)~age,data=cancer,method="breslow")

summary(cox.age )

#####categorizing age to compare group

age2=ifelse(age>65,1,0)

age2

```

```

cancer.data=data.frame(cancer ,age2)

cancer.data

attach(cancer.data)

cox.age2=coxph(Surv(foltime ,even)~ age2+teletherapy+telebrachy ,data=
cancer ,method="breslow")### full model of two categorical covariates
summary(cox.age2 )

test.ph = cox.zph(cox.age2) ####testing proportional hazard assumption
test.ph

cox.age.alone =coxph(Surv(foltime , even) ~ age2,
data=cancer ,method="breslow") ##fitting for age group separately
summary(cox.age.alone )

```

