Hebron University
Faculty of Graduate Studies
Mathematics Department

Parametric and Nonparametric Survival Analysis of Censored Data with Covariates

Submitted by

Tasneem Altose

Supervisor

Dr. Bader Aljawadi

This Thesis is Submitted in Partial Fulfillment of Requirements of the Degree Master of Mathematics, Faculty of Graduate Studies, Hebron University, Hebron, Palestine.

Oct, 2021

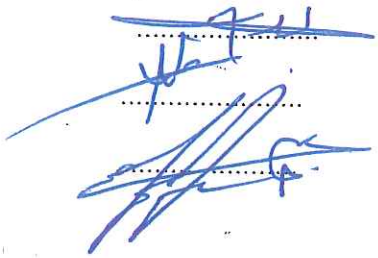
Parametric and Nonparametric Survival Analysis of Censored Data with Covariates

Submitted by

Tasneem Altose

This thesis was defended successfully on 16/12/2021 and approved by:

Committee Members:		Signature
• Dr. Bader Aljawadi	Supervisor	
• Dr. Inad Nawajah	Internal Examiner	
• Dr. Khalid Salah	External Examiner	



DECLARATION

I declare that the master thesis entitled (Parametric and Nonparametric Survival Analysis of Censored Data with Covariates) is my own work, and hereby certify that unless stated, all work contained within this thesis is my own independent research and has not been submitted for the award of any other degree at any institution, except where due acknowledgment is made in the text.

Dedications

I dedicate my thesis to myself and my parents, husband, children, friends, sisters and brother who supported me on each step of the way.

ACKNOWLEDGEMENT

In the name of Allah, the most Gracious, most Merciful.
First and foremost, I thank ALLAH for bestowing me with health, patience, and knowledge to complete this thesis and without ALLAH's grace, we couldn't have done it. So to ALLAH returns all the praise and gratitude.

I would like to express my gratitude to Dr. Bader Aljawadi , the supervisor of my thesis, who was a generous and instructor. I was blessed to be supervised by him. Thanks go to him for his guidance, suggestions and invaluable encouragement throughout the development of this research.

Also, I should thank with great respect and honor all my instructors at Hebron university for their helps and supports during my study.

Finally, I would like to thank my parents and my family members for their encouragement, support, prayers and being always there for me.

Abstract

Survival analysis is a growing branch of statistics that focuses on estimation of expected duration of work, life, graduation or any other studies factor. In this thesis we introduce survival analysis, we explain censoring in survival data as well as the estimation of survival function under different of censoring. There are different approches to estimate the survival function; the parametric and nonparametric, where in parametric the exponential Weibull distributions are employed while in nonparametric some common techniques are used to extract the survival probabilities such as Kaplan Meier and Turbull estimators. The estimation procedure is verified under the most common censoring models; right and interval censoring and a comparison between the two approaches is accomplished in the practical of the thesis.

Contents

Chapter 1

Introduction

1.1 Survival Data and Survival Function

Survival analysis is the branch of statistics for analyzing the expected duration of time till one or more events occur. Particular examples of interested event: death, infection, recurrence of disease, graduation, divorce, malfunctioning of device. survival analysis become widely used in diverse fields, such as medicine, economics and political science. It is called reliability analysis in engineering, duration analysis in economics, and event history in sociology.

This type of analysis aims to answer the questions of: What portion of the population will survive past a certain time? and of those who survive, at what rate will they die or fail? Can multiple causes of death or failure be taken into account? How do particular circumstances or characteristics increase or decrease the likelihood of survival?.

The Survival Function is a function that gives the probability that a patient, device or any other object of interest will survive beyond any specific time(t)[9,15]. so, let T be a continuous variable with cumulative distribution function, $F(t)$, on the interval $[0, \infty)$, then

the survival function can be defined as following:

$$S(t) = P(T > t) = \int_t^{\infty} f(u)du = 1 - F(t)$$

1.2 Hazard Function

Hazard function $h(t)$ is the way to model data distribution in survival analysis, it is the instantaneous risk that the event of interest happens, within a very narrow time frame. The hazard function is a conditional failure rate, in that its conditional a person has actually survived till time t . In other words, The function at year 10 only applies to those who were actually alive in year 10, it doesn't count those who died in previous periods and it is defined as[10]:

$$h(t) = \lim_{dt \to 0} \frac{P[t \leq T < t + dt | T \geq t]}{dt} \tag{1.1}$$

This can be simplified as follows:

$$h(t) = \lim_{dt \to 0} \frac{P(T \leq t + dt) - P(T \leq t)}{S(t)dt}$$

$$h(t) = \lim_{dt \to 0} \left(\frac{F(t + dt) - F(t)}{dt} \right) \frac{1}{S(t)}$$

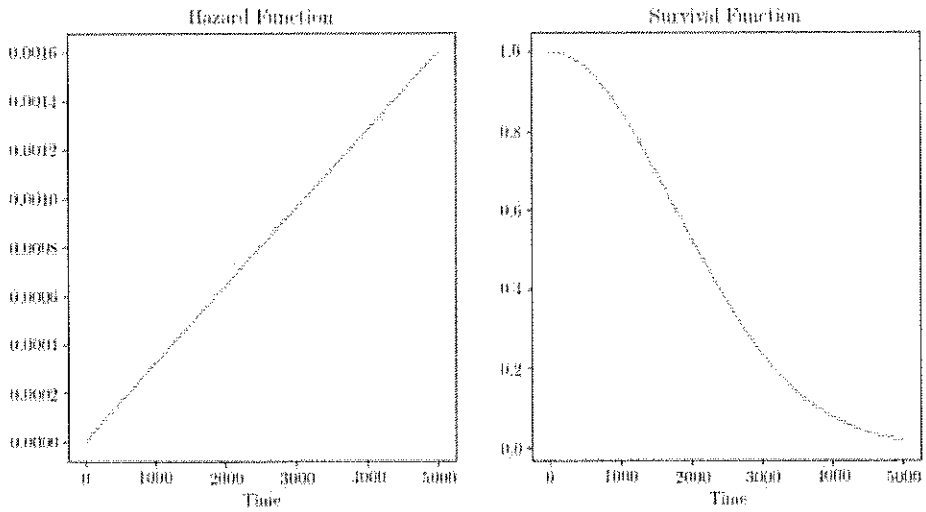
$$h(t) = \frac{dF(t)}{dt} \frac{1}{S(t)}$$

$$h(t) = \frac{f(t)}{S(t)}$$

In other word, the rate of occurrence of the event at duration t equals the density of

events at t divided by the probability of surviving to that duration without experiencing the event.

This figure illustrates behavior of survival and hazard functions



This graph illustrates the behavior of the hazard function which is defined as the instantaneous risk that the vent of interest will happen, and as shown it is inversely related to survival function, which means that when the hazard risk increases, the survival probability goes down.

1.3 Estimation of Survival Function

There are different ways to estimate survival function, including parametric and non parametric methods, and in this section we will discuss both approaches[17,8].

1.3.1 Parametric Approach

There are numbers of popular parametric methods that can be employed to model survival data, and they differ in terms of the assumptions that are made about the distribution of survival times in the population. However, some common distributions can be used to represent the survival function such as:

(1) Normal distribution.

(2) Uniform distribution.

(3) Exponential distribution.

(4) Weibull distribution.

In this thesis, the focus will be on the most two common functions that can be empolyed to represent survival function which they are exponential and wiebull distribution.

1.3.2 Nonparametric Approach

The non parametric survival models describes the specific techniques of survival analysis that covers data being distribution free, or uses distributions but without a specific parameters. These models don't assume that structure of a model to be fixed, in contrast, the model can grow in size to accommodate the complexity of data the non parametric model is needed when data can be ranked, but without numerical interpretation. some common estimators can be employed to estimate $S(t)$ based on the censoring model, such as Kaplan Meier in case if right and Turnbull in case of interva[4,5].

In most applications, the data may be interval-censored. By interval-censored data, we mean that a random variable of interest is known only to lie in an interval, instead of being observed exactly. In such cases, the only information we have for each individual is that their

event time falls in an interval, but the exact time is unknown. A nonparametric estimate of the survival function can also be found in such intervalcensored situations. The survival function is perhaps the most important function in medical and health studies. In this work we describe and illustrate the iterative procedure proposed by Turnbull (1976) to estimate such function.

Kaplan Meier model was named after Edward L. Kaplan and Paul Meier, who each submitted similar manuscripts to the Journal of the American Statistical Association(1958). Then they combined their work into one paper, named with both names. However, Kaplen-Meier estimator is used to estimate survival function from lifetime data. The advantage of this model is that it is very flexible, and model complexity grows with the number of observations. But it is not easy to incorporate covariates, meaning it is difficult to describe how individuals differ in their survival functions, because it uses all cases in series, not just those who were followed up till the selected cut-off. To explain this, when the surviving proportion is multiplied by the surviving proportions for each of the preceding time periods, a probability of surviving to the end of that time period is obtained; the survival probability is then plotted against time. With increasing follow up time, the curve is based on fewer and fewer cases, therefore becoming progressively less reliable. Also, it is not reliable when data are few, such as rare diseases.

Kaplan Meier Method

Kaplan Meier model, also known product limit estimator. It's a series of declining horizontal steps, which represent the true survival function of the whole population.
The kaplan Meier curve shows the probability of an event in a certain time interval. It is used to compare two groups in a study. An important advantage of Kaplen Meier model is that it takes into account some types of censored data, especially right censored data.
Kaplan Meier model is used widely in medical and fundamental research, for example it is

used to measure the fraction of people living a certain time after treatment. However it's limited in estimating survival function adjusted for covariates.

The estimator of the survival function $S(t)$ (the probability that life is longer than t is given by:

$$S(t) = \prod_{i:t_i \leq t} (1 - \frac{d_i}{n_i})$$

with t_i a time when at least one event happened, d_i the number of events that happened at time t_i, and n_i the individuals known to have survived (have not yet had an event or been censored) up to time t_i.

1.4 Censoring in survival data

To perform survival analysis, we need to record time to event, but this is not always possible, and some times we have only partial information about time to event, we speak of censoring, there are generally three reasons why censoring may occur[3]:

(1) A person does not experience the event before the study ends.

(2) A person is lost to follow-up during the study period.

(3) A person withdraws from the study because of some reason.

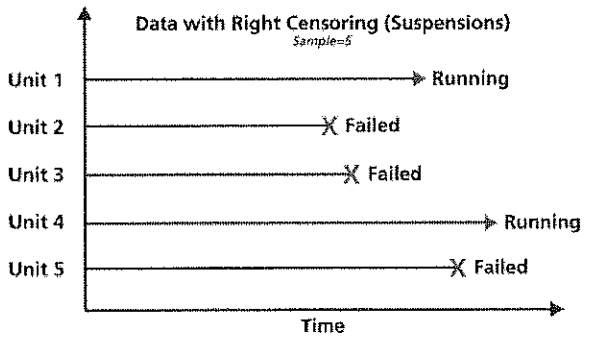
However, an attractive feature of survival analysis is that we are able to include the data

contributed by censored observations right up until they are removed from the risk set. This is an important issue, as Failure to take censoring into account can produce serious bias in estimates of the distribution of survival time and related quantities. Where random censoring is an inevitable feature of a study, it is important to include explanatory variables that are probably related to both censoring and survival time.

1.4.1 Right Censoring

This is the most common type. It occurs when subjects don't experience the event of interest during the study period, so we do know that they survived at lease specific time, but the exact survival time is not known. In other words the survival time will always be equal or greater than the observed time.

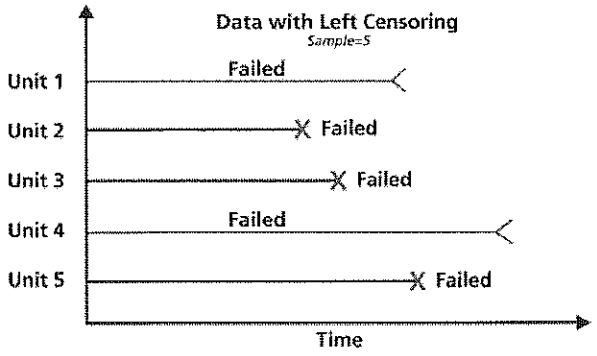
For example, if we test five units, and only three have failed by the end of specific observed period. Then the other 2 units will be right censored, as we don't know the exact time of their failure. This is illustrated in the following figure, as the first and fourth units passed the observed period without experiencing the event of interest, which is their failure.



1.4.2 Left Censoring

The left censoring describes subjects who had experienced the event before being enrolled in the study, or have lost follow up and the exact fate of them is not known. The figure shown below explains how the first and fourth subjects have failed before the observed time.

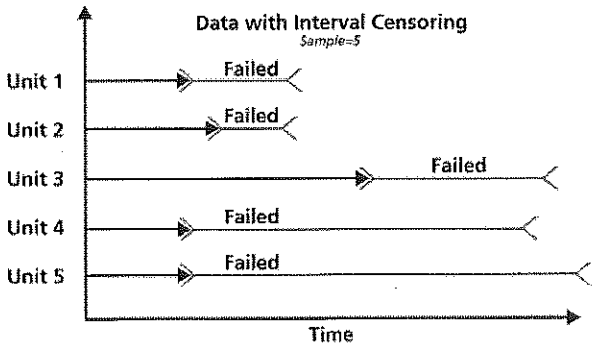
Another example of left censored data, when a physician includes regular clinic patients in a study, to observe the effect of a specific drug on their survival. Some of them may move out of the country, some of them may not be able to continue to follow up, these are specified as left censored data.



1.4.3 Interval Censoring

The interval censoring represents a sampling scheme when the event of interest is known to occur within an interval, instead of being observed exactly, it usually manifests during longitudinal studies, most commonly in medical or health issues, in which patients are followed periodically.

For example, if we are running a test on five units and inspecting them every 100 hours, we only know that a unit failed or did not fail between inspections. Specifically, if we inspect a certain unit at 100 hours and find it operating, and then perform another inspection at 200 hours to find that the unit is no longer operating, then the only information we have is that the unit failed at some point in the interval between 100 and 200 hours.



1.5 Objectives

In view of the importance of the survival function estimation, this thesis examines the efficiency of several methods that are commonly used to estimate survival function in the presence of different types of censored data. Particularly we employ the nonparametric and parametric techniques to estimate survival function under right and interval scenarios of censoring. In the parametric approach two commonly used distribution are considered; exponential and Weibull distributions. Results are compared with each other, shedding light on the most efficient technique for estimation the survival function.

1.6 Thesis structure

This thesis is organized as four chapters. Chapter one gives a brief background and the main objectives for this study. Then the parametric and nonparametric approaches of the survival function estimation will be introduced in this thesis. For right and interval censored data, a parametric estimation technique are introduced in chapter two based on the exponential and Weibull distributions when covariates can be involved or extracted from the analysis but in the simulation part the covariates are not considered as a result of complexity of the analysis. However, in real life scenarios the exact distribution of the data is usually unknown. In such cases, the nonparametric approach is the viable alternative, since the nonparametric estimator does not assume that the data come from a specified distribution. However, in chapter three the Kaplan-Meier estimator is employed to extract the probability of surviving in the case of right-censoring model. Similarly, Turnbull estimation procedure is also considered in the case of interval censoring scenario. In chapter four, a simulation study is conducted to verify the efficiency of the proposed techniques discussed in the theoretical part of the thesis.

Chapter 2

Parametric Estimation of Survival Function

2.1 The Likelihood Function

The Likelihood Function is a method used to estimate parameters of a probability distribution by maximizing a likelihood function. It is an example of point estimates, and answers the question: for which parameter value does the observed data have the biggest probability. The advantages of maximum likelihood estimation(MLE) are that it is often easy to compute, and is intuitive. It has been the main method of statistical inference. So by using MLE we increase the likelihood of our model to reach the true model[6].

Suppose that X is a random variable with probability density function $f(x;\theta)$, θ to be estimated and $x_1, x_2, x_3, ..., x_n$ are iid random sample of size n. The joint probability density function is given by

$$L(x_1, x_2, ..., x_n; \theta) = f(x_1; \theta).f(x_2; \theta)...f(x_n; \theta) = \prod_{i=1}^{n} f(x_i; \theta)$$

The value $\hat{\boldsymbol{\theta}}$ is considered as the maximum likelihood estimate of the parameter θ if

$$L(x_1, x_2, ..., x_n; \hat{\boldsymbol{\theta}}) \geq L(x_1, x_2, ..., x_n; \theta)$$

For all possible choices of θ .

$\hat{\boldsymbol{\theta}}$ will be a function of the sample values since $L(x_1, x_2, ..., x_n; \theta)$ is a function of parameter θ given the sample information $(x_1, x_2, ..., x_n)$

Let K_i be an indicator of censoring such that:

$$K = \begin{cases} 0 & censored \\ 1 & otherwise \end{cases}$$

Given k_i, the complete data are available, then the complete likelihood function is:

$$L_c = \prod_{i=1}^{n} [f(t_i)]^{k_i} . [S(t_i)]^{1-k_i}$$

To maximize L_c we need to maximize the log-likelihood function defined as follow:

$$\begin{aligned} l_c &= \log \prod_{i=1}^{n} [f(t_i)]^{k_i} . [S(t_i)]^{1-k_i} \\ &= \sum_{i=1}^{n} \log[f(t_i)]^{k_i} + \sum_{i=1}^{n} \log[S(t_i)]^{1-k_i} \\ &= \sum_{i=1}^{n} k_i \log[f(t_i)] + \sum_{i=1}^{n} (1 - k_i) \log[S(t_i)] \end{aligned} \tag{2.1}$$

2.2 Maximum Likelihood for Right Censored Data

2.2.1 Expoential function with and without Covariates

In survival analysis, we need to choose the most suitable distribution, but its usually difficult, so we use parametric models, as they don't depend on the original distribution of data. In this section we will discuss the most common distributions that include weibull and exponential models.

Exponential Function without covariates

Exponential function models the behaviour of units that have constant failure rate, for example: the time of failure of electronic components. This leads to memory less characteristic of the exponential distribution. That means the probability of failure next time doesn't change with the age of the person. This model is invested to predict the waiting time for the next event, for example:

(1)The amount of time till the hardware fails.
(2)The amount of time you need till the bus arrives.
(3)The amount of time till an earthquake occurs.

The probability density function (pdf) of an exponential distribution is:

$$f(t,\lambda) = \begin{cases} \lambda e^{-\lambda t} & t \geq 0 \\ 0 & t < 0 \end{cases} \tag{2.2}$$

and the cumulative distribution function is given by:

$$F(t,\lambda) = \begin{cases} 1 - e^{-\lambda t} & t \geq 0 \\ 0 & t < 0 \end{cases} \tag{2.3}$$

and hence the survival function can be defined as follows:

$$S(t) = \begin{cases} e^{-\lambda t_i} & t \geq 0 \\ 0 & t < 0 \end{cases} \tag{2.4}$$

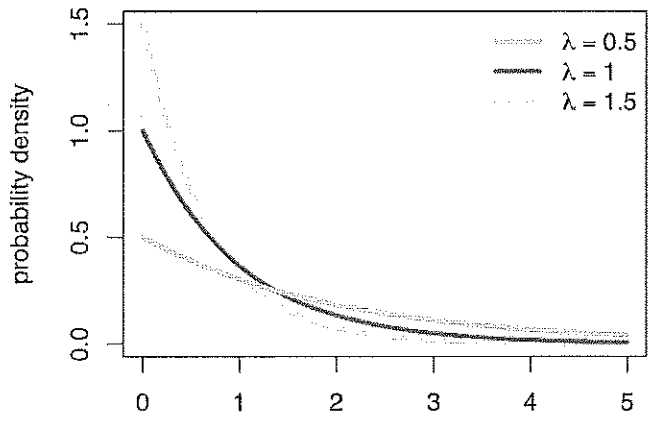


Figure 2.1: Probability density function for exponential

In parametric maximum likelihood method the cumulative distribution function $F(.)$ and the probability density function $f(.)$ for the entire population are known.

In parametric case we will use the exponential function for $S(t)$ and $f(t)$, as shown above in (2.2) and (2.4).

We will use the likelihood function, that is defined in equation (2.1)

$$\begin{aligned} l_c &= \sum_{i=1}^{n} k_i \log[f(t_i)] + \sum_{i=1}^{n} (1-k_i) \log[S(t_i)] \\ &= \sum_{i=1}^{n} k_i \log[\lambda e^{-\lambda t_i}] + \sum_{i=1}^{n} (1-k_i) \log[e^{-\lambda t_i}] \\ &= \sum_{i=1}^{n} k_i [\log(\lambda) + \log(e^{-\lambda t_i})] + \sum_{i=1}^{n} (1-k_i) \log[e^{-\lambda t_i}] \\ &= \sum_{i=1}^{n} k_i [\log(\lambda) + -(\lambda t_i)] + \sum_{i=1}^{n} (1-k_i)(-\lambda t_i)] \\ &= \sum_{i=1}^{n} k_i \log(\lambda) - \sum_{i=1}^{n} k_i \lambda t_i - \sum_{i=1}^{n} \lambda t_i + \sum_{i=1}^{n} k_i \lambda t_i \\ &= \sum_{i=1}^{n} k_i \log(\lambda) - \sum_{i=1}^{n} (\lambda t_i) \end{aligned} \tag{2.5}$$

We will derive this equation with respect to λ such that:

$$\frac{\partial l_c}{\partial \lambda} = \frac{1}{\lambda} \sum_{i=1}^{n} k_i - \sum_{i=1}^{n} t_i = 0$$

From this equation the desired estimate of the parameter λ can be defined as follows:

$$\widehat{\lambda} = \frac{\sum_{i=1}^{n} K_i}{\sum_{i=1}^{n} t_i}$$

Exponential function with Covariates

Covariates are characteristics of participants in a study, and they affect the outcome in a study. A covariate can be independent variable, or unwanted which called the confounding. In survival analysis it allows us to study how specific factors may influence the rate of a particular event at particular point of time. Examples of different covariates include age, gender, medical conditions, the type of treatment of medication given. these specifications help determining how different factors could affect the outcome[9].

Since the covariates involved in the analysis then the shape parameter λ given by :
$\lambda = \exp(B^l Z) = e^{B_1 Z_1 + B_2 Z_2}$ where Z_1, Z_2 are the covariates vector and B_1, B_2 are the coefficients vector.

We will use (2.5), and Subsitute $\lambda = e^{(B_1 Z_1 + B_2 Z_2)}$

$$\begin{aligned} l_c &= \sum_{i=1}^{n} k_i \log(\lambda) - \sum_{i=1}^{n} (\lambda t_i) \\ &= \sum_{i=1}^{n} k_i \log e^{B_1 Z_1 + B_2 Z_2} - \sum_{i=1}^{n} e^{B_1 Z_1 + B_2 Z_2} t_i \\ &= \sum_{i=1}^{n} k_i (B_1 Z_1 + B_2 Z_2) - \sum_{i=1}^{n} e^{B_1 Z_1 + B_2 Z_2} t_i \end{aligned} \tag{2.6}$$

We will derive this equation with respect to B_1 and B_2 such that:

$$\frac{\partial l_c}{\partial B_1} = \sum_{i=1}^{n} K_i Z_{i1} - \sum_{i=1}^{n} Z_{i1} e^{(B_1 Z_1 + B_2 Z_2)} t_i = 0 \tag{2.7}$$

$$\frac{\partial l_c}{\partial B_2} = \sum_{i=1}^{n} K_i Z_{i2} - \sum_{i=1}^{n} Z_{i2} e^{(} B_1 Z_1 + B_2 Z_2) t_i = 0 \tag{2.8}$$

The desired estimates of B_1 and B_2 can be obtained using any numerical techniqe such as Newton-Raphsom method (Appendix A1).

2.2.2 Weibull function with and without Covariates

Weibull distribution is one of the most commonly used lifetime distributions, it was named after Swedish engineer and scientist Ernst Hjalmar weibull, who popularized the distribution in his 1951 to the American society of Mechanical Engineers.

This distribution tests the mean time between failures, and also when the hazardous rate is not constant, that is increasing or decreasing over time, it is particularly well

suited to time series data with heavy tails, Where valued far from the maximum probability are still fairly common. It is described by three parameters:

(1) shape γ .

(2) scale λ.

(3) threshold parameters.

The most general expression of the Weibull pdf is given by the three-parameter Weibull distribution expression:[3]

$$f(t) = \lambda\gamma(t_i)^{\gamma-1}e^{-\lambda t_i^{\gamma}} \tag{2.9}$$

$$F(t) = 1 - e^{-\lambda t_i^{\gamma}} \tag{2.10}$$

$$S(t) = e^{-\lambda t_i^{\gamma}} \tag{2.11}$$

Where $f(t) > 0$ and $t > 0$

Weibull Function without Covariates

In parametric case we will use the Weibull function for $S(t)$ and $f(t)$ as we defined in (2.8), (2.10) and hence the loglikelihood function l_c can be written as follows:

$$\begin{aligned}
l_c &= \sum_{i=1}^{n} k_i \log[f(t_i)] + \sum_{i=1}^{n}(1-k_i)\log[S(t_i)] \\
&= \sum_{i=1}^{n} k_i \log[\lambda\gamma(t_i)^{\gamma-1}e^{-\lambda t_i^{\gamma}}] + \sum_{i=1}^{n}(1-k_i)\log[e^{-\lambda t_i^{\gamma}}] \\
&= \sum_{i=1}^{n} k_i[\log(\lambda)+\log(\gamma)+\log(t_i)^{\gamma-1}+\log e^{-\lambda t_i^{\gamma}}] + \sum_{i=1}^{n}(1-k_i)[-\lambda t_i^{\gamma}] \\
&= \sum_{i=1}^{n} k_i[\log(\lambda)+\log(\gamma)+(\gamma-1)\log(t_i)+-\lambda t_i^{\gamma}] + \sum_{i=1}^{n}(1-k_i)[-\lambda t_i^{\gamma}] \\
&= \sum_{i=1}^{n}[k_i\log(\lambda)+k_i\log(\gamma)+k_i(\gamma-1)\log(t_i)-k_i\lambda t_i^{\gamma}-\lambda t_i^{\gamma}+k_i\lambda t_i^{\gamma}] \\
&= \sum_{i=1}^{n}[k_i\log(\lambda)+k_i\log(\gamma)+k_i(\gamma-1)\log(t_i)-\lambda t_i^{\gamma}] \qquad (2.12)
\end{aligned}$$

We will derive this equation with respect to λ and γ such that:

$$\frac{\partial l_c}{\partial \lambda} = \frac{\partial l_c}{\partial \gamma} = 0$$

$$\begin{aligned}
\frac{\partial l_c}{\partial \lambda} &= \sum_{i=1}^{n}[k_i\frac{1}{\lambda}+0+0+t_i^{\gamma}] = 0 \\
&= \sum_{i=1}^{n}[k_i\frac{1}{\lambda}-t_i^{\gamma}] = 0 \\
&\sum_{i=1}^{n} k_i\frac{1}{\lambda} = \sum_{i=1}^{n} t_i^{\gamma}
\end{aligned}$$

$$\lambda = \frac{\sum_{i=1}^{n} K_i}{\sum_{i=1}^{n} {t_i}^{\gamma}} \qquad (2.13)$$

$$\begin{aligned}\frac{\partial l_c}{\partial \gamma} &= \sum_{i=1}^{n}[0 + k_i\frac{1}{\gamma} + k_i \log t_i - (\gamma - 1)\lambda {t_i}^{\gamma} \log t_i] = 0 \\ &= \sum_{i=1}^{n}[k_i\frac{1}{\gamma} + k_i \log t_i - (\gamma - 1)\lambda {t_i}^{\gamma} \log t_i] = 0 \end{aligned} \tag{2.14}$$

The desired estimate of the parameter can be obtained by solving equation (2.14) using any numerical method.

Weibull function with Covariate

The estimation of survival function will be introduced based weibull distribution in case that some measured covariates are involved in the analysis.

We use (2.11) and subsitute for $\lambda = e^{B_1 Z_1 + B_2 Z_2}$

$$\begin{aligned} l_c &= \sum_{i=1}^{n}[k_i \log(\lambda) + k_i \log(\gamma) + k_i(\gamma - 1)\log(t_i) - \lambda t_i^{\gamma}] \\ &= \sum_{i=1}^{n}[k_i \log e^{B_1 Z_1 + B_2 Z_2} + k_i \log(\gamma) + k_i(\gamma - 1)\log(t_i) - e^{B_1 Z_1 + B_2 Z_2} t_i^{\gamma}] \\ &= \sum_{i=1}^{n}[k_i(B_1 Z_1 + B_2 Z_2) + k_i \log(\gamma) + k_i(\gamma - 1)\log(t_i) - e^{B_1 Z_1 + B_2 Z_2} t_i^{\gamma}] \end{aligned}$$

We will derive this equation with respect to B_1 and B_2 such that:

$$\frac{\partial l_c}{\partial B_1} = \frac{\partial l_c}{\partial B_2} = 0$$

$$\begin{aligned}\frac{\partial l_c}{\partial B_1} &= \sum_{i=1}^{n} K_i Z_{i1} + 0 + 0 + Z_{i1} e^{B_1 Z_1 + B_2 Z_2} t_i^{\gamma} = 0 \\ &= \sum_{i=1}^{n} K_i Z_{i1} + Z_{i1} e^{B_1 Z_1 + B_2 Z_2} t_i^{\gamma} \end{aligned} \tag{2.15}$$

$$\begin{aligned}\frac{\partial l_c}{\partial B_2} &= \sum_{i=1}^{n} K_i Z_{i2} + 0 + 0 + Z_{i2} e^{B_1 Z_1 + B_2 Z_2} t_i^{\gamma} = 0 \\ &= \sum_{k=1}^{n} K_i Z_{i2} + Z_{i2} e^{B_1 Z_1 + B_2 Z_2} t_i^{\gamma} \end{aligned} \tag{2.16}$$

So, find B_1 and B_2 using Newton method.

2.3 Maximum Likelihood for Interval Censored Data

In this section the exponential distribution and wiebull distribution will be employed to represent the distributional function of the data set, where the exact failure time is not observed, but it's known that it lies within the observed interval.

2.3.1 Exponential function with and without covariates

Exponential function without covariates

In case of interval censored data

$$\begin{aligned} S(t_i) &= F(R_i) - F(L_i) \\ &= S(L_i) - S(R_i) \\ &= e^{-\lambda L_i} - e^{-\lambda R_i} \end{aligned} \tag{2.17}$$

The lifetime of uncensored individuals (t_i) is not specified just it is well known that it belongs to an observed interval. So we will use the Mid-point of the interval as an estimation of t_i.

The log-likelihood function defined in equation (2.1) can be written as follows:

$$\begin{aligned} l_c &= \sum_{i=1}^{n} k_i \log[f(t_i)] + \sum_{i=1}^{n} (1-k_i) \log[S(L_i) - S(R_i)] \\ &= \sum_{i=1}^{n} k_i \log[\lambda e^{-\lambda t_i}] + \sum_{i=1}^{n} (1-k_i) \log[e^{-\lambda L_i} - e^{-\lambda R_i}] \\ &= \sum_{i=1}^{n} k_i \log(\lambda) - \sum_{i=1}^{n} k_i \lambda t_i + \sum_{i=1}^{n} (1-k_i) \log[e^{-\lambda L_i} - e^{-\lambda R_i}] \end{aligned} \tag{2.18}$$

The solution of $\frac{\partial l_c}{\partial \lambda} = 0$ is the disired estimates of λ

$$\frac{\partial l_c}{\partial \lambda} = \frac{1}{\lambda}\sum_{i=1}^{n} k_i - \sum_{i=1}^{n} k_i t_i + \sum_{i=1}^{n}(1-k_i)\frac{R_i e^{-\lambda R_i} - L_i e^{-\lambda L_i}}{e_e^{-\lambda L_i - \lambda R_i}} = 0 \tag{2.19}$$

Exponential function with covariates

When the covariates involved in the analysis then the shape parameter λ can be obtained by:

$$\lambda = \exp(B^t Z) = e^{B_1 Z_1 + B_2 Z_2} = M$$

where Z_1, Z_2 are the covariates vector and B_1, B_2 are the coefficients vector.

We use (2.17) and subsitute for λ:

$$\begin{aligned} l_c &= \sum_{i=1}^{n} k_i \log(M) - \sum_{i=1}^{n} k_i M t_i + \sum_{i=1}^{n}(1-k_i)\log[e^{-ML_i} - e^{-MR_i}] \\ &= \sum_{i=1}^{n} k_i[(B_1 Z_1 + B_2 Z_2) - \sum_{i=1}^{n} k_i M t_i] + \sum_{i=1}^{n}(1-k_i)\log[e^{-ML_i} - e^{-MR_i}] \end{aligned}$$

The solution of $\frac{\partial l_c}{\partial B_1} = \frac{\partial l_c}{\partial B_2} = 0$ are the disired estimates of B_1 and B_2

$$\begin{aligned} \frac{\partial l_c}{\partial B_1} &= \sum_{i=1}^{n} k_i[(Z_{1i}) - \sum_{i=1}^{n} Z_{1i} k_i M t_i] \\ &\quad + \sum_{i=1}^{n}(1-k_i)\frac{A_1 - B_1}{e^{-ML_i} - e^{-MR_i}} \end{aligned} \tag{2.20}$$

Where :

$$A_1 = -(Z_{1i})(L_i)(e^{B_1 Z_1 + B_2 Z_2})(e^{L_i e^{B_1 Z_1 + B_2 Z_2}})$$
$$B_1 = -(Z_{1i})(R_i)(e^{B_1 Z_1 + B_2 Z_2})(e^{R_i e^{B_1 Z_1 + B_2 Z_2}})$$

$$\begin{aligned}\frac{\partial l_c}{\partial B_2} &= \sum_{i=1}^{n} k_i[(Z_{2i}) - \sum_{i=1}^{n} Z2ik_i e^{B_1Z_1+B_2Z_2}t_i \\ &+ \sum_{i=1}^{n}(1-k_i)\frac{A_2 - B_2}{e^{-ML_i} - e^{-MR_i}} \end{aligned} \tag{2.21}$$

Where

$$A_2 = -(Z_{2i})(L_i)(e^{B_1Z_1+B_2Z_2})(e^{L_ie^{B_1Z_1+B_2Z_2}})$$
$$B_2 = -(Z_{2i})(R_i)(e^{B_1Z_1+B_2Z_2})(e^{R_ie^{B_1Z_1+B_2Z_2}})$$

2.3.2 Weibull function with and without covariates

Weibull function without Covariates

The log-likelihood function defined in equation (2.1) can be written as follows:

$$\begin{aligned} l_c &= \log\prod_{i=1}^{n}[f(t_i)]^{k_i}.[S(L_i) - S(R_i)]^{1-k_i} \\ &= \sum_{i=1}^{n}\log[f(t_i)]^{k_i} + \sum_{i=1}^{n}\log[S(L_i) - S(R_i)]^{1-k_i} \\ &= \sum_{i=1}^{n} k_i \log[f(t_i)] + \sum_{i=1}^{n}(1-k_i)\log[S(L_i) - S(R_i)] \end{aligned} \tag{2.22}$$

this can be simplified as follows:

$$\begin{aligned} &= \sum_{i=1}^{n} k_i \log[\lambda\gamma(t_i)^{\gamma-1}e^{-\lambda t_i^{\gamma}}] + \sum_{i=1}^{n}(1-k_i)\log[e^{-\lambda L_i^{\gamma}} - e^{-\lambda R_i^{\gamma}}] \\ &= \sum_{i=1}^{n}[k_i\log(\lambda) + k_i\log(\gamma) + k_i(\gamma-1)\log(t_i) - k_i\lambda t_i^{\gamma}] \\ &\quad + \sum_{i=1}^{n}(1-k_i)\log[e^{-\lambda L_i^{\gamma}} - e^{-\lambda R_i^{\gamma}}] \end{aligned} \tag{2.23}$$

We will derive this equation with respect to λ and γ such that:

$$\frac{\partial l_c}{\partial \lambda} = \frac{\partial l_c}{\partial \gamma} = 0$$

$$\begin{aligned}\frac{\partial l_c}{\partial \lambda} &= \sum_{i=1}^{n} k_i \frac{1}{\lambda} - \sum_{i=1}^{n} k_i t_i^{\gamma} \\ &\quad + \sum_{i=1}^{n} (1-k_i) \log \frac{R_i^{\gamma} e^{-\lambda R_i^{\gamma}} - L_i^{\gamma} e^{-\lambda L_i^{\gamma}}}{e^{-\lambda R_i^{\gamma}} - e^{-\lambda L_i^{\gamma}}}\end{aligned} \tag{2.24}$$

$$\begin{aligned}\frac{\partial l_c}{\partial \gamma} &= \frac{1}{\gamma} \sum_{i=1}^{n} k_i + \sum_{i=1}^{n} k_i \log(t_i) - \lambda\gamma \sum_{i=1}^{n} k_i t_i^{\gamma-1} \\ &\quad + \lambda\gamma \sum_{i=1}^{n} (1-k_i) \log \frac{R_i^{\gamma-1} e^{-\lambda R_i^{\gamma}} - L_i^{\gamma-1} e^{-\lambda L_i^{\gamma}}}{e^{-\lambda R_i^{\gamma}} - e^{-\lambda L_i^{\gamma}}}\end{aligned} \tag{2.25}$$

Weibull Function with covariates

We use (2.17) and Subsitute for $\lambda = e^{(B_1 Z_1 + B_2 Z_2)}$

$$\begin{aligned}l_c &= \sum_{i=1}^{n} [k_i \log(\lambda) + k_i \log(\gamma) + k_i(\gamma-1)\log(t_i) - k_i \lambda t_i^{\gamma}] \\ &\quad + \sum_{i=1}^{n} (1-k_i) \log[e^{-\lambda L_i^{\gamma}} - e^{-\lambda R_i^{\gamma}}] \\ &= \sum_{i=1}^{n} [k_i \log(e^{(B_1 Z_1 + B_2 Z_2)}) + k_i \log(\gamma) + k_i(\gamma-1)\log(t_i) - k_i e^{(B_1 Z_1 + B_2 Z_2)} t_i^{\gamma}] \\ &\quad + \sum_{i=1}^{n} (1-k_i) \log[e^{-e^{(B_1 Z_1 + B_2 Z_2)} L_i^{\gamma}} - e^{-e^{(B_1 Z_1 + B_2 Z_2)} R_i^{\gamma}}] \\ &= \sum_{i=1}^{n} [k_i (B_1 Z_1 + B_2 Z_2) + k_i \log(\gamma) + k_i(\gamma-1)\log(t_i) - k_i e^{(B_1 Z_1 + B_2 Z_2)} t_i^{\gamma}] \\ &\quad + \sum_{i=1}^{n} (1-k_i) \log[e^{-e^{(B_1 Z_1 + B_2 Z_2)} L_i^{\gamma}} - e^{-e^{(B_1 Z_1 + B_2 Z_2)} R_i^{\gamma}}]\end{aligned} \tag{2.26}$$

The solutions of the equations $\frac{\partial l_c}{\partial B_1} = \frac{\partial l_c}{\partial B_2} = 0$ are the desired estimates of B_1 and B_2, such that:

Let M $= e^{B_1 Z_1 + B_2 Z_2}$

$$
\begin{aligned}
= & \sum_{i=1}^{n} [k_i(B_1 Z_1 + B_2 Z_2) + k_i \log(\gamma) + k_i(\gamma - 1)\log(t_i) - k_i M t_i^{\gamma}] \\
& + \sum_{i=1}^{n} (1 - k_i) \log[e^{-ML_i^{\gamma}} - e^{-MR_i^{\gamma}}]
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial l_c}{\partial B_1} = & \sum_{i=1}^{n} [k_i Z_{1i} - k_i Z_{1i} M (t_i)^{\gamma} \\
& - (1 - k_i) \log \frac{R_i^{\gamma} Z_{1i} M e^{-MR_i^{\gamma}} - L_i^{\gamma} Z_{1i} M e^{-ML_i^{\gamma}}}{e^{-MR_i^{\gamma}} - e^{-ML_i^{\gamma}}}] = 0
\end{aligned} \tag{2.27}
$$

$$
\begin{aligned}
\frac{\partial l_c}{\partial B_2} = & \sum_{i=1}^{n} [k_i Z_{2i} - k_i Z_{2i} M (t_i)^{\gamma} \\
& - (1 - k_i) \log \frac{R_i^{\gamma} Z_{2i} M e^{-MR_i^{\gamma}} - L_i^{\gamma} Z_{2i} M e^{-ML_i^{\gamma}}}{e^{-MR_i^{\gamma}} - e^{-ML_i^{\gamma}}}] = 0
\end{aligned} \tag{2.28}
$$

Chapter 3

Nonparametric Estimation of Survival Function

Non parametric statistics is the branch of research that does not depend on parametrized probability distributions. It is based mainly on being distribution free, or unspecified distribution parameters. It is being applied to study population that take a ranked order, or numerical distributions, such as studying preferences[8,16].

The wide familiarity of this analysis models is due to the fact that less assumptions are needed, the ease of usage and simplicity.

In this chapter we will discuss non-parametric models with kaplan Meier model and their preferred uses, we will start with Kaplan Meier model, which is used to compare the survival distribution of 2 or more groups, and it can be used for right censored data.

However, for interval censored data the Trumbull model should be used, this will be discussed in further details in this chapter.

3.1 Estimation of survival Function using Right Censoring

3.1.1 The Kaplan Meier Estimator for Survival Function

The greate advantage of Kaplan Meier (KM) estimator is that it is computable for right censored data[12,14].

Let $T_{(1)}, T_{(2)}, ..., T_{(n)}$ are iid survival times with survival function $S(t)$, with $C_1, C_2, ..., C_n$ the censoring times, iid and independent of the T_i, and suppose that our observations are denoted by (U_i, K_i) for $i = 1, 2, ..., n$ such that:

$$U_i = T_i \wedge C_i$$

Suppose that $F(t)$ is discrete with mass points at $0 \leq v_1 < v_2 < ...$ and define the discrete hazard functions [2]:

$$h_1 = P[T = v_1]$$

$$h_2 = P[T = v_2 | T > V_1]$$

and

$$h_j = P[T = v_j | T > v_{j-1}].$$

However, the survival function $S(t)$ can be defined as follows:

$$\begin{aligned} S(t) &= P(T > t) = P(T > v_j) \\ &= P(T > v_j | T > v_{j-1}) P(T > v_{j-1}) \\ &= P(T > v_j | T > v_{j-1}) P(T > v_{j-1} | T > v_{j-2}) P(T > v_{j-2}) \\ &= (1 - h_j)(1 - h_{j-1})...(1 - h_1) = \prod_{i=1}^{j} (1 - h_i) \end{aligned} \tag{3.1}$$

Similarly, define $f_1 = h_1$ and, for $j > 1$,

$$f_j = P(T = v_j) = h_j \prod_{i=1}^{j-1} (1 - h_i)$$

Now making an inference about $F(t)$ based on the likelihood function corresponding to (U_i, k_i) for $i = 1, 2, ..., n$. The likelihood function can be defined as follows:

$$L(F) = \prod_{i=1}^{n} [f(U_i)]^{k_i} . [1 - F(U_i)]^{1-k_i}$$

Substituting the h_j, this becomes

$$L(F) = \prod_{j} h_j^{d_j} (1 - h_j)^{y(v_j) - d_j} \tag{3.2}$$

where $0 \leq h_j \leq 1$,

$d_j = \sum_{i=1}^{n} k_i . 1_{U_i = v_j}$ = who fail at v_j,

$y(v_j) = \sum_{i=1}^{n} 1_{U_i > v_j}$ = who risk at v_j

Maximization of the likelihood function with respect to h will produse the desired estimate of h_j as follows:

$$h_j = \frac{d_j}{y(v_j)}$$

So that:

$$S(t) = \begin{cases} 1 & t < v_j \\ \prod_{i=1}^{j}(1-h_i) & v_j \leq t \leq v_{j-1} \end{cases}$$

Notice that the expression for h_j makes sense: the probability of dying at v_j given you are alive before is estimated by $d_j/Y(v_j)$. Also the expression for $S(t)$ makes sense: the probability of staying alive at v_j if alive before v_j is estimated by

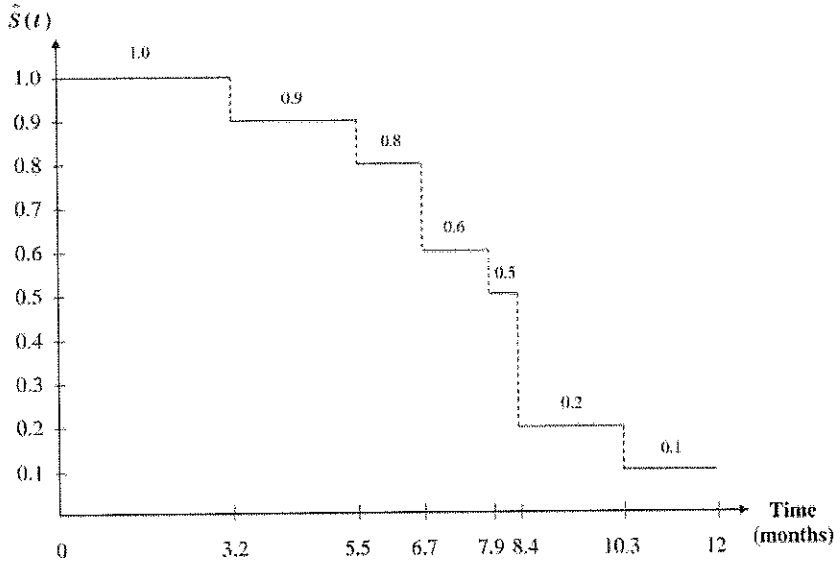
$$\left(1 - \frac{d_j}{y(v_j)}\right).$$

Example 3.1. *Twelve-month cohort study of* $n = 10$ *patients.*

Patient	t_i (months)
1	3.2
2	5.5
3	6.7
4	6.7
5	7.9
6	8.4
7	8.4
8	8.4
9	10.3
10	alive

$\rightarrow$

Interval $[t_i, t_{i+1})$	n_i = # patients at risk at time t_i	d_i = # of deaths at time t_i	$1 - \frac{d_i}{n_i}$	$\hat{S}(t)$
[0,3.2)	10	0	1.00	1.0
[3.2,5.5)	10-0=10	1	0.90	0.9
[5.5,6.7)	10-1=9	1	0.89	0.8
[6.7,7.9)	9-1=8	2	0.75	0.6
[7.9,8.4)	8-2=6	1	0.83	0.5
[8.4,10.3)	6-1=5	3	0.40	0.2
[10.3,12)	5-3=2	1	0.50	0.1
Study Ends	2-1=1	0	1.00	0.1



Variance of the Kaplan-Meier estimator

Delta method :

In order to estimate the variance of the Kaplan-Meier estimator, we need to introduce the delta method. Which is method uses the first order Taylor expansion of a function f of a random variable X around $\mu = E(X)$ to approximate the variance of $f(X)$ [11]:

$$\begin{aligned} f(X) &\simeq f(\mu) + f'(\mu)(X - \mu) \\ VAR(f(X)) &\simeq VAR[f(\mu) + f'(\mu)(X - \mu)] \\ &= f'(\mu)^2 VAR(X - \mu) \\ &= f'(\mu)^2 VAR(X) \\ &= f'(\mu)^2 \sigma^2 \end{aligned} \tag{3.3}$$

where σ^2= VAR(x). The delta method estimator is:

$$\widehat{VAR}(f(X)) = f'^2(\widehat{\mu})\widehat{\sigma}^2$$

where $\widehat{\sigma}$ is an estimator of VAR(X) and $\widehat{\mu}$ is an estimator of $E(X)$

The estimate of the variance is given by Greenwood's formula:

$$\widehat{VAR}(S(t)) = S^2(t) \sum_{t_i < t} \frac{d_i}{n_i(n_i - d_i)}$$

Here we show how to find the Greenwood's formula using the delta method. We need to use the delta method two times:

$$\log(X) \simeq \log(\mu) + (X-\mu)\frac{1}{\mu} \Longrightarrow \widehat{VAR}(\log(X)) \simeq \widehat{\sigma}^2 \frac{1}{\widehat{\mu}^2}$$

and

$$\exp(X) \simeq \exp(\mu) + (X-\mu)\exp(X) \Longrightarrow \widehat{VAR}(\exp(X)) \simeq \exp^2(\widehat{\mu})\widehat{\sigma}^2$$

First we look at $\log \widehat{S}(t)$:

$$\log \widehat{S}_{KM}(t) = \log \prod_{i:t_i \leq t} \left(1 - \frac{d_i}{n_i}\right) = \sum_{i:t_i \leq t} \log\left(1 - \frac{d_i}{n_i}\right)$$

Let $P_i = P(T > t_j | T > t_{i-1})$ then $\widehat{P}_i = \left(1 - \frac{d_i}{n_i}\right)$ is an estimate of this conditional probability. That means we assume that $d_i \sim B(n, 1-P_i)$ Hence, the variance of $\widehat{P}_i$is estimated by $\frac{\widehat{P_i}(1-\widehat{P_i})}{n_i}$.

Moreover, the Binomial variables are independent for all subjects in the study. We have then:

$$\widehat{VAR}\left(\sum_{i:t_i \leq t} \log(\widehat{P}_i)\right) = \sum_{i:t_i \leq t} \widehat{VAR}(\log(\widehat{P}_i)) \tag{3.4}$$

A first use of the delta method gives:

$$\begin{aligned}\widehat{VAR}(\log(\widehat{P}_i)) &\simeq \frac{P_i(1-P_i)}{n}\frac{1}{\widehat{P}_i^2} = \frac{1-(1-\frac{d_i}{n_i})}{n_i(1-\frac{d_i}{n_i})} = \frac{\frac{d_i}{n_i}}{n_i - d_i} = \frac{d_i}{n_i(n_i-d_i)} \\ &\Longrightarrow \log(\widehat{VAR}(\widehat{S}(t)) \simeq \sum_{i:t_i \leq t} \frac{d_i}{n_i(n_i-d_i)}\end{aligned}$$

We use the delta method for the second time and finally find:

$$\begin{aligned}
\widehat{VAR}(\widehat{S}(t) &= \widehat{VAR}\left(\exp[\log(\widehat{S}(t))]\right) \\
&= \exp^2[\log(\widehat{S}(t))] \sum_{i:t_i \leq t} \frac{d_i}{n_i(n_i - d_i)} \\
&= \widehat{S}^2(t)) \sum_{i:t_i \leq t} \frac{d_i}{n_i(n_i - d_i)}
\end{aligned}$$

This last formula had been given in 1926 by Greenwood before Kaplan and Meier published their estimator in 1958.

3.2 Estimation of survival Function using Interval Censoring

Interval censoring occurs when the event is known to occur between two points if time Turnbull models provide, easily verifiable estimation of the distribution function.

3.2.1 Turnbull Estimator of the Survival Function

In some studies the individuals may be followed within time period, so the only known is that interested events has occured within an interval. To solve this

Richard Turnbull formulated an algorithm to estimate the survival function for interval censored data that works on the principle of EM algorithm which also used to estimate the NPMLE for interval-censored data. For this algorithm, equivalence intervals must be defined to determine at what time points the cumulative distribution function F takes jumps $(S_1, S_2, ..., S_m)$ The equivalence intervals are formed by the intervals: $J_1 = (q_1, p_1], J_2 = (q_2, p_2], ..., J_m = (q_m, p_m]$. To find the equivalence intervals, consider all intervals $[L_i, R_i]$ for $i = 1, ..., n$, and order $2n$ endpoints in ascending order, and then each end point "L" that is immediately followed by the end point"R" is an equivalence intervals.

3.2.2 The EM algorithm

Most data analysis situations involve the estimation of unknown parameters from data that consist of observations that represent each distinct case in the study and variables that represent the characteristics that have been either measured or fixed by the experimenter. Each observation provides information for estimating the parameters corresponding to the variables used in the analysis.

A common task is the estimation of the parameters of a probability distribution function.

Perhaps the most frequently encountered estimation problem is the estimation of the mean of a signal in noise. In many parameter estimation problems the situation is more complicated because direct access to the data necessary to estimate the parameters is impossible, or some of the data are missing or unobserved. For example, in a histogram operation, there may also be data dropouts or clustering in such a way that the number of underlying data points is unknown (missing). The *Expectation-Maximization* (EM) algorithm is ideally suited to problems of this sort, in that it produces a maximum-likelihood (ML) estimation of parameters when there are many data were missing or unobserved.

Dempster, Laird and Rubin (1977) showed that how the expectation maximization (EM) algorithm could be used to obtain maximum likelihood estimates of parameters when the observations can be viewed as incomplete data or various levels of generally. They showed the monotone behavior of the likelihood and convergence of the algorithm.

This report represents my attempt at summarizing the EM algorithm (Dempster, Laird and Rubin, 1979). It includes a practical example to provide some intuition.

Description of EM Algorithm

The EM algorithm is a very general iterative algorithm for parameter estimation by maximum likelihood when some of the random variables involved are not observed i.e. considered missing or incomplete. The term EM was introduced in Dempster, Laird and Rubin (1977)[1] where proof of general results about the behavior of the algorithm was given.

For this summarizing report, suppose $x_1, \ldots, x_n$ denotes an observed random sample of size n on some random vector Y with probability density function $g(x_i; \theta)$, and the likelihood

function of Y is $f(Y;\theta)$ is indexed by a p-dimensional parameter $\theta \in \Theta \subseteq R^p$. Then

$$\begin{aligned} Y &= (x_1^T, \ldots, x_n^T)^T \\ f(Y;\theta) &= \prod_{i=1}^{n} g(x_i;\theta). \end{aligned}$$

If the *complete-data* vector Y were observed, it is of interest to compute the maximum likelihood estimate of θ based on the distribution of Y. The log-likelihood function $l(\theta;Y)$ of Y is:

$$l(\theta;Y) = \log f(Y;\theta).$$

Is then required to be maximized.

In the presence of missing data. However, only a function of the complete-data vector Y, is observed. We will denote this by expressing Y as (Y_{obs}, Y_{mis}), where Y_{obs} denotes the observed but "incomplete" data and Y_{mis} denotes the unobserved or "missing" data.

$$\begin{aligned} f(Y;\theta) &= f(Y_{obs}, Y_{mis};\theta) \\ &= f_1(Y_{obs};\theta) \cdot f_2(Y_{mis}|Y_{obs};\theta) \end{aligned}$$

Where f_1 is the joint density of Y_{obs} and f_2 is the joint density of Y_{mis} given the observed data Y_{obs}, respectively.

Thus it follows that

$$\begin{aligned} l(\theta;Y) &= \log f(Y;\theta) = \log\left[f_1(Y_{obs};\theta)\cdot f_2(Y_{mis}|Y_{obs};\theta)\right] \\ &= \log\left[f_1(Y_{obs};\theta)\right] + \log\left[f_2(Y_{mis}|Y_{obs};\theta)\right] \\ \log\left[f_1(Y_{obs};\theta)\right] &= l(\theta;Y) - \log\left[f_2(Y_{mis}|Y_{obs};\theta)\right] \\ l_{obs}(\theta : Y_{obs}) &= l(\theta;Y) - \log\left[f_2(Y_{mis}|Y_{obs};\theta)\right] \end{aligned}$$

Where $l_{obs}(\theta : Y_{obs})$ is the observed-data log-likelihood.

Since Y is not completely observed, $l(\theta;Y)$ cannot be evaluated and hence maximized. The EM algorithm attempts to maximize $l(\theta;Y)$ iteratively, by replacing it by its conditional expectation given the observed data Y_{obs}. This expectation is computed with respect to the distribution of the complete-data evaluated at the current estimate of θ. More specifically, if θ° is an initial value for θ, then on the first iteration it is required to compute

$$Q(\theta;\theta^{\circ}) = E_{\theta^{\circ}}\left[l(\theta;Y)|Y_{obs}\right].$$

$Q(\theta;\theta^{\circ})$ is now maximized with respect to θ, that is $\theta^{(1)}$ is found such that

$$Q(\theta^{(1)};\theta^{\circ}) \geq Q(\theta;\theta^{\circ}).$$

For all $\theta \in \Theta$. Thus the EM algorithm consists of an E-step (Expectation step) followed by an M-step (Maximization step) defined as follows:

- E-step: Compute $Q(\theta;\theta^{(t)})$ where

$$Q(\theta;\theta^{(t)}) = E_{\theta^{(t)}}\left[l(\theta;Y)|Y_{obs}\right].$$

- M-step: Find $\theta^{(t)}$ in Θ such that

$$Q(\theta^{(t+1)}; \theta^{t+1}) \geq Q(\theta; \theta^{t}).$$

for all $\theta \in \Theta$.

The E-step and the M-step are repeated alternately until

$$Q(\theta^{(t+1)}) - Q(\theta^{(t)}) \leq \alpha,$$

Where α is a prescribed small quantity.

3.2.3 The Algorithm of Turnbull Estimator

To estimate the jumps using the given data set, define an $n \times m$ matrix as α to indicate whether the event i can occur in the equivalent interval J or not such that:

$$\alpha_{ij} = \begin{cases} 1 & : if\ [q_i, p_i] \subseteq [L_i, R_i], \quad j = 1, \ldots, m, i = 1, \ldots, n, \\ 0 & : otherwise. \end{cases} \tag{3.5}$$

Then let I be a $n \times m$ to indicate if the event i occurs during the interval J such that:

$$I_{ij} = \begin{cases} 1: & if \quad T_i \in [q_i, p_i] j = 1, ..., m, i = 1, ..., n \\ 0: & \qquad\qquad Otherwise \end{cases}$$

Because of censored data the I_{ij} is not observed, so we work with its expection for known jumps s:

$$\mu_{ij} = E_s[I_{ij}] = \frac{\alpha_{ij} s_j}{\sum_{j=1}^{m} \alpha_{ij} s_j} \quad i = 1, ..., n, \quad j = 1, ..., m \tag{3.6}$$

For given μ_{ij}'s the proportion of data in the J interval is:

$$\pi_j(S) = \frac{\sum_{i=1}^{n} \mu_{ij}}{n} \quad for j = 1, ..., m \tag{3.7}$$

S is a self consistent estimate if $S_j = \pi_j(S)$, Davison(2006) proved that a self consistent estimate is an MLE.

To find μ_{ij} we can use an initialization of S such that$S^k = \left(\frac{1}{m}, ..., \frac{1}{m}\right)$,$k = 0$, then follow up to find $S_j^{k+1} = \pi_j(S^k)$ until stopping conditions such as $\sum_{j=1}^{m} \left(S_j^{k+1} - S_j^k\right)^2 < \epsilon$, $\epsilon > 0$.

The Turnbull estimator of the survival function is given in equation:

$$\hat{S}(T) = \begin{cases} 1 & : if\ T_i < q_1, \\ 1 - \sum_{k=1}^{j} s_k & : if\ p_j \leq T_i < q_{j+1}, \quad j = 1, \ldots, m, i = 1, \ldots, n, \\ 0 & : T_i \geq q_m. \end{cases} \tag{3.8}$$

Example 3.2. *Suppose we have 5 failure times that happen in a trial, the survival times in this virtual study are interval censored data pointes, the data set for* $n = 5$ *failures are supposed to be:*

$$[L_1, R_1] = [1, 2], [L_2, R_2] = [2, 5], [L_3, R_3] = [4, 7], [L_4, R_4] = [3, 8], [L_5, R_5] = [7, 9].$$

To find the equivalence classes, we need to find the ascending order of the $2n$ endpoints. Table shows the order endpoints considering all of the hypothetical intervals.

Initial Endpoints	Corresponding Labels	Ordered Endpoint	Corresponding Labels	Labels
1	L_1	1	L_1	L
2	R_1	2	L_2	L
2	L_2	2	R_1	R
5	R_2	3	L_4	L
4	L_3	4	L_3	L
7	R_3	5	R_2	R
3	L_4	7	L_5	L
8	R_4	7	R_3	R
7	L_5	8	R_4	R
9	R_5	9	R_5	R

Table 3.1: Construction of equivalent classes for the data set of interval censored data.

The first two columns in table are the raw data, and the next two columns contain the endpoints of the interval in ascending order and the corresponding labels. The last column denotes whether the order endpoints are left (L) or right (R) limits. Therefore, the equivalent classes in this hypothetical example are $[2, 2], [4, 5], [7, 7]$.

Chapter 4

Simulation and Results

In this chapter, simulation procedure and results are discussed. Where, simulations involving right and interval-censored data are introduced using R codes and procedures. The simulation in this chapter will be constructed without involving covariates in the data generation as a result of complexity of the analysis.

As stated in the previous chapters, the main goal of this thesis is to examine various techniques for estimation of the survival function. Therefore, in this chapter involving right-censored data, we have focused on estimating the survival function using both nonparametric and parametric estimates, where covariates are excluded from the analysis in both scenarios in this part of the thesis. In each simulation discussed, we have compared biases, relative efficiency (RE) to compare the efficiency of the estimators and also the Mean square error (MSE).

However, following reference [7], we consider the survival estimates associated with equally spaced survival times. Specifically, we have considered times when the true survival estimates are approximately 0.9, 0.75, 0.5, 0.35 and 0.2.

4.1 Design of Simulation for Right Censoring

In this part, we first discuss the data generation and how the bias, mean square error, and relative efficiency can be obtained.

4.1.1 Data Generation

The simulation in this study is constructed based on two commonly used distributions: exponential, and Weibull. Specifically the data generation from exponential distribution with shape parameter $\lambda = 5$ and a Weibull distribution with shape and scale parameters of$\gamma =$ 2 and λ =8, respectively. For each simulation, we generated N = 100 data sets, each with n = 100 elements. Each data set has approximately a 30% chance of having right censored times (Appendix $B2$).

Assuming that the true survival time has an exponential distribution. The data generation procedure is summarized in the following steps:

- Generate the true survival time t from an exponential distribution with $\lambda = 5$.
- Generate a random variable denoted as c for censoring time from Uniform distribution on the interval (0, 0.7) to obtain censoring proportion closed to 30% as possible.
- The true survival time t and censoring time c is compared and the time and indicator variables are denoted as follows:

$$time[i] = \begin{cases} t[i] & if\ t[i] \leq c[i] \\ c[i] & if\ t[i] > c[i] \end{cases}$$

$$indicator[i] = \begin{cases} 0 & if\ time[i] = c[i] \\ 1 & if\ time[i] = t[i] \end{cases}$$

The variables shown in the third step are used to run the simulation. Similarly, the data sets from Weibull distribution with shape and scale parameters of 2 and 8 are generated.

4.1.2 Nonparametric Estimation

Based on the right censored generated data, the nonparametric estimation of survival function is accomplished using Kaplan Meier estimator considering the 5 survival time points of interest for each distribution considered. The Kaplan Meier survival probability is estimated at each of the 5 times using R software (Appendix $C3$), for each of the 100 data sets. Therefore, ate each time point we have 100 estimated survival probabilities and hence the mean and variance of the estimates at each time point of interest is calculated.

4.1.3 Parametric Estimation

In the parametric approach the survival function is estimated using parametric estimation for the same exponential and Weibull data sets we had generated in the previous part. Where the estimated survival probabilities at the same time points mentioned above in the previous section is obtained. This procedure is conducted in R using the suitable packages.
However, based on the nonparametric and parametric estimation techniques discussed above, the simulation results are presented as in the form shown in table 4.1 below:

Time	True Survival Time S(t)	Estimated survival probabilities	Mean of S(t)	Variance of S(t)
t_1	0.90	$\hat{S}_1(t_1), ..., \hat{S}_{100}(t_1)$	$\bar{S}(t_1)$	VAR$\{S_1(t_1), ..., S_{100}(t_1)\}$
t_2	0.75	$\hat{S}_1(t_2), ..., \hat{S}_{100}(t_2)$	$\bar{S}(t_2)$	VAR$\{S_1(t_2), ..., S_{100}(t_2)\}$
t_3	0.50	$\hat{S}_1(t_3), ..., \hat{S}_{100}(t_3)$	$\bar{S}(t_3)$	VAR$\{S_1(t_3), ..., S_{100}(t_3)\}$
t_4	0.35	$\hat{S}_1(t_4), ..., \hat{S}_{100}(t_4)$	$\bar{S}(t_4)$	VAR$\{S_1(t_4), ..., S_{100}(t_4)\}$
t_5	0.20	$\hat{S}_1(t_5), ..., \hat{S}_{100}(t_5)$	$\bar{S}(t_5)$	VAR$\{S_1(t_5), ..., S_{100}(t_5)\}$

Table 4.1: Output table obtained from each estimation method used in a simulation

The bias between the true survival probability and the average of estimated survival probabilities at time t is estimated as follows:

$$Bias\{\hat{S(t)}\} = \bar{S}(t) - S(t) \tag{4.1}$$

The mean square error (MSE) is also estimated based on the following equation:

$$MSE = [Bias\{\hat{S(t)}\}]^2 + VAR\{\hat{S(t)}\} \tag{4.2}$$

Furthermore, the relative efficiency (RE) shown below is also calculated to compare the Kaplan Meier and parametric estimates of the survival function:

$$RE = \frac{MSE(K - MEstimation)}{MSE(Parametric\,Estimation)} \tag{4.3}$$

However, the results of each distribution are summarized in the tables below, where these tables contain 100 parametric and nonparametric survival probabilities for each of the pre-assigned 5 time points. For the nonparametric Kaplan Meier estimates the standard errors are obtained as well as for parametric estimates.

Time	True Survival Time S(t)	Kaplan Meier(Bias)	Parametric Estimates	K-M (VAR)	Parametric Variance	Relative Efficiency
0.019	0.90	0.0081	-0.0162	0.01120	0.01003	1.094552
0.055	0.75	-0.0511	-0.0339	0.02841	0.01921	2.509967
0.141	0.50	-0.0749	-0.0441	0.03145	0.03064	1.137340
0.230	0.35	-0.0604	-0.0389	0.03267	0.03193	1.085965
0.310	0.20	0.0059	-0.0198	0.02597	0.02391	1.070067

Table 4.2: Simulation results based on 100 data sets from exponential distribution for 30 % right-censored observations.

The results in table 4.2 show that relative efficiency values are all greater than 1, which means that the parametric estimator is more efficient than nonparametric estimator in case of right censoring case which is the Kaplan Meier estimator. Furthermore, both the parametric and Kaplan Meier estimators seem to have biases that are closed to zero in case that the true

survival probability closed to 0.2 and 0.9 where the Kaplan Meier bias values are smaller than the parametric bias values. Furthermore, the parametric estimates have lower variance values than the Kaplan Meier variances.

Time	True Survival Time S(t)	Kaplan Meier(Bias)	Parametric Estimates	K-M (VAR)	Parametric Variance	Relative Efficiency
2.496	0.90	0.0157	0.0071	0.01071	0.00644	1.688104
4.321	0.75	-0.0004	0.0159	0.02850	0.01751	1.604485
6.498	0.5	-0.0570	0.0029	0.03075	0.02398	1.417309
8.187	0.35	-0.0253	-0.0129	0.03290	0.02169	1.534565
10.310	0.20	0.0228	-0.0275	0.02480	0.01542	1.565248

Table 4.3: Simulation results based on 100 data sets from Weibull distribution for 30 % right-censored observations

From results in the previous table the situation is similar to the simulation results based on exponential distribution, where the parametric estimation is more efficient referring to the relative efficiency values. The average estimates based on Kaplan Meier estimation technique have smaller biases than the parametric estimator at some time points, but in general the parametric estimator gives less variable estimates than the K-M at all 5 time points.

4.2 Design of Simulation for Interval Censoring

The simulation for interval censoring is similar to the right censored data case, where the parametric and nonparametric approaches are considered with interval censored model (Appendix $D4$ and $E5$). For nonparametric technique using interval censored data the Turnbull estimator is applied to get correct estimates of the survival function, which was calculated through a program in R.

Similar to the right-censored case, we also considered nonparametric and parametric estimators with data that was interval-censored. To correctly estimate the survival function for interval-censored data using a nonparametric technique, we applied the Turnbull estimator. For each estimation technique, we obtained estimated survival probabilities at each of the 5 times of interest for each of the N = 100 data sets with different percentages of interval censoring. However, in interval censored simulation study various sample sizes are considered ranging from 40 to 300 and found that results were often similar. But following reference [2] the sample size is decided to be 100 for each data set. In this chapter we report on a subset of the simulations involving interval-censored data that we have done.

Generating the interval-censored data involved more steps compared to right-censored data as it will be shown in the sequel.

In this study the exponential distribution with shape parameter [0.5,2] is considered for data generation. Each data set contains 100 interval observations of which approximately 30% were censored; here we do not specify left censored observations from interval censored data. Assuming that the true survival time follows an exponential distribution to control the data generation process, the steps used for data generation are as follows:

- Generate the true survival time t from an exponential distribution with different shape parameter values to control the censoring rate.
- Generate a vector V for the checking times, assuming there are 5 time points, in case of exponential distribution the first time v_1 was generated from U(0,0.115). Then the next time v_2 was generated from $U(v_1, v_1 + 0.115)$. The other checking times were generated in the same manner.
- Generate a 1002 empty matrix named "bound" for each data set. The entries of bound

matrix are the intervals endpoints for each individual after comparing the true survival time with the 5 checking times. In case of right censoring the right end point set as "infinity". The formula used for end points determination is:

For $i = 1, \ldots, 100$, $j = 1, \ldots, 5$

$$bounds[i,1] = \begin{cases} 0 & if\ t < V[i,1] \\ V[i,j] & if\ V[i,j] < t < V[i,j+1]\ where\ j = 1,2,3,4,5 \\ V[i,5] & if\ t > V[i,5] \end{cases}$$

$$bounds[i,2] = \begin{cases} V[i,1] & if\ t < V[i,1] \\ V[i,j+1] & if\ V[i,j] < t < V[i,j+1]\ where\ j = 1,2,3,4 \\ 1000 & if\ t > V[i,5] \end{cases}$$

- Generate a 1002 empty matrix named "status". Based on the bound matrix let:

  Status [i,1] $\equiv$ censoring indicator

$$\alpha_i = \begin{cases} 0 & if\ bound[i,2] = inf \\ 1 & if\ otherwise \end{cases}$$

We used similar steps to those given here to generate interval censored data sets where the true survival time was Weibull ($\alpha = 2, \lambda = 8$).

However, the nonparametric Turnbull estimation method is applied for each interval censored data set in R which involved extracting the corresponding equivalence classes and matrix for each generated data set. The parametric estimation is also considered which can accommodate both right and interval censored data, where the K-M estimator is applied in interval censoring by assuming that the midpoint of the interval was the exactly known

survival time when the indicator variable was equal to 1. If the indicator variable was 0, it is assumed that the lower endpoint of the interval was the time point at which the survival time was right-censored. The reason that Kaplan Meier might use this improvised technique with interval-censored data is because the K-M method is widely employed and easily accessible in common statistical software programs.

For estimation techniques considered in interval censoring, the 5 survival time points considered in the right censoring case were of interest for each distribution considered. However, the average of estimated survival probabilities as well as the variance of the survival probabilities at each time point is obtained. Tables similar to right censoring case for each simulation involving interval-censored data are constructed.

4.2.1 Generating Output

To compare Turnbull Estimation and parametric estimation, we defined RE as it defined in the previous section such that:

$$RE = \frac{MSE(TurnbullEstimation)}{MSE(ParametricEstimation)}$$

and in the same manner the comparison of Kaplan Meier estimator with parametric estimation can be verified based on the relative efficiency below:

$$RE = \frac{MSE(TurnbullEstimation)}{MSE(K-M:MIDEstimation)}$$

Moreover, comparing the Turnbull and K-M estimators when the midpoint of the intervals is used as the "true" failure can be verified using the relative efficiency defined as follows:

$$RE = \frac{MSE(TurnbullEstimation)}{MSE(K-M:MIDEstimation)}$$

Similar to tables in right censoring case tables below were constructed to summarize the results for interval censored data simulation.

Table 4.4: Simulation results based on 100 data sets, from exponential distribution, where the midpoint is assumed to be the exact survival time in the K-M estimation.

Time	True Survival Time S(t)	Kaplan Meier:Mid(Bias)	Parametric Bias	K-M VAR	Parametric Variance	Relative Efficiency
0.019	0.90	-0.19649	-0.0098	0.02543	0.00148	40.60998
0.055	0.75	-0.13191	-0.0187	0.02492	0.00770	5.257376
0.141	0.50	-0.08813	-0.0177	0.02789	0.01896	1.850068
0.230	0.35	-0.05148	-0.0074	0.02501	0.02072	1.331432
0.310	0.20	0.00457	0.0089	0.02397	0.01648	1.448794

Table 4.5: Simulation results based on 100 data sets, from Weibull distribution, where the midpoint is assumed to be the exact survival time in the K-M estimation.

Time	True Survival Time S(t)	Kaplan Meier:Mid(Bias)	Parametric Bias	K-M VAR	Parametric Variance	Relative Efficiency
0.019	0.90	0.0891	-0.0095	0.00089	0.00147	5.658587
0.055	0.75	0.1718	-0.0181	0.00870	0.00757	4.838836
0.141	0.50	0.1015	-0.0172	0.02197	0.01895	1.676843
0.230	0.35	0.0658	-0.0069	0.02478	0.02097	1.385012
0.310	0.20	0.0071	0.0084	0.02397	0.01661	1.440024

Tables 4.4 and 4.5 show the results of comparing nonparametric estimation technique

by considering the Kaplan Meier estimation under the assumption that the midpoint of the interval is the exact survival time with the parametric estimation assuming the correct exponential and Weibull distributions. The values of relative efficiency are greater than 1 in all cases based on the two proposed distributions, which means that the parametric estimation is more efficient than the proposed technique to estimate the survival function in case of interval survival data (i.e Kaplan Meier method assuming the midpoint of the intervals). Furthermore, the parametric estimator has a smaller bias and at other time points the K-M estimator has a smaller bias based on the average estimates of the parametric and Kaplan Meier procedures. However, the parametric estimation always gives a smaller variance than the nonparametric estimates even the exponential or Weibull distribution is assumed.

Table 4.6: Simulation results based on 100 data sets, using exponential interval-censored data.

Time	True S(t)	Parametric (Bias)	Turnbull (Bias)	Parametric VAR	Turnbull VAR	Relative Efficiency
0.019	0.90	0.0920	-0.0092	0.07535	0.00149	0.018787
0.055	0.75	-0.0943	-0.0179	0.07397	0.00771	0.096912
0.141	0.50	-0.0717	-0.0175	0.04094	0.01929	0.425258
0.230	0.35	-0.0022	-0.0069	0.03921	0.02092	0.534686
0.310	0.20	0.0877	0.0087	0.03897	0.01635	0.352020

Table 4.7: results based on 100 data sets, using Weibull interval-censored data.

Time	True S(t)	Parametric (Bias)	Turnbull (Bias)	Parametric VAR	Turnbull VAR	Relative Efficiency
0.019	0.90	0.0247	-0.0345	0.02065	0.00712	0.390885
0.055	0.75	0.0629	-0.0429	0.04338	0.01631	0.383434
0.141	0.50	-0.1443	-0.0487	0.03901	0.02157	0.400145
0.230	0.35	-0.0013	-0.0479	0.05376	0.02189	0.449845
0.310	0.20	-0.0548	-0.0301	0.03529	0.01891	0.517483

The results shown in tables 4.6 and 4.7 show that all values of relative efficiency values are less than 1, which indicates that Turnbull estimator in case on interval censoring model is more efficient than parametric estimation. Furthermore, based on the average estimates in interval censoring model, sometimes the Turnbull estimator has a smaller bias and sometimes the parametric estimator has a smaller bias In the following two tables, the simulation results are ordered to compare the efficiency of Kaplan Meier and Turnbull estimators for both Weibull and exponential distributions respectively:

Table 4.8: results based on 100 data sets, using Weibull interval-censored data.

Time	True S(t)	Parametric (Bias)	Turnbull (Bias)	Parametric VAR	Turnbull VAR	Relative Efficiency
0.019	0.90	-0.00389	0.09821	0.00440	0.01008	0.223832
0.055	0.75	-0.01389	0.16291	0.00530	0.01682	0.126683
0.141	0.50	-0.01259	0.11051	0.01902	0.03110	0.442794
0.230	0.35	-0.00409	0.07361	0.02212	0.03422	0.558466
0.310	0.20	0.01121	0.02521	0.01658	0.03342	0.490542

Table 4.9: results based on 100 data sets, using Weibull interval-censored data.

Time	True S(t)	Parametric (Bias)	Turnbull (Bias)	Parametric VAR	Turnbull VAR	Relative Efficiency
0.019	0.90	-0.0079	0.0792	0.00331	0.00107	0.459291
0.055	0.75	-0.0179	0.1439	0.00421	0.00781	0.158866
0.141	0.50	-0.0166	0.0915	0.01793	0.02209	0.597643
0.230	0.35	-0.0081	0.0546	0.02103	0.02521	0.748306
0.310	0.20	0.0072	0.0062	0.01549	0.02441	0.635699

The relative efficiency in the above two tables indicate that Turnbull estimator is more efficient than Kaplan Meier estimator to represent the survival function in the two proposed simulated data sets based on exponential and Weibull distributions, where all values of the relative efficiency are less than 1.

4.3 Conclusion and Results

In this chapter, different techniques to estimate the survival function has been investigated based on the most two common censoring models; right and interval censoring. The analysis is conducted based on both parametric and nonparametric estimation techniques where we compare some of the survival function estimates considering two specific distributions; exponential and Weibull distributions.

In the right censoring simulation the bias values based on the two proposed distributions (exponential and Weibull) indicate that nonparametric survival function estimator known as the Kaplan-Meier estimator provides good estimates of the survival function but the parametric estimation technique seems to show better estimates of the survival function based on the relative efficiency values whatever the distribution assumed in the parametric case.

In the interval censoring model, it is also concluded that nonparametric estimates using Turnbull estimator is more efficient than parametric estimates either the exponential or Weibull distribution assumed to represent the survival time. Furthermore, based on the obtained relative efficiency values shown in tables 4.8 and 4.9, it is noticed that the Turnbull estimator provides more efficient estimate for the survival function using interval censored data compared to the Kaplan Meier estimator using the midpoint as the exact failure time. Therefore, based on these results, the analysts who have considered the Kaplan Meier estimator in case of interval censored data should not be too confident with their results. Thus, the Turnbull estimator is recommended to be used for the survival function in case of interval survival data rather than the Kaplan Meier method.

Appendices

A.1 Newton Raphson method for system of non linear of equations

Newton's Raphson method can be implemented to solve a system of nonlinear equations. This is particularly useful when we try to find maximum likelihood estimators of several unknown parameters. Thus, to solve the system of equations given in 5.7, let $\frac{\partial L_c}{\partial B_j}$ be a vector-valued function of a vector of the parameters $B_1, B_2, \ldots, B_J$, assuming that we have J components. To apply Newton's method to the problem of approximating solutions of

$$\begin{aligned} f_1(B) &= \frac{\partial L_c}{\partial B_1} = 0 \\ &\vdots \\ f_J(B) &= \frac{\partial L_c}{\partial B_J} = 0. \end{aligned} \tag{A.1.1}$$

Let $f(B^T) = \begin{bmatrix} f_1(B) & f_2(B) & \ldots & f_J(B) \end{bmatrix}^T$, and $f'(B^T) = \begin{bmatrix} \frac{\partial f_1}{\partial B_1} & \frac{\partial f_1}{\partial B_2} & \cdots & \frac{\partial f_1}{\partial B_J} \\ \frac{\partial f_2}{\partial B_1} & \frac{\partial f_2}{\partial B_2} & \cdots & \frac{\partial f_2}{\partial B_J} \\ \vdots & \cdots & \vdots & \vdots \\ \frac{\partial f_J}{\partial B_1} & \frac{\partial f_J}{\partial B_2} & \cdots & \frac{\partial f_J}{\partial B_J} \end{bmatrix}$,

we would like to start from initial points B_0^T and then write

$$\begin{aligned} (B^T)^{n+1} &= (B^T)^n - \frac{f(B^T)}{f'(B^T)} \\ \begin{pmatrix} B_1 \\ \vdots \\ B_J \end{pmatrix}^{n+1} &= \begin{pmatrix} B_1 \\ \vdots \\ B_J \end{pmatrix}^{n} - (\tilde{f}(B^T))^{-1} f(B^T). \end{aligned} \tag{A.1.2}$$

where $(\tilde{f}(B^T))^{-1}$ is known as the *Inverse of the Jacobian Matrix.*

Rather than computing the inverse of the Jacobian matrix, we define a vector v, such that

$$\begin{pmatrix} v_1 \\ \vdots \\ v_J \end{pmatrix} = -(\tilde{f}(B^T))^{-1} f(B^T)$$

$$\begin{pmatrix} v_1 \\ \vdots \\ v_J \end{pmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial B_1} & \frac{\partial f_1}{\partial B_2} & \cdots & \frac{\partial f_1}{\partial B_J} \\ \frac{\partial f_2}{\partial B_1} & \frac{\partial f_2}{\partial B_2} & \cdots & \frac{\partial f_2}{\partial B_J} \\ \vdots & \cdots & \vdots & \vdots \\ \frac{\partial f_J}{\partial B_1} & \frac{\partial f_J}{\partial B_2} & \cdots & \frac{\partial f_J}{\partial B_J} \end{bmatrix}^{-1} \begin{pmatrix} f_1(B) \\ \vdots \\ f_J(B) \end{pmatrix}, \tag{A.1.3}$$

Multiply both sides of equation (A.1.3) by $\tilde{f}(B^T)$ then we get

$$\begin{bmatrix} \frac{\partial f_1}{\partial B_1} & \frac{\partial f_1}{\partial B_2} & \cdots & \frac{\partial f_1}{\partial B_J} \\ \frac{\partial f_2}{\partial B_1} & \frac{\partial f_2}{\partial B_2} & \cdots & \frac{\partial f_2}{\partial B_J} \\ \vdots & \cdots & \vdots & \vdots \\ \frac{\partial f_J}{\partial B_1} & \frac{\partial f_J}{\partial B_2} & \cdots & \frac{\partial f_J}{\partial B_J} \end{bmatrix} \begin{pmatrix} v_1 \\ \vdots \\ v_J \end{pmatrix} = - \begin{pmatrix} f_1(B) \\ \vdots \\ f_J(B) \end{pmatrix}. \tag{A.1.4}$$

and then solving the linear system of equations (A.1.4) for $(\; v_1 \quad \ldots \quad v_J \;)$.

Once $\begin{pmatrix} v_1 \\ \vdots \\ v_J \end{pmatrix}$ is known, the next iteration is computed according to the rule

$$\begin{pmatrix} B_1 \\ \vdots \\ B_J \end{pmatrix}^{n+1} = \begin{pmatrix} B_1 \\ \vdots \\ B_J \end{pmatrix}^{n} - \begin{pmatrix} v_1 \\ \vdots \\ v_J \end{pmatrix}. \tag{A.1.5}$$

Repeat until converge.

B.2 Parametric for right censoring

```
############### Generation of the Data ################
cen=cens=array()
 realcensoring= Bias= array()
Error=Censoring=LossFollow=array()
Bias1=Bias2=MSE=array()
 ####################################
  for (w in 1:20)
    {
    dat1<-data.frame(time=rexp( 100,rate=5), Censored=rbinom(100,1,0.90),
    dat2<-dat1[order(dat1[,1]),]                    # order the data #
     for (i in 1:10)
            {
          dat2$Censored[i+90]=0
          dat2$time[i+90]=dat2$time[90]
                        }

     cens<-c(dat2$Censored)      #censored status#
     tim<-c(dat2$time)           #lifetimes#
     L1<-length(cens)           #number of censored#
      for (j in 1:L1)
         {
    if ((cens[j]==1) {(cen[j]=1)}
           else {(cen[j]=cens[j])}
                                                    }
       L3<-length(cen[cen<1])              #number of censored#
     Realcen=(L3/length(cen))
         RealCensoring<-sprintf("%.0f%%", 100*Realcen)
          realcensoring[w]=RealCensoring   #Percent of censored#
##################### Connections ##########################
    data=data.frame(Ti=dat1$time,Censored=cen(

```

```
32  ##################### Basics  ############################
38 n=length(data$Ti)                  # Length of the whole data #
34  m<-length(data$Censored[data$Censored>0])     # Number of uncensored
      individuals #
35  d=(n-m)              # number of censored individuals #
36 p=T.cen=Stat.cen=array(1,d)
37  Lambda=Theta=pp=ppp=pppp=outL=outT=array(1,40)
38  T.uncen=array(1,m)
39  Stat.uncen=array(1,m)
40  p=array(1,d(
41
42  dat2<-c(split(data[1:3],data$Censored==1))  # Split the data #
43
44  T.uncen<-c((dat2$'TRUE')$Ti)          # Life times data set for uncensored
      #
45  Stat.uncen<-c((dat2$'TRUE')$Censored)        # censoring Status data set #
46
47  T.cen<-c((dat2$'FALSE')$Ti)           # Life times data set for censored #
48  Stat.cen<-c((dat2$'FALSE')$Censored)         # censoring Status data set #
49 ###### Initial values ######
50 Lambda=1
51  Theta=1
52
53 for (i in 1:40)
54  {
55    for (j in 1:d)
56     {
57       p[j]=((1)/(1+((1-exp(-Theta[i]))*(exp(Theta[i]-(Lambda[i]*T.cen[j]))
      ))))
58           }
59     pp<-c(p(
60     ppp<-c(1-p(
61     pppp<-c(T.cen*p(
```

```
62
63    S1=sum(ppp)
64    S2=sum(pp(
65    S3=sum(pppp(
66  Lambda[i+1]=((m)/(sum(T.uncen)+S3))
67   Theta[i+1]=log(((m+S2)/S1)+1(          ########
68    Th=Theta[41]
69    Lam=Lambda[41]
70                                  }
71    outL[w]<-c(Lambda(
72    outT[w]<-c(Theta)
73
74  Censoring[w]=sprintf("%.0f%%", 100*(d/n))
75   LossFollow[w]=sprintf("%.0f%%", 100*((d/n)))    # Loss follow up #
76
77   ########################## MSE ######################################
78   Bias1[w]=((Real-exp(-Th))^2)    # Bias square #
79                                                                        }
80     ##################### Mean Square Error (MSE ( ################
81      V=var(Expected)            # variance of the estimate values #
82      Bias2=c(Bias1)                  # set of Bias square #
83
84      MSE=round((V+Bias2)*1000)
85     ###################################################################
86
87   Result1<-data.frame(Lambda=Lambda,Theta=Theta)
88   Result2<-data.frame(CensorRate=Censoring ,LossFollow=LossFollow,
89   Expected=Exp,Bias=Bias,MSEx1000=MSE)
90     print(Result1)  #Theta and Lambda#
91     Result3<-Result2[order(Result2[,1]),]
92     print(data.frame(Result3))
```

C.3 Nonparametric for right censoring

```
1
2  ############### Generation of the Data ################
3 cen=array(1,100)
4  cens=array(1,100)
5 realcensoring=array(1,20)
6 Error=array(1,20)
7   for (w in 1:20)
8    {
9     dat1<-data.frame(time=rexp(100),Censored=rbinom(100,1,0.8))
10    datcen<-dat1[order(dat1[,1]),]                    # order the data #
11    cens<-c(dat1$Censored)        #censored status#
12     for (i in 1:15)
13            { cens[i+85]=0  }
14     L1<-length(cens[cens<1])      #number of censored#
15     L3<-length(cen[cen<1])              #number of censored#
16     Realcen=(L3/length(cen))
17            RealCensoring<-sprintf("%.0f%%", 100*Realcen)
18            realcensoring[w]=RealCensoring           #Percent of censored#
19 ################################
20  n<-length(cen)
21  m=n
22  k=array(1,n)
23  for (i in 1:n)
24     {if (cen[i]==0) d=0 else d=1
25      k[i]<-1-(d/m[i])     #the estimate of the KM survival function
26              m[i+1]=m[i]-1
27                                }
28              W<-c(k)             #the KM estimate for each individual
29                   p=W[1]
30                                  for (j in 2:n) {p[j]=p[j-1]*W[j]   #
    computation of the product
```

```
                    N<-c(p)                          # Estimation of S(t)
                    M=1-N                            # Estimation of F(t)
                  sump<-(1-N[n])                   # Summation of the jumps
                                          }
                           # print(W)                 # (1-(d/n)for each
individual)
                           # print(N)                 # KM Estimator
                           # print(M)

############################ EM Algorithm ######################
   fn<-function(T)
         {
              for (i in 1:n)
               {
 Q1=((exp(-T)-(M[i]*exp(-T*M[i])))/((exp(-T*M[i]))-exp(-T))) #
Derivative #
 Q2=log((exp(-T*M[i]))-exp(-T))
                            }
 QQ1<-c(Q1)
 QQ2<-c(Q2)

     fn<-sum((T*(M[1]-1)*exp(T*(M[i]-1)))-((1-(exp(T*(M[1]-1)))))+
            (((1-((1-(exp(T*(M[1]-1))))))*sum(QQ1)))+
            ((M[1]-1)*exp(T*(M[1]-1))*sum(QQ2)))
                                      }        # end of function #
          Z<-uniroot(fn,interval=c(0.1,5))
          Th<-c(Z$root)
Result<-data.frame(Censoring=RealCen , Error=Eror)
print(Result)
#write.table(Result,"e:/Result.csv", sep=",", row.names=FALSE) #send the
  file to Execl#
```

D.4 Parametric for interval censoring

```
########## 100 individuals in each sample ###############
  ################ Step 1  (true survival time:tt) ##################
RealCens= Bias=array()
for (w in 1:20)
{
    time<-c(data.frame(time=rexp(100,rate=0.6))$time)
 ################ Step 2  (20 visits:V) ##################
  v<-array(1,20)
  for (i in 1:19)
   {
     v[1]=runif(1,0,0.215)
     v[i+1]=runif(1,v[i],v[i]+0.215)
                                        }
     V<-c(v)
################ Step 3  (generating 100x2 matrix) (bounds) ############
  bounds<-matrix(nrow=100,ncol=2)
 ########## Step 4 (compare t and V) to determine the bounds ########
           ###### left and right bounds #######
   for (i in 1:length(time))
    {
        if(time[i]<V[1]){(bounds[i,1]=0);(bounds[i,2]=V[1])}else

        if(time[i]>V[20]){(bounds[i,1]=sample(V,1,rep=FALSE));(bounds[i
   ,2]=Inf)}else
      for (j in 1:(length(V)-1))
        {
          if(time[i]>=V[j] && time[i]<=V[j+1]){(bounds[i,1]=V[j]);(bounds[
   i,2]=V[j+1])}
                                              }
                                                    }
   #print(bounds)
```

```
################ Generating Censoring ###############
Left=Leftun=Rightun=tun=Right=array()
  censoring<-matrix(nrow=length(time),ncol=1)
  for (i in 1:length(time))
    {
      if (bounds[i,2]==Inf)    #right censored#
        {Left[i]=bounds[i,1];Right[i]=5}
           else {}
      if (bounds[i,2]!=Inf)    #right censored#
        {Leftun[i]=bounds[i,1];Rightun[i]=bounds[i,2]}
           else {}
                          }

     L0<-c(na.omit(Left))    #Left endpoints for censored individuals#
     R0<-c(na.omit(Right))    #Right endpoints for censored individuals#
     L1<-c(na.omit(Leftun))    #Left endpoints for uncensored individuals#
     R1<-c(na.omit(Rightun))    #Right endpoints for uncensored
   individuals#
    for (i in 1:length(L1))
     {
       tun[i]=(L1[i]+R1[i])/2
                               }
    ################ Solving for Parameters ###############
    n=length(time)
    d=length(R0)    #number of censored individuals
    m=length(R1)    #number of uncensored individuals
          #P[1]=Theta , p[2]=Lammbda
 Lik=function(p)
     {
      -((m*log(p[2]))-(p[2]*sum(tun))+(m*log(1-exp(-p[1])))-
      (p[1]* sum(1/(1+((exp(p[1])-1)*(exp(-p[2]*L0)-exp(-p[2]*R0))))))+
      log(1-exp(-p[1]))*sum(1-((1/(1+((exp(p[1])-1)*(exp(-p[2]*L0)-exp(-p
   [2]*R0)))))))+
```

```
61        sum((1-((1/(1+((exp(p[1])-1)*(exp(-p[2]*L0)-exp(-p[2]*R0)))))))*(log
      ((exp(-p[1]*L0))-(exp(-p[1]*R0))))))
62                                    }
63   Max=optim(c(1,1),Lik)
64   param=c(Max$par)
65  Bias[w]=sprintf("%.0f%%",100*abs((((length(R0)/length(time))))-exp(-param
      [1])))    }
66  Result<-data.frame(CensorRate=RealCens, Bias=Bias, MeanError= Mean square
      error)
67    print(Result)
```

E.5 Nonparametric for interval censoring

```
########## 100 individuals in each sample ##############
  ################ Step 1  (true survival time:tt) ##################


    time<-c(data.frame(time=rexp(100,rate=1.5))$time)
 ################ Step 2  (20 visits:V) ##################
  v<-array(1,20)
  for (i in 1:19)
   {
     v[1]=runif(1,0,0.115)
     v[i+1]=runif(1,v[i],v[i]+0.115)
                                      }
     V<-c(v)
################ Step 3  (generating 100x2 matrix) (bounds) ############
  bounds<-matrix(nrow=100,ncol=2)
 ######## Step 4 (compare t and V) to determine the bounds ############
           ###### left and right bounds #######
   for (i in 1:100)
    {
        if(time[i]<V[1]){(bounds[i,1]=0);(bounds[i,2]=V[1])}else        if(
   time[i]>V[20]){(bounds[i,1]=sample(V,1,rep=FALSE));(bounds[i,2]=Inf)}
   else
      for (j in 1:(length(V)-1))
        {
   if(time[i]>=V[j] &&time[i]<=V[j+1]){(bounds[i,1]=V[j]);(bounds[i,2]=V[j
   +1])}
                                                 } }
 ################ Generating Censoring ###############
   censoring <-matrix(nrow=100,ncol=1)
   for (i in 1:100)
     {
       if (bounds[i,2]==Inf)
```

```
      else {(censoring[i,1]=1) }
                                }
  fff<-data.frame(censoring=censoring)
    censor<-length(censoring[censoring<1])    #number of censored#
        RealCens<-sprintf("%.0f%%", 100*(censor/length(time)))  #Real
  censoring fraction#
  ############## Arrange the data file in a data frame ################
      leftL=array(1:100)        #create left label to data#
       for (i in 1:100)
         {
           leftL[i]="L"
                           }
       rightL=array(1:100)      #create right label to data#
         for (i in 1:100)
           {
             rightL[i]="R"
                              }
 data1=data.frame(left=bounds[ ,1],Label1=leftL,right=bounds[ ,2],
                  Label2=rightL,cens=censoring)

########################### Estimation process ##################
    L<-data.frame(bound=data1$left,label=data1$Label1)
    R<-data.frame(bound=data1$right,label=data1$Label2)
 ########################################################
  common<- intersect(colnames(L), colnames(R))
 data2<-rbind(L[,common], R[,common])          # combine data to sort it #
 sortedLU<-data2[order(data2$bound, decreasing=FALSE), ]  #sort the data#
 eq1<-c(sortedLU$label)  ## Sorted labels of boundaries (1:L),(2:R) ##
 eq2<-c(sortedLU$bound)  ## Sorted values of boundaries ##
   L2=array(1,(length(eq2)-1))
   R2=array(1,(length(eq2)-1))
   for (i in 1:(length(eq1)-1))   # define the Equivalent Intervals #
     {
```

```
  if ((eq1[i]=="L") && (eq1[i+1]=="R")) {(L2[i]=eq2[i]);(R2[i]=eq2[i
+1])}
      else {(L2[i]=NA) ; (R2[i]=NA)}     #1:left, 2:right#
                                        }


l2<-c(L2)          ### **** set of equivalent Intervals **** ###
r2<-c(R2)
L22<-c(na.omit(l2))       ## Omit NA values ##
R22<-c(na.omit(r2))
       for (i in 1:(length(R22))) #cancel Inf from right endpionts#
     {
       if(R22[i]==Inf) {R22[i]=max(V)} else{R22[i]=R22[i]}
                                        }


EqClass<-data.frame(L22,R22)  ## Net Equivalent Classes ##
 n<-length(c(data1$left))
m=length(L22)
a=1
#################### Step 1 :(Define Alpha Matrix) ##########
alpha<-matrix(0,nrow=n,ncol=m,byrow=TRUE)
left1<-c(L$bound)     #### original classes ****#
right1<-c(R$bound)
left2<-c(L22)              #### Equivalent classes ****#
right2<-c(R22)
  for (j in 1:m)
   {
    for(i in 1:n)
     {
      if ((left2[j]>=left1[i]) && (right2[j]<=right1[i]))
         {alpha[i,j]=1}
             else {alpha[i,j]=0}
                                  }
                                    }
```

```
############ Step 2:(Define mu matrix) and loop ################
  out<-matrix(nrow=m,ncol=40,byrow=TRUE)  ## Store Loop output ##
  S<-matrix(1/m,nrow=m,ncol=1,byrow=TRUE)  ## initial values for s (jumps)
    ##

  for (w in 1:40)
   {
   mu<-matrix(nrow=n,ncol=m,byrow=TRUE)       # mu matrix #
   num<-matrix(nrow=n,ncol=m,byrow=FALSE)    ## numerator in mu matrix ##
den<-matrix((alpha%*%S),nrow=n,ncol=1,byrow=TRUE)    ## denominator in mu
   matrix ##
   for (j in 1:m)
     {
      for (i in 1:n)
       {
         num[i,j]=alpha[i,j]*S[j]
           mu[i,j]<-num[i,j]/den[i]
                                    }
                                      }
 ###################### Step 3:Define Pie=sj ######################
Pie<-matrix(colSums(mu),nrow=m,ncol=1,byrow=TRUE)/n    ### sj=Mu/n ###
     S=Pie
      for (u in 1:length(S))
          {
            out[u,w]=round(S[u],6)
                                  }   }
      Net<-c(out[,40])
      #print(Net)            # print the last iteration only #

  ###################### Cumulative Jumps ####################
    CumF=array(1:length(Net))
     CumF[1]=Net[1]
      for (i in 2: length(Net))
```

```
124       {CumF[i]=CumF[i-1]+Net[i]      # CumF: Cumulative jumps #
125                                      }
126   NetS=data.frame(EqClass=EqClass,St=abs((1-CumF)))       #print(NetS)
127  ############## Define F(x) the cumulative function ##############
128    t=array( ,n)
129    F=array( ,n)
130    Intervals<-data.frame(left=L[ ,1],right=R[ ,1])   # original Intervals,
       #
131     LL<-c(Intervals[ ,1])      # Left endpoints of intervals #
132     RR<-c(Intervals[ ,2])      # Right endpoints of intervals #
133        q<-c(EqClass[,1])         # Left endpoints of Equivalent intervals
     #
134        p<-c(EqClass[,2])         # Right endpoints of Equivalent intervals
       #
135
136   for (i in 1:n)
137      {                          # define the t:comparable point #
138       if (RR[i]==Inf) {t[i]=sample(LL[i]:max(V),1)}
139        else (t[i]=runif(1,LL[i],RR[i]))   # t is random number from the
     intervals #
140                           }
141       ##################################################################
142       ###### Define F(x) based on the generated values of t ####
143       ##################################################################
144   d=array()
145   for (i in 1:n)
146     {
147       if(t[i]<q[1]) {F[i]=0.01} else   #first endpiont#
148       if(t[i]>=p[m]) {F[i]=0.99} else  #last endpiont#
149   for(j in 1:m-1)
150         {   d[j]=(p[j]-q[j])
151          if((d[j]=0)&&(t[i]=p[j])) {F[i]=CumF[j]} else     #if the jump
     occurred at one point#
```

```
152          if ((t[i]>=p[j]) && (t[i]<q[j+1])) {F[i]=CumF[j]} else (F[i]=F[i
    ])
153                                        }   }
154       for (i in 1:n)        # S(t) when t belongs to equivalence intervals #
155       {
156          for(j in 1:m)
157            {
158              if ((t[i]>q[j]) && (t[i]<=p[j]))
159         {
160         tN<-c(seq((q[j]+0.00001),(p[j]-0.00001),len=100))  # generate 100 t
    between p and q #
161          if(j==1) {FN<-c(seq(0,CumF[j],len=100))}else
162         {FN<-c(seq((CumF[j-1]),(CumF[j]),len=100))}   # generate 100 FN(1-S)
     between S(q) and S(p) #
163             for(e in 1:99)
164                {
165                  if ((t[i]>=tN[e]) && (t[i]<tN[e+1]))
166                      {F[i]=mean(c(FN[e],FN[e+1]))}  else(F[i]=F[i])
167                                              }
168                                                }else ((F[i]=F[i]))
169                                                                    }    }
170    Sx<-data.frame(Surf.Func=1-F)    # The Survival Function #
171
172  ########################## Solving for Theta ####################
173  Surv1=array()
174  censoring<-c(censoring)  #censoring status#
175  survival<-c(1-F)         #survival function#
176  for(i in 1:length(censoring))  # Determine the survival function of
     censored ind.#
177     {
178        if (censoring[i]==0) {Surv1[i]=survival[i]}
179        Surv2<-c(Surv1)    }
180        Si<-c(na.omit(Surv2))
```

```
181        mm<-(length(Si))     ### number of censored individuals ###
182 ##################################################################
183    ############# Numerical solution of theta Equation ##############
       ##################################################################
184                        ########## pi=1-ci=1-gi **********#
185     s1=s2=s3=pi=array()
186      T=1
187          for(j in 1:20)
188            {
189              for (i in 1:mm)
190                 {
191                  s1[i]=pi[i]=exp(-T[j]*Si[i])
192                  s2[i]=1-s1[i]
193                                      }
194               s11=sum(s1)
195               s22=sum(s2)
196               Pi=c(pi)
197     Q1=Q2=Q3=array()
198     fn<-function(T)
199             {   for (i in 1:mm)
200                 {
201     Q1[i]=(Si[i]/(1+exp(-T*Si[i])))
202     Q2[i]=(1-Pi[i])*Q1[i]
203                               }
204         fn=(n-mm)+s22-(exp(T)*(mm))+s11+((exp(T)-1)*sum(Q2))
205                                     }   # end of function #
206     Z<-uniroot(fn,interval=c(0.001,100))
207     T[j+1]=(Z$root)
208                             }
209     Result<-data.frame(Bias=bias, MSE=Mean error)
210     print(Result)
```

Bibliography

[1] A. Dempster, N. Laird and D. Rubin, Maximum likelihood from incomplete data via the EM algorithm. *Journal of royal statistical Society*, 39(1):1-38, 1977.

[2] C. Regina Elandt-Johnson and Norman L. Johnson *Survival Models and Data Analysis.* Wilew, 1980.

[3] D. Bernoulli *Essai d'une nouvelle analyse de la mortalité causée par la petite vérole.* Mem. Math. Phy. Acad. Roy. Sci., Paris, 1971.

[4] D. M. Witten, R. Tibshirani, Survival analysis with high-dimensional covariates. *Stat. Methods Med. Res.*, 19, 29–51, 2010.

[5] Dudley, William N., Wickham, Rita, Coombs, Nicholas. (2016). An Introduction to Survival Statistics: Kaplan-Meier Analysis. *Journal of Advanced Practitioner in Oncology*, 7(1).

[6] G. E. Dinse. An Alternative to Efron's Redistribution-of-Mass Construction of the Kaplan-Meier Estimator. *The American Statistician*, 39(4):299-300, 1985.

[7] Goodall, Dunn and Babiker. Interval-censored survival time data: confidence intervals for the non-parametric survivor function.. *Statistics in Medicine*, 23:1131-1145, 2004.

[8] J. Kim and D. Pollard. Cube root asymptotics. *Annals of Statistics*, 18: 191-219, 1990.

[9] J. P. Ska. The EM algorithm and its implementation for the estimation frequencies of snp-haplotypes. *Int. J. Appl. Math. Compute Sci.*, 13(3):419-429, 2003.

[10] J. Rossi Richard. *Mathematical Statistics : An Introduction to Likelihood Based Inference.* John Wiley and Sons, New York, p.227, 2018.

[11] K. Doehler and M. Davidian. *Smooth' Inference for Survival Functions with Arbitrarily Censored Data.* Statistics in Medicine, 2008.

[12] Kleinbaum, G. David, Klein and Mitchel *Survival analysis.* Third edition, 2012.

[13] R. Maller and S. Zhou. *Survival Analysis with Long-Term survivors.* First Edition, John Wiley and Sons, New York, 1996.

[14] R. Lira, R. Antunes-Foschini and R. Rocha. Survival analysis (Kaplan-Meier curves): a method to predict the future. *Arq Bras Oftalmol,* 83(2):V-VII, 2020.

[15] Tableman, Mara, Kim and Jong Sung. *Survival Analysis Using S* Chapman and Hall/CRC, First Edition, 2003.

[16] T. Hastie, R. Tibrishani and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Berlin, 2017.

[17] Z. Khalid and B.J.T Morgan. Cross-sectional and longitudinal Approaches in a survival Mixture Model. *Matematika,* 24: 231-242, 2008.