

COMPETING CAUSES OF FAILURE FOR
WEIBULL LIFETIMES UNDER PROGRESSIVE
TYPE II CENSORING

COMPETING CAUSES OF FAILURE FOR WEIBULL
LIFETIMES UNDER TYPE II PROGRESSIVE
CENSORING

By
MALLIKARJUNA RAO RETTIGANTI, M.Sc.

A Project
Submitted to the School of Graduate Studies
in Partial Fulfillment of the Requirements
for the Degree
Master of Science

McMaster University
© Copyright by Mallikarjuna Rao Rettiganti, 2005

MASTER OF SCIENCE (2005)
(Statistics)

McMaster University
Hamilton, Ontario

TITLE: Competing Causes of Failure for Weibull Lifetimes under
Type II Progressive Censoring

AUTHOR: Mallikarjuna Rao Rettiganti

SUPERVISOR: Professor Narayanaswami Balakrishnan

NUMBER OF PAGES: xi, 63

*To Amma, Nanna, Akka and Ramu:
All this would not have been possible
without your love and support.*

ℰ

*To K. N. Rao Taatagaru:
I did it!!*

Table of Contents

Table of Contents	v
List of Tables	vii
Acknowledgements	ix
Abstract	xi
1 Introduction	1
1.1 Introduction to Competing Risks	1
1.2 The Need for Progressive Censoring	2
1.3 The Model	4
1.3.1 Notations	5
1.3.2 Description	5
1.4 Algorithm for Generating Progressively Type II Censored Data under competing risks for Weibull Lifetimes	6
2 Maximum Likelihood Estimation under Competing Risks	9
2.1 Maximum Likelihood Estimation	9
2.2 MLE for Weibull Distribution with Equal Shape Parameters	11
2.2.1 Relative Risk	11
2.2.2 Finding the MLEs	12
2.2.3 Algorithm for finding the MLEs	14
3 Confidence Intervals	15
3.1 Asymptotic Confidence Intervals	15

3.2	Bootstrap- p Confidence Intervals	17
3.3	Studentized or Bootstrap- t Confidence Intervals	18
4	Simulation Study	21
4.1	Probability Coverages and Average Coverage Lengths (ACLs)	21
4.2	Performance and Comparison of Confidence Intervals	22
4.2.1	Comparisons based on Probability Coverages and ACLs	23
4.2.2	Comparison based on Asymptotic Bias, MSE and Others	24
4.3	The Results	26
5	Numerical Example	37
5.1	Results for Each Censoring Scheme	37
5.1.1	Censoring Scheme - C_1	38
5.1.2	Censoring Scheme - C_2	39
5.1.3	Censoring Scheme - C_3	40
5.1.4	Censoring Scheme - C_4	41
6	Conclusion	43
6.1	Extensions	44
A	R Functions Used in this Project	45
A.1	Functions used for Simulation	45
A.2	Functions Used for the Numerical Example	55
	Bibliography	60

List of Tables

4.1	Probability Coverages and ACLs for: $n = 200, m = 80$	27
4.2	Probability Coverages and ACLs for: $n = 100, m = 40$	28
4.3	Probability Coverages and ACLs for: $n = 50, m = 20$	29
4.4	Probability Coverages and ACLs for: $n = 30, m = 15$	30
4.5	Probability Coverages and ACLs for: $n = 30, m = 10$	31
4.6	Bias, MSE, Mean and Variance for: $n = 200, m = 80$	32
4.7	Bias, MSE, Mean and Variance for: $n = 100, m = 40$	33
4.8	Bias, MSE, Mean and Variance for: $n = 50, m = 20$	34
4.9	Bias, MSE, Mean and Variance for: $n = 30, m = 15$	35
4.10	Bias, MSE, Mean and Variance for: $n = 30, m = 10$	36
5.1	Autopsy data for 99 RFM conventional male mice which received a radiation dose of 300r at age 5-6 weeks	38
5.2	Confidence intervals and lengths for censoring scheme C_1	39
5.3	Confidence intervals and lengths for censoring scheme C_2	40
5.4	Confidence intervals and lengths for censoring scheme C_3	41
5.5	Confidence intervals and lengths for censoring scheme C_4	42

Acknowledgements

There are great many people who have been responsible for the succesful completion of this project among many other things. I am glad to be able to thank them at this time.

First and foremost, I would like to thank my supervisor, Dr. N. Balakrishnan, for his wonderful help and suggestions provided throughtout the course of not only my project, but also my entire M.Sc. here at McMaster. Without his help, this project would not have been possible. I would like to thank all my professors here at McMaster and Loyola College, Chennai, India. Teaching is a highly noble profession and you all have inspired me to adopt it at some point in my future.

I am grateful to all my friends that I have made after coming to McMaster, whose association I will always cherish. I have always relied on Enayet's help for almost everything, and I am thankful to him. I would also like to thank Dharmesh, who made me believe in myself. Binod, Shah, Nilesh, Sanjay and Yuvraj: you all have been really great. I hope to continue my association with you.

My father, mother, sister and brother: you all have been responsible for my every effort until now and I do not think I can give you enough thanks. I would like to thank my grand father, Dr. K. N. Rao, for everything he has done. His contagious love for reading and writing has infected me.

Last, but not the least, my heartfelt thanks to all those whom I have not mentioned!!

Abstract

In life-testing experiments, specimens may fail due to more than one cause. Usually, the researcher who is interested in the primary cause of failure, treats the other causes of failure as censored data. In recent times, however, analysis is being done in the presence of competing causes of failure. Also, censoring becomes inevitable since it saves time and resources. In this project, we discuss inference for competing risks under Type II progressive censoring for Weibull lifetimes. We have considered a simple model where we assume equal shape parameters.

This report is organized into six chapters. In Chapter 1, the concepts of progressive censoring and competing risks are introduced. The model is presented and then described. Notations used throughout the project are also given. Maximum likelihood equations are derived in Chapter 2 and the algorithm for solving them is also presented. Different confidence intervals for the parameters are discussed in Chapter 3 and the simulation results comparing the probability coverages and the lengths of the different confidence intervals are provided in Chapter 4. The performance of these different intervals is illustrated using a real dataset in Chapter 5. Finally, in Chapter 6, we provide a summary of the results and some extensions that may be carried out in the future.

Chapter 1

Introduction

1.1 Introduction to Competing Risks

In the analysis of reliability data, the failure of items may be due to more than one cause. Such cases give rise to the analysis of competing risks. There are many examples available where items or individuals can fail due to one of several causes. For example, Hoel (1972) gave data based on a laboratory experiment in which mice were given a dose of radiation at 6 weeks of age. The causes of death were recorded as Thymic Lymphoma, Reticulum Cell Sarcoma, or other. Another example is the very famous Boag's data (Boag 1949), where, in a study of breast cancer patients, the cause of death were recorded as "cancer" or "others". Traditionally, the researcher who is interested in the primary cause of failure, treats the other causes of failure as censored data. In recent years, however, lifetimes have been modelled in the presence of other competing risk factors. The data consists of a failure time and an indicator variable indicating the cause corresponding to that failure. The causes of failure may be assumed to be either independent or dependent. Usually the causes are assumed to act independently of each other, even though the the assumption of dependence may be more realistic. The identifiability of the model is a major concern in assuming dependent lifetimes. According to Kalbfleisch and Prentice (1980), Crowder (2001) and several other authors, without information on covariates, it is not possible using

data, to test the assumption of independent failure times. The reader can refer to Crowder (2001) and the monograph by David and Moeschberger (1978) who have discussed the different competing risks models in an exhaustive manner.

In this project, we shall develop inference for the competing risk model under a general type of censoring. Censoring is unavoidable in life-testing experiments since it is not always possible to collect the data for each individual. For example, the individual may withdraw from a study which may result in loss of data. Also, the test facilities may be needed for other purposes or it may not be possible to test all individuals due to cost constraints. In such cases, censoring becomes inevitable. The two most common types of censoring are Type I and Type II censoring. Consider n items in a life-testing experiment. In Type I censoring, the experiment is continued up to a specified time T . Failures that occur after T are not observed. This termination point T is assumed to be independent of the failure times. However, in Type II censoring, the experiment is continued until a specified number of units, say $m \leq n$, have failed. That is, only the smallest m lifetimes are observed. In Type I censoring, the number of failures is random and the end point of the experiment is fixed, whereas in Type II censoring the number of failures is fixed and the end point of the experiment is random.

Conventional Type I and Type II censoring have been discussed in much detail by numerous authors including Harter (1992), Mann, Schafer and Singpurwalla (1974), Bain (1978), Lawless (1982), Nelson (1982), Balakrishnan and Cohen (1991) and Bain and Engelhardt (1991).

1.2 The Need for Progressive Censoring

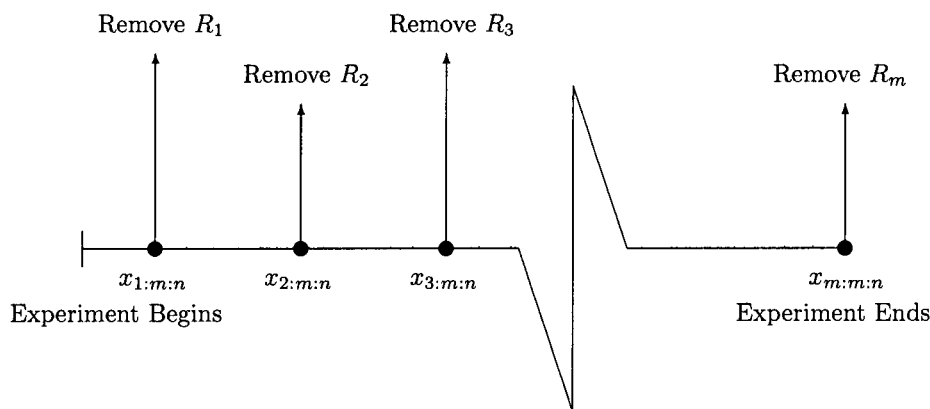
Conventional Type I and II censoring (discussed above), though useful, does not have a lot of flexibility because all the items have to be censored at the terminal point of the experiment only. This was a major drawback of censoring studies until Cohen (1963) and Cohen (1966) first studied a more general type of censoring. He allowed

for items to be censored at different time points during the experiment. Such stages arise naturally when removals of test specimens are required for various reasons, such as to collect degradation-related information from live specimen or to release scarce test facilities for other use. Fix m censoring times T_1, \dots, T_m . At time point T_i , remove R_i of the remaining units (uncensored and units that have not yet failed) randomly. The experiment terminates at time T_m with R_m units still surviving. This is called Progressive Type I censoring. A more detailed view of this mode of censoring is discussed by Cohen (1991).

In this project, we are interested in a different type of censoring called progressive Type II censoring. This allows for items to be removed at different points during the experiment. Also, we accomodate competing risks here. The censoring scheme is as follows: Consider n individuals in a life-testing experiment and let us assume there are K causes of failure competing against each other independently. At the time of each failure, the failure time and the cause of failure is noted and one or more of the surviving units *may* be removed from the experiment at random. Thus, the data from the progressively Type II censored sample under competing risks can be denoted as follows:

$$(X_{1:m:n}, \delta_1, R_1), \dots, (X_{m:m:n}, \delta_m, R_m),$$

where $X_{1:m:n} < \dots < X_{m:m:n}$ denote the m observed failure times, $\delta_1, \dots, \delta_m$ denote the causes of failure and R_1, \dots, R_m denote the number of items removed from the study at the failure times $X_{1:m:n}, \dots, X_{m:m:n}$. At the time of the m^{th} failure, the remaining units in the study are censored. This implies $R_1 + \dots + R_m = n - m$. This censoring scheme can be pictorially depicted as follows.



Progressive Type II censoring has been discussed exhaustively in the book by Balakrishnan and Aggarwala (2000). We shall concentrate on the analysis of competing risks under such a progressive Type II censoring.

1.3 The Model

We shall now discuss the model for lifetimes for competing risks under progressive Type II censoring in detail. We shall also assume that the lifetimes under the competing risks have independent Weibull distributions with the same shape parameter. We consider a very simple model where we assume equal shape parameters for both causes of failure. Note that Kundu, Kannan and Balakrishnan (2004) have already done the analysis for competing risks under progressive Type II censoring for exponential lifetimes.

Without loss of generality we shall assume that there are only two causes of failure,

i.e., $K = 2$. The results can be easily extended for $K > 2$. We shall now introduce the following notations.

1.3.1 Notations

1. X_{ji} : lifetime of the i^{th} individual under Cause j , $j = 1, 2$;
2. $F(\cdot)$: cumulative distribution function of X_i ;
3. $F_j(\cdot)$: cumulative distribution function of X_{ji} ;
4. $\bar{F}_j(\cdot)$: survival function of X_{ji} ;
5. δ_i : the indicator variable denoting the cause of failure of the i^{th} individual;
6. m : number of complete failures before the experiment is terminated;
7. $x_{i:m:n}$: i^{th} observed failure time, $i = 1, \dots, m$;
8. R_i : number of units removed at the time of the i^{th} failure with $R_i \geq 0$ and $\sum_{i=1}^m R_i + m = n$;
9. $C_1 - C_4$: The four major types of censoring schemes discussed later.
10. $\text{Weibull}(\alpha, \beta)$: Weibull distribution with shape parameter α and scale parameter β .

1.3.2 Description

We assume here that (X_{1i}, X_{2i}) , $i = 1, \dots, n$, are n independent and identically distributed Weibull random variables with parameters (α, β_1) and (α, β_2) respectively. Here, we shall assume equal shape parameters. $X_i = \min\{X_{1i}, X_{2i}\}$ and the observed sample is thus denoted:

$$(X_{1:m:n}, \delta_1, R_1), \dots, (X_{m:m:n}, \delta_m, R_m). \quad (1.3.1)$$

We also assume that all the causes of failure are known and act independently to each other. The density function of X_{ji} is given by

$$f_{X_{ji}}(x) = \frac{\alpha}{\beta_j^\alpha} x^{\alpha-1} \exp \left\{ - \left(\frac{x}{\beta_j} \right)^\alpha \right\}, \quad x > 0, \quad \alpha, \beta_j > 0, \quad j = 1, 2, \quad (1.3.2)$$

and the corresponding distribution function is given by

$$F_{X_{ji}}(x) = P(X_{ji} \leq x) = 1 - \exp \left\{ - \left(\frac{x}{\beta_j} \right)^\alpha \right\}, \quad j = 1, 2, \quad (1.3.3)$$

which implies

$$\bar{F}_{X_{ji}}(x) = P(X_{ji} > x) = \exp \left\{ - \left(\frac{x}{\beta_j} \right)^\alpha \right\}, \quad j = 1, 2. \quad (1.3.4)$$

Note here that $X_{i:m:n}$ is not the same as $X_{i:n}$, the i^{th} usual order statistic from a sample of size n , for $i \geq 2$. This can be easily seen since there is a possibility that $X_{i:n}$ may be censored by the time $X_{i:m:n}$ is observed. However, $X_{1:m:n} = X_{1:n}$ since no items have been censored prior to the first failure.

1.4 Algorithm for Generating Progressively Type II Censored Data under competing risks for Weibull Lifetimes

Since we now have the model, we can see how to generate a dataset from this model. Again, we assume $K = 2$. To generate a progressively Type II censored dataset under two competing causes of failure for Weibull lifetimes, we use the following algorithm:

1. Generate two *i.i.d* samples of size n for each cause of failure as follows:

$$\begin{aligned} (X_{11}, \dots, X_{1n}) &\stackrel{iid}{\sim} \text{Weibull}(\alpha, \beta_1) \quad \text{and} \\ (X_{21}, \dots, X_{2n}) &\stackrel{iid}{\sim} \text{Weibull}(\alpha, \beta_2). \end{aligned}$$

2. For each $i = 1, \dots, n$,

$$\begin{aligned} &\text{if } X_{1i} \leq X_{2i}, && \text{set } \delta_i = 1 \text{ and } X_i = X_{1i}, \\ &\text{else, if } X_{1i} > X_{2i}, && \text{set } \delta_i = 2 \text{ and } X_i = X_{2i}. \end{aligned}$$

3. We now have the data as follows.

$$(X_1, \delta_1), \dots, (X_n, \delta_n). \quad (1.4.1)$$

This is now ordered irrespective of the cause of failure, but without losing track of the corresponding cause of failure. Thus we now have the *n usual order statistics*

$$(X_{1:n}, \delta_1^*), \dots, (X_{n:n}, \delta_n^*). \quad (1.4.2)$$

We introduce the notation $*$ since δ_i^* 's are concomitants of the order statistics. Thus, δ_i may not be equal to δ_i^* .

4. We now have to progressively censor this sample. First we fix $m \leq n$ and the censoring scheme (R_1, \dots, R_m) so that $\sum_{i=1}^m R_i = n - m$. That is, we will observe m failures and the rest are progressively censored.
5. To begin with $X_{1:n}$ is observed as the first failure and thus $X_{1:m:n} = X_{1:n}$. Then R_1 of the remaining surviving $n - 1$ units are randomly selected and dropped from the study. At this stage the next smallest lifetime of the remaining units is observed as the second failure, i.e., $(X_{2:m:n})$, so that R_2 of the remaining $n - R_1 - 2$ units are randomly censored from the study. This process is continued until, at the time of the m^{th} observed failure $(X_{m:m:n})$, the remaining $R_m = n - R_1 - R_2 - \dots - R_{m-1} - m$ units are all censored.

This leads to the progressively censored competing risk sample given by (1.3.1). They are also the *progressively Type II censored order statistics* from the distribution $F_{X_i}(x_i)$.¹ Note that, we have dropped the $*$'s for simplicity but it is understood that

¹Derived later in the next chapter.

they are still concomitants of the original lifetimes given in (1.4.1). The function to generate the data (*datagen*), written in **R** (R Development Core Team 2004), is given in Appendix A.1. After the data is progressively censored, we calculate n_1 and n_2 . If either of them is equal to zero, then we drop the dataset from consideration and generate a new one. This is because, we need n_1 and n_2 to be strictly greater than zero to be able to solve for MLEs (as we see in the next chapter).

Chapter 2

Maximum Likelihood Estimation under Competing Risks

In this chapter we construct the general likelihood function for a competing risk model under progressive Type II censoring. We then use this to find the likelihood for weibull lifetimes with equal shape parameters. The required components of the likelihood is derived. We present the likelihood equations to be solved to get the MLEs of the parameters. Finally we present an algorithm to derive the MLEs.

2.1 Maximum Likelihood Estimation

First, we shall look at likelihood function for a progressively Type II censored random sample from a continuous distribution and then proceed to incorporate the competing risk factor into it. We now note that the observed lifetimes under progressive Type II censoring are denoted by

$$(X_{1:m:n}, R_1), \dots, (X_{m:m:n}, R_m).$$

The joint probability density function of all the above m progressive Type II censored order statistics is as follows (see Balakrishnan and Aggarwala (2000), p. 8):

$$f_{X_{1:m:n}, \dots, X_{m:m:n}}(x_1, \dots, x_m) = c \prod_{i=1}^m f(x_i)[1 - F(x_i)]^{R_i}, \quad 0 < x_1 < \dots < x_m < \infty, \quad (2.1.1)$$

where $c = n(n - R_1 - 1)(n - R_1 - R_2 - 2) \dots (n - R_1 - R_2 - \dots - R_{m-1} - m + 1)$. This is also the likelihood function.

Notice that immediately preceding the first observed failure, n items are still on test; immediately preceding the second observed failure, $n - R_1 - 1$ items are still on test, and so on; immediately preceding the m^{th} observed failure, $n - R_1 - R_2 - \dots - R_{m-1} - m + 1$ are still on test. Thus the constant c , in addition to being simply the normalizing constant to make the joint pdf integrate to one, is also the number of ways in which the m progressive Type II censored order statistics may occur if the observed failure times are x_1, \dots, x_m .

Now, if we take competing risks into consideration, as seen in the previous section, the observed sample is denoted by

$$(X_{1:m:n}, \delta_1, R_1), \dots, (X_{m:m:n}, \delta_m, R_m).$$

Here, we know that $X_i = \min\{X_{1i}, X_{2i}\}$. Let us denote $\pi_1 = P(X_{1i} \leq X_{2i})$ and $\pi_2 = P(X_{2i} < X_{1i})$; that is, π_1 is the relative risk due to Cause 1 and π_2 is the relative risk due to Cause 2. Then the likelihood of the m progressively Type II censored order statistics under these two competing risks becomes

$$f_{X_{1:m:n}, \dots, X_{m:m:n}}(x_1, \dots, x_m) = c \prod_{i=1}^m f_{X_i}(x_i)[1 - F_{X_i}(x_i)]^{R_i} (\pi_1)^{n_1} (\pi_2)^{n_2}, \quad 0 < x_1 < \dots < x_m < \infty \quad (2.1.2)$$

where n_1 and n_2 are the number of failures due to Cause 1 and Cause 2 respectively, and $X_i = \min\{X_{1i}, X_{2i}\}$. This is because each of the m observed failures could have occurred due to Cause 1 with probability π_1 and Cause 2 with probability π_2 .

2.2 MLE for Weibull Distribution with Equal Shape Parameters

In this section, we shall find the likelihood discussed in the previous chapter assuming Weibull lifetimes. First we need to find $F_{X_i}(x_i)$, $f_{X_i}(x_i)$, π_1 and π_2 . The distribution function $F_{X_i}(x_i)$ can be found as follows:

$$\begin{aligned}
 F_{X_i}(x_i) = P(X_i \leq x_i) &= P(\min\{X_{1i}, X_{2i}\} \leq x_i) \\
 &= 1 - P(\min\{X_{1i}, X_{2i}\} > x_i) \\
 &= 1 - P(X_{1i} > x_i, X_{2i} > x_i) \\
 &= 1 - P(X_{1i} > x_i)P(X_{2i} > x_i) \\
 &= 1 - \exp\left\{-\left(\frac{x_i}{\beta_1}\right)^\alpha\right\} \exp\left\{-\left(\frac{x_i}{\beta_2}\right)^\alpha\right\} \\
 &\quad \text{(follows easily from (1.3.4))} \\
 &= 1 - \exp\left\{-x_i^\alpha \left\{\frac{1}{\beta_1^\alpha} + \frac{1}{\beta_2^\alpha}\right\}\right\}. \tag{2.2.1}
 \end{aligned}$$

Clearly, from (1.3.3), this is again a Weibull distribution with shape α and scale $\left(\frac{\beta_1^\alpha \beta_2^\alpha}{\beta_1^\alpha + \beta_2^\alpha}\right)^{\frac{1}{\alpha}}$. Thus, the probability density function of the observed data (obtained by differentiating the above cdf) is given by

$$f(x_i) = \alpha \left(\frac{1}{\beta_1^\alpha} + \frac{1}{\beta_2^\alpha}\right) x_i^{\alpha-1} \exp\left\{-x_i^\alpha \left(\frac{1}{\beta_1^\alpha} + \frac{1}{\beta_2^\alpha}\right)\right\}, \quad x_i > 0. \tag{2.2.2}$$

2.2.1 Relative Risk

We now have to find π_1 . This can be found using integration of the joint pdf of X_{1i} and X_{2i} . Since X_{1i} and X_{2i} are independent, their joint pdf is equal to the product

of the marginal pdf of X_{1i} and X_{2i} . Thus, from (1.3.2) we have

$$\begin{aligned}
\pi_1 &= P(X_{1i} \leq X_{2i}) \\
&= \int_{y=0}^{\infty} \int_{x=0}^y f_{X_{1i}, X_{2i}}(x, y) dx dy \\
&= \frac{\alpha}{\beta_2^\alpha} \int_{y=0}^{\infty} y^{\alpha-1} \exp \left\{ - \left(\frac{y}{\beta_2} \right)^\alpha \right\} \frac{\alpha}{\beta_1^\alpha} \int_{x=0}^y x^{\alpha-1} \exp \left\{ - \left(\frac{x}{\beta_1} \right)^\alpha \right\} dx dy \\
&= \frac{\beta_2^\alpha}{\beta_1^\alpha + \beta_2^\alpha}.
\end{aligned} \tag{2.2.3}$$

This implies

$$\pi_2 = P(X_{2i} < X_{1i}) = 1 - \pi_1 = \frac{\beta_1^\alpha}{\beta_1^\alpha + \beta_2^\alpha} \tag{2.2.4}$$

Thus, likelihood equation given in (2.1.2) now becomes

$$\begin{aligned}
L &\propto c \alpha^m \left(\frac{1}{\beta_1^\alpha} + \frac{1}{\beta_2^\alpha} \right)^m \left(\prod_{i=1}^m x_i^{\alpha-1} \right) \exp \left\{ - \left(\frac{1}{\beta_1^\alpha} + \frac{1}{\beta_2^\alpha} \right) \sum_{i=1}^m (R_i + 1) x_i^\alpha \right\} \\
&\quad \times \left(\frac{\beta_2^\alpha}{\beta_1^\alpha + \beta_2^\alpha} \right)^{n_1} \left(\frac{\beta_1^\alpha}{\beta_1^\alpha + \beta_2^\alpha} \right)^{n_2} \\
&\propto c \frac{\alpha^m}{\beta_1^{\alpha n_1} \beta_2^{\alpha n_2}} \left(\prod_{i=1}^m x_i^{\alpha-1} \right) \exp \left\{ - \left(\frac{1}{\beta_1^\alpha} + \frac{1}{\beta_2^\alpha} \right) \sum_{i=1}^m (R_i + 1) x_i^\alpha \right\}.
\end{aligned} \tag{2.2.5}$$

2.2.2 Finding the MLEs

Now that we have the likelihood function, we can obtain the maximum likelihood estimates of the three parameters α , β_1 and β_2 . Taking the logarithm of (2.2.5), we obtain

$$\begin{aligned}
\ln L &= \ln c + m \ln \alpha - \alpha n_1 \ln \beta_1 - \alpha n_2 \ln \beta_2 \\
&\quad + (\alpha - 1) \sum_{i=1}^m \ln x_i - \left\{ \frac{\beta_1^\alpha + \beta_2^\alpha}{\beta_1^\alpha \beta_2^\alpha} \right\} \sum_{i=1}^m (R_i + 1) x_i^\alpha.
\end{aligned} \tag{2.2.6}$$

Differentiating¹ (2.2.6) with respect to α , β_1 and β_2 and simplifying the expressions, we get

$$\begin{aligned} \frac{\partial \ln L}{\partial \alpha} &= \frac{m}{\alpha} - n_1 \ln \beta_1 - n_2 \ln \beta_2 + (\beta_1^{-\alpha} \ln \beta_1 + \beta_2^{-\alpha} \ln \beta_2) \sum_{i=1}^m (R_i + 1) x_i^\alpha \\ &\quad + \sum_{i=1}^m \ln x_i - (\beta_1^{-\alpha} + \beta_2^{-\alpha}) \sum_{i=1}^m (R_i + 1) x_i^\alpha \ln x_i, \end{aligned} \quad (2.2.7)$$

$$\frac{\partial \ln L}{\partial \beta_1} = -\frac{n_1 \alpha}{\beta_1} + \frac{\alpha}{\beta_1^{\alpha+1}} \sum_{i=1}^m (R_i + 1) x_i^\alpha \quad (2.2.8)$$

and

$$\frac{\partial \ln L}{\partial \beta_2} = -\frac{n_2 \alpha}{\beta_2} + \frac{\alpha}{\beta_2^{\alpha+1}} \sum_{i=1}^m (R_i + 1) x_i^\alpha. \quad (2.2.9)$$

The maximum likelihood estimates of α , β_1 and β_2 , namely $\hat{\alpha}$, $\hat{\beta}_1$ and $\hat{\beta}_2$, can be obtained by equating the above three equations (2.2.7), (2.2.8) and (2.2.9) to 0 and then solving them iteratively. This yields the following three equations:

$$\frac{m}{\alpha} + \sum_{i=1}^m \ln x_i - (\beta_1^{-\alpha} + \beta_2^{-\alpha}) \sum_{i=1}^m (R_i + 1) x_i^\alpha \ln x_i = 0, \quad (2.2.10)$$

$$\hat{\beta}_1 = \left(\frac{\sum_{i=1}^m (R_i + 1) x_i^\alpha}{n_1} \right)^{\frac{1}{\alpha}} \quad (2.2.11)$$

and

$$\hat{\beta}_2 = \left(\frac{\sum_{i=1}^m (R_i + 1) x_i^\alpha}{n_2} \right)^{\frac{1}{\alpha}}. \quad (2.2.12)$$

We can see clearly that explicit expressions can be derived for $\hat{\beta}_1$ and $\hat{\beta}_2$. From equations (2.2.11) and (2.2.12), it is evident that for the MLEs of β_1 and β_2 to exist, n_1 and n_2 have to be greater than 0, respectively. Thus we can say that $\hat{\beta}_1$ and $\hat{\beta}_2$ are the conditional MLEs.

¹Note: The first order derivatives and the second order derivatives that we are about to see in the next section were verified using Maple (2003).

2.2.3 Algorithm for finding the MLEs

From the three equations (2.2.10), (2.2.11) and (2.2.12) we can clearly see that they can be solved only iteratively. Explicit expressions for $\hat{\beta}_1$ and $\hat{\beta}_2$ makes our job easier. The algorithm is as follows:

1. Give an initial estimate for α say α_0 , which can be any reasonable value chosen from previous experience.
2. Using α_0 , find $\hat{\beta}_1^{(1)}$ and $\hat{\beta}_2^{(1)}$ from equations (2.2.11) and (2.2.12) respectively.
3. Use these two values and solve (2.2.10) for α . Again, this equation cannot be solved directly. Another iterative method has to be employed to solve for the value of α from this. One way of solving this equation is to minimize the square of that function using standard optimization packages (for example, function ‘*optimize*’ can be used in R). This should give the solution for α say $\hat{\alpha}^{(1)}$. Use this updated value of $\hat{\alpha}$ and proceed to Step 2 to find $\hat{\beta}_1^{(2)}$ and $\hat{\beta}_2^{(2)}$.

This process is to be repeated until a desired level of accuracy is reached. The function (*mleweibull*) to find the MLEs, written in **R**, has been provided in Appendix A.1.

Chapter 3

Confidence Intervals

In this chapter, we look at three different types of confidence intervals for the parameters. The first method is the asymptotic confidence interval based on the asymptotic distributions of the MLEs. The next two methods are the bootstrap percentile method and the bootstrap- t method. We shall look at these methods in detail. These three methods will be compared using a simulation study in the next chapter.

3.1 Asymptotic Confidence Intervals

In this section, we shall construct asymptotic confidence intervals based on the Fisher information matrix. Let $\mathbf{X} = (X_1, \dots, X_n)$ be a random sample, and let $f(\mathbf{X}|\boldsymbol{\theta})$ denote the probability density function of \mathbf{X} from a distribution with parametric vector $\boldsymbol{\theta} = (\theta_1, \dots, \theta_k)$. Then the Fisher information matrix $I_n(\boldsymbol{\theta})$ of sample size n is based on the expected values of the second order partial derivatives, and is given by

$$I_n(\boldsymbol{\theta})_{i,j} = -E \left[\frac{\partial^2 \ln f(\mathbf{X}|\boldsymbol{\theta})}{\partial \theta_i \partial \theta_j} \right]. \quad (3.1.1)$$

Strictly, this definition corresponds to the *expected* Fisher information. If taking the expectation is not possible or very complicated, we can obtain a data-dependent quantity that is called the *observed* Fisher information.

For this model, the second order partial derivatives (obtained by differentiating equations (2.2.7), (2.2.8) and (2.2.9) again with respect to the three parameters α, β_1

and β_2 respectively) are presented below:

$$\begin{aligned} \frac{\partial^2 \log L}{\partial \alpha^2} &= -\frac{m}{\alpha^2} - (\beta_1^{-\alpha} + \beta_2^{-\alpha}) \sum_{i=1}^m (R_i + 1) x_i^\alpha \log^2 x_i \\ &\quad + 2 \times (\beta_1^{-\alpha} \log \beta_1 + \beta_2^{-\alpha} \log \beta_2) \sum_{i=1}^m (R_i + 1) x_i^\alpha \log x_i \\ &\quad - (\beta_1^{-\alpha} \log^2 \beta_1 + \beta_2^{-\alpha} \log^2 \beta_2) \sum_{i=1}^m (R_i + 1) x_i^\alpha \end{aligned} \quad (3.1.2)$$

$$\begin{aligned} \frac{\partial^2 \log L}{\partial \alpha \partial \beta_1} &= \frac{\partial^2 \log L}{\partial \beta_1 \partial \alpha} = -\frac{n_1}{\beta_1} + \frac{\alpha}{\beta_1^{\alpha+1}} \sum_{i=1}^m (R_i + 1) x_i^\alpha \log x_i \\ &\quad + \left(\frac{1 - \alpha \log \beta_1}{\beta_1^{\alpha+1}} \right) \sum_{i=1}^m (R_i + 1) x_i^\alpha \end{aligned} \quad (3.1.3)$$

$$\begin{aligned} \frac{\partial^2 \log L}{\partial \alpha \partial \beta_2} &= \frac{\partial^2 \log L}{\partial \beta_2 \partial \alpha} = -\frac{n_2}{\beta_2} + \frac{\alpha}{\beta_2^{\alpha+1}} \sum_{i=1}^m (R_i + 1) x_i^\alpha \log x_i \\ &\quad + \left(\frac{1 - \alpha \log \beta_2}{\beta_2^{\alpha+1}} \right) \sum_{i=1}^m (R_i + 1) x_i^\alpha \end{aligned} \quad (3.1.4)$$

$$\frac{\partial^2 \log L}{\partial \beta_1^2} = \frac{\alpha n_1}{\beta_1^2} - \frac{\alpha(\alpha + 1)}{\beta_1^{\alpha+2}} \sum_{i=1}^m (R_i + 1) x_i^\alpha \quad (3.1.5)$$

$$\frac{\partial^2 \log L}{\partial \beta_2^2} = \frac{\alpha n_2}{\beta_2^2} - \frac{\alpha(\alpha + 1)}{\beta_2^{\alpha+2}} \sum_{i=1}^m (R_i + 1) x_i^\alpha \quad (3.1.6)$$

$$\frac{\partial^2 \log L}{\partial \beta_1 \partial \beta_2} = \frac{\partial^2 \log L}{\partial \beta_2 \partial \beta_1} = 0. \quad (3.1.7)$$

We use here the observed Fisher information matrix, since obtaining the expected Fisher information matrix for this model is not feasible. The observed Fisher information matrix is given by the negative of matrix of the second order derivatives evaluated at the MLEs as follows:

$$I_{obs} = - \begin{pmatrix} \frac{\partial^2 \log L}{\partial \alpha^2} & \frac{\partial^2 \log L}{\partial \alpha \partial \beta_1} & \frac{\partial^2 \log L}{\partial \alpha \partial \beta_2} \\ \frac{\partial^2 \log L}{\partial \beta_1 \partial \alpha} & \frac{\partial^2 \log L}{\partial \beta_1^2} & \frac{\partial^2 \log L}{\partial \beta_1 \partial \beta_2} \\ \frac{\partial^2 \log L}{\partial \beta_2 \partial \alpha} & \frac{\partial^2 \log L}{\partial \beta_2 \partial \beta_1} & \frac{\partial^2 \log L}{\partial \beta_2^2} \end{pmatrix}_{\alpha=\hat{\alpha}, \beta_1=\hat{\beta}_1, \beta_2=\hat{\beta}_2}. \quad (3.1.8)$$

The inverse of the observed Fisher information matrix gives the variance-covariance matrix of the MLEs which can be used to construct asymptotic confidence intervals for the parameters. We shall make use of the asymptotic normality property of the MLEs to obtain this confidence interval.

Thus if

$$V(\alpha, \beta_1, \beta_2) = I_{obs}^{-1} = ((v_{ij}(\alpha, \beta_1, \beta_2))), \quad i, j = 1, 2, 3$$

is the variance-covariance matrix, the $100(1 - 2\xi)\%$ confidence intervals for α, β_1, β_2 can be given by,

$$\begin{aligned} \hat{\alpha} &\pm z_\xi \sqrt{v_{11}} \\ \hat{\beta}_1 &\pm z_\xi \sqrt{v_{22}} \\ \hat{\beta}_2 &\pm z_\xi \sqrt{v_{33}}, \end{aligned} \tag{3.1.9}$$

where z_ξ is the upper ξ^{th} percentile point of the standard normal distribution. As the name itself suggests, this holds good only when n is large. The asymptotic confidence intervals cannot be relied upon for small sample sizes. This will be shown in the next section with the help of some simulation work.

3.2 Bootstrap- p Confidence Intervals

In this section, we construct bootstrap percentile confidence intervals. This method was first proposed by Efron (1982). We shall illustrate the procedure for the parameter α . Intervals for the parameters β_1 and β_2 can be obtained in the same manner.

To obtain the percentile bootstrap (bootstrap- p) confidence interval for α , we use the following algorithm:

- (1) Determine $\hat{\alpha}$, $\hat{\beta}_1$, and $\hat{\beta}_2$ from the sample $\{(x_{1:m:n}, \delta_1, R_1), \dots, (x_{m:m:n}, \delta_m, R_m)\}$ as discussed in Chapter 2.
- (2) Generate a progressively Type II censored competing risks data set using the MLEs $\hat{\alpha}$, $\hat{\beta}_1$ and $\hat{\beta}_2$. For this data, compute the bootstrap estimates of α , β_1 and β_2 , namely $\hat{\alpha}^*$, $\hat{\beta}_1^*$ and $\hat{\beta}_2^*$.

- (3) Repeat Step 2 R times. This gives R estimates for each of the parameters α , β_1 and β_2 .
- (4) Then, a $100(1 - 2\xi)\%$ confidence interval for α is given by

$$\left(\hat{\alpha}_{(R+1)\xi}^*, \hat{\alpha}_{(R+1)(1-\xi)}^*\right). \quad (3.2.1)$$

That is, we sort the R $\hat{\alpha}^*$'s in ascending order and take the $(R + 1)\xi^{\text{th}}$ and $(R + 1)(1 - \xi)^{\text{th}}$ values. In other words, we take the ξ^{th} and $(1 - \xi)^{\text{th}}$ percentile point of the distribution of $\hat{\alpha}^*$. Percentile bootstrap confidence intervals for β_1 and β_2 are obtained in an analogous manner.

3.3 Studentized or Bootstrap- t Confidence Intervals

In this section, we shall construct Bootstrap- t confidence intervals for the parameters α , β_1 and β_2 . This method was first proposed by Hall (1988). Again, we shall construct confidence interval for the parameter α and the intervals for other parameters can be obtained in a similar manner. The algorithm is as follows:

- (1) Determine $\hat{\alpha}$, $\hat{\beta}_1$, and $\hat{\beta}_2$ from the sample $\{(x_{1:m:n}, \delta_1, R_1), \dots, (x_{m:m:n}, \delta_m, R_m)\}$ as discussed in Chapter 2. Also calculate the variance of $\hat{\alpha}$, say v using the Fisher information as discussed in Section 3.1.
- (2) Generate a progressively Type II censored competing risks data set using the MLEs $\hat{\alpha}$, $\hat{\beta}_1$ and $\hat{\beta}_2$. For this data, compute the bootstrap estimates of α , β_1 and β_2 namely $\hat{\alpha}^*$, $\hat{\beta}_1^*$ and $\hat{\beta}_2^*$. For this sample again, calculate the variance estimate of $\hat{\alpha}^*$, say v^* , again using Section 3.1.
- (3) Now we use the form of the normal approximation confidence limit, but replace the $N(0, 1)$ approximation for $Z = (\hat{\alpha} - \alpha)/\sqrt{v}$ by a bootstrap approximation.

Thus we calculate the bootstrap statistic

$$z^* = \frac{\hat{\alpha}^* - \hat{\alpha}}{\sqrt{v^*}}$$

- (4) Repeat Steps 2 and 3 R times. This gives R bootstrap versions of z^* .
- (5) Order the R simulated values of z^* and the p^{th} quantile of Z is estimated by the $(R + 1)p^{\text{th}}$ of these ordered z^* 's.

Then, a $100(1 - 2\xi)\%$ confidence interval for α is given by

$$\left(\hat{\alpha} - \sqrt{v} z_{((R+1)(1-\xi))}^*, \hat{\alpha} - \sqrt{v} z_{((R+1)\xi)}^* \right). \quad (3.3.1)$$

These we refer to as *studentized bootstrap confidence limits*. They are also known as bootstrap- t limits, by analogy with the Student- t confidence limits for the mean of a normal distribution, to which they are equal under infinite simulation in that problem. A more detailed description of various types of parametric bootstrap confidence intervals can be found in the book of Davison and Hinkley (1997).

Chapter 4

Simulation Study

A simulation study was conducted to compare the coverage probabilities and the coverage lengths of the three types of confidence intervals discussed in the previous chapter. The results for different values of n , m and different censoring schemes are presented in tables below.

The *coverage probability* for an interval estimator $I(\mathbf{X})$ for a parameter θ is the probability that the random interval contains the true parameter value, i.e.

$$P_{\theta}(\theta \in I(\mathbf{X})).$$

We shall first see how these are calculated for the three confidence intervals discussed in Chapter 3 before we take a look at the results.

4.1 Probability Coverages and Average Coverage Lengths (ACLs)

The concept of computing probability coverages and *average coverage lengths* is quite simple. We present here the procedure for finding the probability coverage and average coverage lengths for asymptotic confidence intervals. Probability coverages and average lengths for the other two types of confidence intervals are computed in an analogous manner.

1. We first generate N datasets from a specified set of parameters. In the tables given below, we have chosen to generate $N=1000$ datasets from $\alpha = 2$, $\beta_1 = 3$, and $\beta_2 = 5$.
2. Then we compute the asymptotic confidence intervals for each of the parameters as discussed in Section 3.1.
3. Once we have N confidence intervals for each of the parameters, we then calculate the proportion of these confidence intervals that contain the true parameter value. For example, if we wish to find the probability coverage for the parameter α above, we shall compute the proportion of confidence intervals that contain the true parameter value of 2.
4. To compute average coverage lengths, we calculate the lengths for each of the N confidence intervals and then take the average of those. So, for the asymptotic confidence coverage lengths, say for parameter α , we just compute the average of $2 \times z_\xi \sqrt{v_{11}}$, where v_{11} is the asymptotic variance of $\hat{\alpha}$ and z_ξ is the upper ξ^{th} percentile point of the standard normal distribution as seen in the previous chapter.

4.2 Performance and Comparison of Confidence Intervals

In this section, we shall discuss the performance of the three confidence intervals first and then compare them. The results were obtained for the following four major censoring schemes:

1. C_1 - Drop some units after each failure (not necessarily all same).
2. C_2 - Drop units equally (as far as possible) after the first and last failure.
3. C_3 - Drop all the $n - m$ units after the first failure.

4. C_4 - Drop all the $n - m$ units after the last failure. This is the conventional Type -II censoring.

We present the results for $n = 200, 100, 50$ and 30 . For $n = 30$, we have used two different cases, one where $m = 15$ and the other where $m = 10$. The tables are presented at the end of this Chapter. We shall analyze and compare the performance of these confidence intervals first based on probability coverages and coverage lengths and then based on the bias and MSE.

4.2.1 Comparisons based on Probability Coverages and ACLs

The following were observed from the Tables 4.1 – 4.5:

1. We can clearly see from Tables 4.1 and 4.2 that the probability coverages for parameters β_1 and β_2 are pretty close to the nominal level of 95%. But as we reduce the sample size (Tables 4.3- 4.5), as expected, we can see that the coverages are not even satisfactory.
2. The intervals for the common shape parameter α perform well even for small sample sizes for asymptotic and bootstrap- t confidence intervals. This is natural since we use the same shape parameter for both causes of failure. Had we used different shape parameters, this would not necessarily have been the case.
3. Surprisingly, the probability coverages for the shape parameter α reduce with sample size for the percentile bootstrap confidence intervals.
4. We also observe that the average lengths of the confidence intervals increases as we decrease m . This is clearly evident from Tables 4.4 and 4.5. In Table 4.4 we have considered $n = 30$ and $m = 15$ and in Table 4.5 we have considered $n = 30$ and $m = 10$. We can clearly see that even for the same sample size n , if we reduce m the performance of the intervals goes down dramatically. This is evident very clearly, not so much in terms of the probability coverages but in

terms of the coverage lengths. This is again true for all three types of confidence intervals.

5. We also note that the confidence intervals for β_2 perform not as good as the confidence intervals for β_1 for asymptotic and bootstrap- t confidence intervals. This is because we simulated the data using $\beta_2 = 5$ and $\beta_1 = 3$. Surprisingly again, for bootstrap- p , intervals for β_2 seem to perform better than for α and β_1 for all sample sizes.
6. The confidence intervals seem to perform better for the censoring scheme C_3 and worst for censoring scheme C_4 . This is reflected in the probability coverages and the average coverage lengths for all sample sizes.
7. The performance of all three confidence intervals are comparable for large sample sizes when compared using probability coverages. When it comes to small samples, the bootstrap- t confidence intervals seem to perform better than the other two. We observe from Tables 4.3-4.5 that though the confidence intervals for the parameters α and β_2 are comparable between asymptotic and bootstrap- t methods, the latter always performs better for parameter β_1 .
8. When we compare using coverage lengths, however, the asymptotic confidence intervals perform better than the other two for all sample sizes and for all censoring schemes.

4.2.2 Comparison based on Asymptotic Bias, MSE and Others

The bias and the MSE provide a measure of how close the estimator is to the true value of the parameter. In this section, we shall take a look at the bias, MSE and the mean and variances of the estimators, the average of the asymptotic variance and the relative risk due to Cause 1. The results are presented in Tables 4.6 - 4.10.

These results were also obtained using 1000 replications. The following points were observed:

1. For large sample sizes ($n = 200, 100$) we can see that the bias is relatively close to zero. But as the sample size reduces the bias moves away from zero. Also, the bias increases as m is decreased. This is quite evident from Tables 4.9 and 4.10 where we have presented the results for the same value of $n = 30$ but for different number of failures, i.e., $m = 15$ and $m = 10$. The same is true for the MSE for all sample sizes and for all censoring schemes. This should explain why the probability coverages are very good for large sample sizes and not so good for small sample sizes.
2. We observe a negative bias for the parameter β_1 for all sample sizes and for all censoring schemes. This means that we are underestimating β_1 . However, this is not true for parameters α and β_2 .
3. The mean and variances of the estimates of the parameters over 1000 simulations have also been computed. We can easily see that, for large sample sizes, the means of the estimates of the parameters α , β_1 and β_2 are very close to the true values i.e., 2, 3 and 5 respectively. However, this is not the case for smaller sample sizes.
4. Finally, we take the variances of the MLEs of α , β_1 and β_2 and compare them with the average asymptotic variance. Here again, the variances are comparable for large values of n and m but not quite close for smaller sample sizes.
5. Here again, a closer look at the results for the four censoring schemes suggests that the censoring scheme C_3 might be better than the other three.

4.3 The Results

The following Tables 4.1 – 4.5 present the probability coverages and average coverage lengths (ACL) for $100(1 - 0.05)\%$ confidence intervals. The data sets have been simulated using $\alpha = 2$, $\beta_1 = 3$, and $\beta_2 = 5$. The bias and mean square errors of the estimators for different sample sizes and censoring schemes are also presented in Tables 4.6 – 4.10. The **R** functions for computing Tables 4.1 – 4.10 are presented in Appendix A.1.

Table 4.1: Probability Coverages and ACLs for: $n = 200, m = 80$

Censoring Scheme	Methods	Parameters	Coverage Lengths	Coverage Probabilities
$2, \dots, 2, 1, \dots, 1$	Asymptotic	α	0.66785	0.953
		β_1	0.83090	0.933
		β_2	2.60607	0.927
	Bootstrap- p	α	0.69228	0.922
		β_1	0.83707	0.933
		β_2	2.81785	0.925
	Bootstrap- t	α	0.66784	0.951
		β_1	0.87320	0.949
		β_2	2.80637	0.921
$60, 0, \dots, 0, 60$	Asymptotic	α	0.79204	0.958
		β_1	0.86830	0.940
		β_2	2.78715	0.930
	Bootstrap- p	α	0.82466	0.956
		β_1	0.86670	0.944
		β_2	2.99292	0.956
	Bootstrap- t	α	0.79359	0.962
		β_1	0.90685	0.955
		β_2	2.99292	0.939
$120, 0, \dots, 0$	Asymptotic	α	0.65955	0.945
		β_1	0.76324	0.938
		β_2	2.29992	0.926
	Bootstrap- p	α	0.67797	0.942
		β_1	0.77014	0.953
		β_2	2.49798	0.943
	Bootstrap- t	α	0.65798	0.955
		β_1	0.78595	0.953
		β_2	2.41769	0.931
$0, \dots, 0, 120$	Asymptotic	α	0.84555	0.936
		β_1	1.02354	0.922
		β_2	3.16026	0.921
	Bootstrap- p	α	0.88207	0.929
		β_1	1.03902	0.939
		β_2	3.33657	0.943
	Bootstrap- t	α	0.84231	0.936
		β_1	1.10442	0.938
		β_2	3.41220	0.940

Table 4.2: Probability Coverages and ACLs for: $n = 100, m = 40$

Censoring Scheme	Methods	Parameters	Coverage Lengths	Coverage Probabilities
2, ..., 2, 1, ..., 1	Asymptotic	α	0.95939	0.950
		β_1	1.16251	0.920
		β_2	3.99758	0.913
	Bootstrap- p	α	1.03483	0.902
		β_1	1.18249	0.916
		β_2	4.31628	0.940
	Bootstrap- t	α	0.96435	0.948
		β_1	1.29083	0.943
		β_2	4.19841	0.928
30, 0, ..., 0, 30	Asymptotic	α	1.12324	0.948
		β_1	1.23046	0.917
		β_2	4.13918	0.903
	Bootstrap- p	α	1.20846	0.916
		β_1	1.24226	0.917
		β_2	4.71525	0.938
	Bootstrap- t	α	1.12274	0.958
		β_1	1.35772	0.938
		β_2	4.59081	0.915
60, 0, ..., 0	Asymptotic	α	0.92007	0.963
		β_1	1.08316	0.948
		β_2	3.49499	0.928
	Bootstrap- p	α	0.97231	0.943
		β_1	1.08926	0.934
		β_2	3.98826	0.947
	Bootstrap- t	α	0.91661	0.957
		β_1	1.13464	0.936
		β_2	3.66439	0.920
0, ..., 0, 60	Asymptotic	α	1.23194	0.948
		β_1	1.42446	0.905
		β_2	4.52260	0.906
	Bootstrap- p	α	1.32796	0.913
		β_1	1.46778	0.916
		β_2	5.19191	0.947
	Bootstrap- t	α	1.20982	0.945
		β_1	1.65245	0.948
		β_2	5.33766	0.941

Table 4.3: Probability Coverages and ACLs for: $n = 50, m = 20$

Censoring Scheme	Methods	Parameters	Coverage Lengths	Coverage Probabilities
2, ..., 2, 1, ..., 1	Asymptotic	α	1.41045	0.949
		β_1	1.63647	0.900
		β_2	5.94362	0.880
	Bootstrap- p	α	1.64209	0.881
		β_1	1.64912	0.896
		β_2	6.82765	0.939
	Bootstrap- t	α	1.41253	0.949
		β_1	1.95292	0.929
		β_2	7.51082	0.871
15, 0, ..., 0, 15	Asymptotic	α	1.64607	0.946
		β_1	1.69688	0.889
		β_2	6.19892	0.882
	Bootstrap- p	α	1.87051	0.888
		β_1	1.77429	0.910
		β_2	7.42411	0.934
	Bootstrap- t	α	1.61129	0.955
		β_1	2.10801	0.946
		β_2	8.18001	0.877
30, 0, ..., 0	Asymptotic	α	1.31926	0.947
		β_1	1.48714	0.918
		β_2	5.36563	0.906
	Bootstrap- p	α	1.42579	0.911
		β_1	1.58030	0.937
		β_2	6.32928	0.933
	Bootstrap- t	α	1.28351	0.956
		β_1	1.70434	0.941
		β_2	6.42113	0.866
0, ..., 0, 30	Asymptotic	α	1.82845	0.955
		β_1	1.97489	0.883
		β_2	7.08252	0.865
	Bootstrap- p	α	2.13484	0.888
		β_1	2.12633	0.900
		β_2	8.55059	0.922
	Bootstrap- t	α	1.76260	0.947
		β_1	2.70017	0.944
		β_2	10.31887	0.870

Table 4.4: Probability Coverages and ACLs for: $n = 30, m = 15$

Censoring Scheme	Methods	Parameters	Coverage Lengths	Coverage Probabilities
1, ..., 1	Asymptotic	α	1.76345	0.937
		β_1	1.85314	0.889
		β_2	7.04248	0.854
	Bootstrap- p	α	2.18543	0.855
		β_1	1.90484	0.880
		β_2	7.27211	0.923
	Bootstrap- t	α	1.75464	0.947
		β_1	2.35553	0.924
		β_2	10.10409	0.849
8, 0, ..., 0, 7	Asymptotic	α	1.91445	0.951
		β_1	1.84573	0.896
		β_2	7.15659	0.859
	Bootstrap- p	α	2.28819	0.882
		β_1	1.92499	0.904
		β_2	7.53835	0.926
	Bootstrap- t	α	1.86179	0.951
		β_1	2.34496	0.930
		β_2	9.96274	0.858
15, 0, ..., 0	Asymptotic	α	1.57852	0.959
		β_1	1.74166	0.922
		β_2	6.26151	0.900
	Bootstrap- p	α	1.83311	0.872
		β_1	1.79699	0.904
		β_2	6.38686	0.951
	Bootstrap- t	α	1.57142	0.954
		β_1	1.99154	0.919
		β_2	7.40166	0.859
0, ..., 0, 15	Asymptotic	α	2.11672	0.974
		β_1	2.06551	0.874
		β_2	8.02907	0.842
	Bootstrap- p	α	2.66759	0.850
		β_1	2.14919	0.895
		β_2	8.50245	0.913
	Bootstrap- t	α	2.07017	0.944
		β_1	2.82347	0.932
		β_2	12.27736	0.868

Table 4.5: Probability Coverages and ACLs for: $n = 30, m = 10$

Censoring Scheme	Methods	Parameters	Coverage Lengths	Coverage Probabilities
$2, \dots, 2$	Asymptotic	α	2.32210	0.964
		β_1	2.54982	0.847
		β_2	8.64345	0.827
	Bootstrap- p	α	3.13171	0.838
		β_1	2.70349	0.854
		β_2	8.57585	0.871
	Bootstrap- t	α	2.23531	0.946
		β_1	4.03219	0.938
		β_2	18.62305	0.842
$10, 0, \dots, 0, 10$	Asymptotic	α	2.51655	0.962
		β_1	2.55154	0.853
		β_2	9.52874	0.823
	Bootstrap- p	α	3.26937	0.844
		β_1	2.86809	0.863
		β_2	9.07476	0.874
	Bootstrap- t	α	2.40373	0.954
		β_1	4.07523	0.912
		β_2	18.36085	0.846
$20, 0, \dots, 0$	Asymptotic	α	1.84073	0.958
		β_1	2.20369	0.900
		β_2	7.65283	0.871
	Bootstrap- p	α	2.20501	0.895
		β_1	2.31266	0.917
		β_2	6.41350	0.922
	Bootstrap- t	α	1.83360	0.960
		β_1	2.65071	0.908
		β_2	10.58682	0.803
$0, \dots, 0, 20$	Asymptotic	α	2.87804	0.949
		β_1	3.14303	0.816
		β_2	11.48173	0.789
	Bootstrap- p	α	4.14486	0.818
		β_1	3.26263	0.851
		β_2	10.92282	0.856
	Bootstrap- t	α	2.76066	0.949
		β_1	5.26677	0.916
		β_2	25.91460	0.840

Table 4.6: Bias, MSE, Mean and Variance for: $n = 200, m = 80$

Censoring Scheme	$\hat{\pi}_1$ $\pi_1 = 0.73529$	Parameters	Bias	MSE	Mean	Variance	Asymptotic Variance
2, ..., 2, 1, ..., 1	0.73504	α	0.03945	0.03057	2.03945	0.02904	0.02924
		β_1	-0.01775	0.04605	2.98224	0.04578	0.04590
		β_2	0.01032	0.47331	5.01032	0.47367	0.47558
60, 0, ..., 0, 60	0.73441	α	0.03756	0.04217	2.03755	0.04079	0.04123
		β_1	-0.00571	0.04731	2.99428	0.04732	0.05037
		β_2	0.03204	0.52164	5.03204	0.52113	0.54834
120, 0, ..., 0	0.73318	α	0.03442	0.03092	2.03442	0.02976	0.02857
		β_1	-0.00152	0.04028	2.99847	0.04031	0.03845
		β_2	0.00825	0.35418	5.00824	0.35446	0.36664
0, 0, ..., 120	0.73560	α	0.04547	0.05200	2.04547	0.04998	0.04708
		β_1	-0.01631	0.07147	2.98369	0.07127	0.07084
		β_2	0.03613	0.71273	5.03613	0.71213	0.71735

Table 4.7: Bias, MSE, Mean and Variance for: $n = 100, m = 40$

Censoring Scheme	$\hat{\pi}_1$ $\pi_1 = 0.73529$	Parameters	Bias	MSE	Mean	Variance	Asymptotic Variance
2, ..., 2, 1, ..., 1	0.74085	α	0.06516	0.07097	2.06516	0.06679	0.06090
		β_1	-0.04198	0.08582	2.95802	0.08414	0.09186
		β_2	0.14399	1.52425	5.14399	1.50502	1.37792
30, 0, ..., 0, 30	0.73530	α	0.06585	0.09072	2.06585	0.08647	0.08382
		β_1	-0.01854	0.10869	2.98146	0.10845	0.10482
		β_2	0.10394	1.29493	5.10394	1.28541	1.34359
60, 0, ..., 0	0.73580	α	0.03947	0.05603	2.03947	0.05452	0.05601
		β_1	-0.00252	0.07154	2.99747	0.07160	0.07828
		β_2	0.13623	0.95314	5.13623	0.93551	0.93091
0, 0, ..., 60	0.73587	α	0.10740	0.12490	2.10740	0.11348	0.10128
		β_1	-0.04026	0.13914	2.95973	0.13766	0.14241
		β_2	0.02949	1.41868	5.02949	1.41923	1.60785

Table 4.8: Bias, MSE, Mean and Variance for: $n = 50, m = 20$

Censoring Scheme	$\hat{\pi}_1$ $\pi_1 = 0.73529$	Parameters	Bias	MSE	Mean	Variance	Asymptotic Variance
2, ..., 2, 1, ..., 1	0.73635	α	0.13629	0.17475	2.13629	0.15633	0.13424
		β_1	-0.06094	0.17546	2.93905	0.17191	0.19166
		β_2	0.16826	2.83819	5.16826	2.81269	3.60833
15, 0, ..., 0, 15	0.73510	α	0.16891	0.24120	2.16891	0.21288	0.18460
		β_1	-0.05716	0.19126	2.94283	0.18818	0.20913
		β_2	0.14707	3.03746	5.14707	3.01885	3.90477
30, 0, ..., 0	0.74085	α	0.12240	0.14113	2.12240	0.12627	0.11733
		β_1	-0.02976	0.16498	2.97023	0.16426	0.15341
		β_2	0.23670	2.18304	5.23671	2.12914	2.72633
0, 0, ..., 30	0.73775	α	0.21028	0.28249	2.21028	0.23851	0.22814
		β_1	-0.06804	0.25960	2.93195	0.25523	0.29307
		β_2	0.17963	4.02357	5.17963	3.99530	5.41318

Table 4.9: Bias, MSE, Mean and Variance for: $n = 30, m = 15$

Censoring Scheme	$\hat{\pi}_1$ $\pi_1 = 0.73529$	Parameters	Bias	MSE	Mean	Variance	Asymptotic Variance
1, ..., 1	0.73120	α	0.20501	0.31244	2.20501	0.27068	0.21426
		β_1	-0.04683	0.25153	2.95316	0.24959	0.25365
		β_2	0.20995	4.17684	5.20995	4.13689	5.73031
8, 0, ..., 0, 7	0.73087	α	0.22609	0.33462	2.22609	0.28378	0.25284
		β_1	-0.03106	0.23706	2.96893	0.23633	0.25488
		β_2	0.22225	4.30610	5.22226	4.26096	6.16741
15, 0, ..., 0	0.73247	α	0.15754	0.20195	2.15754	0.17731	0.16940
		β_1	-0.00328	0.20918	2.99671	0.20938	0.21537
		β_2	0.22147	2.88982	5.22147	2.84362	4.11519
0, 0, ..., 15	0.73373	α	0.25751	0.39247	2.25751	0.32648	0.31023
		β_1	-0.06204	0.32050	2.93795	0.31697	0.33963
		β_2	0.23463	5.56654	5.23463	5.51700	8.53191

Table 4.10: Bias, MSE, Mean and Variance for: $n = 30, m = 10$

Censoring Scheme	$\hat{\pi}_1$ $\pi_1 = 0.73529$	Parameters	Bias	MSE	Mean	Variance	Asymptotic Variance
2, ..., 2	0.72160	α	0.34611	0.58091	2.34611	0.46157	0.38196
		β_1	-0.07125	0.49340	2.92874	0.48881	0.54876
		β_2	-0.01876	5.02178	4.98123	5.02645	9.14268
10, 0, ..., 0, 10	0.72360	α	0.36278	0.71184	2.36278	0.58080	0.45627
		β_1	-0.07303	0.45782	2.92696	0.45294	0.55302
		β_2	0.13415	7.41573	5.13415	7.40514	15.0254
20, 0, ..., 0	0.71760	α	0.18791	0.29776	2.18791	0.26271	0.23570
		β_1	-0.00227	0.36475	2.99773	0.36511	0.37172
		β_2	0.13173	3.72488	5.13173	3.71124	6.49764
0, 0, ..., 20	0.72630	α	0.43348	1.00959	2.43348	0.82250	0.61401
		β_1	-0.07506	0.70456	2.92493	0.69963	0.90356
		β_2	0.26950	9.68316	5.26950	9.62015	20.0416

Chapter 5

Numerical Example

In this chapter, we shall illustrate the methods considered in the previous section with the help of a real-life dataset. We consider a dataset first considered by Hoel (1972). This is a mortality data provided by Dr. H. E. Walburg, Jr., of the Oak Ridge National Laboratory. The data (Table 5.1) came from a laboratory experiment performed on RFM strain male mice which had received a radiation dose of 300r at an age of 5-6 weeks. Hoel considered two major causes of deaths, namely, thymic lymphoma and reticulum cell sarcoma and combined all the other causes into a single group. However, for the purpose of analysis we consider reticulum cell sarcoma as Cause 1 and combine the others as Cause 2. Biologists believe that both of these two diseases are lethal and act independently of each other. Thus, the assumption of independence of the causes of failure is satisfied. The data is presented in Table 5.1.

5.1 Results for Each Censoring Scheme

There were $n = 99$ mice and we used $m = 35$ for four different censoring schemes again based largely on the schemes discussed in the previous chapter. For each of these censoring schemes, we calculate the MLEs of the parameters, the relative risk due to Cause 1 and all three types of confidence intervals and lengths for the three

Table 5.1: Autopsy data for 99 RFM conventional male mice which received a radiation dose of 300r at age 5-6 weeks

Thymic Lymphoma (22%)	159, 189, 191, 198, 200, 207, 220, 235, 245, 250, 256, 261, 265, 266, 280, 343, 356, 383, 403, 414, 428, 432
Reticulum Cell Sarcoma (38%)	317, 318, 399, 495, 525, 536, 549, 552, 554, 557, 558, 571, 586, 594, 596, 605, 612, 621, 628, 631, 636, 643, 647, 648, 649, 661, 663, 666, 670, 695, 697, 700, 705, 712, 713, 738, 748, 753
Other Causes (39%)	40, 42, 51, 62, 163, 179, 206, 222, 228, 252, 249, 282, 324, 333, 341, 366, 385, 407, 420, 431, 441, 461, 462, 482, 517, 517, 524, 564, 567, 586, 619, 620, 621, 622, 647, 651, 686, 761, 763

parameters. We shall present the results for each censoring scheme. We also constructed 95% confidence intervals based on the three methods discussed in Chapter 3. They are presented in the form of four tables, one for each censoring scheme. The **R** functions used to calculate these results are presented in Appendix A.2.

5.1.1 Censoring Scheme - C_1

We used the censoring scheme $R_1 = 2, R_2 = 2, \dots, R_{31} = 2, R_{32} = 1, R_{33} = 0, R_{34} = 1, R_{35} = 0$. After the censoring is done, we get $n_1 = 5$ and $n_2 = 30$. The MLEs were computed numerically as described in Chapter 2. They are

$$\hat{\alpha} = 2.17606, \hat{\beta}_1 = 1240.057, \hat{\beta}_2 = 544.316.$$

The variance covariance matrix of the MLEs is

$$V(\hat{\alpha}, \hat{\beta}_1, \hat{\beta}_2) = \begin{bmatrix} 0.08294103 & -55.2855 & -7.185028 \\ -55.28550518 & 101798.1246 & 4789.280950 \\ -7.18502772 & 4789.2810 & 2707.997690 \end{bmatrix}$$

Table 5.2: Confidence intervals and lengths for censoring scheme C_1 .

Method	Parameters	Lower Limit	Upper Limit	Length
Asymptotic	α	1.61164	2.74056	1.12892
	β_1	614.71470	1865.40000	1250.68500
	β_2	442.32260	646.30940	203.98690
Bootstrap- p	α	1.74101	2.97268	1.23167
	β_1	783.04460	2394.46700	1611.42200
	β_2	439.18860	637.82560	198.63700
Bootstrap- t	α	1.57495	2.74004	1.16509
	β_1	886.29970	2345.71800	1459.41800
	β_2	469.72480	698.51350	228.78880

The relative risk due to Cause 1 is

$$\hat{\pi}_1 = \frac{\hat{\beta}_2^{\hat{\alpha}}}{\hat{\beta}_1^{\hat{\alpha}} + \hat{\beta}_2^{\hat{\alpha}}} = 0.14286$$

5.1.2 Censoring Scheme - C_2

Here, we censored 32 observations after the first failure and the remaining 32 after the last failure. There is no censoring done for failures that occur inbetween the first and the last. i.e., $R_1 = 32, R_2 = 0, \dots, R_{34} = 0, R_{35} = 32$. Here we get $n_1 = 3$ and $n_2 = 32$. Here we observed $n_1 = 17$ and $n_2 = 18$. The MLEs obtained for this censoring scheme are

$$\hat{\alpha} = 1.86530, \hat{\beta}_1 = 2155.581, \hat{\beta}_2 = 605.9378.$$

The variance covariance matrix of the MLEs is

$$V(\hat{\alpha}, \hat{\beta}_1, \hat{\beta}_2) = \begin{bmatrix} 0.07516 & -138.62740 & -7.98173 \\ -138.62742 & 700828.27330 & 14720.66090 \\ -7.98173 & 14720.66090 & 4145.28772 \end{bmatrix}$$

Table 5.3: Confidence intervals and lengths for censoring scheme C_2 .

Method	Parameters	Lower Limit	Upper Limit	Length
Asymptotic	α	1.32794	2.40264	1.07470
	β_1	514.78730	3796.37400	3281.58700
	β_2	479.74770	732.12800	252.38030
Bootstrap- p	α	1.43332	2.63938	1.20606
	β_1	1127.07800	5166.92100	4039.84300
	β_2	479.82300	738.21870	258.39580
Bootstrap- t	α	1.32957	2.44520	1.11562
	β_1	1428.51400	5402.94200	3974.42800
	β_2	509.83370	799.25630	289.42260

The relative risk due to Cause 1 is

$$\hat{\pi}_1 = 0.08571$$

5.1.3 Censoring Scheme - C_3

For this censoring scheme, we censored all the $n - m$, i.e., 64 observations after the first failure. i.e., $R_1 = 64, R_2 = 0, \dots, R_{35} = 0$. The MLEs obtained thus are

$$\hat{\alpha} = 2.42692, \hat{\beta}_1 = 688.2123, \hat{\beta}_2 = 672.193.$$

The variance covariance matrix of the MLEs is

$$V(\hat{\alpha}, \hat{\beta}_1, \hat{\beta}_2) = \begin{bmatrix} 0.11475 & -5.68505 & -4.80415 \\ -5.68505 & 5011.91406 & 238.00582 \\ -4.80416 & 238.00582 & 4463.04658 \end{bmatrix}$$

The relative risk due to Cause 1 is

$$\hat{\pi}_1 = 0.48571$$

Table 5.4: Confidence intervals and lengths for censoring scheme C_3 .

Method	Parameters	Lower Limit	Upper Limit	Length
Asymptotic	α	1.76298	3.09086	1.32788
	β_1	549.45690	826.96770	277.51080
	β_2	541.25560	803.13040	261.87490
Bootstrap- p	α	1.94040	3.19122	1.25083
	β_1	561.87020	855.42760	293.55750
	β_2	547.25770	822.25570	274.99790
Bootstrap- t	α	1.77179	3.09607	1.32428
	β_1	578.06500	870.16390	292.09880
	β_2	562.73600	855.81630	293.08020

5.1.4 Censoring Scheme - C_4

For this censoring scheme we censored all the $n - m$, i.e., 64 observations after the last or the m^{th} failure. i.e., $R_1 = 0, \dots, R_{34} = 0, R_{35} = 64$. This is exactly the conventional Type II censoring where the remaining units in the study are censored after a particular number of failures have been observed. Here, we get $n_1 = 2$ and $n_2 = 33$. The MLEs obtained for this censoring scheme are

$$\hat{\alpha} = 1.93957, \hat{\beta}_1 = 2454.446, \hat{\beta}_2 = 578.4216.$$

The variance covariance matrix of the MLEs is

$$V(\hat{\alpha}, \hat{\beta}_1, \hat{\beta}_2) = \begin{bmatrix} 0.09768 & -242.6721 & -15.08662 \\ -242.67213 & 1403600.7307 & 37481.85039 \\ -15.08662 & 37481.8504 & 5025.23995 \end{bmatrix}$$

The relative risk due to Cause 1 is

$$\hat{\pi}_1 = 0.05714$$

Comparing the interval lengths we observe that there is not much difference between the three methods. This is true for each censoring scheme. However, when

Table 5.5: Confidence intervals and lengths for censoring scheme C_4 .

Method	Parameters	Lower Limit	Upper Limit	Length
Asymptotic	α	1.32702	2.55212	1.22511
	β_1	132.40540	4776.48700	4644.08200
	β_2	439.48190	717.36140	277.87950
Bootstrap- p	α	1.44317	2.77218	1.32901
	β_1	1096.66100	5971.04100	4874.38000
	β_2	448.22160	730.67190	282.45020
Bootstrap- t	α	1.35592	2.60524	1.24933
	β_1	1533.12200	7750.88100	6217.75900
	β_2	476.73410	790.07730	313.34330

we compare the confidence interval for each censoring scheme, the lengths for the censoring scheme C_3 are lesser than those for the other censoring schemes as also observed from the simulation study done in Chapter 4.

Remark 1: For censoring scheme C_3 , the relative risk due to Cause 1 is 0.48571. This is significantly higher than those for the other three censoring schemes because when we drop all the $n - m$ objects after the first failure, there is a greater chance that a lot of the observations due to Cause 2 get censored, thus resulting in more failures due to Cause 1. For censoring scheme C_3 , the number of failures due to Cause 1 is $n_1 = 17$ which is significantly higher than those for the other censoring schemes. Also, a closer look at the data tells us that the lifetimes of the rats are significantly higher when subjected to Cause 1 than the other causes. This should explain why the relative risk is high for censoring scheme C_3 and not so much for other censoring schemes.

Chapter 6

Conclusion

In this report, we have considered the competing risks model when the data is progressively Type II censored. We have also assumed that the lifetimes under the two different causes of failures have independent weibull distributions with equal shape parameters. The MLEs are obtained for the three parameters. Three different types of confidence intervals are constructed and compared using probability coverages and average coverage lengths. A numerical example has also been considered to illustrate the methods discussed in this project.

The asymptotic intervals seem to perform better in the case of large sample sizes. The average coverage lengths also seem to decrease as m increases. The same is true for even bootstrap intervals. The bootstrap probability coverages also seem to be affected by sample size. The bootstrap studentized intervals seem to perform better than the percentile intervals in terms of coverage probabilities. The simulation results show that the censoring scheme C_3 may be better than the other three. This can be seen in terms of probability coverages and ACLs. Also, the bias and MSE are lower for this type of censoring scheme. This is true for almost all the sample sizes considered in this project.

6.1 Extensions

A natural extension of this model would be to consider the more general case by using different shape parameters. That is, we can assume that the distribution corresponding to Cause 1 as $\text{Weibull}(\alpha_1, \beta_1)$ and the distribution corresponding to Cause 2 as $\text{Weibull}(\alpha_2, \beta_2)$. In this case, the relative risk due to Cause 1 is given by:

$$\begin{aligned}
 \pi_1 &= P(X_{1i} \leq X_{2i}) \\
 &= \int_0^\infty \int_{x=0}^y f_{X_{1i}, X_{2i}}(x, y) \, dx \, dy \\
 &= \int_{y=0}^\infty \frac{\alpha_2}{\beta_2^{\alpha_2}} y^{\alpha_2-1} \exp \left\{ - \left(\frac{y}{\beta_2} \right)^{\alpha_2} \right\} \int_{x=0}^y \frac{\alpha_1}{\beta_1^{\alpha_1}} y^{\alpha_1-1} \exp \left\{ - \left(\frac{x}{\beta_1} \right)^{\alpha_1} \right\} \, dx \, dy \\
 &= 1 - \int_{y=0}^\infty \frac{\alpha_2}{\beta_2^{\alpha_2}} y^{\alpha_2-1} \exp \left[- \left(\frac{y^{\alpha_2}}{\beta_2^{\alpha_2}} + \frac{y^{\alpha_1}}{\beta_1^{\alpha_1}} \right) \right] \, dy \tag{6.1.1}
 \end{aligned}$$

Computation of the integral given in (6.1.1) is not straightforward. It has to be expressed as an infinite series involving gamma functions. The derivatives of the likelihood function will get quite complicated because it involves the derivatives of gamma function. For this reason, we have not carried out any implementation of this more general model. However, this may be pursued in the future. Gamma life-times may also be considered in the future.

We can also consider dependent causes of failure although identifiability is a concern as explained in Chapter 1. If there is any evidence that the causes of failure are dependent, we could use models such as bivariate exponential model (among others) suggested by Marshall and Olkin (see Kotz, Balakrishnan and Johnson (2000)).

Appendix A

R Functions Used in this Project

A.1 Functions used for Simulation

```
datagen<-  
function (n=18,m=6,r=c(2,1,3,1,3,2),ini=c(1,0.8,0.7),...)  
{  
  #program to generate progressive competing risks data  
  #generate data until you get a dataset which has n1!=0 and n2!=0  
  alpha<-ini[1]  
  beta1<-ini[2]  
  beta2<-ini[3]  
  
  #p<-beta2^alpha/(beta1^alpha+beta2^alpha)  
  #validate the inputs  
  if (m!=length(r))  
  {  
    stop("The total number of failures should be equal  
    to the number of stages of censoring")  
  }  
  
  #function to generate progressive censoring data  
  #create arrays to store the generated values first  
  repeat  
  {
```

```

x<-array(dim=c(n,2))
x1i<-rweibull(n,shape=alpha,scale=beta1)
x2i<-rweibull(n,shape=alpha,scale=beta2)
z<-matrix(cbind(x1i,x2i),nrow=n)
#function to return the min of two values in a 2-d array
mini<-function(a) if (a[1]<=a[2]) {return(c(a[1],1))}
else {return(c(a[2],2))}
x<-apply(z,1,mini)
xd<-data.frame(t(x))

#ordering the data without losing the concomitants
xdo<-xd[order(xd[,1]),]

#create a data frame to store the index, the data,
the concomitants and censoring
data<-data.frame(index=c(1:n),x=xdo$X1,cause=xdo$X2,cen=c(rep(0,n)))

# the data has to be progressively type II censored now..

for(i in 1:m)
{
  if (r[i] != 0)
  {
    #first to check if there are enough observations to
    take the random sample from
    check<-data$x[data$cen==0][i]
    #print(data$index[data$x>check & data$cen==0])
    if(length(data$index[(data$x>check)&(data$cen==0)])>=r[i])
    {
      if ((length(data$index[data$x>check &
        data$cen==0]))>1)
      {
        c<-sample(data$index[(data$x>check)
          &(data$cen==0)],r[i])
        for(j in 1:length(c))
        {
          data$cen[data$index==c[j]]<-1
        }
      }
    }
  }
}

```

```

    }
    else if ((length(data$index[data$x>check &
data$cen==0]))==1)
    {
        data$cen[data$index == data$index[data$x>check
& data$cen==0]]<-1
    }
}
else break
}
}
n1<-length(data$x[data$cen==0 & data$cause==1])
m<-length(data$x[data$cen==0])
if (n1!=0 & n1!=m) break
}
return(data)
}

mleweibull<-
function (ini=c(1,2,3),data=data,m=6,n=18,r=c(2,1,3,1,3,2),...)
{
#get the data from the Type II progressively censored sample..

data<-datagen(m=m,n=n,r=r,p=p,ini=ini,...)
observed<-data$x[data$cen==0]
m<-length(observed)
n1<-length(data$x[data$cen==0 & data$cause==1])
#n2<-length(data$x[data$cen==0 & data$cause==2])
n2<-m-n1

if (n1==0 | n2==0)
{
    print(data)
    stop("n1 or n2 or both is 0. cannot find mle..")
}
}

```

```

#mycode
a.old<-ini[1]
b1.old<-ini[2]
b2.old<-ini[3]
repeat
{
    a<-a.old
    sumxa<-sum((r+1)*observed^a)
    sumxalogx<-sum((r+1)*observed^a*log(observed))

    b1<-(sumxa/n1)^(1/a)
    b2<-(sumxa/n2)^(1/a)
    doealpha<-function(x) (m/x -n1*log(b1)-n2*log(b2) +
    sum(log(observed))-(b1^(-x)+b2^(-x))*sum((r+1)*
    observed^x*log(observed))+(b1^(-x)*log(b1)+b2^(-x)*log(b2))
    *sum((r+1)*observed^x))^2
    a.object<-optimize(doealpha,lower=a-0.05,upper=a+0.05)
    a.old<-a.object$minimum
    if ((abs(a.old-a)<1e-10) & (abs(b1.old-b1)<1e-10)
    & (abs(b2.old-b2)<1e-10))
    {
        break
    }
    b1.old<-b1
    b2.old<-b2
}
sumxa<-sum((r+1)*observed^a)
sumxalogx<-sum((r+1)*observed^a*log(observed))
sumxalogx2<-sum((r+1)*observed^a*(log(observed))^2)
hessian<-matrix(c(rep(0,12)),nrow=3)
#record the mles in the first column
hessian[1,1]<-a
hessian[2,1]<-b1
hessian[3,1]<-b2
#record the hessian matrix in the columns 2,3,4 of the matrix 'hessian'

hessian[1,2]<- -m/(a^2) - (b1^(-a)+b2^(-a))*sumxalogx2 +
                2*(b1^(-a)*log(b1)+b2^(-a)*log(b2))*sumxalogx -

```



```

      (b1^(-a)*(log(b1))^2 + b2^(-a)*(log(b2))^2)*sumxa

hessian[1,3]<- -(n1/b1) + a/b1^(a+1)*sumxalogx +
              ((1-a*log(b1))/b1^(a+1))*sumxa
hessian[1,4]<- -(n2/b2) + a/b2^(a+1)*sumxalogx +
              ((1-a*log(b2))/b2^(a+1))*sumxa
hessian[2,3]<- a*n1/b1^2 - (a*(a+1)/b1^(a+2))*sumxa
hessian[3,4]<- a*n2/b2^2 - (a*(a+1)/b2^(a+2))*sumxa
hessian[2,2]<-hessian[1,3]
hessian[3,2]<-hessian[1,4]
return(hessian)
}

simulation<-
function (ini=c(1,2,2),R=999,n=18,
m=6,r=c(2,1,3,1,3,2),...)
{
#function to do the simulation
mle<-data.frame(index=1:R,a=0,b1=0,b2=0,a.var=1,
b1.var=1,b2.var=1,a.pivot=0,b1.pivot=0,b2.pivot=0,n1=0)
for (i in 1:R)
{
  data<-datagen(m=m,n=n,r=r,p=p,alpha=ini[1],
  beta1=ini[2],beta2=ini[3],...)
  mle$n1[i]<-length(data$x[data$cause==1 & data$cen==0])
  mle.object<-mleweibull(ini=ini,data=data,n=n,m=m,r=r)
  mle$a[i]<-mle.object[1,1]
  mle$b1[i]<-mle.object[2,1]
  mle$b2[i]<-mle.object[3,1]

  hessian<-mle.object[,-1]
  varcov<-solve(-hessian)
  mle$a.var[i]<-varcov[1,1]
  mle$b1.var[i]<-varcov[2,2]
  mle$b2.var[i]<-varcov[3,3]
}
}

```

```

mle$a.pivot<-(mle$a-ini[1])/sqrt(mle$a.var)
mle$b1.pivot<-(mle$b1-ini[2])/sqrt(mle$b1.var)
mle$b2.pivot<-(mle$b2-ini[3])/sqrt(mle$b2.var)

p95.alpha<-round(length(mle$a.pivot[(mle$a.pivot>-qnorm(0.975))
  & (mle$a.pivot<qnorm(0.975))])/R,3)
p95.beta1<-round(length(mle$b1.pivot[(mle$b1.pivot>-qnorm(0.975))
  & (mle$b1.pivot<qnorm(0.975))])/R,3)
p95.beta2<-round(length(mle$b2.pivot[(mle$b2.pivot>-qnorm(0.975))
  & (mle$b2.pivot<qnorm(0.975))])/R,3)
}

bootstrap<-
function (ini=c(2,3,5),N=1000,n=100,m=40,r=c(rep(2,20),
rep(1,20)),al=0.025,R=999)
{
#function to do the bootstrap confidence intervals and probabilities
main.m<-matrix(data=NA,nrow=N,ncol=6)
basic.boot<-data.frame(a.l=main.m[,1],a.u=main.m[,2],b1.l=main.m[,3],
b1.u=main.m[,4],b2.l=main.m[,5],b2.u=main.m[,6],
a.count=0,b1.count=0,b2.count=0)

stud.boot<-data.frame(a.l=main.m[,1],a.u=main.m[,2],b1.l=main.m[,3],
b1.u=main.m[,4],b2.l=main.m[,5],b2.u=main.m[,6],
a.count=0,b1.count=0,b2.count=0)
perc.boot<-data.frame(a.l=main.m[,1],a.u=main.m[,2],b1.l=main.m[,3],
b1.u=main.m[,4],b2.l=main.m[,5],b2.u=main.m[,6],
a.count=0,b1.count=0,b2.count=0)

for(i in 1:N)
{
  data<-datagen(ini=ini,n=n,m=m,r=r)
  mle.boot<-mleweibull(data=data,ini=ini,n=n,m=m,r=r)
  a.hat<-mle.boot[1,1]
  b1.hat<-mle.boot[2,1]
  b2.hat<-mle.boot[3,1]
  hessian<-mle.boot[,-1]

```

```

varcov<-solve(-hessian)
a.var<-varcov[1,1]
b1.var<-varcov[2,2]
b2.var<-varcov[3,3]
sim.boot<-simulation(ini=c(a.hat,b1.hat,b2.hat),n=n,m=m,r=r)

#basic boot strap intervals
basic.boot$a.l[i]<-2*a.hat - sort(sim.boot$a)[(R+1)*(1-a1)]
basic.boot$a.u[i]<-2*a.hat - sort(sim.boot$a)[(R+1)*(a1)]
if (basic.boot$a.l[i]<ini[1] & basic.boot$a.u[i]>ini[1])
  basic.boot$a.count[i]<-1

basic.boot$b1.l[i]<-2*b1.hat - sort(sim.boot$b1)[(R+1)*(1-a1)]
basic.boot$b1.u[i]<-2*b1.hat - sort(sim.boot$b1)[(R+1)*(a1)]
if (basic.boot$b1.l[i]<ini[2] & basic.boot$b1.u[i]>ini[2])
  basic.boot$b1.count[i]<-1

basic.boot$b2.l[i]<-2*b2.hat - sort(sim.boot$b2)[(R+1)*(1-a1)]
basic.boot$b2.u[i]<-2*b2.hat - sort(sim.boot$b2)[(R+1)*(a1)]
if (basic.boot$b2.l[i]<ini[3] & basic.boot$b2.u[i]>ini[3])
  basic.boot$b2.count[i]<-1

#studentized bootstrap intervals
stud.boot$a.l[i]<-a.hat - sqrt(a.var)*sort(sim.boot$a.pivot)
[(R+1)*(1-a1)]
stud.boot$a.u[i]<-a.hat - sqrt(a.var)*sort(sim.boot$a.pivot)
[(R+1)*(a1)]
if (stud.boot$a.l[i]<ini[1] & stud.boot$a.u[i]>ini[1])
  stud.boot$a.count[i]<-1

stud.boot$b1.l[i]<-b1.hat - sqrt(b1.var)*sort(sim.boot$b1.pivot)
[(R+1)*(1-a1)]
stud.boot$b1.u[i]<-b1.hat - sqrt(b1.var)*sort(sim.boot$b1.pivot)
[(R+1)*(a1)]
if (stud.boot$b1.l[i]<ini[2] & stud.boot$b1.u[i]>ini[2])
  stud.boot$b1.count[i]<-1

```

```

stud.boot$b2.l[i]<-b2.hat - sqrt(b2.var)*sort(sim.boot$b2.pivot)
[(R+1)*(1-a1)]
stud.boot$b2.u[i]<-b2.hat - sqrt(b2.var)*sort(sim.boot$b2.pivot)
[(R+1)*(a1)]
if (stud.boot$b2.l[i]<ini[3] & stud.boot$b2.u[i]>ini[3])
stud.boot$b2.count[i]<-1

#percentile bootstrap intervals
perc.boot$a.l[i]<-sort(sim.boot$a)[(R+1)*(a1)]
perc.boot$a.u[i]<-sort(sim.boot$a)[(R+1)*(1-a1)]
if (perc.boot$a.l[i]<ini[1] & perc.boot$a.u[i]>ini[1])
  perc.boot$a.count[i]<-1

perc.boot$b1.l[i]<-sort(sim.boot$b1)[(R+1)*(a1)]
perc.boot$b1.u[i]<-sort(sim.boot$b1)[(R+1)*(1-a1)]
if (perc.boot$b1.l[i]<ini[2] & perc.boot$b1.u[i]>ini[2])
  perc.boot$b1.count[i]<-1

perc.boot$b2.l[i]<-sort(sim.boot$b2)[(R+1)*(a1)]
perc.boot$b2.u[i]<-sort(sim.boot$b2)[(R+1)*(1-a1)]
if (perc.boot$b2.l[i]<ini[3] & perc.boot$b2.u[i]>ini[3])
  perc.boot$b2.count[i]<-1
}

# to calculate the pr coverages
pr.cov<-matrix(data=NA,nrow=3,ncol=6)
boot.pr.cov<-data.frame(par=c("alpha","beta1","beta2"),
basic=pr.cov[,1],stud=pr.cov[,2],perc=pr.cov[,3],
basic.length=pr.cov[,4],stud.length=pr.cov[,5],
perc.length=pr.cov[,6])

boot.pr.cov$basic[1]<-sum(basic.boot$a.count)/N
boot.pr.cov$stud[1]<-sum(stud.boot$a.count)/N
boot.pr.cov$perc[1]<-sum(perc.boot$a.count)/N

#coverage lengths for alpha
boot.pr.cov$basic.length[1]<-mean(basic.boot$a.u-basic.boot$a.l)

```

```

boot.pr.cov$stud.length[1]<-mean(stud.boot$a.u-stud.boot$a.l)
boot.pr.cov$perc.length[1]<-mean(perc.boot$a.u-perc.boot$a.l)

boot.pr.cov$basic[2]<-sum(basic.boot$b1.count)/N
boot.pr.cov$stud[2]<-sum(stud.boot$b1.count)/N
boot.pr.cov$perc[2]<-sum(perc.boot$b1.count)/N

#coverage lengths for beta1
boot.pr.cov$basic.length[2]<-mean(basic.boot$b1.u-basic.boot$b1.l)
boot.pr.cov$stud.length[2]<-mean(stud.boot$b1.u-stud.boot$b1.l)
boot.pr.cov$perc.length[2]<-mean(perc.boot$b1.u-perc.boot$b1.l)

boot.pr.cov$basic[3]<-sum(basic.boot$b2.count)/N
boot.pr.cov$stud[3]<-sum(stud.boot$b2.count)/N
boot.pr.cov$perc[3]<-sum(perc.boot$b2.count)/N

#coverage lengths for beta2
boot.pr.cov$basic.length[3]<-mean(basic.boot$b2.u-basic.boot$b2.l)
boot.pr.cov$stud.length[3]<-mean(stud.boot$b2.u-stud.boot$b2.l)
boot.pr.cov$perc.length[3]<-mean(perc.boot$b2.u-perc.boot$b2.l)

#print(boot.pr.cov)
return(boot.pr.cov)
}

bias.simulation<-
function (ini=c(1,2,2),R=999,n=18,m=6,r=c(2,1,3,1,3,2),...)
{
  #function to do the simulation
  mle<-data.frame(index=1:R,a=0,b1=0,b2=0,a.var=1,b1.var=1,b2.var=1,
    a.pivot=0,b1.pivot=0,b2.pivot=0,n1=0)
  for (i in 1:R)
  {
    data<-datagen(m=m,n=n,r=r,p=p,alpha=ini[1],beta1=ini[2],
      beta2=ini[3],...)
  }
}

```

```

mle$n1[i]<-length(data$x[data$cause==1 & data$cen==0])
mle.object<-mleweibull(ini=ini,data=data,n=n,m=m,r=r)
mle$a[i]<-mle.object[1,1]
mle$b1[i]<-mle.object[2,1]
mle$b2[i]<-mle.object[3,1]

hessian<-mle.object[, -1]
varcov<-solve(-hessian)
mle$a.var[i]<-varcov[1,1]
mle$b1.var[i]<-varcov[2,2]
mle$b2.var[i]<-varcov[3,3]
}
mle$a.pivot<-(mle$a-ini[1])/sqrt(mle$a.var)
mle$b1.pivot<-(mle$b1-ini[2])/sqrt(mle$b1.var)
mle$b2.pivot<-(mle$b2-ini[3])/sqrt(mle$b2.var)

p95.alpha<-round(length(mle$a.pivot[(mle$a.pivot>-qnorm(0.975)) &
(mle$a.pivot<qnorm(0.975))])/R,3)
p95.beta1<-round(length(mle$b1.pivot[(mle$b1.pivot>-qnorm(0.975)) &
(mle$b1.pivot<qnorm(0.975))])/R,3)
p95.beta2<-round(length(mle$b2.pivot[(mle$b2.pivot>-qnorm(0.975)) &
(mle$b2.pivot<qnorm(0.975))])/R,3)

alpha.length<-mean(2*qnorm(0.975)*sqrt(mle$a.var))
beta1.length<-mean(2*qnorm(0.975)*sqrt(mle$b1.var))
beta2.length<-mean(2*qnorm(0.975)*sqrt(mle$b2.var))

a.bias<-(mean(mle$a)-ini[1])
b1.bias<-(mean(mle$b1)-ini[2])
b2.bias<-(mean(mle$b2)-ini[3])
a.mse<-mean((mle$a-ini[1])^2)
b1.mse<-mean((mle$b1-ini[2])^2)
b2.mse<-mean((mle$b2-ini[3])^2)
coverage<-data.frame(p95.alpha=p95.alpha,p95.beta1=p95.beta1,
p95.beta2=p95.beta2)
print(coverage)
covLength<-data.frame(alpha.length=alpha.length,
beta1.length=beta1.length,beta2.length=beta2.length)

```

```

print(covLength)
bias<-data.frame(a.bias=a.bias,b1.bias=b1.bias,b2.bias=b2.bias)
print(bias)
mse<-data.frame(a.mse=a.mse,b1.mse=b1.mse,b2.mse=b2.mse)
print(mse)
probX1X2<-data.frame(probX1X2=mean(mle$b2~mle$a/
(mle$b1~mle$a+mle$b2~mle$a)))
print(probX1X2)
var.mle<-var(matrix(c(mle$a,mle$b1,mle$b2),nrow=R))
print(var.mle)
mle.mean<-data.frame(a.mean=mean(mle$a),b1.mean=mean(mle$b1),
b2.mean=mean(mle$b2))
print(mle.mean)
mle.var<-data.frame(a.var=mean(mle$a.var),b1.var=mean(mle$b1.var),
b2.var=mean(mle$b2.var))
print(mle.var)
}

```

A.2 Functions Used for the Numerical Example

```

numerical.datagen<-
function (n=99,m=35,r=c(rep(2,31),1,0,1,0))
{
  xd<-hoel
  xdo<-xd[order(xd[,1]),]
  #create a data frame to store the index, the data,
  the concomitants and censoring
  data<-data.frame(index=c(1:n),x=xdo$X1,cause=xdo$X2,cen=c(rep(0,n)))
  for(i in 1:m)
  {
    if (r[i] != 0)
    {
      #first to check if there are enough observations to
      take the random sample from
      check<-data$x[data$cen==0][i]
      if(length(data$index[(data$x>check)&(data$cen==0)])>=r[i])
      {

```

```

    if ((length(data$index[data$x>check & data$cen==0]))>1)
    {
        c<-sample(data$index[(data$x>check)&
        (data$cen==0)],r[i])
        for(j in 1:length(c))
        {
            data$cen[data$index==c[j]]<-1
        }
    }
    else if ((length(data$index[data$x>check &
    data$cen==0]))==1)
    {
        data$cen[data$index == data$index[data$x>check
        & data$cen==0]]<-1
    }
    }
    else break
}
return(data)
}

```

```

numerical.simulation<-
function (n=99,m=35,r=c(rep(2,31),1,0,1,0),R=999,al=0.05)
{
#function for simulating for the numerical example

#get the mle and the hessian matrix from the dataset using
numerical.mleweibull (almost same as mleweibull)
mle.main<-numerical.mleweibull(n=n,m=m,r=r)
hessian.main<-mle.main[,-1]
mle.a<-mle.main[1,1]
mle.b1<-mle.main[2,1]
mle.b2<-mle.main[3,1]

#function to do the simulation

```



```

mle<-data.frame(index=1:R,a=0,b1=0,b2=0,a.var=1,
b1.var=1,b2.var=1,a.pivot=0,b1.pivot=0,b2.pivot=0,n1=0)
for (i in 1:R)
{
    data<-datagen(m=m,n=n,r=r,alpha=mle.a,beta1=mle.b1,beta2=mle.b2)
    mle$n1[i]<-length(data$x[data$cause==1 & data$cen==0])
    mle.object<-mleweibull(ini=c(mle.a,mle.b1,mle.b2),
    data=data,n=n,m=m,r=r)
    mle$a[i]<-mle.object[1,1]
    mle$b1[i]<-mle.object[2,1]
    mle$b2[i]<-mle.object[3,1]

    hessian<-mle.object[,-1]
    varcov<-solve(-hessian)
    mle$a.var[i]<-varcov[1,1]
    mle$b1.var[i]<-varcov[2,2]
    mle$b2.var[i]<-varcov[3,3]
}
mle$a.pivot<-(mle$a-mle.a)/sqrt(mle$a.var)
mle$b1.pivot<-(mle$b1-mle.b1)/sqrt(mle$b1.var)
mle$b2.pivot<-(mle$b2-mle.b2)/sqrt(mle$b2.var)

#Now to calculate the Asymptotic Confidence intervals...
varcov.main<-solve(-hessian.main)

asy.a.l<-mle.a-qnorm(1-al/2)* sqrt(varcov.main[1,1])
asy.a.u<-mle.a+qnorm(1-al/2)* sqrt(varcov.main[1,1])
asy.a.len<-asy.a.u-asy.a.l

asy.b1.l<-mle.b1-qnorm(1-al/2)* sqrt(varcov.main[2,2])
asy.b1.u<-mle.b1+qnorm(1-al/2)* sqrt(varcov.main[2,2])
asy.b1.len<-asy.b1.u-asy.b1.l

asy.b2.l<-mle.b2-qnorm(1-al/2)* sqrt(varcov.main[3,3])
asy.b2.u<-mle.b2+qnorm(1-al/2)* sqrt(varcov.main[3,3])
asy.b2.len<-asy.b2.u-asy.b2.l

#Now to calculate the percentile bootstrap Confidence intervals..

```

```

per.a.l<-sort(mle$a)[(R+1)*(a1/2)]
per.a.u<-sort(mle$a)[(R+1)*(1-a1/2)]
per.a.len<-per.a.u-per.a.l
per.b1.l<-sort(mle$b1)[(R+1)*(a1/2)]
per.b1.u<-sort(mle$b1)[(R+1)*(1-a1/2)]
per.b1.len<-per.b1.u-per.b1.l

per.b2.l<-sort(mle$b2)[(R+1)*(a1/2)]
per.b2.u<-sort(mle$b2)[(R+1)*(1-a1/2)]
per.b2.len<-per.b2.u-per.b2.l

#Now to calculate the Studentized bootstrap Confidence intervals..
stu.a.l<-mle.a - sqrt(varcov.main[1,1])*sort(mle$a.pivot)
[(R+1)*(1-a1/2)]
stu.a.u<-mle.a - sqrt(varcov.main[1,1])*sort(mle$a.pivot)
[(R+1)*(a1/2)]
stu.a.len<-stu.a.u-stu.a.l

stu.b1.l<-mle.b1 - sqrt(varcov.main[2,2])*sort(mle$b1.pivot)
[(R+1)*(1-a1/2)]
stu.b1.u<-mle.b1 - sqrt(varcov.main[2,2])*sort(mle$b1.pivot)
[(R+1)*(a1/2)]
stu.b1.len<-stu.b1.u-stu.b1.l

stu.b2.l<-mle.b2 - sqrt(varcov.main[3,3])*sort(mle$b2.pivot)
[(R+1)*(1-a1/2)]
stu.b2.u<-mle.b2 - sqrt(varcov.main[3,3])*sort(mle$b2.pivot)
[(R+1)*(a1/2)]
stu.b2.len<-stu.b2.u-stu.b2.l

#to present the results in a dataframe..
asy.a<-data.frame(lower=asy.a.l,upper=asy.a.u,length=asy.a.len)
asy.b1<-data.frame(lower=asy.b1.l,upper=asy.b1.u,length=asy.b1.len)
asy.b2<-data.frame(lower=asy.b2.l,upper=asy.b2.u,length=asy.b2.len)

per.a<-data.frame(lower=per.a.l,upper=per.a.u,length=per.a.len)
per.b1<-data.frame(lower=per.b1.l,upper=per.b1.u,length=per.b1.len)
per.b2<-data.frame(lower=per.b2.l,upper=per.b2.u,length=per.b2.len)

```

```
stu.a<-data.frame(lower=stu.a.l,upper=stu.a.u,length=stu.a.len)
stu.b1<-data.frame(lower=stu.b1.l,upper=stu.b1.u,length=stu.b1.len)
stu.b2<-data.frame(lower=stu.b2.l,upper=stu.b2.u,length=stu.b2.len)

#to calculate the relative risk due to cause #1

pi1<-mle.b2^mle.a/(mle.b1^mle.a+mle.b2^mle.a)

#printing the results

list(mle.a=mle.a,mle.b1=mle.b1,mle.b2=mle.b2,varcov.main=varcov.main,
asy.a=asy.a,asy.b1=asy.b1,asy.b2=asy.b2,per.a=per.a,per.b1=per.b1,
per.b2=per.b2,stu.a=stu.a,stu.b1=stu.b1,stu.b2=stu.b2,pi1=pi1)

}
```

Bibliography

- Bain, L. J. (1978), *Statistical analysis of reliability and life-testing models - Theory and practice*, Marcel Dekker, New York.
- Bain, L. J. and Engelhardt, M. (1991), *Statistical analysis of reliability and life-testing models, Second edition*, Marcel Dekker, New York.
- Balakrishnan, N. and Aggarwala, R. (2000), *Progressive censoring: Theory, methods and applications*, Birkhauser, Boston.
- Balakrishnan, N. and Cohen, A. C. (1991), *Order statistics and inference: Estimation methods*, Academic Press, San Diego.
- Boag, J. W. (1949), ‘Maximum likelihood estimates of the proportion of patients cured by cancer therapy’, *Journal of the Royal Statistical Society. Series B* **11**, 15–44.
- Cohen, A. C. (1963), ‘Progressively censored samples in life testing’, *Technometrics* **5**, 327–329.
- Cohen, A. C. (1966), ‘Life testing and early failure’, *Technometrics* **8**, 539–549.
- Cohen, A. C. (1991), *Truncated and censored samples: theory and applications*, Marcel Dekker, New York.

- Crowder, M. J. (2001), *Classical competing risks*, Chapman and Hall, Boca Raton, FL.
- David, H. A. and Moeschberger, M. L. (1978), *Theory of competing risks*, Griffin, London.
- Davison, A. C. and Hinkley, D. V. (1997), *Bootstrap methods and their applications*, Cambridge University Press, Cambridge.
- Hall, P. (1988), 'Theoretical comparison of bootstrap confidence intervals', *Annals of Statistics* **16**, 927–953.
- Harter, H. L. (1992), *Order statistics and their uses in testing and estimation, Vol. 2: Estimates based on order statistics of samples from various populations*, U.S. Government Printing Office, Washington, D. C.
- Hoel, D. G. (1972), 'A representation of mortality data by competing risks', *Biometrics* **28**, 475–488.
- Kalbfleisch, J. D. and Prentice, R. L. (1980), *The statistical analysis of failure time data*, Wiley, New York.
- Kotz, S., Balakrishnan, N. and Johnson, N. L. (2000), *Continuous multivariate distributions*, Vol. 1, Wiley, New York.
- Kundu, D., Kannan, N. and Balakrishnan, N. (2004), 'Analysis of progressively censored competing risks data', *Handbook of Statistics* **23**, 331–348.
- Lawless, J. F. (1982), *Statistical models and methods for lifetime data*, John Wiley and Sons, New York.

- Mann, N. R., Schafer, R. E. and Singpurwalla, N. D. (1974), *Methods for statistical analysis of reliability and life data*, John Wiley and Sons, New York.
- Maple (2003), *Maplesoft, a division of Waterloo Maple Inc. 1981-2003*, Waterloo Maple Inc.
- Nelson, W. (1982), *Applied life data analysis*, John Wiley and Sons, New York.
- R Development Core Team (2004), *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.