# Likelihood Inference of Some Cure Rate Models and Applications

By

XiaoFeng Liu

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Master of Science

McMaster University

MASTER OF SCIENCE (2011) McMaster University

(Statistics) Hamilton, Ontario

| | |
|---|---|
| TITLE: | Likelihood Inference of Some Cure Rate Models and Applications |
| AUTHOR: | XiaoFeng Liu<br>(McMaster University, Canada) |
| SUPERVISOR: | Dr. N. Balakrishnan |
| NUMBER OF PAGES: | ix, 64 |

# Acknowledgements

I would like to take this opportunity to express my gratitude towards my supervisor Dr. N. Balakrishnan for his many helpful suggestions, and infinite patience throughout the work of my project. Special thanks to Dr. J. Rodrigues for his generous help during this study.

I am grateful to have Dr. R. Viveros-Aguilera and Dr. A. Childs as my examiners. Thanks for their time and helpful remarks. I would like to thank the Department of Mathematics and Statistics at McMaster University for providing a gracious learning environment.

Most importantly, I want to thank my family, especially my parents, for their love and support that have helped me to continue my studies.

# Abstract

In this thesis, we perform a survival analysis for right-censored data of populations with a cure rate. We consider two cure rate models based on the Geometric distribution and Poisson distribution, which are the special cases of the Conway-Maxwell distribution. The models are based on the assumption that the number of competing causes of the event of interest follows Conway-Maxwell distribution. For various sample sizes, we implement a simulation process to generate samples with a cure rate. Under this setup, we obtain the maximum likelihood estimator (MLE) of the model parameters by using the gamlss R package. Using the asymptotic distribution of the MLE as well as the parametric bootstrap method, we discuss the construction of confidence intervals for the model parameters and their performance is then assessed through Monte Carlo simulations.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

In reliability and survival analysis, models for survival data with a survival fraction (also known as cure rate models or long-term survival models) play an important role. Our study is an extension of the work of Rodrigues et al. (2009). These authors proposed Conway-Maxwell Poisson (COM-Poisson) cure rate model as a flexible alternative to the cure rate model discussed by Yin and Ibrahim (2005), wherein the COM-Poisson distribution is a special case of the Weighted Poisson distribution and it can account for over-dispersion and under-dispersion that is commonly encountered in discrete data. The objective of our study is to consider some well-known special cases of the COM-Poisson cure rate model under high levels of censoring rate and examine whether the asymptotic normality of the maximum likelihood estimators (MLEs) of the model parameters holds. In terms of coverage probability for the model parameters, we compare the interval estimation based on asymptotic normality and by the parametric bootstrap method.

Cure rate models include the sampling objects that are not susceptible to the occurrence of an event of interest, which can happen due to different competing causes. Without the survival fraction, the cure rate model will just become a regular survival model. The cure rate model is quite useful in biomedical studies. For instance, an event of interest can be patient's death, which can occur due to metastasis-component tumor cells in an individual left active after an initial treatment. Some patients are cured after the initial treatment. However, some are still

under the risk of tumor cells relapsing or metastasizing. The metastasis-component tumor cells are the cells which have the potential to metastasize (Yakovlev and Tsodikov, 1996).

Let $M$ be a COM-Poisson random variable denoting the number of competing causes of the event of interest, with probability mass function (p.m.f.) $P_m = P[M = m]$ for $m = 0, 1, 2, \ldots$. Given $M = m$, let $W_j$ (for $j = 1, 2, \ldots, m$) be a random variable denoting the time-to-event for the $jth$ competing cause. The random variable $W_j$ shares the common distribution function $F(y) = 1 - S(y)$, where $S(y)$ is a proper survival function.

In the competing causes scenario, both the number of competing causes ($M$) and the lifetime ($W_j$) of a particular competing cause are not observable. Only the shortest lifetime of all the competing causes is observed. However, the survival data includes the portion of the sample that are not susceptible (cured fraction) to the event. In order to accommodate the cured portion to the occurrence of the event, the lifetime is defined as

$$Y = \min\{W_0, W_1, W_2, \ldots, W_m\}, \tag{1.1}$$

where $W_0 = \infty$ with probability 1. When $M = 0$, there is no competing cause to trigger the event of interest, and so the sample individuals whose lifetimes are $W_0$ would lead to the cured proportion $P_0$, which is also termed cure rate.

This study concentrates on two of the most popular special cases of the COM-Poisson distribution, which are the Geometric and Poisson distributions. In the former, the number of competing causes is modeled by a Geometric distribution, while in the latter, the number of competing causes is assumed to follow a Poisson distribution. Our interest lies on the inference for the model parameters. Under some regularity conditions, the asymptotic distribution of the maximum likelihood estimators of the model parameters is multivariate normal. However, for a real biomedical data, the accuracy of the point and interval estimation may be questionable as the estimates of the variances of the MLEs are not so reliable in this case. In order to find an alternate method, the bootstrap method is employed. We conducted a simulation study to compare the interval estimation based on the normal approximation and the bootstrap method.

The remainder of this thesis is organized as follows. In Chapter 2, we formulate the Conway-Maxwell Poisson cure rate model, and introduce some of its special cases. Maximum likelihood estimation and standard errors of the parameters of the COM-Poisson cure rate model are then discussed in Chapter 3. In Chapter 4, the Monte Carlo simulation algorithm is described along with the simulational results. Finally, in Chapter 5, we introduce the parametric bootstrap method, and the interval estimation is based on it.

# Chapter 2

# Cure Rate Model Formulation

In this chapter, we formulate the COM-Poisson cure rate model by stating and proving three theorems that are necessary for this model formulation. Let us assume $M$ follows a weighted Poisson distribution with probability mass function

$$p(m;\eta;\phi) = P[M = m;\eta,\phi] = \frac{w(m;\phi)p^*(m;\eta)}{E_\eta[w(M;\phi)]}, \qquad m = 0,1,2,\ldots, \tag{2.1}$$

where $p^*(\cdot;\eta)$ is the p.m.f. of a Poisson distribution with parameter $\eta > 0$, $w(\cdot;\phi)$ is a positive weight function with parameter $\phi > 0$, and $E_\eta[\cdot]$ is the expectation taken with respect to a Poisson distribution with mean parameter $\eta$. We call (2.1) the weighted Poisson distribution with parameter $\eta$ and weight function $w(\cdot;\phi)$.

The sampling lifetime $Y$ in (1.1) then has the cure rate survival function (Tsodikov et al., 2003; Rodrigues et al., 2009) of the from

$$S_{pop}(y) = P[Y \geq y] = \sum_{m=0}^{\infty} P[M = m]\{S(y)\}^m = A_M(S(y)), \tag{2.2}$$

where $S(y)$ indicates the common survival function, and $A_M(\cdot)$ denotes the probability generation function (p.g.f.) of the random variable $M$ denoting the number of competing causes.

**Theorem 2.1.** *For $\phi \geq 0$ and $\theta \in \Theta \subset \mathbb{R}$, and a discrete random variable $M$ with p.m.f*

$$p(m;\theta,\phi) = \varphi(m;\phi)e^{\theta m - K(\theta,\phi)}, \qquad m = 0,1,2,\ldots, \tag{2.3}$$

*the p.g.f. is given by*

$$A(s) = e^{-\eta(1-s)}\frac{E_{s\eta}[w(M;\phi)]}{E_{\eta}[w(M;\phi)]}, \qquad where \quad \eta = e^{\theta}.$$

*Proof.* In Kokonendji et al. (2008), we know that the weight function $w(\cdot;\phi)$ is given by $w(m;\phi) = m!\varphi(m;\phi)$. Let $\eta = e^{\theta}$, from (2.3), we can obtain

$$\begin{aligned} p(m;\theta,\phi) &= \varphi(m;\phi)e^{\theta m - K(\theta,\phi)} \\ &= m!\varphi(m;\phi)\frac{e^{-\eta}\eta^m}{m!}e^{-K(\theta,\phi)}e^{\eta} \\ &= w(m;\phi)\frac{e^{-\eta}\eta^m}{m!}e^{\eta - K(\theta,\phi)} \end{aligned}$$

Since $p(m;\theta,\phi)$ is a p.m.f., then we get

$$\sum_{m=0}^{\infty} p(m;\theta,\phi) = \sum_{m=0}^{\infty} w(m;\phi)\frac{e^{-\eta}\eta^m}{m!}e^{\eta - K(\theta,\phi)} = 1$$

Consequently, we have

$$\sum_{m=0}^{\infty} w(m;\phi)\frac{e^{-\eta}\eta^m}{m!} = e^{-\eta + K(\theta,\phi)} = E_{\eta}[w(M;\phi)] \tag{2.4}$$

Now, from (2.2) and (2.4), we get

$$\begin{aligned} A(s) &= E(s^m) \\ &= \sum_{m=0}^{\infty} w(m;\phi)\frac{e^{-\eta}\eta^m}{m!}e^{\eta-K(\theta,\phi)}s^m \\ &= e^{\eta s-K(\theta,\phi)}\sum_{m=0}^{\infty} w(m;\phi)\frac{e^{-\eta s}(\eta s)^m}{m!} \\ &= e^{\eta s-\eta}e^{\eta-K(\theta,\phi)}E_{s\eta}[w(M;\phi)] \\ &= e^{-\eta(1-s)}\frac{E_{s\eta}[w(M;\phi)]}{E_{\eta}[w(M;\phi)]} \end{aligned}$$

□

By Theorem 2.1 and (2.2), the cure rate survival function can be written as

$$S_{pop}(y) = e^{-\eta(1-S(y))}\frac{E_{S(y)\eta}[w(M;\phi)]}{E_{\eta}[w(M;\phi)]}, \tag{2.5}$$

which is in fact an improper survival function. For a proper survival function, we should have $\lim_{y\to\infty} S(y) = 0$, but $\lim_{y\to\infty} S_{pop}(y) > 0$, which is verified by the following theorem.

**Theorem 2.2.** *Given the weight function $w(0;\phi) > 0$ and a proper survival function $S(y)$, we have*

$$\lim_{y\to\infty} S_{pop}(y) = P[M=0] = p_0 = e^{-\eta}\frac{w(0;\phi)}{E_{\eta}[w(M;\phi)]},$$

*where $p_0$ denotes the immune or cured proportion in the sample data.*

*Proof.* It is intuitively clear that $\lim_{y\to\infty} S_{pop}(y) = P[M=0]$, since as the lifetime approaches infinity, the survival probability is equal to the cure rate. Then, we take the limit of (2.5) as

follows:

$$\begin{aligned}
\lim_{y\to\infty} S_{pop}(y) &= \lim_{y\to\infty}\left\{e^{-\eta(1-S(y))}\frac{E_{S(y)\eta}[w(M;\phi)]}{E_{\eta}[w(M;\phi)]}\right\} \\
&= e^{-\eta}\frac{\lim_{y\to\infty} E_{S(y)\eta}[w(M;\phi)]}{E_{\eta}[w(M;\phi)]} \\
&= e^{-\eta}\frac{\sum_{m=0}^{\infty} w(0;\phi)\frac{0^m}{m!}}{E_{\eta}[w(M;\phi)]} \\
&= e^{-\eta}\frac{w(0;\phi)}{E_{\eta}[w(M;\phi)]}.
\end{aligned}$$

□

In the paper by Conway and Maxwell (1961), the COM-Poisson distribution was originally proposed. By accounting for both under-dispersion and over-dispersion, the COM-Poisson distribution generalizes the Poisson distribution in a simple and flexible way. This distribution can be presented in the form (2.3), and therefore from the proof of Theorem 2.1 it can be regarded as a weighted Poisson distribution with weight function $w(M;\phi) = (m!)^{1-\phi}$. For $m = 0, 1, 2, \ldots$, we have the p.m.f. of the random variable $M$ to be

$$P[M = m; \eta, \phi] = \frac{1}{Z(\eta,\phi)}\frac{\eta^m}{(m!)^{\phi}}, \tag{2.6}$$

where $Z(\eta,\phi) = \sum_{j=0}^{\infty}\frac{\eta^j}{(j!)^{\phi}}$, which serves as a normalization constant. Therefore, from (2.6), we can easily obtain the cure rate of the COM-Poisson distribution, which is $p_0 = \frac{1}{Z(\eta,\phi)}$. The COM-Poisson distribution accounts for under-dispersion for parameter $\phi > 1$ and accounts over-dispersion when $\phi < 1$. Taking different values for the parameter $\phi$, COM-Poisson distribution includes Poisson and geometric distributions as special cases and Bernoulli distribution as a limiting distribution. From (2.6), we observe the following:

(i) When $\phi = 1$, we have

$$P[M = m; \eta, \phi = 1] = \frac{1}{Z(\eta, \phi = 1)} \frac{\eta^m}{m!} = \frac{1}{\sum_{j=0}^{\infty} \frac{\eta^j}{j!}} \frac{\eta^m}{m!} = \frac{\eta^m}{m!} e^{-\eta},$$

which is the p.m.f. of the Poisson distribution with parameter $\eta$;

(ii) When $\phi = 0$, we have

$$P[M = m; \eta, \phi = 0] = \frac{\eta^m}{Z(\eta, \phi = 0)} = \frac{\eta^m}{\sum_{j=0}^{\infty} \eta^j} = \eta^m (1 - \eta), for \quad 0 < \eta < 1.$$

Thus, the COM-Poisson distribution reduces to a Geometric distribution with success probability $1 - \eta$;

(iii) When $\phi \to \infty$, we find

1) For $m = 0$,

$$\lim_{\phi \to \infty} P[M = 0; \eta, \phi] = \lim_{\phi \to \infty} \frac{1}{Z(\eta, \phi)} \frac{\eta^0}{(0!)^\phi} = \frac{1}{1 + \eta}$$

2) For $m = 1$,

$$\lim_{\phi \to \infty} P[M = 1; \eta, \phi] = \lim_{\phi \to \infty} \frac{1}{Z(\eta, \phi)} \frac{\eta^1}{(1!)^\phi} = \frac{\eta}{1 + \eta}$$

3) For $m > 1$,

$$\lim_{\phi \to \infty} P[M; \eta, \phi] = 0$$

Therefore, as $\phi \to \infty$, the limiting distribution of COM-Poisson distribution is Bernoulli with success probability $\frac{\eta}{1 + \eta}$.

**Theorem 2.3.** *If a random variable M follows a COM-Poisson distribution, then the long-term survival function is*

$$S_{pop}(y) = \frac{Z(\eta S(y); \phi)}{Z(\eta; \phi)}, \qquad with \quad \eta = e^{\theta}. \tag{2.7}$$

*Proof.* From (2.5), we have

$$\begin{aligned} S_{pop}(y) &= e^{-\eta(1-S(y))}\frac{E_{S(y)\eta}[(m!)^{1-\phi}]}{E_{\eta}[(m!)^{1-\phi}]} \\ &= e^{-\eta(1-S(y))}\frac{\sum_{m=0}^{\infty}(m!)^{1-\phi}\frac{e^{-\eta S(y)}[\eta S(y)]^m}{m!}}{\sum_{m=0}^{\infty}(m!)^{1-\phi}\frac{e^{-\eta}\eta^m}{m!}} \\ &= e^{-\eta(1-S(y))}\frac{e^{-\eta S(y)}\sum_{m=0}^{\infty}\frac{[\eta S(y)]^m}{(m!)^{\phi}}}{e^{-\eta}\sum_{m=0}^{\infty}\frac{\eta^m}{(m!)^{\phi}}} \\ &= \frac{Z(\eta S(y);\phi)}{Z(\eta;\phi)}. \end{aligned}$$

□

The long-term survival function in Theorem 2.3 is an improper survival function, from which we can derive the improper density function as follows:

$$f_{pop}(y) = -S'_{pop}(y) = \frac{1}{Z(\eta;\phi)}\frac{f(y)}{S(y)}\sum_{m=0}^{\infty}\frac{[\eta S(y)]^m}{(m!)^{\phi}}, \tag{2.8}$$

where $f(y)$ and $S(y)$ are the common (proper) density and survival function of the lifetime $W$ respectively.

The COM-Poisson cure rate model can encompass under-dispersion and over-dispersion by taking various values of $\phi$. In this thesis, our focus is on the two special distributions, which are Poisson and Geometric. In the Poisson case, the cure rate is obtained as $p_0 = \frac{1}{Z(\eta,1)} = e^{-\eta}$, while in the Geometric case, we can easily get the cure fraction as $p_0 = \frac{1}{Z(\eta,0)} = 1-\eta$.

# Chapter 3

# Maximum Likelihood Estimation and Illustration

## 3.1 Inference

In practice, the lifetimes of all the sampling units may not be observable, due to censoring that could be present in the data. Now, let us consider the case that the minimum lifetime of all the competing causes $Y$ is not completely observed and is subject to right censoring. We assume that the $W_j$s are $i.i.d.$ with common density function $f(y)$. Let $C_i$ denotes the censoring time, thereby, the observed individual lifetime will be $T_i = \min(Y_i, C_i)$. We can also observe the censoring indicator $\delta_i = I(Y_i \leq C_i)$ in which $\delta_i = 1$ if observed $T_i$ is a time-to-event $(Y_i \leq C_i)$ and $\delta_i = 0$ if $T_i$ is right censored $(Y_i > C_i)$. We include the $k$ covariates through the link function for the parameter $\eta$ in (2.7) by the log-linear link function $\eta_i = \exp(x_i^{'}\beta)$, $i = 1, \ldots, n$, where $\beta$ is a $(k+1) \times 1$ vector of regression coefficients. For a given sample of size $n$, the complete data likelihood function of parameters $\vartheta = (\beta^{'}, \nu^{'}, \phi)$ under non-informative censoring can then be written as

$$L(\vartheta; t, \delta) \propto \prod_{i=1}^{n} \left\{ f_{pop}(\vartheta; t_i, \delta_i) \right\}^{\delta_i} \left\{ S_{pop}(\vartheta; t_i, \delta_i) \right\}^{1-\delta_i}, \tag{3.1}$$

where $t = (t_1, t_2, \ldots, t_n)'$ and $\delta = (\delta_1, \delta_2, \ldots, \delta_n)'$. We now assume the time-to-event ($W$) follows a Weibull distribution with

$$F(w; \nu_1, \nu_2) = 1 - \exp(-w^{\nu_1} e^{\nu_2}) \quad and \quad f(w; \nu_1, \nu_2) = \nu_1 w^{\nu_1 - 1} \exp(\nu_2 - w^{\nu_1} e^{\nu_2}) \tag{3.2}$$

for $w > 0$, $\nu_1 > 0$, and $\nu_2 \in \mathbb{R}$.

Then from (3.1), the maximum likelihood estimation of the parameters $\vartheta = (\beta', \nu', \phi)$ is performed. However, since the log-likelihood function $l(\vartheta; t, \delta) = \log L(\vartheta; t, \delta)$ is highly sensitive to the initial values of the parameters, the regular log-likelihood function maximization methods (such as Newton-Raphson method) do not work satisfactorily. So, the RS method (Rigby and Stasinopoulos, 2005) is employed to maximize $l(\vartheta; t, \delta)$ numerically. This method is available in the *gamlss* package (Stasinopoulos and Rigby 2007). The RS method only needs the improper functions ($S_{pop}(t)$ and $f_{pop}(t)$ from (2.7) and (2.8)) and the first derivatives of $log f_{pop}(t)$. These derivatives are presented in Appendix A.

The corresponding summary function in R for *gamlss* has two ways to produce the standard errors (SEs). One, which takes into account the relationship between the distribution parameters, calculates the SEs based on the observed fisher information matrix and the other one uses individual fits of the parameters. If the former method failed, then by default the latter is employed. However, standard errors produced by the latter method are found to be not so reliable. In this case, SEs can be obtained through the second-order derivatives of the log-likelihood function. We notice that unlike the usual likelihood functions, (3.1) involves improper density and improper survival functions. In order to avoid the possible differentiability problems, we consider another way of setting up the likelihood function, which only involves proper density and survival functions.

Let $F^*_{pop}(t)$ and $S^*_{pop}(t)$ be the overall population (including the cure fraction) lifetime cumulative distribution function ($c.d.f.$) and survival function, respectively. Also, let $F^*_s(t)$ and $S^*_s(t)$ be the c.d.f. and survival function for the lifetime of susceptible (non-cured) population, where $F^*_s(t)$ and $S^*_s(t)$ are now proper functions. Let the cured proportion in the population be $\pi$. Given

the observed data $t = (t_1, t_2, \ldots, t_n)$,where $t_i = \min(Y_i, C_i)$, we have $P(T \leq t \mid Cured) = 0$ for $0 \leq t < \infty$. Then, we have

$$\begin{aligned} F^*_{pop}(t) &= P(T \leq t) \\ &= P(\text{Cured})P(T \leq t \mid \text{Cured}) + P(\text{Susceptible})P(T \leq t \mid \text{Susceptible}) \\ &= (1-\pi)F^*_s(t). \end{aligned}$$

Thus, we obtain

$$f^*_{pop}(t) = (1-\pi)f^*_s(t) \tag{3.3}$$

$$S^*_{pop}(t) = 1 - F^*_{pop}(t) = \pi + (1-\pi)S^*_s(t) \tag{3.4}$$

Now let $I_0 = \{i \in (1, 2, \ldots, n) : \delta_i = 0\}$ and $I_1 = \{i \in (1, 2, \ldots, n) : \delta_i = 1\}$, where $\delta_i = 0$ indicates censoring and $\delta_i = 1$ denotes failure of the observing subject. Then, with (3.3) and (3.4), the likelihood function can be written in the form

$$L(t) \propto \prod_{i \in I_1} f^*_{pop}(t_i) \prod_{i \in I_0} S^*_{pop}(t_i) = \prod_{i \in I_1} (1-\pi_i) f^*_s(t_i) \prod_{i \in I_0} \left\{\pi_i + (1-\pi_i)S^*_s(t_i)\right\}.$$

The log-likelihood function for the COM-Poisson distribution can then be expressed as

$$l(t) = \sum_{i \in I_1} log\left\{(1-\pi_i)f^*_s(t_i)\right\} + \sum_{i \in I_0} log\left\{\pi_i + (1-\pi_i)S^*_s(t_i)\right\}. \tag{3.5}$$

Taking different values for the parameter $\phi$, we can derive the log-likelihood function for the two special cases.

**Case 1: Geometric Model**

For $\phi = 0$ and $\eta < 1$, the COM-Poisson distribution converges to the Geometric distribution with p.m.f. $P(M = m) = (1-\eta)\eta^m$, for $m = 0, 1, 2, \ldots$, and cured proportion $\pi = 1 - \eta$. $W_j$s

are assumed to be Weibull with c.d.f. and p.d.f. given in (3.2). Then, the survival function for the non-cured population can be obtained as

$$\begin{aligned} S_s^*(t) &= P(T > t \mid \text{Individual is susceptible}) \\ &= P(\min(W_1, W_2, \ldots, W_m) > t \mid m \geq 1) \\ &= \sum_{m=1}^{\infty} \left\{ P(W_1 > t) \cdots P(W_m > t) \frac{P(M = m)}{1 - P(M = 0)} \right\} \\ &= \sum_{m=1}^{\infty} [S(t)]^m \frac{(1-\eta)\eta^m}{\eta} \\ &= \frac{(1-\eta)S(t)}{1 - \eta S(t)}. \end{aligned}$$

Therefore, the corresponding density function is $f_s^*(t) = -[S_s^*(t)]' = \dfrac{(1-\eta)f(t)}{[1-\eta S(t)]^2}$. We then obtain the log-likelihood function in (3.5) as

$$\begin{aligned} l(t) &= \sum_{i \in I_1} \log \eta_i \frac{(1-\eta)f(t)}{[1-\eta S(t)]^2} + \sum_{i \in I_0} \log \left\{ 1 - \eta_i + \eta_i \frac{(1-\eta)S(t)}{1-\eta S(t)} \right\} \\ &= \sum_{i=1}^{n} \{\delta_i [\log \eta_i + \log f(t_i) - \log(1 - \eta_i S(t_i))] + \log(1-\eta_i) - \log(1 - \eta_i S(t_i))\} \end{aligned}$$

Under suitable regularity conditions, the asymptotic distribution of the $MLE$ $\hat{\vartheta}$ is normally distributed with covariance matrix $\Sigma(\hat{\vartheta})$ that can be estimated by $\hat{\Sigma}(\hat{\vartheta}) = \left\{ -\dfrac{\partial^2 l(\vartheta; t)}{\partial \vartheta \partial \vartheta'} \right\}^{-1}$, evaluated at $\hat{\vartheta}$, where the second derivatives are presented in Appendix A.

Furthermore, we can obtain the cure rate for the $jth$ group of a generated sample of the Geometric model estimated by $\widehat{p_{0_j}} = 1 - \exp(\hat{\beta}_0 + j\hat{\beta}_1)$. Let $\beta = (\beta_0, \beta_1)'$, then the corresponding estimate of the variance is obtained by $(\nabla p_{0_j})' Var(\beta) \nabla p_{0_j}$, where $Var(\beta)$ is approximated by $\hat{\Sigma}(\hat{\beta}) = \left\{ -\dfrac{\partial^2 l(\vartheta; t)}{\partial \beta \partial \beta'} \right\}^{-1}$, and $\nabla p_{0_j} = (-\exp(\beta_0 + j\beta_1), -j\exp(\beta_0 + j\beta_1))'$ evaluated at $\hat{\vartheta}$.

**Case 2: Poisson Model**

When $\phi = 1$, the random variable $M$ follows Poisson distribution with $p.m.f.$ $P(M = m) = \dfrac{\eta^m}{m!} e^{-\eta}$ for $m = 0, 1, 2, \ldots$, and the cure rate is $\pi = e^{-\eta}$. The survival function for susceptible

population is

$$\begin{aligned} S_s^*(t) &= P(T > t \mid \text{Individual is susceptible}) \\ &= P(\min(W_1, W_2, \ldots, W_m) > t \mid m \geq 1) \\ &= \sum_{m=1}^{\infty} \left\{ P(W_1 > t) \cdots P(W_m > t) \frac{P(M = m)}{1 - P(M = 0)} \right\} \\ &= \sum_{m=1}^{\infty} [S(t)]^m \frac{\frac{\eta^m}{m!} e^{-\eta}}{1 - e^{-\eta}} \\ &= \frac{e^{\eta S(t)} - 1}{e^{\eta} - 1}. \end{aligned}$$

The corresponding density function is $f_s^*(t) = -[S_s^*(t)]' = \frac{\eta f(t) e^{\eta S(t) - \eta}}{1 - e^{-\eta}}$. Then, the log-likelihood function in (3.5) in this case is

$$\begin{aligned} l(t) &= \sum_{i \in I_1} log \left\{ (1 - e^{-\eta}) \frac{\eta f(t) e^{\eta S(t) - \eta}}{1 - e^{-\eta}} \right\} + \sum_{i \in I_0} log \left\{ e^{-\eta} + (1 - e^{-\eta}) \frac{e^{\eta S(t)} - 1}{e^{\eta} - 1} \right\} \\ &= \sum_{i=1}^{n} \left\{ \delta_i log \eta_i + \delta_i log f(t_i) + \eta_i S(t_i) - \eta_i \right\}. \end{aligned}$$

As in Case 1, under suitable regularity conditions, the asymptotic distribution of the $MLE$ $\hat{\vartheta}$ is normally distributed with covariance matrix $\Sigma(\hat{\vartheta})$ that can be estimated by $\hat{\Sigma}(\hat{\vartheta}) = \left\{ -\frac{\partial^2 l(\vartheta; t)}{\partial \vartheta \partial \vartheta'} \right\}^{-1}$, where the second derivatives are presented in Appendix A.

To obtain the cure rate for the $jth$ group of a generated sample of the Poisson model, $p_{0_j}$ is estimated by $\widehat{p_{0_j}} = e^{-e^{\hat{\beta_0} + j\hat{\beta_1}}}$. Let $\beta = (\beta_0, \beta_1)'$, then the corresponding estimate of the variance is obtained by $(\nabla p_{0_j})' Var(\beta) \nabla p_{0_j}$, where $Var(\beta)$ is approximated by $\hat{\Sigma}(\hat{\beta}) = \left\{ -\frac{\partial^2 l(\vartheta; t)}{\partial \beta \partial \beta'} \right\}^{-1}$, and $\nabla p_{0_j} = (-e^{-e^{\beta_0 + j\beta_1}} e^{\beta_0 + j\beta_1}, -j e^{-e^{\beta_0 + j\beta_1}} e^{\beta_0 + j\beta_1})'$ evaluated at $\hat{\vartheta}$.

## 3.2 Data Illustration

In this section, we provide an application of the methodology discussed in the previous section to a real data set. This data set is part of an assay, which evaluates the performance of

postoperative treatment on cutaneous melanoma (a type of malignant cancer) by using a high dosage of a particular drug (interferon alfa-2b). The data were taken from Ibrahim et al. (2001) (also see Kirkwood et al., 2000). The data set presents the survival times, recording the time (in years) until the patient's death or the censoring time, whichever comes first. The following data were collected from each patient as covariates: treatment (0: observation, n=204; 1: interferon, n=213); age (in years), nodule category (1: n=82; 2: n=87; 3: n=137; 4: n=111); sex (0: male, n=263; 1: female, n=154); performance status (0: fully active, n=363; 1: other, n=54); and tumor thickness (in $mm$). The original sample contains 427 patients, but 10 of them were removed from this study since their tumor thickness data were missing. The percentage of censored observations was 56%. In our illustration, we picked the nodule category as the only covariate $x$. Nodule category ranging from 1 to 4, respectively, is coded from the number of lymph nodes involved in the disease $(0, 1, 2-3, and \geq 4)$. In this case, the link function between the parameter $\eta$ and the covariate $x$ is taken to be in the form

$$\eta_i = \exp(\beta_0 + x_i\beta_1), \quad i = 1, 2, \ldots, 417.$$

The numerical calculation is carried out in the R system (R Development Core Team, 2009), and the computation codes are presented in Appendix B.

Since the normalization constant $Z(\eta; \phi)$ is not in a closed form, we approximate the value by truncating the numerical series. Based on (2.6), the ratio of $kth$ and $(k-1)th$ terms of the series is $\eta/k^{\phi}$. By choosing a small positive number $\varepsilon$, we take the smallest $k$ such that $k \geq (\eta/\varepsilon)^{1/\phi}$, using which we get the numerical approximation of the normalizing constant as $Z(\eta, \phi) \approx \sum_{j=0}^{k} \frac{\eta^j}{(j!)^{\phi}}$. All the remaining computation is based on this approximation.

After running the program in R, we obtained for the COM-Poisson model $\hat{\phi} \approx 0$. Thus, for the given data set, it is reasonable to assume a Geometric distribution for $M$. Table 3.1 presents the estimates and standard errors of the model parameters. Estimate of the cure rate $(p_0)$, grouped by the nodule category, is $\widehat{p_{0_j}} = 1 - \exp(\hat{\beta}_0 + j\hat{\beta}_1)$, for $j = 1, 2, 3, 4$. Then, the cure rates for the four nodule categories (1-4) are obtained to be 0.650, 0.557, 0.438, and 0.289,

Table 3.1: Maximum likelihood estimates of the parameters of the COM-Poisson (Geometric) cure rate model.

| Parameter | Estimate | Standard Error | p-Value |
|---|---|---|---|
| $\nu_1$ | $\exp(0.633) = 1.833$ | [0.125](0.027) | $< 0.0005$ |
| $\nu_2$ | -2.041 | [0.160](0.037) | $< 0.0005$ |
| $\beta_0$ | -1.287 | [0.142](0.133) | $< 0.0005$ |
| $\beta_1$ | 0.237 | [0.041](0.042) | $< 0.0005$ |

respectively. Note that, in Table 3.1, the standard errors (in the form of $(\cdot)$), obtained from the *gamlss* package seem to be severely under-estimated and are therefore not reliable in this case. However, the ones in the form of $[\cdot]$, computed from the second-order derivatives of the log-likelihood function (Geometric), seem more reasonable.

# Chapter 4

# Monte Carlo Simulation Algorithm and Results

## 4.1 Introduction

In this chapter, a Monte Carlo simulation study is performed to study the estimates of the unknown model parameters. Two model simulations, based on Geometric cure model and Poisson cure model, are carried out. The simulation process is based on the given cutaneous melanoma data. Therefore, we break each sample into four nodule groups (1-4) with corresponding sizes of $n_1, n_2, n_3$, and $n_4$, with $n = n_1 + n_2 + n_3 + n_4$ being the total sample size.

## 4.2 Monte Carlo Simulation Set-up, Bias and MSE for Geometric Cure Model

**Monte Carlo simulation set-up**:

**Step 1**: By choosing some values for the regression parameters $\beta_0$ and $\beta_1$, we fix the cure rates for the four nodule groups to be $p_{0_1}$, $p_{0_2}$, $p_{0_3}$, and $p_{0_4}$, where $p_{0_1} > p_{0_2} > p_{0_3} > p_{0_4}$.

Note 1: For the Geometric case, we have $\eta = \exp(\beta_0 + x\beta_1)$, where $x$=1, 2, 3, 4, and so $\beta_0 + x\beta_1$ must

less than 0.

Note 2: Since $p_{0_i}$s are in descending order and $p_0 = 1 - \eta$, we have $\beta_1 > 0$ and $\beta_0 < 0$.

**Step 2**: Fix the censoring rates for the four nodule groups to be $q_1$, $q_2$, $q_3$, and $q_4$, and also the distribution parameters $\nu_1$ and $\nu_2$ of the Weibull (in (3.2)), where $\nu_1 > 0$.

**Step 3**: For group $j$ ($j$=1, 2, 3, and 4), generate the competing cause random variable $M_i^j$ ($i = 1, \ldots, n_j$) from Geometric distribution of size $n_j$ with the cure rates determined from Step 1.

**Step 4**: Take the generated values from Step 3 by groups (from group 1 to group 4) and store them as $M = (M_1, M_2, \ldots, M_n)$.

**Step 5**: Record the nodule category $x$ as $(x_1, x_2, \ldots, x_n)$, where $x_1 = \cdots = x_{n_1} = 1, x_{n_1+1} = \cdots = x_{n_1+n_2} = 2, x_{n_1+n_2+1} = \cdots = x_{n_1+n_2+n_3} = 3$, and $x_{n_1+n_2+n_3+1} = \cdots = x_n = 4$.

**Step 6**: From Step 4, for the *ith* observation, if $M_i = 0$, set the lifetime as $Y_i = \infty$; otherwise, generate $M_i$ Weibull random variables $W = (W_1, W_2, \ldots, W_{M_i})$ with shape parameter=$\nu_1$ and scale parameter=$\exp(-\nu_2/\nu_1)$. Then, set $Y_i = \min(W_1, W_2, \ldots, W_{M_i}), i = 1, 2, \ldots, n$.

**Step 7**: For the desired censoring scheme $C = (C_1, C_2, \ldots, C_n)$, we assume $c_i \sim exponential(\theta)$, where $\theta$ is the scale parameter, $i = 1, 2, \ldots, n$.

**Step 8**: Let $T_i = \min(Y_i, C_i)$.

**Step 9**: If $Y_i < C_i$, set $\delta_i = 1$; otherwise, set $\delta_i = 0$.

**Step 10**: Run *gamlss* function, based on the simulated data ($T$, $\delta$ and $x$), to determine the MLEs of $\beta_0$, $\beta_1$, $\nu_1$, and $\nu_2$.

**Step 11**: Repeat Steps 3-10 B times.

In Step 7 above, the exponential parameter $\theta$ is calculated numerically. For each group, $\theta$ is different. Let us look at group $j$ ($j = 1, \ldots, 4$). From Steps 1 and 2, we know $P(\text{Cured}) = p_{0_j}$

and $P(\text{Censoring}) = q_j$. And

$$\begin{aligned} P(\text{Censoring}) &= P(\text{Censoring} \cap \text{Cured}) + P(\text{Censoring} \cap \text{ Not cured}) \\ &= P(\text{Censoring} \mid \text{Cured})P(\text{Cured}) + P(\text{Censoring} \mid \text{Not cured})P(\text{Not cured}). \end{aligned}$$

Thus, we get

$$P(\text{Censoring} \mid \text{Not cured}) = \frac{P(\text{Censoring}) - P(\text{Cured})}{P(\text{Not cured})} = \frac{q_j - p_{0_j}}{1 - p_{0_j}},$$

and so for given $\nu_1$ and $\nu_2$ from Step 2, we obtain

$$\begin{aligned} P(\text{Censoring} \mid \text{Not cured}) &= P(Y_i > C_i) \\ &= \int_0^\infty P(Y_i > C_i) f_{C_i}(C_i)\, dC_I \\ &= \int_0^\infty \exp(-C_i^{\nu_1} e^{\nu_2}) \frac{1}{\theta_j} e^{-C_i/\theta_j}\, dC_i \\ &= \int_0^\infty \exp(-z^{\nu_1} \exp(\nu_2 + \nu_1 log(\theta_j))) e^{-z}\, dz. \end{aligned}$$

Then, $\theta_j$ can be obtained by Monte Carlo estimation method.

For the purpose of cross-checking the simulation results, we produced two parallel chains of simulations of different model parameter choices with the average censoring rates to be 57.5% and 58.75%. The R code is presented in Appendix B. For both simulation processes, we took equal sample sizes for each nodule group, and let $n$=100, 200, 400 and 800. We simulated B=1000 times for each sample size. The marginal parameter histograms are produced in Figures 4.1 and 4.2, where the vertical line indicates the location of the true parameter. As can be seen, both $\nu_1$ and $\nu_2$ are biased for all sample sizes; on the other hand, the normal approximation for the regression parameters ($\beta_0$ and $\beta_1$) is quite satisfactory and the bias are negligible. Within each configuration, we estimated the bias and mean square error (MSE) for each model parameter. The results are summarized in Tables 4.1 and 4.2, from which we can observe that

the MSE decreases with increasing sample size. However, we observed that, even with different parameter choices, the estimates $\nu_1$ and $\nu_2$ are still biased, and for both choices the censoring rates are close to 60%, which is very high. So, it is reasonable to presume that the large bias in the estimates of the Weibull parameters is due to the high censoring rate. To verify this, we carried out another simulation. With careful selection of the model parameters, we adjusted the average censoring rate to be 17.5%. We conducted 1000 replications each with average censoring rate to be $15 - 20\%$. The corresponding histogram is presented in Figure 4.5, from which we observe that the bias in the estimates of $\nu_1$ and $\nu_2$ are greatly diminished. These simulation results display that the bias can be substantial under high censoring rate.

Table 4.1: Bias and MSE for the Geometric cure model with $\vartheta = (\beta_0 = -1.301, \beta_1 = 0.251, \nu_1 = \exp(0.4), \nu_2 = -2)$ and censoring rate=55-60%.

| | $\nu_1$ | | $\nu_2$ | | $\beta_0$ | | $\beta_1$ | |
|---|---|---|---|---|---|---|---|---|
| Size | Bias | MSE | Bias | MSE | Bias | MSE | Bias | MSE |
| n=100 | -0.1917 | 0.0453 | 0.2025 | 0.0885 | -0.0118 | 0.0775 | -0.005 | 0.0061 |
| n=200 | -0.1705 | 0.0347 | 0.2056 | 0.0714 | -0.0173 | 0.0436 | -0.0034 | 0.0033 |
| n=400 | -0.1495 | 0.0252 | 0.1955 | 0.0564 | -0.0175 | 0.0204 | -0.0032 | 0.0015 |
| n=800 | -0.1247 | 0.0175 | 0.172 | 0.0402 | -0.0118 | 0.0105 | -0.0029 | 0.0008 |

Table 4.2: Bias and MSE for the Geometric cure model with $\vartheta = (\beta_0 = -1, \beta_1 = 0.15, \nu_1 = \exp(0.25), \nu_2 = -1.5)$ and censoring rate=55-60%.

| | $\nu_1$ | | $\nu_2$ | | $\beta_0$ | | $\beta_1$ | |
|---|---|---|---|---|---|---|---|---|
| Size | Bias | MSE | Bias | MSE | Bias | MSE | Bias | MSE |
| n=100 | -0.096 | 0.0176 | 0.0423 | 0.0503 | -0.0204 | 0.0793 | 0.0001 | 0.007 |
| n=200 | -0.0908 | 0.0129 | 0.0652 | 0.0278 | -0.0116 | 0.0346 | -0.0006 | 0.0031 |
| n=400 | -0.0839 | 0.0098 | 0.0627 | 0.017 | -0.0078 | 0.0199 | -0.0013 | 0.0017 |
| n=800 | -0.0672 | 0.006 | 0.0682 | 0.0115 | -0.0037 | 0.0094 | -0.0021 | 0.0008 |

## 4.3 Monte Carlo Simulation Set-up, Bias, and MSE for Poisson Cure Model

**Monte Carlo simulation set-up**

**Step 1**: By choosing some values for the regression parameters $\beta_0$ and $\beta_1$, we fix the cure rates for the four nodule groups to be $p_{0_1}, p_{0_2}, p_{0_3}$, and $p_{0_4}$, where $p_{0_1} > p_{0_2} > p_{0_3} > p_{0_4}$.

Note: Given $\eta = \exp(\beta_0 + x\beta_1)$ for $x = 1, 2, 3$, and 4, since $p_{0_i}$s are in descending order and $p_0 = \exp(-\eta)$, we have $\beta_1 > 0$.

**Step 2**: Fix the censoring rates for the four nodule groups to be $q_1, q_2, q_3$, and $q_4$, and also the distribution parameters $\nu_1$ and $\nu_2$ of the Weibull (in (3.2)), where $\nu_1 > 0$.

**Step 3**: For group $j$ ($j = 1, 2, 3$, and 4), generate the competing cause random variable $M_i^j$ ($i = 1, \ldots, n_j$) from Poisson distribution of size $n_j$ with the cure rates determined from Step 1.

**Step 4**: Take the generated values from Step 3 by groups (from group 1 to group 4) and store them as $M = (M_1, M_2, \ldots, M_n)$.

**Step 5**: Record the nodule category $x$ as $(x_1, x_2, \ldots, x_n)$, where $x_1 = \cdots = x_{n_1} = 1, x_{n_1+1} = \cdots = x_{n_1+n_2} = 2, x_{n_1+n_2+1} = \cdots = x_{n_1+n_2+n_3} = 3$, and $x_{n_1+n_2+n_3+1} = \cdots = x_n = 4$.

**Step 6**: From Step 4, for the $ith$ observation, if $M_i = 0$, set the lifetime as $Y_i = \infty$; otherwise, generate $M_i$ Weibull random variables $W = (W_1, W_2, \ldots, W_{M_i})$ with shape parameter=$\nu_1$ and scale parameter=$\exp(-\nu_2/\nu_1)$. Then, set $Y_i = \min(W_1, W_2, \ldots, W_{M_i}), i = 1, 2, \ldots, n$.

**Step 7**: For the desired censoring scheme $C = (C_1, C_2, \ldots, C_n)$, we assume $c_i \sim exponential(\theta)$, where $\theta$ is the scale parameter, $i = 1, 2, \ldots, n$. (The parameter $\theta$ is calculated numerically same as in the Geometric case; see to Section 4.1.)

**Step 8**: Let $T_i = \min(Y_i, C_i)$.

**Step 9**: If $Y_i < C_i$, set $\delta_i = 1$; otherwise, set $\delta_i = 0$.

**Step 10**: Run *gamlss* function, based on the simulated data ($T$, $\delta$ and $x$), to determine the MLEs of $\beta_0$, $\beta_1$, $\nu_1$, and $\nu_2$.

**Step 11**: Repeat Steps 3-10 B times.

As discussed in previous section, in order to cross-check the simulation results, we produced two parallel chains of simulations of different model parameter choices with the average censoring rates to be 58.75% and 57.5%. For both simulation processes, we took equal sample sizes for each nodule group, and let $n$=100, 200, 400 and 800. We simulated B=1000 times for each sample size. The marginal parameter histograms are produced in Figures 4.3 and 4.4, where the vertical line indicate the location of the true parameter. As can be seen, once again the estimates of both $\nu_1$ and $\nu_2$ are seriously biased; on the other hand, the bias in the estimates $\beta_0$ and $\beta_1$ is insignificant. So, we attempted to reduce the bias by first-order jackknife (Adams et al, 1971) defined as

$$J^{(1)}[\hat{\vartheta}] \quad = \quad n\hat{\vartheta} - \frac{n-1}{n}\sum_{i-1}^{n}\hat{\vartheta}^i,$$

where $\hat{\vartheta}^i$ is the MLE with $ith$ observation removed. However, the jackknife process was not effective as can be seen in Table 4.3 produced with true parameter as $\vartheta = (\beta_0 = -1.17, \beta_1 = 0.3277, \nu_1 = \exp(0.6), \nu_2 = -0.1)$.

Table 4.3: Jackknifing of the estimates.

| | $\hat{\nu_1}$ | $\hat{\nu_2}$ | $\hat{\beta_0}$ | $\hat{\beta_1}$ |
|---|---|---|---|---|
| Before jackknifing | 1.5555 | -0.1045 | -1.1968 | 0.3445 |
| After jackknifing | 1.5425 | -0.1233 | -1.1626 | 0.3338 |

Within each configuration, we estimated the bias and MSE for each model parameter. These results are summarized in Tables 4.4 and 4.5, from which we observe that MSE decreases as sample size increases. However, we also note that even with different parameter choices, the estimates of $\nu_1$ and $\nu_2$ are still biased, and for both choices the censoring rates are close to 60%,

which is very high. So, it is reasonable to presume that the bias in the estimates of the Weibull parameters is due to high censoring rate. To check this, we carried out another simulation. With another selection of the model parameters, we set the average censoring rate to be 17.5%. We produced 1000 replications each with average censoring rate to be 15-20%. The histogram is presented in Figure 4.6. Upon comparing Figures 4.3, 4.4, and 4.6, we observe that the bias in the estimates of $\nu_1$ and $\nu_2$ are evidently reduced in this case. Therefore, it is reasonable to state that heavy censoring results in biasing the estimates of $\nu_1$ and $\nu_2$.

Table 4.4: Bias and MSE for the Poisson cure model with $\vartheta = (\beta_0 = -1.17, \beta_1 = 0.3277, \nu_1 = \exp(0.6), \nu_2 = -0.1)$ and censoring rate=55-60%.

| | $\nu_1$ | | $\nu_2$ | | $\beta_0$ | | $\beta_1$ | |
|---|---|---|---|---|---|---|---|---|
| Size | Bias | MSE | Bias | MSE | Bias | MSE | Bias | MSE |
| n=100 | -0.351 | 0.1381 | -0.1047 | 0.0312 | -0.022 | 0.2116 | 0.0089 | 0.0228 |
| n=200 | -0.3229 | 0.1123 | -0.0939 | 0.0204 | 0.0174 | 0.0981 | -0.0047 | 0.0109 |
| n=400 | -0.2815 | 0.0841 | -0.0735 | 0.0115 | 0.0018 | 0.0463 | 0.0025 | 0.005 |
| n=800 | -0.2471 | 0.0635 | -0.0536 | 0.0062 | 0.0067 | 0.0243 | -0.0022 | 0.0026 |

Table 4.5: Bias and MSE for the Poisson cure model with $\vartheta = (\beta_0 = -1.4835, \beta_1 = 0.4525, \nu_1 = \exp(0.3), \nu_2 = -1.2)$ and censoring rate=55-60%.

| | $\nu_1$ | | $\nu_2$ | | $\beta_0$ | | $\beta_1$ | |
|---|---|---|---|---|---|---|---|---|
| Size | Bias | MSE | Bias | MSE | Bias | MSE | Bias | MSE |
| n=100 | -0.1053 | 0.0163 | 0.049 | 0.0309 | -0.0165 | 0.2248 | 0.0061 | 0.0232 |
| n=200 | -0.1021 | 0.0139 | 0.0548 | 0.0183 | -0.0009 | 0.1102 | -0.0007 | 0.0112 |
| n=400 | -0.0937 | 0.011 | 0.0628 | 0.0125 | -0.0059 | 0.0526 | 0.0014 | 0.0054 |
| n=800 | -0.0809 | 0.0081 | 0.0616 | 0.0082 | 0.0025 | 0.0252 | -0.0015 | 0.0027 |

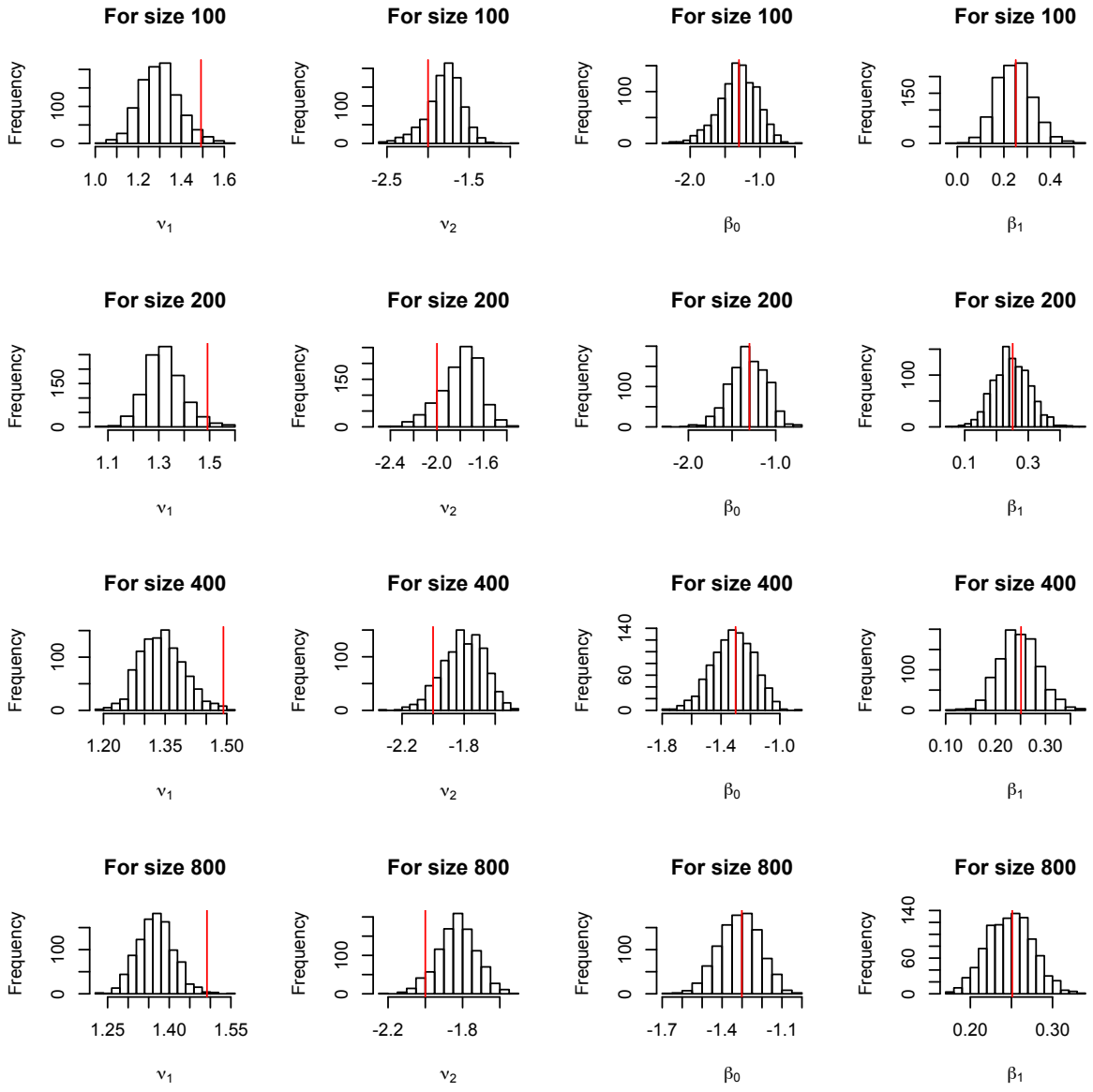


Figure 4.1: Geometric case: histograms of estimates for the parameter set $\vartheta = (\beta_0 = -1.301, \beta_1 = 0.251, \nu_1 = \exp(0.4), \nu_2 = -2)$ with censoring rate=55-60%.

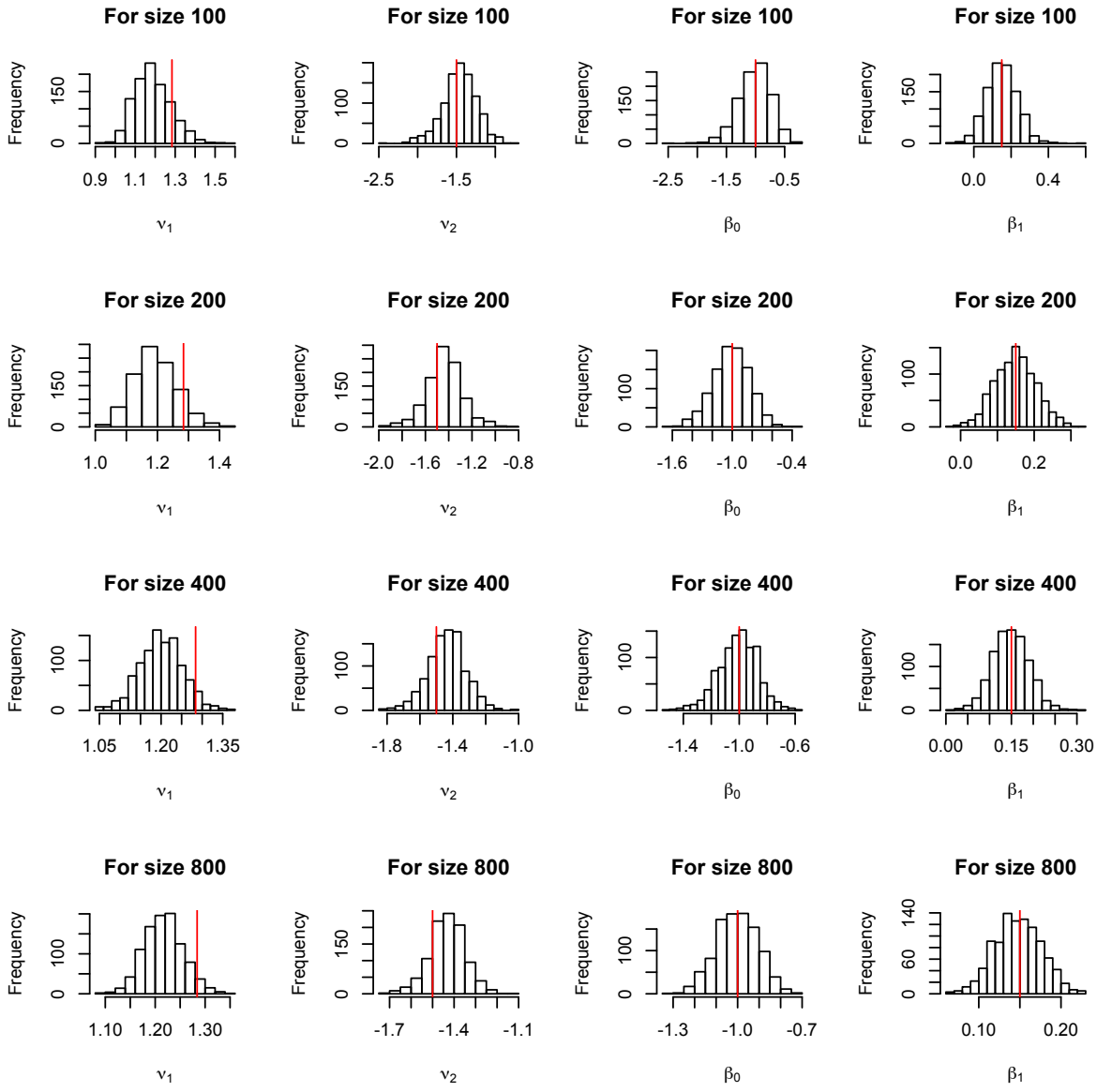


Figure 4.2: Geometric case: histograms of estimates for the parameter set $\vartheta = (\beta_0 = -1, \beta_1 = 0.15, \nu_1 = \exp(0.25), \nu_2 = -1.5)$ with censoring rate=55-60%.

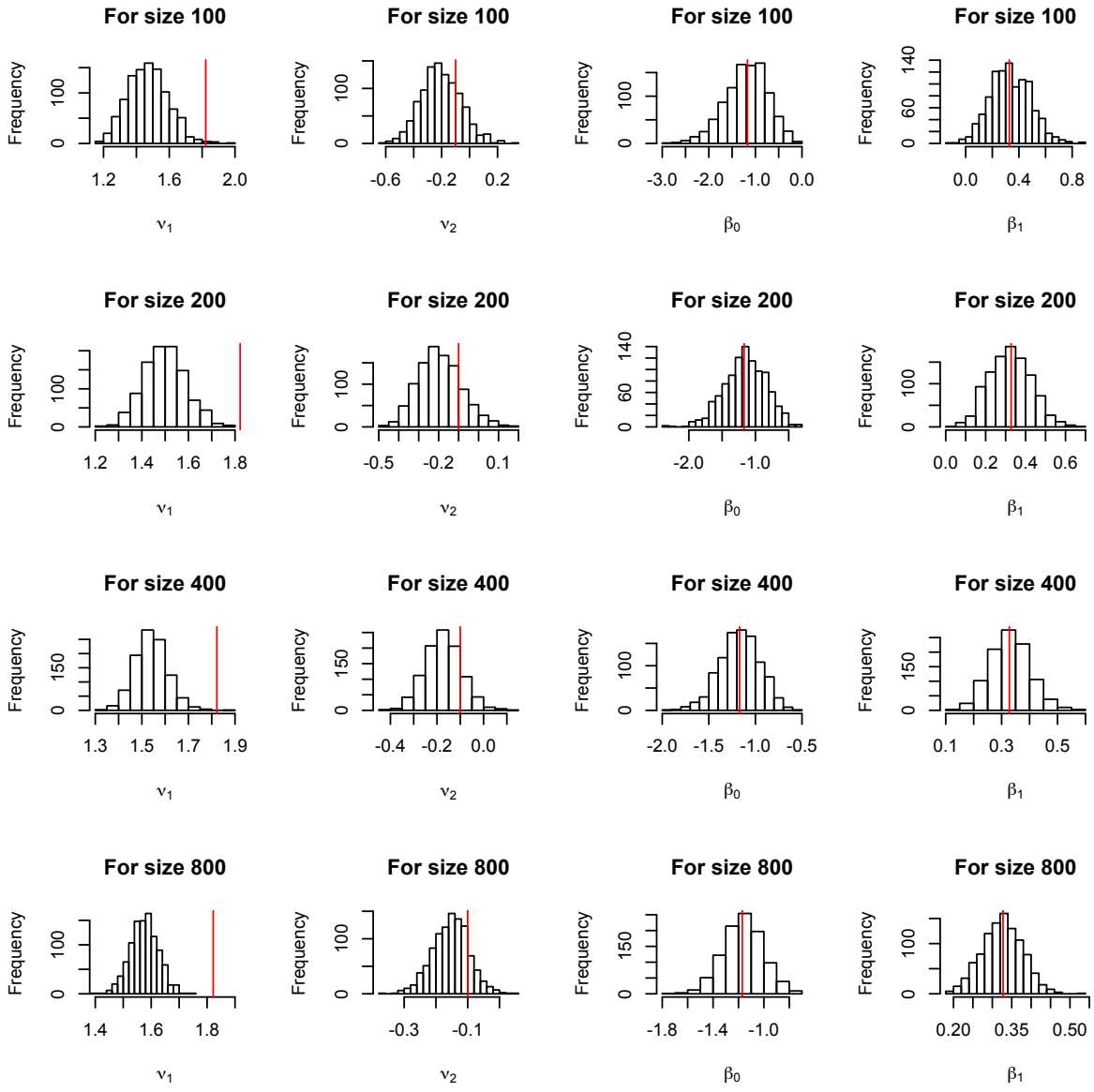


Figure 4.3: Poisson case: histograms of estimates for the parameter set $\vartheta = (\beta_0 = -1.17, \beta_1 = 0.3277, \nu_1 = \exp(0.6), \nu_2 = -0.1)$ with censoring rate=55-60%.

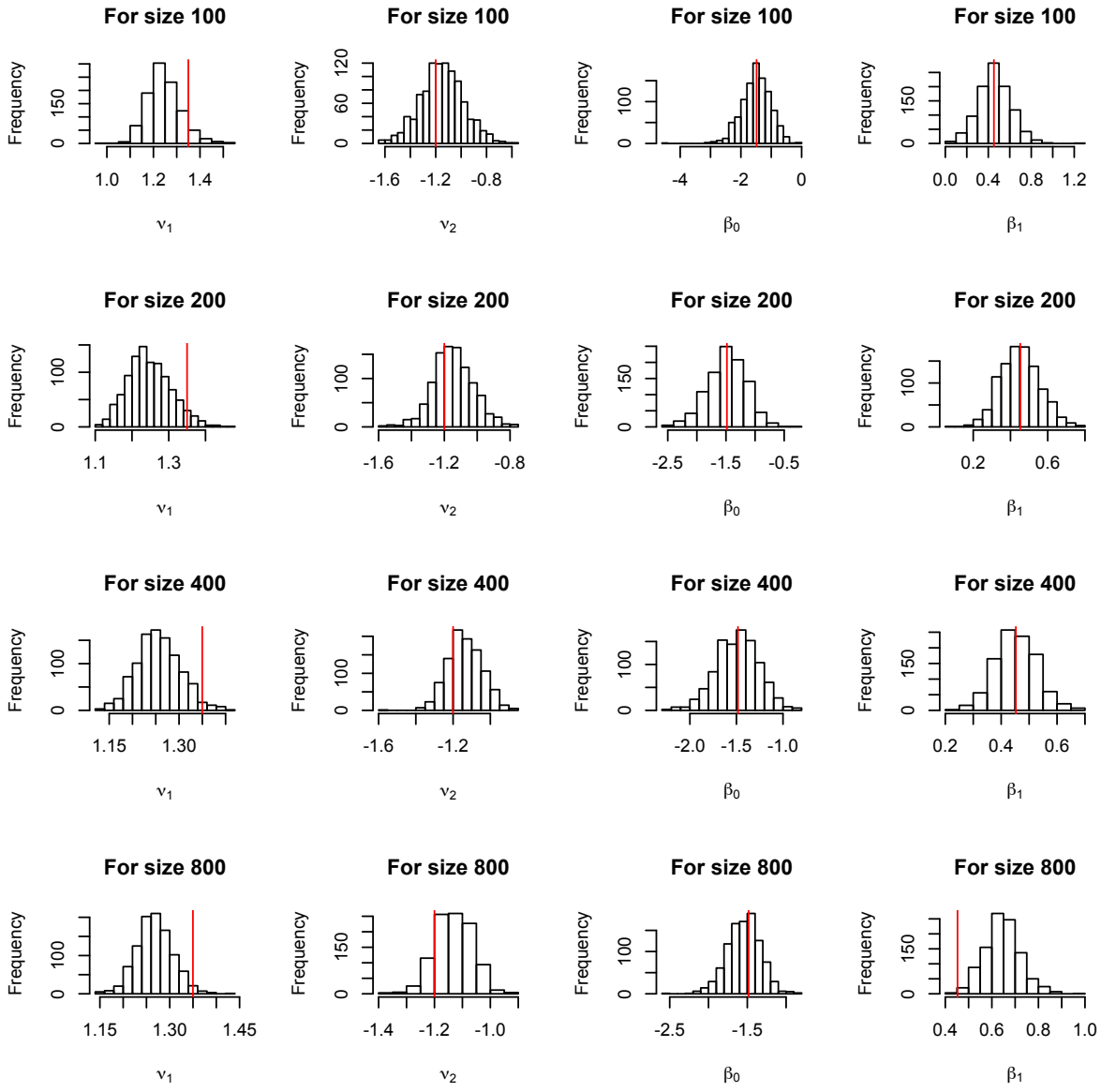


Figure 4.4: Poisson case: histograms of estimates for the parameter set $\vartheta = (\beta_0 = -1.4835, \beta_1 = 0.4525, \nu_1 = \exp(0.3), \nu_2 = -1.2)$ with censoring rate=55-60%.

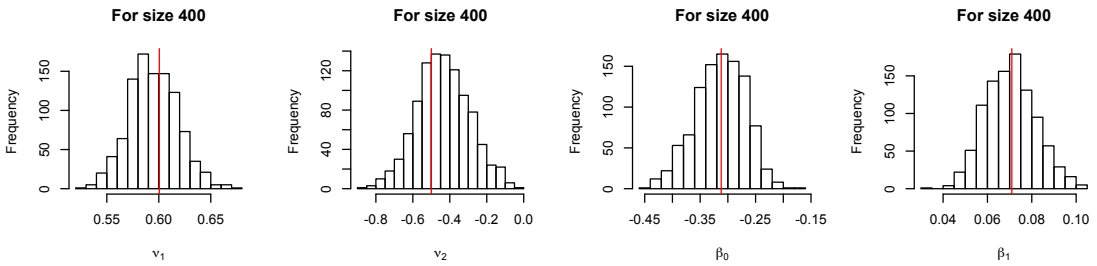


Figure 4.5: Geometric case: histograms of estimates for the parameter set $\vartheta = (\beta_0 = -0.312, \beta_1 = 0.071, \nu_1 = \exp(-0.51), \nu_2 = -0.5)$ with censoring rate=15-20%.

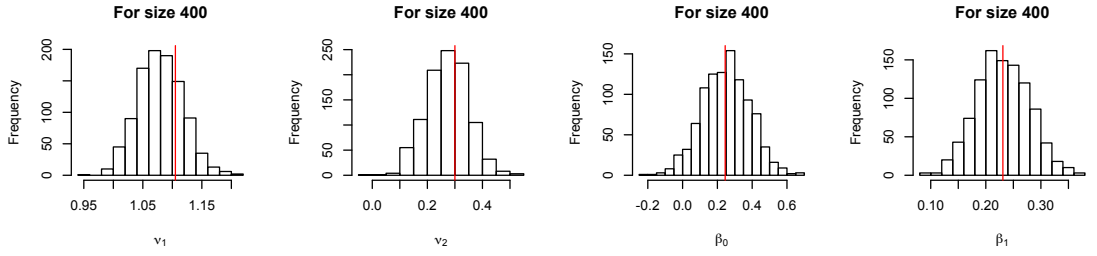


Figure 4.6: Poisson case: histograms of estimates for the parameter set $\vartheta = (\beta_0 = 0.2448, \beta_1 = 0.2310, \nu_1 = \exp(0.1), \nu_2 = 0.3)$ with censoring rate=15-20%.

# Chapter 5

# Interval Estimation and Bootstrap

## 5.1 Interval Estimation based on Normal Approximation

As mentioned earlier, under certain regularity conditions, the MLEs are asymptotically unbiased and efficient. As the sample size approaches infinity, the bias tends to zero and the variances approaches the Cramer-Rao lower bound. Furthermore, their limiting distribution is normal with variance-covariance matrix approximated by the inverse of the Fisher information matrix (see Silvey, 1975; Casella and Berger, 2002). Therefore, inference about the parameters can be based on the asymptotic normality of the MLEs. Under these assumptions, we present an approximate method to construct CIs for $\vartheta$ by using the large-sample properties of the MLEs.

**Model 1: Geometric Cure Model**

The simulation is carried out with two sets of parameter configurations, each with sample sizes 100, 200, 400 and 800. Refer to the simulation process and the results in Section 4.1, from where we calculate the MLE of $\vartheta$. As discussed in Section 3.1, the standard errors can be obtained in two ways, with one is based on the summary function of the *gamlss* package and the other based on the Fisher information matrix. In this case, we found the SEs produced by the former to be not reliable, and so we applied the second method for this purpose. The variance-

covariance is estimated by $\hat{\Sigma}(\hat{\vartheta}) = \left\{ -\frac{\partial^2 l(\vartheta; t)}{\partial \vartheta \partial \vartheta'} \right\}^{-1}$. The computational R code is presented in Appendix B. The coverage probabilities for $\vartheta$ are presented in Tables 5.1 and 5.2. We observe the coverage for the regression parameters be to excellent for various sample sizes, but the coverage probabilities for the Weibull parameters decrease as sample size increases. Although the coverage is somewhat close to 95% for small sample sizes in the second configuration, but as mentioned earlier in Section 4.1, the bias of the estimates of the Weibull parameters are fairly consistent, and the normality assumption fails in the case of Geometric cure model.

Table 5.1: Coverage probability with 95% CIs for the Geometric cure model with $\vartheta = (\beta_0 = -1.301, \beta_1 = 0.251, \nu_1 = \exp(0.4), \nu_2 = -2)$ and censoring rate=55-60%.

| | Coverage Probability | | | |
|---|---|---|---|---|
| Size | $\nu_1$ | $\nu_2$ | $\beta_0$ | $\beta_1$ |
| 100 | 0.875 | 0.961 | 0.962 | 0.961 |
| 200 | 0.733 | 0.882 | 0.961 | 0.959 |
| 400 | 0.545 | 0.704 | 0.969 | 0.969 |
| 800 | 0.403 | 0.608 | 0.968 | 0.968 |

Table 5.2: Coverage probability with 95% CIs for the Geometric cure model with $\vartheta = (\beta_0 = -1, \beta_1 = 0.15, \nu_1 = \exp(0.25), \nu_2 = -1.5)$ and censoring rate=55-60%.

| | Coverage Probability | | | |
|---|---|---|---|---|
| Size | $\nu_1$ | $\nu_2$ | $\beta_0$ | $\beta_1$ |
| 100 | 0.983 | 0.983 | 0.959 | 0.953 |
| 200 | 0.943 | 0.966 | 0.968 | 0.959 |
| 400 | 0.857 | 0.958 | 0.946 | 0.945 |
| 800 | 0.807 | 0.931 | 0.954 | 0.963 |

**Model 2: Poisson Cure Model**

The simulation set-up for this model is similar to that of the Geometric cure model. Two sets of parameter configurations, each with sample sizes of 100, 200, 400 and 800, are generated. The MLEs are produced as in Section 4.2, also with the simulation set-up. The standard errors are determined from the variance-covariance matrix calculated by $\hat{\Sigma}(\hat{\vartheta}) = \left\{ -\frac{\partial^2 l(\vartheta; t)}{\partial \vartheta \partial \vartheta'} \right\}^{-1}$. The computational R code is presented in Appendix B. The coverage probability for $\vartheta$ is shown in Tables 5.3 and 5.4. As in the Geometric case, the coverage for the Weibull parameters is higher

for smaller sample sizes. The coverage for the regression parameters is consistent and close to 95%, while the coverage for $\nu_1$ and $\nu_2$ are quite poor due to the corresponding estimates being seriously biased. Thus, the normal approximation seems to be unsatisfactory in this case.

Table 5.3: Coverage probability with 95% CIs for the Poisson cure model with $\vartheta = (\beta_0 = -1.17, \beta_1 = 0.3277, \nu_1 = \exp(0.6), \nu_2 = -0.1)$ and censoring rate=55-60%.

| | Coverage Probability | | | |
|---|---|---|---|---|
| Size | $\nu_1$ | $\nu_2$ | $\beta_0$ | $\beta_1$ |
| 100 | 0.587 | 0.989 | 0.947 | 0.952 |
| 200 | 0.3 | 0.979 | 0.949 | 0.951 |
| 400 | 0.12 | 0.953 | 0.957 | 0.955 |
| 800 | 0.024 | 0.931 | 0.95 | 0.948 |

Table 5.4: Coverage probability with 95% CIs for the Poisson cure model with $\vartheta = (\beta_0 = -1.4835, \beta_1 = 0.4525, \nu_1 = \exp(0.3), \nu_2 = -1.2)$ and censoring rate=55-60%.

| | Coverage Probability | | | |
|---|---|---|---|---|
| Size | $\nu_1$ | $\nu_2$ | $\beta_0$ | $\beta_1$ |
| 100 | 0.995 | 0.99 | 0.957 | 0.948 |
| 200 | 0.969 | 0.968 | 0.95 | 0.949 |
| 400 | 0.869 | 0.955 | 0.954 | 0.947 |
| 800 | 0.736 | 0.918 | 0.962 | 0.958 |

## 5.2 Bootstrap Algorithm and Interval estimation

In this section, two bootstrap methods (parametric and non-parametric) are described. We then construct CI for $\vartheta$ by using the parametric bootstrap method and then examine the coverage probability for $\vartheta$ based on this CI. Following are the two algorithms that are implemented to carry out the computation based on the samples that are generated from the Monte Carlo simulation described earlier in Chapter 4.

**Method 1: Non-parametric Bootstrap**

**Step 1**: For a given sample that is generated (see Chapter 4) with MLEs ($\hat{\beta}_0, \hat{\beta}_1, \hat{\nu_1}$, and $\hat{\nu_2}$), we sample the observations $(T_i, \delta_i, x_i)$ with replacement of size $n$.

**Step 2**: Run *gamlss* function with this sample to obtain the new MLEs, denoted by $(\hat{\beta_0}^*, \hat{\beta_1}^*, \hat{\nu_1}^*$, and $\hat{\nu_2}^*)$.

**Step 3**: Repeat Steps 1-2 B times. Then, arrange all the new MLEs in ascending order to obtain the bootstrap samples

$$\left\{\hat{\beta_0}^{*[1]} < \hat{\beta_0}^{*[2]} < \cdots < \hat{\beta_0}^{*[B]}\right\},$$
$$\left\{\hat{\beta_1}^{*[1]} < \hat{\beta_1}^{*[2]} < \cdots < \hat{\beta_1}^{*[B]}\right\},$$
$$\left\{\hat{\nu_1}^{*[1]} < \hat{\nu_1}^{*[2]} < \cdots < \hat{\nu_1}^{*[B]}\right\},$$
$$\left\{\hat{\nu_2}^{*[1]} < \hat{\nu_2}^{*[2]} < \cdots < \hat{\nu_2}^{*[B]}\right\}.$$

With these bootstrap samples, we now obtain the two-sided $100(1-\alpha)\%$ bootstrap CIs for $\beta_0, \beta_1, \nu_1$, and $\nu_2$ as $\left(\hat{\beta_0}^*_{(\alpha/2)}, \hat{\beta_0}^*_{(1-\alpha/2)}\right)$, $\left(\hat{\beta_1}^*_{(\alpha/2)}, \hat{\beta_1}^*_{(1-\alpha/2)}\right)$, $\left(\hat{\nu_1}^*_{(\alpha/2)}, \hat{\nu_1}^*_{(1-\alpha/2)}\right)$, and $\left(\hat{\nu_2}^*_{(\alpha/2)}, \hat{\nu_2}^*_{(1-\alpha/2)}\right)$ respectively. Record whether the true parameters ($\beta_0, \beta_1, \nu_1$, and $\nu_2$) fall inside the intervals.

**Step 4**: Repeat Steps 1-3 for every sample within the same configuration (from Chapter 4), and then calculate the coverage probabilities for the four parameters.

**Method 2: Parametric Bootstrap**

**Step 1**: Obtain one generated sample with MLEs ($\hat{\beta_0}, \hat{\beta_1}, \hat{\nu_1}$, and $\hat{\nu_2}$) of the corresponding configuration with true parameter$\vartheta$=($\beta_0, \beta_1, \nu_1$, and $\nu_2$) detailed in chapter 4. And then calculate the MLE for $\theta$, $\hat{\theta}$=($\hat{\theta_1}$, $\hat{\theta_2}$, $\hat{\theta_3}$, and $\hat{\theta_4}$)$'$.

**Step 2**: For $jth$ nodule group ($j$=1, 2, 3, and 4), generate the competing cause random variable $M_i^j$ $(i = 1, \ldots, n_j)$ from Geometric (or Poisson) distribution of size $n_j$. Take the generated values and store them as $M = (M_1, M_2, \ldots, M_n)$.

**Step 3**: Record the nodule category $x$ as $(x_1, x_2, \ldots, x_n)$, where $x_1 = \cdots = x_{n_1} = 1, x_{n_1+1} = \cdots = x_{n_1+n_2} = 2, x_{n_1+n_2+1} = \cdots = x_{n_1+n_2+n_3} = 3$, and $x_{n_1+n_2+n_3+1} = \cdots = x_n = 4$.

**Step 4**: From Step 2, for *ith* observation, if $M_i = 0$, set the lifetime as $Y_i = \infty$; otherwise, generate $M_i$ Weibull random variables $W = (W_1, W_2, \ldots, W_{M_i})$ with shape parameter=$\hat{\nu_1}$ and scale parameter=$\exp(-\hat{\nu_2}/\hat{\nu_1})$. Then, set $Y_i = \min(W_1, W_2, \ldots, W_{M_i}), i = 1, 2, \ldots, n$.

**Step 5**: For the desired censoring scheme $C = (C_1, C_2, \ldots, C_n)$, we assume $c_i \sim exponential(\hat{\theta})$, where $\hat{\theta}$ is the scale parameter, $i = 1, 2, \ldots, n$.

**Step 6**: Let $T_i = \min(Y_i, C_i)$ and if $Y_i < C_i$, set $\delta_i = 1$; otherwise, set $\delta_i = 0$.

**Step 7**: By the obtained sample $(T, \delta, x)$, run *gamlss* function to calculate the new $MLEs$, denoted by $(\hat{\beta_0}^*, \hat{\beta_1}^*, \hat{\nu_1}^*$, and $\hat{\nu_2}^*)$.

**Step 8**: Repeat Steps 2-7 B times. Then, arrange all the new $MLEs$ in ascending order to obtain the bootstrap samples of

$$\left\{\hat{\beta_0}^{*[1]} < \hat{\beta_0}^{*[2]} < \cdots < \hat{\beta_0}^{*[B]}\right\},$$
$$\left\{\hat{\beta_1}^{*[1]} < \hat{\beta_1}^{*[2]} < \cdots < \hat{\beta_1}^{*[B]}\right\},$$
$$\left\{\hat{\nu_1}^{*[1]} < \hat{\nu_1}^{*[2]} < \cdots < \hat{\nu_1}^{*[B]}\right\},$$
$$\left\{\hat{\nu_2}^{*[1]} < \hat{\nu_2}^{*[2]} < \cdots < \hat{\nu_2}^{*[B]}\right\}.$$

With these bootstrap samples, we now obtain the two-sided $100(1-\alpha)\%$ bootstrap CIs for $\beta_0, \beta_1, \nu_1$, and $\nu_2$ are $\left(\hat{\beta_0}^*_{(\alpha/2)}, \hat{\beta_0}^*_{(1-\alpha/2)}\right)$, $\left(\hat{\beta_1}^*_{(\alpha/2)}, \hat{\beta_1}^*_{(1-\alpha/2)}\right)$, $\left(\hat{\nu_1}^*_{(\alpha/2)}, \hat{\nu_1}^*_{(1-\alpha/2)}\right)$, and $\left(\hat{\nu_2}^*_{(\alpha/2)}, \hat{\nu_2}^*_{(1-\alpha/2)}\right)$, respectively. Record whether ($\beta_0, \beta_1, \nu_1$, and $\nu_2$) fall inside the intervals.

**Step 9**: Repeat Steps 1-8 $M$ times with a different sample within the same configuration, and then obtain the coverage probabilities of the parameters.

In Step 1, the $MLE$ $\hat{\theta}$ is obtained in the following way:

We assumed $C_i \sim exponential(\theta)$, where $\theta$ is the scale parameter and $T_i = \min(Y_i, C_i)$. Let $I_0 = \{i \in (1, 2, \ldots, n) : \delta_i = 0\}$ and $I_1 = \{i \in (1, 2, \ldots, n) : \delta_i = 1\}$, so that the likelihood function

can be written as

$$\begin{aligned} L(\theta) &= \prod_{i \in I_1} e^{-Y_i/\theta} \prod_{i \in I_0} \frac{1}{\theta} e^{-C_i/\theta} \\ &= \frac{1}{\theta^{|I_0|}} \exp\left( -\sum_{i=1}^{n} \frac{T_i}{\theta} \right), \end{aligned}$$

where $|I_0|$ is the cardinality of the set $I_0$. So, the log-likelihood function is obtained as

$$l(\theta) = |I_0| log(\theta) - \sum_{i=1}^{n} \frac{T_i}{\theta},$$

and consequently the MLE is derived as

$$\hat{\theta} = \frac{1}{|I_0|} \sum_{i=1}^{n} T_i.$$

By comparing the two bootstrap methods, we observe the parametric bootstrap method to be a preferable one. Since the sampling method for non-parametric bootstrap involves replacement, if there is heavy censoring, it results in a significant loss of information. So, the parametric bootstrap is chosen for the numerical study. Ideally, in order to get a sufficient interval counts, we should set B=1000 and M=1000. However, the calculation is enormously large and it will take tremendous amount of time to complete this simulation and consequently the computation process cannot be finished with a limited time span. In order to get a general idea of how well the method performs, we conducted a small bootstrap calculation with the Poisson cure model and a sample of size 400. In order to meet the minimum requirement of obtaining a 95% CI, we let B=200 and M=10. The corresponding result is presented in Table 5.5.

Table 5.5: Comparison between the large-sample normal approximation and the bootstrap method.

| | Coverage Probability | | | |
|---|---|---|---|---|
| Method | $\nu_1$ | $\nu_2$ | $\beta_0$ | $\beta_1$ |
| Approximate | 0.12 | 0.953 | 0.957 | 0.955 |
| Bootstrap | 1/10 | 9/10 | 9/10 | 8/10 |

## 5.3 Comments on the Comparison

In Table 5.5, we observe that the coverage probabilities for the model parameters are quite similar for the normal approximation and parametric bootstrap methods. Although some other methods can also brough into this comparison such as the Bias-corrected and accelerated (BCa) percentile bootstrap method and Studentized-t bootstrap method, the computational time will also be tremendous, but this would still be worthwhile doing. Furthermore, as a future work, it could also be desirable to develop a new method for a computationally efficient maximization process of the log-likelihood function. In addition, since in this cure rate context we are likely to face high censoring rate in the data, work can also be done to reduce the bias of the MLEs of the model parameters, which would greatly improve the coverage probability. All these problems would be of interest for further study.

# Appendix A

# Derivations

## A.1 First derivatives of $\log f_{pop}(t)$ (from Section 3.1)

We have

$$\frac{\partial}{\partial \eta} \log f_{pop}(t;\eta,\phi,\nu) = S(t;\nu) \frac{\sum_{j=1}^{\infty} \frac{j^2 \{\eta S(t;\nu)\}^{j-1}}{(j!)^{\phi}}}{\sum_{j=1}^{\infty} \frac{j \{\eta S(t;\nu)\}^{j}}{(j!)^{\phi}}} - \frac{1}{\eta Z(\eta,\phi)} \sum_{j=1}^{\infty} \frac{j\eta^j}{(j!)^{\phi}},$$

$$\frac{\partial}{\partial \phi} \log f_{pop}(t;\eta,\phi,\nu) = -\frac{\sum_{j=1}^{\infty} \frac{j \log(j!) \{\eta S(t;\nu)\}^{j}}{(j!)^{\phi}}}{\sum_{j=1}^{\infty} \frac{j \{\eta S(t;\nu)\}^{j}}{(j!)^{\phi}}} - \frac{1}{Z(\eta,\phi)} \sum_{j=1}^{\infty} \frac{j \log(j!)}{(j!)^{\phi}},$$

and

$$\frac{\partial}{\partial \nu_l} \log f_{pop}(t;\eta,\phi,\nu) = \frac{1}{f(t;\nu)} \frac{\partial}{\partial \nu_l} f(t;\nu) + \frac{\partial}{\partial \nu_l} F(t;\nu) \left\{ \frac{1}{S(t;\nu)} - \frac{\sum_{j=1}^{\infty} \frac{j^2 \{\eta S(t;\nu)\}^{j-1}}{(j!)^{\phi}}}{\sum_{j=1}^{\infty} \frac{j \{\eta S(t;\nu)\}^{j}}{(j!)^{\phi}}} \right\}$$

for $l$=1 and 2. In addition

$$\frac{\partial}{\partial \nu_1} f(t;\nu) = t^{\nu_1 - 1} \exp(\nu_2 - t^{\nu_1} e^{\nu_2}) \left\{1 + \nu_1 \log t (1 - t^{\nu_1} e^{\nu_2})\right\},$$
$$\frac{\partial}{\partial \nu_2} f(t;\nu) = \nu_1 t^{\nu_1 - 1} \exp(\nu_2 - t^{\nu_1} e^{\nu_2})(1 - t^{\nu_1} e^{\nu_2}),$$
$$\frac{\partial}{\partial \nu_2} F(t;\nu) = t^{\nu_1} \exp(\nu_2 - t^{\nu_1} e^{\nu_2}),$$

*and*

$$\frac{\partial}{\partial \nu_1} F(t;\nu) = \log t \frac{\partial}{\partial \nu_2} F(t;\nu)$$

## A.2 Second derivatives of the log-likelihood function for the Geometric cure model

$$\frac{\partial^2 l(\vartheta;t)}{\partial \nu_m \partial \nu_n} = \sum_{i=1}^{n} \left\{ \delta_i \frac{\partial^2 \log f(t_i)}{\partial \nu_m \partial \nu_n} + (1+\delta_i) \frac{\eta_i \frac{\partial^2 S(t_i)}{\partial \nu_m \partial \nu_n}(1 - \eta_i S(t_i)) + {\eta_i}^2 \frac{\partial S(t_i)}{\partial \nu_m} \frac{\partial S(t_i)}{\partial \nu_n}}{(1 - \eta_i S(t_i))^2} \right\},$$

$$\frac{\partial^2 l(\vartheta;t)}{\partial \nu_m \partial \beta_l} = \sum_{i=1}^{n} \left\{ (1+\delta_i) \frac{\frac{\partial S(t_i)}{\partial \nu_m} \frac{\partial \eta_i}{\partial \beta_l}(1 - \eta_i S(t_i)) + \eta_i S(t_i) \frac{\partial S(t_i)}{\partial \nu_m} \frac{\partial \eta_i}{\partial \beta_l}}{(1 - \eta_i S(t_i))^2} \right\},$$

and

$$\frac{\partial^2 l(\vartheta;t)}{\partial \beta_l \partial \beta_k} = \sum_{i=1}^{n} \left\{ -\frac{\frac{\partial^2 \eta_i}{\partial \beta_l \partial \beta_k}(1 - \eta_i) + \frac{\partial \eta_i}{\partial \beta_l} \frac{\partial \eta_i}{\partial \beta_k}}{(1 - \eta_i)^2} + (1+\delta_i) \frac{S(t_i) \frac{\partial^2 \eta_i}{\partial \beta_l \partial \beta_k}(1 - \eta_i S(t_i)) + [S(t_i)]^2 \frac{\partial \eta_i}{\partial \beta_l} \frac{\partial \eta_i}{\partial \beta_k}}{(1 - \eta_i S(t_i))^2} \right\}$$

for $n, m$=1, 2; $l, k$=0, 1, where

$$\frac{\partial \eta}{\partial \beta_0} = e^{\beta_0 + x\beta_1}, \quad \frac{\partial \eta}{\partial \beta_1} = x e^{\beta_0 + x\beta_1}, \quad \frac{\partial^2 \eta}{\partial {\beta_0}^2} = e^{\beta_0 + x\beta_1}, \quad \frac{\partial^2 \eta}{\partial {\beta_1}^2} = x^2 e^{\beta_0 + x\beta_1},$$
$$\frac{\partial^2 \eta}{\partial \beta_0 \partial \beta_1} = x e^{\beta_0 + x\beta_1}, \quad \frac{\partial^2 \log f(t)}{\partial {\nu_1}^2} = -\frac{1}{{\nu_1}^2} - e^{\nu_2} t^{\nu_1} (\log t)^2, \quad \frac{\partial^2 \log f(t)}{\partial {\nu_2}^2} = -e^{\nu_2} t^{\nu_1},$$
$$\frac{\partial^2 \log f(t)}{\partial \nu_1 \partial \nu_2} = -e^{\nu_2} t^{\nu_1} (\log t)^2, \quad \frac{\partial S(t)}{\partial \nu_1} = -t^{\nu_1} \exp(\nu_2 - e^{\nu_2} t^{\nu_1}) \log t, \quad \frac{\partial S(t)}{\partial \nu_2} = -t^{\nu_1} \exp(\nu_2 - e^{\nu_2} t^{\nu_1}),$$
$$\frac{\partial^2 S(t)}{\partial {\nu_1}^2} = t^{\nu_1} \exp(\nu_2 - e^{\nu_2} t^{\nu_1})(e^{\nu_2} t^{\nu_1} - 1)(\log t)^2, \quad \frac{\partial^2 S(t)}{\partial {\nu_1}^2} = t^{\nu_2} \exp(\nu_2 - e^{\nu_2} t^{\nu_1})(e^{\nu_2} t^{\nu_1} - 1),$$

*and*

$$\frac{\partial^2 S(t)}{\partial \nu_1 \partial \nu_2} = t^{\nu_2} \exp(\nu_2 - e^{\nu_2} t^{\nu_1})(e^{\nu_2} t^{\nu_1} - 1) \log t.$$

## A.3 Second derivatives of the log-likelihood function for the Poisson Model

$$\frac{\partial^2 l(\vartheta; t)}{\partial \nu_m \partial \nu_n} = \sum_{i=1}^{n} \left\{ \delta_i \frac{\partial^2 \log f(t_i)}{\partial \nu_m \partial \nu_n} + \eta_i \frac{\partial^2 S(t_i)}{\partial \nu_m \partial \nu_n} \right\},$$

$$\frac{\partial^2 l(\vartheta; t)}{\partial \nu_m \partial \beta_l} = \sum_{i=1}^{n} \left\{ \frac{\partial S(t_i)}{\partial \nu_m} \frac{\partial \eta_i}{\partial \beta_l} \right\},$$

and

$$\frac{\partial^2 l(\vartheta; t)}{\partial \beta_l \partial \beta_k} = \sum_{i=1}^{n} \left\{ \delta_i \frac{\partial^2 \log \eta_i}{\partial \beta_l \partial \beta_k} + \frac{\partial^2 \eta_i}{\partial \beta_l \partial \beta_k} (S(t_i) - 1) \right\},$$

for $n, m$=1, 2 and $l, k$=0, 1, where

$$\frac{\partial^2 \log \eta_i}{\partial \beta_l \partial \beta_k} = 0.$$

# Appendix B

# R Code

## B.1 Defining the COM-Poisson cure rate model

```
##The GAMLSS framework
library(gamlss)
library(gamlss.dist)
library(gamlss.cens)
library(survival)

## Weibull distribution
dWEI4 = function (y, mu = 1, sigma = 0, log = FALSE)
{
    if (any(mu <= 0))
        stop(paste("mu must be positive", "\n", ""))
    if (any(y <= 0))
        stop(paste("y must be greater than 0 ", "\n", ""))
    sigma2 = exp(-sigma / mu)
    fy = dweibull(y, shape = mu, scale = sigma2, log = log)
    fy
```

```
}

pWEI4 = function (q, mu = 1, sigma = 0, lower.tail = TRUE, log.p = FALSE)
{
    if (any(mu <= 0))
        stop(paste("mu must be positive", "\n", ""))
    if (any(q <= 0))
        stop(paste("y must be greater than 0 ", "\n", ""))
    sigma2 = exp(-sigma / mu)
    cdf = pweibull(q, shape = mu, scale = sigma2, lower.tail = lower.tail,
        log.p = log.p)
    cdf
}

# Relative difference for truncations and maximum term
eps = 0.1
maxk0 = 50

# Truncated Z function
# nu = eta, and tau = phi
Zk = function(nu, tau, k) {
  rangek = 0:k
  sum(nu^rangek / factorial(rangek)^tau)
}

## Truncated series in the derivative of the Z function
# nu = eta and tau = phi
```

```
Znuk = function(nu, tau, k) {
  rangek = 1:k
  sum(rangek * nu^rangek / factorial(rangek)^tau)
}

## Improper p.d.f.
# Competing causes ~ COM-Poisson
# Time to event ~ Weibull
# mu = gamma1, sigma = gamma2, nu = eta, and tau = phi
dCOMWEI4 = function(y, mu = 2, sigma = 3, nu = 0.5, tau = 1, log = FALSE)
{
    if (any(mu <= 0))
        stop(paste("mu must be greater than 0 ", "\n", ""))
    if (any(nu <= 0))
        stop(paste("nu must be greater than 0 ", "\n", ""))
    if (any(tau <= 0))
        stop(paste("tau must be greater than 0 ", "\n", ""))
    if (any(y <= 0))
        stop(paste("y must be greater than 0 ", "\n", ""))
    ly = max(length(y), length(mu), length(sigma), length(nu), length(tau))
    y = rep(y, length = ly)
    sigma = rep(sigma, length = ly)
    mu = rep(mu, length = ly)
    nu = rep(nu, length = ly)
    tau = rep(tau, length = ly)

    vf = dWEI4(y, mu = mu, sigma = sigma, log = TRUE) # p.d.f.
```

```
    vS = pWEI4(y, mu = mu, sigma = sigma, lower.tail = FALSE) # 1 - c.d.f.

    # Truncation in Z(nu = eta, tau = phi)
    k0nu = min(ceiling((max(nu) / eps)^(1 / tau[1])), maxk0)

    logfy = vf + log(mapply(function(x1, x2) Znuk(x1,x2,k0nu), nu*vS, tau)) -
       log(mapply(function(x1, x2) Zk(x1,x2,k0nu), nu, tau)) - log(vS)
    if (log == FALSE)
        fy = exp(logfy)
    else fy = logfy
    fy
}

## Improper c.d.f.
# Competing causes ~ COM-Poisson
# Time to event ~ Weibull
# mu = gamma1, sigma = gamma2, nu = eta, and tau = phi
pCOMWEI4 = function(q, mu = 2, sigma = 3, nu = 0.5, tau = 1,
  lower.tail = TRUE,   log.p = FALSE)
{
    if (any(mu <= 0))
        stop(paste("mu must be greater than 0 ", "\n", ""))
    if (any(nu <= 0))
        stop(paste("nu must be greater than 0 ", "\n", ""))
    if (any(tau <= 0))
        stop(paste("tau must be  greater than 0 ", "\n", ""))
    if (any(q <= 0))
```

```
        stop(paste("y must be >0", "\n", ""))
    lq = max(length(q), length(mu), length(sigma), length(nu), length(tau))
    q = rep(q, length = lq)
    sigma = rep(sigma, length = lq)
    mu = rep(mu, length = lq)
    nu = rep(nu, length = lq)
    tau = rep(tau, length = lq)

    vS = pWEI4(q, mu = mu, sigma = sigma, lower.tail = FALSE) # 1 - c.d.f.

    k0nu = min(ceiling((max(nu) / eps)^(1 / tau[1])), maxk0)

    cdf = 1 - mapply(function(x1, x2) Zk(x1,x2,k0nu), nu*vS, tau) /
      mapply(function(x1, x2) Zk(x1,x2,k0nu), nu, tau)  # = 1 - Spop

    if (lower.tail == TRUE)
        cdf = cdf
    else cdf = 1 - cdf
    if (log.p == FALSE)
        cdf = cdf
    else cdf = log(cdf)
    cdf
}

COMWEI4 = function (mu.link = "log", sigma.link = "identity", nu.link = "log",
  tau.link = "log")
{
```

```
mstats = checklink("mu.link", "COM-Poisson cure rate model",
    substitute(mu.link), c("log","identity", "inverse", "sqrt"))
dstats = checklink("sigma.link", "COM-Poisson cure rate model",
    substitute(sigma.link), c("identity"))
vstats = checklink("nu.link", "COM-Poisson cure rate model",
    substitute(nu.link), c("log", "identity", "sqrt"))
tstats = checklink("tau.link", "COM-Poisson cure rate model",
    substitute(tau.link), c("inverse", "log", "identity", "own"))
structure(
    list(family = c("COMWEI4", "COM-Poisson cure rate model"),
    parameters = list(mu = TRUE, sigma = TRUE, nu = TRUE, tau = TRUE),
    nopar = 4,
    type = "Continuous",
    mu.link = as.character(substitute(mu.link)),
    sigma.link = as.character(substitute(sigma.link)),
    nu.link = as.character(substitute(nu.link)),
    tau.link = as.character(substitute(tau.link)),
    mu.linkfun = mstats$linkfun,
    sigma.linkfun = dstats$linkfun,
    nu.linkfun = vstats$linkfun,
    tau.linkfun = tstats$linkfun,
    mu.linkinv = mstats$linkinv,
    sigma.linkinv = dstats$linkinv,
    nu.linkinv = vstats$linkinv,
    tau.linkinv = tstats$linkinv,
    mu.dr = mstats$mu.eta,
    sigma.dr = dstats$mu.eta,
```

```
nu.dr = vstats$mu.eta,
tau.dr = tstats$mu.eta,

dldm = function(y, mu, sigma, nu, tau) {
    nd = gamlss:::numeric.deriv(dCOMWEI4(y, mu, sigma, nu, tau,
        log = TRUE), "mu", delta = 1e-04)
    dldm = as.vector(attr(nd, "gradient"))
    dldm },

dldd = function(y, mu, sigma, nu, tau) {
    nd = gamlss:::numeric.deriv(dCOMWEI4(y, mu, sigma, nu, tau,
        log = TRUE), "sigma", delta = 1e-04)
    dldd = as.vector(attr(nd, "gradient"))
    dldd },

dldv = function(y, mu, sigma, nu, tau) {
    nd = gamlss:::numeric.deriv(dCOMWEI4(y, mu, sigma, nu, tau,
        log = TRUE), "nu", delta = 1e-04)
    dldv = as.vector(attr(nd, "gradient"))
    dldv },
dldt = function(y, mu, sigma, nu, tau) {
    nd = gamlss:::numeric.deriv(dCOMWEI4(y, mu, sigma, nu, tau,
        log = TRUE), "tau", delta = 1e-04)
    dldt = as.vector(attr(nd, "gradient"))
    dldt },

d2ldm2 = function(y, mu, sigma, nu, tau) {
```

```
    nd = gamlss:::numeric.deriv(dCOMWEI4(y, mu, sigma, nu, tau,
        log = TRUE), "mu", delta = 1e-04)
    dldm = as.vector(attr(nd, "gradient"))
    d2ldm2 = -dldm * dldm
    d2ldm2 },

d2ldmdd = function(y, mu, sigma, nu, tau) {
    nd = gamlss:::numeric.deriv(dCOMWEI4(y, mu, sigma, nu, tau,
        log = TRUE), "mu", delta = 1e-04)
    dldm = as.vector(attr(nd, "gradient"))
    nd = gamlss:::numeric.deriv(dCOMWEI4(y, mu, sigma, nu, tau,
        log = TRUE), "sigma", delta = 1e-04)
    dldd = as.vector(attr(nd, "gradient"))
    d2ldmdd = -dldm * dldd
    d2ldmdd },

d2ldmdv = function(y, mu, sigma, nu, tau) {
    nd = gamlss:::numeric.deriv(dCOMWEI4(y, mu, sigma, nu, tau,
        log = TRUE), "mu", delta = 1e-04)
    dldm = as.vector(attr(nd, "gradient"))
    nd = gamlss:::numeric.deriv(dCOMWEI4(y, mu, sigma, nu, tau,
        log = TRUE), "nu", delta = 1e-04)
    dldv = as.vector(attr(nd, "gradient"))
    d2ldmdv = -dldm * dldv
  d2ldmdv },

d2ldmdt = function(y, mu, sigma, nu, tau) {
```

```
  nd = gamlss:::numeric.deriv(dCOMWEI4(y, mu, sigma, nu, tau,
       log = TRUE), "mu", delta = 1e-04)
  dldm = as.vector(attr(nd, "gradient"))
  nd = gamlss:::numeric.deriv(dCOMWEI4(y, mu, sigma, nu, tau,
      log = TRUE), "tau", delta = 1e-04)
  dldt = as.vector(attr(nd, "gradient"))
  d2ldmdt = -dldm * dldt
  d2ldmdt },

d2ldd2 = function(y, mu, sigma, nu, tau) {
    nd = gamlss:::numeric.deriv(dCOMWEI4(y, mu, sigma, nu, tau,
       log = TRUE), "sigma", delta = 1e-04)
    dldd = as.vector(attr(nd, "gradient"))
    d2ldd2 = -dldd * dldd
    d2ldd2 },

 d2ldddv = function(y, mu, sigma, nu, tau) {
    nd = gamlss:::numeric.deriv(dCOMWEI4(y, mu, sigma, nu, tau,
       log = TRUE), "sigma", delta = 1e-04)
    dldd = as.vector(attr(nd, "gradient"))
    nd = gamlss:::numeric.deriv(dCOMWEI4(y, mu, sigma, nu, tau,
        log = TRUE), "nu", delta = 1e-04)
    dldv = as.vector(attr(nd, "gradient"))
    d2ldddv = -dldd * dldv
    d2ldddv },

 d2lddt = function(y, mu, sigma, nu, tau) {
```

```
    nd = gamlss:::numeric.deriv(dCOMWEI4(y, mu, sigma, nu, tau,
       log = TRUE), "sigma", delta = 1e-04)
    dldd = as.vector(attr(nd, "gradient"))
    nd = gamlss:::numeric.deriv(dCOMWEI4(y, mu, sigma, nu, tau,
        log = TRUE), "tau", delta = 1e-04)
    dldt = as.vector(attr(nd, "gradient"))
    d2ldddt = -dldd * dldt
    d2ldddt },

d2ldv2 = function(y, mu, sigma, nu, tau) {
  nd = gamlss:::numeric.deriv(dCOMWEI4(y, mu, sigma, nu, tau,
      log = TRUE), "nu", delta = 1e-04)
  dldv = as.vector(attr(nd, "gradient"))
  d2ldv2 = -dldv * dldv
  d2ldv2 },

d2ldvdt = function(y, mu, sigma, nu, tau) {
  nd = gamlss:::numeric.deriv(dCOMWEI4(y, mu, sigma, nu, tau,
      log = TRUE), "nu", delta = 1e-04)
  dldv = as.vector(attr(nd, "gradient"))
  nd = gamlss:::numeric.deriv(dCOMWEI4(y, mu, sigma, nu, tau,
      log = TRUE), "tau", delta = 1e-04)
  dldt = as.vector(attr(nd, "gradient"))
  d2ldvdt = -dldv * dldt
  d2ldvdt },

d2ldt2 = function(y, mu, sigma, nu, tau) {
```

```
    nd = gamlss:::numeric.deriv(dCOMWEI4(y, mu, sigma, nu, tau,
        log = TRUE), "tau", delta = 1e-04)
    dldt = as.vector(attr(nd, "gradient"))
    d2ldt2 = -dldt * dldt
    d2ldt2 },

  G.dev.incr = function(y, mu, sigma, nu, tau,...)
    -2 * dCOMWEI4(y, mu = mu, sigma = sigma, nu = nu, tau = tau,
    log = TRUE),
  rqres = expression(rqres(pfun = "pCOMWEI4", type = "Continuous",
      ymin = 0, y = y, mu = mu, sigma = sigma, nu = nu, tau = tau)),
  mu.initial = expression(mu = rep(1, length(y))),
  sigma.initial = expression(sigma = rep(-log(mean(y)), length(y))),
  nu.initial = expression(nu = rep(0.5, length(y))),
  tau.initial = expression(tau = rep(0.5, length(y))),
  mu.valid = function(mu) all(mu > 0),
  sigma.valid = function(sigma) TRUE,
  nu.valid = function(nu)  all(nu > 0),
  tau.valid = function(tau) all(tau > 0),
  y.valid = function(y) all(y > 0)), class = c("gamlss.family",
  "family"))
}
```

**Remark:** Above R code is provided by Dr. J. Rodrigues.

## B.2 Computation code for the Geometric case:

```
##To simulate samples for Geometric cure model
```

```
samp.geo=function (m,n,qr,par,r1,r2)
{
mat1<-data.frame(t=c(0),d=c(0),x=c(0))
for(i in 1:m){
d<-c(0)
T<-c(0)
t<-c(0)
nodule<-c(rep(1,n[1]),rep(2,n[2]),rep(3,n[3]),rep(4,n[4]))
nd1<-rgeom(n[1],qr[1])
nd2<-rgeom(n[2],qr[2])
nd3<-rgeom(n[3],qr[3])
nd4<-rgeom(n[4],qr[4])

for(i in (1:n[1])){
if (nd1[i]<=0)
T[i]=10000
else if (nd1[i]>0)
T[i]<-min(rweibull(nd1[i],shape=exp(r1),scale=exp(-r2/exp(r1))))
}
for(i in (1:n[2])){
if (nd2[i]<=0)
T[n[1]+i]<-10000
else if (nd2[i]>0)
T[n[1]+i]<-min(rweibull(nd2[i],shape=exp(r1),scale=exp(-r2/ exp(r1))))
}
for(i in (1:n[3])){
if (nd3[i]<=0)
```

```
T[n[1]+n[2]+i]<-10000
else if (nd3[i]>0)
T[n[1]+n[2]+i]<-min(rweibull(nd3[i],shape=exp(r1),scale=exp(-r2/ exp(r1))))
}
for(i in (1:n[4])){
if (nd4[i]<=0)
T[n[1]+n[2]+n[3]+i]<-10000
else if (nd4[i]>0)
T[n[1]+n[2]+n[3]+i]<-min(rweibull(nd4[i],shape=exp(r1),scale=
        exp(-r2/ exp(r1))))
}
 C<-c(rexp(n[1],par[1]),rexp(n[2],par[2]),rexp(n[3],par[3]),rexp(n[4],par[4]))
for(i in (1:sum(n))){
t[i]<-min(T[i],C[i])
if (T[i]>C[i])
d[i]=0
else d[i]=1
}
mat2<-data.frame(t=t,d=d,x=nodule)
mat1<-rbind(mat1,mat2)
}
mat1<-mat1[-1,]
mat1
}

##To calculate the estimates for Geometric cure model
estim.geo=function (m,a,x)
```

```
{
nu1<-c(0)
nu2<-c(0)
beta0<-c(0)
beta1<-c(0)
for(i in 1:m){
m4g = gamlss(Surv(t[(a*i-(a-1)):(a*i)], d[(a*i-(a-1)):(a*i)]) ~ 1, family =cens
(COMWEI4),nu.formula = ~ 1 + x[(a*i-(a-1)):(a*i)], data = x, tau.start =
.Machine$double.eps,tau.fix = TRUE,  c.crit = 0.1, n.cyc = 100)
nu1[i]<-exp(m4g$mu.coefficients)
nu2[i]<-m4g$sigma.coefficients
beta0[i]<-m4g$nu.coefficients[1]
beta1[i]<-m4g$nu.coefficients[2]
print(i)
print(date())
}
data<-data.frame(nu1=nu1,nu2=nu2,beta0=beta0,beta1=beta1)
write.table(data,file="estimates.txt")
}

###To calculate SEs for Geometric cure model
inmat.geo=function (a,r1,r2,b0,b1,t,d,x)
{
n<-length(r1)
nu1SE<-c(0)
nu2SE<-c(0)
beta0SE<-c(0)
```

```
beta1SE<-c(0)
for(i in 1:n){
eta<-exp(b0[i]+x[(a*i-(a-1)):(a*i)]*b1[i])
f<-r1[i]*t[(a*i-(a-1)):(a*i)]^(r1[i]-1)*exp(r2[i]--t[(a*i-(a-1)):(a*i)]^r1[i]*
exp(r2[i]))
s<-exp(-t[(a*i-(a-1)):(a*i)]^r1[i]*exp(r2[i]))
dndb0<-exp(b0[i]+x[(a*i-(a-1)):(a*i)]*b1[i])
dndb1<-x[(a*i-(a-1)):(a*i)]*exp(b0[i]+x[(a*i-(a-1)):(a*i)]*b1[i])
dnd2b0<-exp(b0[i]+x[(a*i-(a-1)):(a*i)]*b1[i])
dndb0b1<-x[(a*i-(a-1)):(a*i)]*exp(b0[i]+x[(a*i-(a-1)):(a*i)]*b1[i])
dnd2b1<-x[(a*i-(a-1)):(a*i)]^2*exp(b0[i]+x[(a*i-(a-1)):(a*i)]*b1[i])
dlfd2r1<- -1/r1[i]^2-exp(r2[i])*t[(a*i-(a-1)):(a*i)]^r1[i]*(log(t[(a*i-(a-1)):
(a*i)]))^2
dlfdr1r2<- -exp(r2[i])*t[(a*i-(a-1)):(a*i)]^r1[i]*log(t[(a*i-(a-1)):(a*i)])
dlfd2r2<- -exp(r2[i])*t[(a*i-(a-1)):(a*i)]^r1[i]
dsdr1<- -t[(a*i-(a-1)):(a*i)]^r1[i]*exp(r2[i])*log(t[(a*i-(a-1)):(a*i)])*exp
(-t[(a*i-(a-1)):(a*i)]^r1[i]*exp(r2[i]))
dsdr2<- -t[(a*i-(a-1)):(a*i)]^r1[i]*exp(r2[i])*exp(-t[(a*i-(a-1)):(a*i)]^r1
[i]*exp(r2[i]))
dsd2r1<- -t[(a*i-(a-1)):(a*i)]^r1[i]*(log(t[(a*i-(a-1)):(a*i)]))^2*exp(-t[(a*i
-(a-1)):(a*i)]^r1[i]*exp(r2[i])+r2[i])+t[(a*i-(a-1)):(a*i)]^(2*r1[i])*(log(t[
(a*i-(a-1)):(a*i)]))^2*exp(-t[(a*i-(a-1)):(a*i)]^r1[i]*exp(r2[i])+2*r2[i])
dsdr1r2<- -t[(a*i-(a-1)):(a*i)]^r1[i]*log(t[(a*i-(a-1)):(a*i)])*exp(-t[(a*i-
(a-1)):(a*i)]^r1[i]*exp(r2[i])+r2[i])+t[(a*i-(a-1)):(a*i)]^(2*r1[i])*log(t[(a*i
-(a-1)):(a*i)])*exp(-t[(a*i-(a-1)):(a*i)]^r1[i]*exp(r2[i])+2*r2[i])
dsd2r2<- -t[(a*i-(a-1)):(a*i)]^r1[i]*exp(-t[(a*i-(a-1)):(a*i)]^r1[i]*exp(r2[i])
+r2[i])+t[(a*i-(a-1)):(a*i)]^(2*r1[i])*exp(-t[(a*i-(a-1)):(a*i)]^r1[i]*exp
```

```
(r2[i])+2*r2[i])
j11<- -sum(d[(a*i-(a-1)):(a*i)]*dlfd2r1+(1+d[(a*i-(a-1)):(a*i)])*(eta*dsd2r1*
(1-eta*s)+(eta*dsdr1)^2)/(1-eta*s)^2)
j12<- -sum(d[(a*i-(a-1)):(a*i)]*dlfdr1r2+(1+d[(a*i-(a-1)):(a*i)])*(eta*dsdr1r2*
(1-eta*s)+eta^2*dsdr1*dsdr2)/(1-eta*s)^2)
j22<- -sum(d[(a*i-(a-1)):(a*i)]*dlfd2r2+(1+d[(a*i-(a-1)):(a*i)])*(eta*dsd2r2*
(1-eta*s)+(eta*dsdr2)^2)/(1-eta*s)^2)
j13<- -sum((1+d[(a*i-(a-1)):(a*i)])*dsdr1*dndb0/(1-eta*s)^2)
j14<- -sum((1+d[(a*i-(a-1)):(a*i)])*dsdr1*dndb1/(1-eta*s)^2)
j23<- -sum((1+d[(a*i-(a-1)):(a*i)])*dsdr2*dndb0/(1-eta*s)^2)
j24<- -sum((1+d[(a*i-(a-1)):(a*i)])*dsdr2*dndb1/(1-eta*s)^2)
j33<- -sum(d[(a*i-(a-1)):(a*i)]*(eta*dnd2b0-(dndb0)^2)/eta^2-(dnd2b0*(1-eta)+
(dndb0)^2)/(1-eta)^2+(1+d[(a*i-(a-1)):(a*i)])*s*(dnd2b0*(1-eta*s)+(dndb0)^2*s)
/(1-eta*s)^2)
j44<- -sum(d[(a*i-(a-1)):(a*i)]*(eta*dnd2b1-(dndb1)^2)/eta^2-(dnd2b1*(1-eta)+
(dndb1)^2)/(1-eta)^2+(1+d[(a*i-(a-1)):(a*i)])*s*(dnd2b1*(1-eta*s)+(dndb1)^2*s)
/(1-eta*s)^2)
j34<- -sum(d[(a*i-(a-1)):(a*i)]*(eta*dndb0b1-(dndb1*dndb0))/eta^2-(dndb0b1*
(1-eta)+(dndb0*dndb1))/(1-eta)^2+(1+d[(a*i-(a-1)):(a*i)])*s*(dndb0b1*(1-eta*s)
+(dndb0*dndb1)*s)/(1-eta*s)^2)
J<-matrix(c(j11,j12,j13,j14,j12,j22,j23,j24,j13,j23,j33,j34,j14,j24,j34,j44)
,nrow=4,byrow=F)
inv<-solve(J)
nu1SE[i]<-sqrt(inv[1,1])
nu2SE[i]<-sqrt(inv[2,2])
beta0SE[i]<-sqrt(inv[3,3])
beta1SE[i]<-sqrt(inv[4,4])
```

```
}
data.frame(nu1=r1,nu1SE=nu1SE,nu2=r2,nu2SE=nu2SE,beta0=b0,beta0SE=beta0SE,
beta1=b1,beta1SE=beta1SE)
}

###To calculate coverage probability
coverage=function (par,nu1,nu1se,nu2,nu2se,b0,b0se,b1,b1se)
{
n<-length(nu1)
count1<-0
count2<-0
count3<-0
count4<-0
nu1ci<-matrix(rep(0,2*n),ncol=2)
nu2ci<-matrix(rep(0,2*n),ncol=2)
b0ci<-matrix(rep(0,2*n),ncol=2)
b1ci<-matrix(rep(0,2*n),ncol=2)
for(i in 1:n){
nu1ci[i,]<-nu1[i]+c(qnorm(0.025),qnorm(0.975))*nu1se[i]
if(nu1ci[i,1]<exp(par[1])&&nu1ci[i,2]>exp(par[1]))
count1=count1+1

nu2ci[i,]<-nu2[i]+c(qnorm(0.025),qnorm(0.975))*nu2se[i]
if(nu2ci[i,1]<par[2]&&nu2ci[i,2]>par[2])
count2=count2+1

b0ci[i,]<-b0[i]+c(qnorm(0.025),qnorm(0.975))*b0se[i]
```

```
if(b0ci[i,1]<par[3]&&b0ci[i,2]>par[3])
count3=count3+1

b1ci[i,]<-b1[i]+c(qnorm(0.025),qnorm(0.975))*b1se[i]
if(b1ci[i,1]<par[4]&&b1ci[i,2]>par[4])
count4=count4+1
}
data.frame(nu1COVER=count1/n,nu2COVER=count2/n,beta0COVER=count3/n,beta1COVER
=count4/n)
}
```

## B.3 Computation code for the Poisson case:

```
###To simulate samples for Poisson cure model
samp.poi=function (m,n,qr,par,r1,r2)
{
mat1<-data.frame(t=c(0),d=c(0),x=c(0))
for(i in 1:m){
d<-c(0)
T<-c(0)
t<-c(0)
nodule<-c(rep(1,n[1]),rep(2,n[2]),rep(3,n[3]),rep(4,n[4]))
nd1<-rpois(n[1],-log(qr[1]))
nd2<-rpois(n[2],-log(qr[2]))
nd3<-rpois(n[3],-log(qr[3]))
nd4<-rpois(n[4],-log(qr[4]))

for(i in (1:n[1])){
```

```
if (nd1[i]<=0)
T[i]=10000
else if (nd1[i]>0)
T[i]<-min(rweibull(nd1[i],shape=exp(r1),scale=exp(-r2/exp(r1))))
}
for(i in (1:n[2])){
if (nd2[i]<=0)
T[n[1]+i]<-10000
else if (nd2[i]>0)
T[n[1]+i]<-min(rweibull(nd2[i],shape=exp(r1),scale=exp(-r2/ exp(r1))))
}
for(i in (1:n[3])){
if (nd3[i]<=0)
T[n[1]+n[2]+i]<-10000
else if (nd3[i]>0)
T[n[1]+n[2]+i]<-min(rweibull(nd3[i],shape=exp(r1),scale=exp(-r2/ exp(r1))))
}
for(i in (1:n[4])){
if (nd4[i]<=0)
T[n[1]+n[2]+n[3]+i]<-10000
else if (nd4[i]>0)
T[n[1]+n[2]+n[3]+i]<-min(rweibull(nd4[i],shape=exp(r1),scale=exp(-r2/ exp(r1))))
}
 C<-c(rexp(n[1],par[1]),rexp(n[2],par[2]),rexp(n[3],par[3]),rexp(n[4],par[4]))
for(i in (1:sum(n))){
t[i]<-min(T[i],C[i])
if (T[i]>C[i])
```

```
d[i]=0
else d[i]=1
}
mat2<-data.frame(t=t,d=d,x=nodule)
mat1<-rbind(mat1,mat2)
}
mat1<-mat1[-1,]
mat1
}

###To calculate estimates for Poisson cure model
estim.poi=function (m,a,y)
{
nu1<-c(0)
nu2<-c(0)
beta0<-c(0)
beta1<-c(0)
for(i in 1:m){
m4p = gamlss(Surv(t[(a*i-(a-1)):(a*i)], d[(a*i-(a-1)):(a*i)]) ~ 1, family = cens
(COMWEI4),nu.formula = ~ 1 + x[(a*i-(a-1)):(a*i)], data = y, tau.start = 1.00001,
tau.fix = TRUE, c.crit = 0.1, n.cyc = 200)
nu1[i]<-exp(m4p$mu.coefficients)
nu2[i]<-m4p$sigma.coefficients
beta0[i]<-m4p$nu.coefficients[1]
beta1[i]<-m4p$nu.coefficients[2]
print(i)
print(date())
```

```
}
data<-data.frame(nu1=nu1,nu2=nu2,beta0=beta0,beta1=beta1)
write.table(data,file="estimates.txt")
data
}

###To calculate SEs for Poisson cure model
inmat.poi=function (a,r1,r2,b0,b1,t,d,x)
{
n<-length(r1)
nu1SE<-c(0)
nu2SE<-c(0)
beta0SE<-c(0)
beta1SE<-c(0)
u1<-c(0)
u2<-c(0)
u3<-c(0)
u4<-c(0)
for(i in 1:n){
eta<-exp(b0[i]+x[(a*i-(a-1)):(a*i)]*b1[i])
s<-exp(-t[(a*i-(a-1)):(a*i)]^r1[i]*exp(r2[i]))
dndb0<-exp(b0[i]+x[(a*i-(a-1)):(a*i)]*b1[i])
dndb1<-x[(a*i-(a-1)):(a*i)]*exp(b0[i]+x[(a*i-(a-1)):(a*i)]*b1[i])
dnd2b0<-exp(b0[i]+x[(a*i-(a-1)):(a*i)]*b1[i])
dndb0b1<-x[(a*i-(a-1)):(a*i)]*exp(b0[i]+x[(a*i-(a-1)):(a*i)]*b1[i])
dnd2b1<-x[(a*i-(a-1)):(a*i)]^2*exp(b0[i]+x[(a*i-(a-1)):(a*i)]*b1[i])
dlfd2r1<- -1/r1[i]^2-exp(r2[i])*t[(a*i-(a-1)):(a*i)]^r1[i]*(log(t[(a*i-
```

```
        (a-1)):(a*i)]))^2
dlfdr1r2<- -exp(r2[i])*t[(a*i-(a-1)):(a*i)]^r1[i]*log(t[(a*i-(a-1)):(a*i)])
dlfd2r2<- -exp(r2[i])*t[(a*i-(a-1)):(a*i)]^r1[i]
dsdr1<- -t[(a*i-(a-1)):(a*i)]^r1[i]*exp(r2[i])*log(t[(a*i-(a-1)):(a*i)])
        *exp(-t[(a*i-(a-1)):(a*i)]^r1[i]*exp(r2[i]))
dsdr2<- -t[(a*i-(a-1)):(a*i)]^r1[i]*exp(r2[i])*exp(-t[(a*i-(a-1)):(a*i)]
        ^r1[i]*exp(r2[i]))
dsd2r1<- -t[(a*i-(a-1)):(a*i)]^r1[i]*(log(t[(a*i-(a-1)):(a*i)]))^2*exp
        (-t[(a*i-(a-1)):(a*i)]^r1[i]*exp(r2[i])+r2[i])+t[(a*i-(a-1)):(a*i)]^
        (2*r1[i])*(log(t[(a*i-(a-1)):(a*i)]))^2*exp(-t[(a*i-(a-1)):(a*i)]^r1[i]
        *exp(r2[i])+2*r2[i])
dsdr1r2<- -t[(a*i-(a-1)):(a*i)]^r1[i]*log(t[(a*i-(a-1)):(a*i)])*exp
        (-t[(a*i-(a-1)):(a*i)]^r1[i]*exp(r2[i])+r2[i])+t[(a*i-(a-1)):(a*i)]
        ^(2*r1[i])*log(t[(a*i-(a-1)):(a*i)])*exp(-t[(a*i-(a-1)):(a*i)]^r1[i]
        *exp(r2[i])+2*r2[i])
dsd2r2<- -t[(a*i-(a-1)):(a*i)]^r1[i]*exp(-t[(a*i-(a-1)):(a*i)]^r1[i]*exp
        (r2[i])+r2[i])+t[(a*i-(a-1)):(a*i)]^(2*r1[i])*exp(-t[(a*i-(a-1)):(a*i)]
        ^r1[i]*exp(r2[i])+2*r2[i])
j11<--sum(d[(a*i-(a-1)):(a*i)]*dlfd2r1+dsd2r1*eta)
j12<--sum(d[(a*i-(a-1)):(a*i)]*dlfdr1r2+dsdr1r2*eta)
j22<--sum(d[(a*i-(a-1)):(a*i)]*dlfd2r2+dsd2r2*eta)
j13<--sum(dsdr1*dndb0)
j14<--sum(dsdr1*dndb1)
j23<--sum(dsdr2*dndb0)
j24<--sum(dsdr2*dndb1)
j33<--sum(dnd2b0*(s-1))
j44<--sum(dnd2b1*(s-1))
```

```
j34<--sum(dndb0b1*(s-1))
J<-matrix(c(j11,j12,j13,j14,j12,j22,j23,j24,j13,j23,j33,j34,j14,j24,j34
        ,j44,nrow=4,byrow=F)
inv<-solve(J)
nu1SE[i]<-sqrt(inv[1,1])
nu2SE[i]<-sqrt(inv[2,2])
beta0SE[i]<-sqrt(inv[3,3])
beta1SE[i]<-sqrt(inv[4,4])
}
 data.frame(nu1=r1,nu1SE=nu1SE,nu2=r2,nu2SE=nu2SE,beta0=b0,beta0SE=beta0SE,
beta1=b1,beta1SE=beta1SE)
}

###To calculate coverage probability
coverage=function (par,nu1,nu1se,nu2,nu2se,b0,b0se,b1,b1se)
{
n<-length(nu1)
count1<-0
count2<-0
count3<-0
count4<-0
nu1ci<-matrix(rep(0,2*n),ncol=2)
nu2ci<-matrix(rep(0,2*n),ncol=2)
b0ci<-matrix(rep(0,2*n),ncol=2)
b1ci<-matrix(rep(0,2*n),ncol=2)
for(i in 1:n){
nu1ci[i,]<-nu1[i]+c(qnorm(0.025),qnorm(0.975))*nu1se[i]
```

```
if(nu1ci[i,1]<exp(par[1])&&nu1ci[i,2]>exp(par[1]))
count1=count1+1

nu2ci[i,]<-nu2[i]+c(qnorm(0.025),qnorm(0.975))*nu2se[i]
if(nu2ci[i,1]<par[2]&&nu2ci[i,2]>par[2])
count2=count2+1

b0ci[i,]<-b0[i]+c(qnorm(0.025),qnorm(0.975))*b0se[i]
if(b0ci[i,1]<par[3]&&b0ci[i,2]>par[3])
count3=count3+1

b1ci[i,]<-b1[i]+c(qnorm(0.025),qnorm(0.975))*b1se[i]
if(b1ci[i,1]<par[4]&&b1ci[i,2]>par[4])
count4=count4+1
}
data.frame(nu1COVER=count1/n,nu2COVER=count2/n,beta0COVER=count3/n,beta1COVER=count4/n)
}
```

# Bibliography


[1] Adams, J. E., Gray, H. L., Watkins, T.A., 1971. An asymptotic characterization of bias reduction by jackknifing. *The Annals of Mathematical Statistics* **42**(5), 1606-1612.

[2] Casella, G., Berger, R. L., 2002. *Statistical Inference.* Second edition, Duxbury Press.

[3] Conway, R. W., Maxwell, W. L., 1961. A queuing model with state dependent service rates. *The Journal of Industrial Engineering* **XII**(2), 132-136.

[4] Ibrahim, J. G., Chen, M-H., Sinha, D., 2001. *Bayesian Survival Analysis.* Springer, New York.

[5] Kirkwood, J. M., Ibrahim, J. G., Sondak, V. K., Richards, J., Faherty, L. E., Ernstoff, M. S., Smith, T. J., Rao, U., Steele, M., Blum, R. H., 2000. High- and low-dise interferon alfa-2b in high-risk melanoma: first analysis of Intergroup Trial E1690/S9111/S9190. *Journal of Clinical Oncology* **18**(12), 2444-2458.

[6] Kokonendji, C. C., Mizere, D., Balakrishnan, N., 2008. Connections of the Poisson weight function to overdispersion and underdispersion. *Journal of Statistical Planning and Inference* **138**(5), 1287-1296.

[7] R Development Core Team, 2009. R: A Language and Environment for Statistical Computing. R Foundation for the Statistical Computing, Vienna,Austria.

[8] Rigby, R. A., Stasinopoulos, D. M., 2005. Generalized additive models for location, scale and shape (with discussion). *Applied Statistics* **54**(3), 507-554.

[9] Rodrigues, J., Cancho, V. G., de Castro, M., Louzada-Neto, F., 2009. On the unification of the long-term survival models. *Statistics and Probability Letters* **79**, 753-759.

[10] Rodrigues, J., de Castro, M., Cancho, V. G., Balakrishnan, N., 2009. COM-Poisson cure rate survival models and an application to a cutaneous melanoma data. *Journal of Statistical Planning and Inference* **139**, 3605-3611.

[11] Silvey, S. D., 1975. *Statistical Inference.* Chapman & Hall, Boca Raton, Florida.

[12] Stasinopoulos, D. M., Rigby, R. A., 2007. Generalized additive models for location, scale and shape (GAMLSS) in R. *Journal of Statistical Software* **23**(7), 1-46.

[13] Tsodikov, A. D., Ibrahim, J. G., Yakovlev, A. Y., 2003. Estimating cure rates from survival data: an alternative to two-component mixture models. *Journal of the American Statistical Association* **98**(464), 1063-1078.

[14] Yakovlev, A. Y., Tsodikov, A. D., 1996. *Stochastic Models of Tumor Latercy and their Biostatistical Applications.* World Scientific, Singapore.

[15] Yin, G., Ibrahim, J.G., 2005. Cure rate models: a unified approach. *The Canadian Journal of Statistics* **33**(4), 559-570.