University of Alberta

Informative Random Censoring in Parametric Survival Models

by

Weihong Li

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science
in
Biostatistics

Department of Mathematical and Statistical Sciences

Fall 2009
Edmonton, Alberta

Examining Committee

Keumhee Carrière Chough, Mathematical and Statistical Sciences

Narasimha Prasad, Mathematical and Statistical Sciences

A “Sentil” Senthilselvan, Public Health Sciences, University of Alberta

Abstract

Informative random censoring survival data are often seen in clinical trials. However, the methodology to deal with this kind of data has not been well developed due to difficulty of identifying the information. Several methods were proposed, for example, by Siannis et al. (2005). We use simulation studies to investigate sensitivity of these methods and show that the maximum likelihood estimation (MLE) method provides narrower confidence intervals than Siannis et al. (2005). This is true and expected under the same assumption as in Siannis et al. (2005). However, we were able to give practical guidelines on how to guess at the missing information of random censoring. We give conditions to obtain more precise estimators for survival data analyses, providing a user-friendly R program. Two real-life data sets are used to illustrate the application of this methodology.

Acknowledgements

I am grateful for having been Dr. K. C. Carrière's student. Without her prompt responses, insightful guidance, consistent encouragement, and endless support, I would not have been able to finish this thesis. My special gratitude also goes to Dr. Prasad and Dr. Senthilselvan, my thesis supervisory committee members.

I would like to thank all the professors, staff, my fellow graduate students, and all my friends in the department for their moral support. I appreciate my inspiring discussion with Dr. Peng Zhang, Dr. Pengfei Li, and Dr. Wanhua Su.

I want to express my special thanks to Dr. Steve Cumming, who supervised me during my internship when I was an undergraduate student. What I learned by working with him has benefitted my graduate studies.

I am thankful for all the help from my mother, my parents-in-law, my friend Hairong Du, and all the encouragement from my father.

Finally, I am most grateful to my husband, Peng Li, for his constant understanding and unselfish support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Survival analysis is an important topic in many areas, including biomedical, engineering, and social sciences. One of the special features in survival data is censoring. When the end point of interest for an individual has not been observed after a certain length of time for many different reasons in a variety of forms, we say that right censoring occurs. Right censoring is divided into three types: type I, type II, and type III. Type I and type II are also called singly censored data (Lee and Wang, 2003). Type III is also called progressively censored data, or random censoring (Lee and Wang, 2003; Allison, 1995), which includes both noninformative and informative censoring types. Type I, type II, and noninformative random censoring can be handled by using standard methodologies, which are based on the assumption of independence of the failure end points and the censoring mechanism. However, when the independence assumption is questionable, inferences based on standard methodologies may be

biased and possibly misleading. Therefore, informative random censoring is one of the most sensitive problems in survival data analysis.

In typical clinical trials, one is often interested in the analysis of an overall measure of time to failure. For example, when a new drug is introduced to cancer patients, researchers want to compare it with a standard treatment and conclude whether patients taking a new drug will survive significantly longer than those taking a standard drug. The analysis of such data might be complicated because of potentially selective dropout prior to the main end point or the end of the study; i.e., patients may withdraw from the study for various unknown reasons, including inadequate treatment experience, worsening of symptoms, severe treatment toxicity (Ruan and Gray, 2008), or simple loss to followup. Withdrawal due to the worsening of symptoms could mean patients facing a higher risk of failure than other patients. That is, if the patient had not withdrawn from the study, he or she would have reached the end point sooner than expected otherwise. Withdrawal due to severe toxicity could mean that the patient has received treatment benefit early on and, hence, is possibly at a lower risk of failure than other patients.

Standard survival analysis methods assume independent and noninformative censoring. That is, withdrawn patients have the same risk of failure as those remaining in the study. If those withdrawn were at higher risk, then the estimators generated by using standard methods would overestimate the survival time because only patients in good condition (those who stayed in the study) could be considered. Similarly, if those who withdrew were at lower risk, the

standard method would underestimate their survival time.

In an early study, Tsiatis (1972) discussed that the joint distribution of failure (T) and censoring (C) time is not identifiable from the observed data. Later, Siannis et al. (2005) introduced an assumption about the dependence structure between T and C in a parametric survival model in order to deal with the problem. As well, Siannis et al. (2005) proposed an approximation to conduct sensitivity analysis. However, this approximation is valid only when the degree of dependence δ between T and C is small. In this thesis, we investigate the performance of Siannis et al.'s approximation with maximum likelihood estimation method under the same assumption for parametric survival models. As well, a simulation study illustrates the pattern of the maximum likelihood estimators.

The aim of this thesis is to investigate and provide guidelines for practitioners on what to do for the situation with informative censoring.

This thesis is organized as follows. In Chapter 2, the methodologies for type I, type II, and random censoring, focusing on the method proposed by Siannis et al for nonindependent random censoring, are reviewed. In Chapter 3, graphical analysis explores dependent relationships between T and C. In Chapter 4, simulation studies are presented, which support the maximum likelihood methods under Siannis et al.'s assumption about the joint distribution between T and C. In Chapter 5, the strategy developed under this assumption is applied to two real data examples. Chapter 6 presents our conclusions and recommendations.

Chapter 2

Review of Literature

The survival analysis of failure time data is one of biostatisticians' major tasks. The failure time is the period of time between the beginning of a study and the occurrence of some event (Kalbfleisch and Prentice, 2002). In such data, the end points of interest for some individuals are often not observed during the study period for a variety of reasons. Whenever the end point is not observed, we say that the data for an individual are censored. Censoring can be classified into three forms: right censoring, left censoring, and interval censoring. Since right censoring is the most frequently seen form in clinical trials, we will cover only the methodologies dealing with this type of censoring. Because the distribution of the lifetime (T) and censoring time (C) is independent for type I, type II and noninformative random censoring, likelihood-based methodology has been well developed and widely used in standard parametric and nonparametric analysis and has been implemented in popular statistics software such as SAS, R, and

S-Plus. For the informative random censoring mechanism, the joint distribution of T and C is not identifiable from the observed data (Tsiatis, 1972). Therefore, most studies (Scharfstein et al., 2001; Siannis et al., 2005; Zhang and Heitjan, 2005) introduce additional assumptions about and models for the dependence structure between these two variables to make the problem identifiable.

We first introduce some notation. Suppose that n individuals are followed from $t = 0$ until each fails or is censored. Their lifetimes are represented by random variables $T_1, \cdots, T_n$, and the censoring time is $C_1, \cdots, C_n$. We have a time variable t_i, which is either the lifetime or a censoring time, $t_i = \min(T_i, C_i)$. I_i is a random variable that equals 1 if $T_i = t_i$ and 0 if $T_i > t_i$, $I_i = I(T_i = t_i)$. Let $f_T(t,\theta), S_T(t,\theta), h_T(t,\theta), s_T(t,\theta) = \frac{\partial}{\partial\theta}\log f_T(t,\theta), i_\theta = \mathrm{Var}_T(s_T(t,\theta))$ be T's probability density function, survival function, hazard function, score function, and information function, respectively. Similarly, let $f_C(c,\gamma), S_C(c,\gamma), h_C(c,\gamma), s_C(c,\gamma), i_\gamma$ be C's corresponding functions.

2.1 Type I Censoring

Type I censoring means that the censoring time is prespecified and the same for all individuals. For instance, in a clinical trial, if all individuals enter the study at the same time, and the study is planned to follow the occurrence of an event of interest for up to two years only, we say type I censoring happens for those whom the event does not occur at the end of the study. Since C_i is a fixed constant, the information we have from the censored data is that the survival

time is greater than C_i. Assuming that the lifetimes $T_1, \cdots, T_n$ are statistically independent, we obtain the likelihood function as (Lawless, 2003)

$$L = \prod_{i=1}^{n} f_T(t_i)^{I_i} S_T(t_i)^{1-I_i}. \tag{2.1}$$

2.2 Type II Censoring

Type II censoring refers to the situation in which a study stops after a prespecified number of events have been observed. In the previous example, if the study stops once r events have been observed, the rest of the individuals are experiencing type II censoring. As the total observation time is unknown at the beginning, type II is less common than the other types in a planned study. r is a prespecified integer between 1 and n, and it is chosen before the data are collected. Again, we receive no information for people who are censored. Thus, the likelihood function is (Lawless, 2003)

$$\begin{aligned} L &= \frac{n!}{(n-r)!}\left[\prod_{i=1}^{r} f_T(t_{(i)})\right] S_T(t_{(r)})^{n-r} \\ &\propto \prod_{i=1}^{n} f_T(t_i)^{I_i} S_T(t_i)^{1-I_i}. \end{aligned} \tag{2.2}$$

2.3 Random Censoring

Type III censoring, or random censoring, occurs when observations are terminated for reasons that are not under the control of the investigator (Allison, 1995). In the example for type I censoring, if some individuals withdraw from the study before two years are up, or if they die due to other reasons than the

one we are looking for, we say random censoring happens. In other situations, clinical trails are planned to run, for example, for three years, with patients entering the trial over the first year. The lifetime variable T_i for a patient is the duration of an event of interest observed from the time of entry to the study, and C_i would be the time between the patient's date of entry and the end of the study. We say this censoring is random censoring because individuals entered the study randomly according to their time of diagnosis.

2.3.1 Noninformative Random Censoring

When the lifetimes and censoring times are assumed to be mutually independent, that is, when $S_C(c, \gamma)$ does not depend on any of the parameters of $S_T(t, \theta)$, noninformative random censoring, or independent random censoring, happens. To assume independence, the lifetimes must have absolutely no bearing on the likelihood of the censoring time, not even through some indirect link (Crowder, 2001). This assumption is reasonable if patients withdraw from the study due to reasons that do not change the risk of failure. Therefore, the likelihood function is (Lawless, 2003)

$$\begin{aligned} L &= \prod_{i=1}^{n} [f_T(t_i) S_C(t_i)]^{I_i} [f_C(t_i) S_T(t_i)]^{1-I_i} \\ &\propto \prod_{i=1}^{n} f_T(t_i)^{I_i} S_T(t_i)^{1-I_i}. \end{aligned} \tag{2.3}$$

Because the likelihood functions for type I, type II, and noninformative random censoring are all reduced to the same form, standard methods of survival analysis do not distinguish among them and treat them all as generic right

censored observations.

2.3.2 Informative Random Censoring

Although independent censoring is assumed in many clinical trails for the sake of easy calculation, the estimator generated by this mechanism is biased if censoring is indeed dependent. Informative random censoring can happen when individuals are censored selectively or withdraw from the study because they appear to be at a higher or lower risk of failure compared to others at risk with the same covariate values (Kalbfleisch and Prentice, 2002). For example, a patient may withdraw from the study because of inadequate treatment response, or a deteriorating condition which needs an alternative treatment. In this case, withdrawal may indicate failure is likely to occur sooner than might have been expected otherwise. Ignoring this information will produce an overestimated survival function.

In the case where patients enter the study at different times, the patients' survival may have time trends that can induce a correlation between the failure and the censoring times. For example, if patients who enroll later tend to survive longer, a negative correlation will occur between the survival and censoring times. This correlation could occur if treatments improve over time, as might be expected if a patient receives a new surgical intervention. Because censoring is more likely to happen in those who enter later, the censoring is heavier for the better survivors, and ignoring this information will produce an underestimated survival function.

These examples clearly represent situations where the reasons for censoring might be related to the ultimate failure time, and so the use of only standard methods of analysis based on independent censoring would be inappropriate.

One easy way to deal with this problem is to include the entry time as a covariate, but doing so might cause computational difficulties if a high proportion of censoring time occurs (Allison, 1995), or if estimating a robust model is difficult (Zhang and Heitjan, 2005).

Another possibility is to directly estimate a joint model for the survival and censoring distributions. Theoretically, this estimation can be done by assuming that the joint density of the failure and censoring times is equal to the product of the desired marginal density of the failure time and the conditional density of the censoring time given the failure time. However, Tsiatis (1972) proved this problem is a nonidentifiability problem, so that the pair (t, I_i) provides insufficient information to determine the joint distribution of T and C. Crowder (2001) further showed that even when information on the pair (t, I_i) and the marginal distribution of T is known, the joint distribution of T and C is still not identifiable.

Since a test cannot be done for informative censoring versus noninformative censoring, and an assumption of independence cannot be verified (Peterson, 1976), recent studies have focused on sensitivity analysis, which is intended to evaluate to what extent the conclusions of a study can be affected by different assumptions concerning the censoring, after introducing additional assumptions on and models for the dependence structure between the failure time and the

dependent censoring.

Scharfstein et al. (2001) provided a semi-parametric approach to estimate the distributions of survival with dependent censored data, introducing a sensitivity analysis based on unmeasured factors. In order to draw an inference about the distribution of T, these researchers assumed that the relationships between the nonidentifiable term $f(T|t_i = t, \delta = 0)$ and the identifiable term $f(T|t_i > t)$ can be represented by

$$f(T|t_i = t, \delta = 0) = f(T|t_i > t)\frac{e^{q_t(T)}}{E(e^{q_t(T)}|t_i > t)},$$

where $q_t(T)$ is a known function of t and T, called the censoring bias function. When $q_t(T) = 0$ for all t, the assumed model is identical to the assumption of noninformative censoring. In order to facilitate the sensitivity analysis, a parameter vector α is introduced into the censoring bias function. Letting $q_t(T) = \alpha[T - (t + 1)]$ and varying α over a plausible range, we will have a different estimator by solving the estimating equation $\sum_{i=1}^{n} \mathbf{U(O_i; S, \gamma, q)} = \mathbf{0}$ (Scharfstein et al., 2001). When α goes to $-\infty$, the estimator attains the empirical version of the lower bound, corresponding to the assumption that improvement occurs immediately after censoring; when α goes to ∞, the estimator attains the upper bound, corresponding to the assumption that censored subjects never improve. The estimator becomes the Kaplan-Meier estimate when $\alpha = 0$. This method involves many complicated calculations and focuses on estimating a definitive nonignorable model.

Zhang and Heitjan (2005) proposed a sensitivity analysis that aims to estimate the effect of nonignorability censoring due to the end-of-study. They

argued that, in many clinical trials, some time trends in survival might induce a correlation between the survival and censoring times. Thus, these researchers modeled the joint distribution of the survival and censoring times as the marginal distribution of survival multiplied by a conditional model for censoring given survival by assuming that the distribution of the nonignorable end-of-study censoring given survival is beta with parameters $\alpha(t_i, \gamma), \beta(t_i, \gamma)$:

$$f(c_i|t_i) = \frac{1}{B[\alpha(t_i,\gamma),\beta(t_i,\gamma)]m}\left(\frac{c_i}{m}\right)^{\alpha(t_i,\gamma)-1}\left(1-\frac{c_i}{m}\right)^{\beta(t_i,\gamma)-1},$$

where

$$\begin{aligned}\alpha(t_i,\gamma) &= \frac{2}{1+e^{\gamma_1(t_i-\gamma_0)}},\\ \beta(t_i,\gamma) &= \frac{2e^{\gamma 1(t_i-\gamma_0)}}{1+e^{\gamma_1(t_i-\gamma 0)}}, \quad t_i \geq 0, \quad 0 \leq \frac{c_i}{m} \leq 1.\end{aligned}$$

Then, the likelihood function becomes (Zhang and Heitjan, 2005)

$$L = \prod_{i=1}^{n}\left[f_\theta(t_i)f_\gamma(c_i|t_i)\right]^{\delta_i}\left[\int_{c_i}^{\infty} f_\theta(u)f_\gamma(c_i|u)du\right]^{1-\delta_i},$$

where $\gamma = (\gamma_0, \gamma_1)^\top$, γ_1 is a nonignorability parameter that takes the value 0 when T and C are independent, γ_0 is included to improve interpretability, and $f_\theta(t_i)$ can be any parametric survival density function.

Under the assumption of $f_\gamma(c|t)$, Zhang and Heitjan (2005) defined the Index of Sensitivity to Nonignorability (ISNI), which measures the effect of small departures from the ignorability of the MLE, as the derivative of $\hat{\theta})(\gamma_1)$ with respect to γ_1 evaluated at $\gamma_1 = 0$:

$$\text{ISNI}(\hat{\theta}) = \left.\frac{\partial\hat{\theta}(\gamma_1)}{\partial\gamma_1}\right|_{\gamma_1=0}.$$

In general, the magnitude of ISNI increases with the fraction of observations that are censored, but its interpretation depends on the measurement units of the data. In order to overcome this problem, Zhang and Heitjan (2005) proposed a graphical local sensitivity analysis, based on the ISNI statistic, to access the sensitivity of inferences to nonignorable censoring. This access is achieved by looking at a plot of the expected standardized censoring time $E[\frac{c}{m}|t,\gamma]$ versus the survival time t. If the difference between $E[\frac{c}{m}|\min(t),\gamma]$ and $E[\frac{c}{m}|\max(t),\gamma]$ exceeds the real likely difference, the estimate of θ is insensitive to nonignorability. Otherwise, an additional modeling technique is required.

Siannis et al. (2005) proposed a sensitivity analysis for a lost-to-follow-up informative censoring parametric survival model. This analysis is based on the assumption that the conditional distribution of C given T has exactly the same parametric form as its marginal distribution $f_C(c,\gamma)$, but with the parameter allowed to depend on T through a bias function $B(t,\theta)$ and a dependence parameter δ:

$$P(C=c|T=t) = f_C(c, \gamma + \delta i_\gamma^{-\frac{1}{2}} B(t,\theta)), \tag{2.4}$$

where θ and γ are assumed to be scalar parameters, parameter δ represents the level of dependence between the failure and the censoring processes, and the scalar function $B(t,\theta)$ measures the pattern of this dependence. Note that true δ denotes as δ, assumed δ denotes as δ_*, and estimator of δ denotes as $\hat{\delta}$. Then

the joint density function of T and C is

$$\begin{aligned}
f_{T,C}(t,c) &= f_T(t,\theta) f_{C|T}(c|t) \\
&= f_T(t,\theta) f_C(c, \gamma + \delta i_\gamma^{-\frac{1}{2}} B(t,\theta)) \qquad (2.5) \\
&= f_T(t,\theta) f_C(c+0, \gamma + \delta i_\gamma^{-\frac{1}{2}} B(t,\theta)) \\
&\simeq f_T(t,\theta)[f_C(c,\gamma) + \frac{\partial f_C(c,\gamma)}{\partial c} 0 + \frac{\partial f_C(c,\gamma)}{\partial \gamma} \delta i_\gamma^{-\frac{1}{2}} B(t,\theta)] \\
&= f_T(t,\theta)[f_C(c,\gamma) + \frac{f_C(c,\gamma)}{f_C(c,\gamma)} \frac{\partial f_C(c,\gamma)}{\partial \gamma} \delta i_\gamma^{-\frac{1}{2}} B(t,\theta)] \\
&= f_T(t,\theta) f_C(c,\gamma)[1 + \delta i_\gamma^{-\frac{1}{2}} s_C(c,\gamma) B(t,\theta)]. \qquad (2.6)
\end{aligned}$$

By plugging (2.6) into the log-likelihood function, we have

$$
\begin{aligned}
L_\delta(\theta,\gamma) &= \sum_{i=1}^{n}\{I_i \log P(T=t_i \cap T<C) + (1-I_i)\log P(C=t_i \cap C<T)\} \qquad (2.7)\\
&\simeq \sum_{i=1}^{n}\Big\{I_i \log \int_{t_i}^{\infty} f_T(t_i,\theta) f_C(c,\gamma)[1+\delta i_\gamma^{-\frac{1}{2}} s_C(c,\gamma)B(t_i,\theta)]dc\\
&\quad +(1-I_i)\log \int_{t_i}^{\infty} f_T(t,\theta) f_C(t_i,\gamma)[1+\delta i_\gamma^{-\frac{1}{2}} s_C(t_i,\gamma)B(t,\theta)]dt\Big\}\\
&= \sum_{i=1}^{n}\Big\{I_i \log\Big[f_T(t_i,\theta)S_C(t_i,\gamma) + f_T(t_i,\theta)\delta i_\gamma^{-\frac{1}{2}} B(t_i,\theta)\int_{t_i}^{\infty} f_C(c,\gamma)s_C(c,\gamma)dc\Big]\\
&\quad +(1-I_i)\log\Big[f_C(t_i,\gamma)S_T(t_i,\theta) + f_C(t_i,\gamma)\delta i_\gamma^{-\frac{1}{2}} s_C(t_i,\gamma)\int_{t_i}^{\infty} f_T(t,\theta)B(t,\theta)dt\Big]\Big\}\\
&= \sum_{i=1}^{n}\Big\{I_i \log\Big[f_T(t_i,\theta)S_C(t_i,\gamma) + f_T(t_i,\theta)\delta i_\gamma^{-\frac{1}{2}} B(t_i,\theta)\frac{\partial}{\partial\gamma}S_C(t_i,\gamma)\Big]\\
&\quad +(1-I_i)\log\Big[f_C(t_i,\gamma)S_T(t_i,\theta) + f_C(t_i,\gamma)\delta i_\gamma^{-\frac{1}{2}} s_C(t_i,\gamma)\int_{t_i}^{\infty} f_T(t,\theta)B(t,\theta)dt\Big]\Big\}\\
&= \sum_{i=1}^{n}\Big\{I_i \log f_T(t_i,\theta)S_C(t_i,\gamma)\Big[1+\delta i_\gamma^{-\frac{1}{2}} B(t_i,\theta)\frac{\frac{\partial}{\partial\gamma}S_C(t_i,\gamma)}{S_C(t_i,\gamma)}\Big]\\
&\quad +(1-I_i)\log f_C(t_i,\gamma)S_T(t_i,\theta)\Big[1+\delta i_\gamma^{-\frac{1}{2}} s_C(t_i,\gamma)\frac{\int_{t_i}^{\infty} f_T(t,\theta)B(t,\theta)dt}{S_T(t_i,\theta)}\Big]\Big\}\\
&= \sum_{i=1}^{n}\Big\{I_i \log f_T(t_i,\theta)S_C(t_i,\gamma)\Big[1-\delta i_\gamma^{-\frac{1}{2}} B(t_i,\theta)\frac{\partial H_C(t_i,\gamma)}{\partial\gamma}\Big]\\
&\quad +(1-I_i)\log f_C(t_i,\gamma)S_T(t_i,\theta)\Big[1+\delta i_\gamma^{-\frac{1}{2}} s_C(t_i,\gamma)\mu(t_i,\theta)\Big]\Big\}\\
&= \sum_{i=1}^{n}\big\{I_i \log\big[f_T(t_i,\theta)S_C(t_i,\gamma)\big] + (1-I_i)\log\big[f_C(t_i,\gamma)S_T(t_i,\theta)\big]\big\}\\
&\quad +\delta i_\gamma^{-\frac{1}{2}}\sum_{i=1}^{n}\big\{(1-I_i)\mu(t_i,\theta)s_C(t_i,\gamma) - I_i B(t_i,\theta)\frac{\partial H_C(t_i,\gamma)}{\partial\gamma}\big\}\\
&= L_0(\theta,\gamma) + \delta i_\gamma^{-\frac{1}{2}}\sum_{i=1}^{n}\big\{(1-I_i)\mu(t_i,\theta)s_C(t_i,\gamma) - I_i B(t_i,\theta)\frac{\partial H_C(t_i,\gamma)}{\partial\gamma}\big\} \qquad (2.8)
\end{aligned}
$$

where

$$\mu(t_i,\theta) = \frac{\int_{t_i}^{\infty} f_T(t,\theta)B(t,\theta)dt}{S_T(t_i,\theta)},$$

and

$$
\begin{aligned}
L_0(\theta,\gamma) &= \sum_{i=1}^{n}\big\{I_i\log\big[f_T(t_i,\theta)S_C(t_i,\gamma)\big]+(1-I_i)\log\big[f_C(t_i,\gamma)S_T(t_i,\theta)\big]\big\}\\
&= \sum_{i=1}^{n}\big[I_i\log f_T(t_i,\theta)+I_i\log S_C(t_i,\gamma)\\
&\quad +\log f_C(t_i,\gamma)+\log S_T(t_i,\theta)-I_i\log f_C(t_i,\gamma)-I_i\log S_T(t_i,\theta)\big]\\
&= \sum_{i=1}^{n}\big[I_i\log\frac{f_T(t_i,\theta)}{S_T(t_i,\theta)}-I_i\log\frac{f_C(t_i,\gamma)}{S_C(t_i,\gamma)}+\log f_C(t_i,\gamma)+\log S_T(t_i,\theta)\big]\\
&= \sum_{i=1}^{n}\big[I_i\log\frac{f_T(t_i,\theta)}{S_T(t_i,\theta)}-I_i\log\frac{f_C(t_i,\gamma)}{S_C(t_i,\gamma)}+\log f_C(t_i,\gamma)+\log S_T(t_i,\theta)\\
&\quad +\log\frac{f_C(t_i,\gamma)}{S_C(t_i,\gamma)}-\log\frac{f_C(t_i,\gamma)}{S_C(t_i,\gamma)}\big]\\
&= \sum_{i=1}^{n}\big[I_i\log h_T(t_i,\theta)+(1-I_i)\log h_C(t_i,\gamma)+\log S_C(t_i,\gamma)+\log S_T(t_i,\theta)\\
&= \sum_{i=1}^{n}\big[I_i\log h_T(t_i,\theta)+(1-I_i)\log h_C(t_i,\gamma)-H_C(t_i,\gamma)-H_T(t_i,\theta)\big] \qquad (2.9)
\end{aligned}
$$

is the likelihood function for all the parameters when T and C are independent.

Recall equation (2.8). It is equivalent to

$$
L_\delta(\theta,\gamma)-L_0(\theta,\gamma)=\delta i_\gamma^{-\frac{1}{2}}\sum_{i=1}^{n}\big[(1-I_i)\mu(t_i,\theta)s_C(t_i,\gamma)-I_iB(t_i,\theta)\frac{\partial H_C(t_i,\gamma)}{\partial\gamma}\big].
$$

If we take derivative $\frac{\partial}{\partial\theta}|_{\theta=\hat\theta_\delta}$ on both sides of the above equation, we obtain

$$
\begin{aligned}
&\frac{\partial L_\delta(\theta,\gamma)}{\partial\theta}|_{\theta=\hat\theta_\delta}-\frac{\partial L_0(\theta,\gamma)}{\partial\theta}|_{\theta=\hat\theta_\delta}\\
&= \delta i_\gamma^{-\frac{1}{2}}\sum_{i=1}^{n}\big[(1-I_i)\frac{\partial\mu(t_i,\theta)}{\partial\theta}s_C(t_i,\gamma)-I_i\frac{\partial B(t_i,\theta)}{\partial\theta}\frac{\partial H_C(t_i,\gamma)}{\partial\gamma}\big]_{\theta=\hat\theta_\delta} \qquad (2.10)
\end{aligned}
$$

Using the fact that $\theta=\hat\theta_\delta$ gives the maximum of $L_\delta(\theta,\gamma)$, we obtain the left hand side of equation (2.10), which is equal to

$$
\frac{\partial L_\delta(\theta,\gamma)}{\partial\theta}|_{\theta=\hat\theta_\delta}-\frac{\partial L_0(\theta,\gamma)}{\partial\theta}|_{\theta=\hat\theta_\delta}=-\frac{\partial L_0(\theta,\gamma)}{\partial\theta}|_{\theta=\hat\theta_\delta}.
$$

Furthermore, since $\hat{\theta}_0$ is in the neighbourhood of $\theta = \hat{\theta}_\delta$, we can use Taylor expansion to obtain

$$\begin{aligned} -\frac{\partial L_0(\theta,\gamma)}{\partial\theta}|_{\theta=\hat{\theta}_\delta} &= -\frac{\partial L_0(\theta,\gamma)}{\partial\theta}|_{\theta=\hat{\theta}_0} - \frac{\partial^2 L_0(\theta,\gamma)}{\partial^2\theta}|_{\theta=\hat{\theta}_0}(\hat{\theta}_\delta - \hat{\theta}_0) - \cdots \\ &\simeq -\frac{\partial^2 L_0(\theta,\gamma)}{\partial\theta^2}|_{\theta=\hat{\theta}_0}(\hat{\theta}_\delta - \hat{\theta}_0). \end{aligned}$$

Therefore, we obtain

$$\begin{aligned} &-\frac{\partial^2 L_0(\theta,\gamma)}{\partial\theta^2}|_{\theta=\hat{\theta}_0}(\hat{\theta}_\delta - \hat{\theta}_0) \\ &\simeq \delta i_\gamma^{-\frac{1}{2}} \sum_{i=1}^{n} \big[(1-I_i)\frac{\partial\mu(t_i,\theta)}{\partial\theta}s_C(t_i,\gamma) - I_i\frac{\partial B(t_i,\theta)}{\partial\theta}\frac{\partial H_C(t_i,\gamma)}{\partial\gamma}\big]_{\theta=\hat{\theta}_\delta}. \end{aligned}$$

Dividing the factor $\frac{\partial^2 L_0(\theta,\gamma)}{\partial^2\theta}|_{\theta=\hat{\theta}_0}$ on both sides of the above equation, we obtain

$$\hat{\theta}_\delta - \hat{\theta}_0 \simeq \delta i_\gamma^{-\frac{1}{2}} \iota(\theta)^{-1} \sum_{i=1}^{n} \big\{(1-I_i)\frac{\partial\mu(t_i,\theta)}{\partial\theta}s_C(t_i,\gamma) - I_i\frac{\partial B(t_i,\theta)}{\partial\theta}\frac{\partial H_C(t_i,\gamma)}{\partial\gamma}\big\} \quad (2.11)$$

where

$$\iota(\theta) = -\frac{\partial^2 L_0(\theta,\gamma)}{\partial\theta^2}. \quad (2.12)$$

In (2.11), $\hat{\theta}_\delta$ is the value of θ which maximizes (2.8), and $\hat{\theta}_0$ is the usual maximum likelihood estimate from (2.9). Therefore, the maximum likelihood estimator for a given small value of δ is approximately equal to the estimate that would be obtained by assuming ignorable censoring plus δ multiplied by a sensitivity index U, which depends on the observed pattern of the censored observations:

$$\hat{\theta}_\delta = \hat{\theta}_0 + \delta U + O(\delta^2). \quad (2.13)$$

Siannis et al. (2005) called δU the bias correction. The bias function $B(t,\theta)$ has been suggested to be taken as the standardized score function of the distribution

of T unless an expert judgement can be made of the practical context. By applying this method, a complicated analysis can be avoided if the censoring is slight, and the robustness to small departures from ignorability can be accessed quickly. However, Siannis' approximation is valid under two conditions: (1) the true dependence between T and C is small, and (2) the linear approximation is accurate for a small value of δ. When the dependence between T and C is highly correlated, using an explicit dependent risk model is more appropriate. Therefore, in this thesis, we use likelihood-based methodology to estimate the parameters based on Siannis' assumption about the conditional density of T and C (2.4).

Note that formulas (2.6), (2.7), (2.8), (2.9), (2.11), and (2.13) are results presented by Siannis et al. (2005). All we did here was to provide a detailed demonstration of their accuracy.

2.3.3 Group Comparison

Assume there are two study groups, 1 and 2, with the failure times in group i governed by an exponential distribution with parameter θ_i. To compare the survival experience of these two groups, we need to compare θ_1 and $\theta_2 = \theta_1 + \beta$. It follows that $\beta = 0$ corresponds to no difference in survival experience in the two groups.

Because these two groups are independent, the log-likelihood function for the observed data from the two study groups is given by

$$L = L_1 + L_2,$$

where L_1 is the log-likelihood function for group 1, and L_2 is the log-likelihood function for group 2. Once we have this log-likelihood function, we can write out the score vector and observed information matrix routinely. Consider a test of the hypothesis $\beta = 0$; since the asymptotic distribution of $\hat{\beta}$ is $\mathrm{N}(\beta, \mathrm{Var}(\hat{\beta}))$, the appropriate test statistic would be $\frac{|\hat{\beta}|}{\mathrm{SD}(\hat{\beta})}$. See Chapter 5, Data Analysis, for more details.

Chapter 3

Exploratory Analysis

To study a parametric model where informative random censoring exists, Siannis et al. (2005) introduced parameter δ and bias function $B(t, \theta)$ to reflect the association between the failure time T and the censoring processes C. The parameter δ can be thought of as measuring the size of the dependence between T and C, and the bias function $B(t, \theta)$ as measuring the pattern of this dependence. A relatively simple sensitivity analysis on the parameter of interest is approximated by the estimate that would be obtained by assuming ignorable censoring plus the bias correction (2.13). In order to understand how accurate the approximation is by comparing it to what can be obtained by maximizing the likelihood function, we use 3D plots to demonstrate the difference. As well, we use graphs to illustrate what the probability of a failure time T is given a different censoring time C by varying δ. In addition, we explore the association between the dependence parameter δ and the Spearman's rank correlation

coefficient ρ to understand what other parameters might effect δ.

3.1 Exact Method vs. Siannis' Approximation

Under Siannis et al.'s assumption (2.4), sensitivity analysis can be conducted by using two methods: the exact method and Siannis et al.'s approximation. For the exact method, we estimate parameters by maximizing the profile likelihood function for δ (2.7). For Siannis et al.'s approximation, we estimate parameters by using formula (2.11). In order to have a way to understand how accurate the approximation method is, we compare the conditional distribution of C given T under different δ under these two methods.

For simplification, we assume the special case where the marginal distribution of T and C has the parametric proportional hazard structure

$$h_T(t,\theta) = e^{\theta} h_T^*(t); \qquad h_C(c,\gamma) = e^{\gamma} h_C^*(c),$$

and the marginal models are the exponential distribution

$$f_T(t,\theta) = e^{\theta} e^{-e^{\theta} t}; \qquad f_C(c,\gamma) = e^{\gamma} e^{-e^{\gamma} c}. \tag{3.1}$$

Therefore, the score function is

$$\begin{aligned} s_T(t,\theta) &= \frac{\partial}{\partial\theta} \log f_T(t,\theta) \\ &= \frac{\partial}{\partial\theta} \log h_T(t,\theta) S_T(t,\theta) \\ &= \frac{(\frac{\partial}{\partial\theta} h_T(t,\theta)) S_T(t,\theta) + h_T(t,\theta) \frac{\partial}{\partial\theta} S_T(t,\theta)}{h_T(t,\theta) S_T(t,\theta)} \\ &= \frac{h_T(t,\theta) S_T(t,\theta) - h_T(t,\theta) e^{-H_T(t,\theta)} H_T(t,\theta)}{h_T(t,\theta) S_T(t,\theta)} \\ &= 1 - H_T(t,\theta). \end{aligned} \tag{3.2}$$

Similarly, $s_C(c,\gamma) = 1 - H_C(c,\gamma)$ with variance $i_\theta = i_\gamma = 1$. Thus, in the case of exponential distribution, the exact conditional distribution of C given T becomes

$$\begin{aligned} f_{C|T}(c,t) &= e^{\gamma+\delta i_\gamma^{-\frac{1}{2}} B(t,\theta)} e^{-e^{\gamma+\delta i_\gamma^{-\frac{1}{2}} B(t,\theta)} c} \\ &= e^{\gamma+\delta(1-e^\theta t)} e^{-e^{\gamma+\delta(1-e^\theta t)} c}. \end{aligned} \tag{3.3}$$

By using Siannis' approximation, the conditional distribution becomes

$$\begin{aligned} f_{C|T}(c,t) &\simeq f_C(c,\gamma)[1+\delta i_\gamma^{-\frac{1}{2}} s_C(c,\gamma) B(t,\theta)] \\ &= e^\gamma e^{-e^\gamma c}[1+\delta(1-e^\gamma c)(1-e^\theta t)]. \end{aligned} \tag{3.4}$$

Figures 3.1 to 3.5 help us to understand how accurate the approximation is under different values of δ. We take the special case where $\theta = -3$, and $\gamma = -4$; then the median survival time of T is 14 months and that of C is 38 months. In this setting, the censoring rate is around 30% from a simulation study with 5000 repeated data. T and C range between 0 and 80 months as for clinical trials' study periods are longer than six years.

Figure 3.1 is a 3D plot when δ is 0. Note that if $\delta = 0$, T and C are independent and the censoring is ignorable. Therefore, both methods are reduced to the same formula, no matter what value T is, and the probability that a censoring event will happen follows exponential distribution:

$$\begin{aligned} f_{C|T}(C=c|T=t) &= f_C(c, \gamma + \delta i_\gamma^{-\frac{1}{2}} B(t,\theta)) \\ &= f_C(c,\gamma) \\ &= e^\gamma e^{-e^\gamma c}. \end{aligned}$$

For example, when a new drug is introduced to cancer patients, researchers want to study their survival time. If a dropout happens because the patient moves to another city, we say the dropout (C) and the survival time (T) are independent. Under both methods, the probability of this kind of dropout follows the same exponential distribution regardless of the survival time. Figure 3.6 reflects this result.

Figure 3.2 presents the 3D plot when $\delta = 0.3$. In this case, both methods have similar conditional density when $T < 60$ months. When $T > 60$ months, Siannis et al.'s approximation has lower probability of censoring during the early stage of the study compared to the exact method. This difference means Siannis et al.'s approximation is close to that of the exact method for the first 5 years of the study. Figure 3.7 reflects this result. In this figure, the dashed line represents Siannis et al.'s method, and the solid line represents the exact method.

Figure 3.3 presents the 3D plot when $\delta = 0.5$. As the dependence parameter δ gets larger, Siannis et al.'s approximation is less close to that of the exact method for a large value of T. Figure 3.8 shows that the two methods agree with each other when $T < 50$ months.

When a negative dependent relation exists, the two methods agree with each other in a similar manner. Figures 3.4 and 3.5 present the 3D plots when $\delta = -0.3$ and $\delta = -0.5$, respectively. Figures 3.9 and 3.10 present the 2D plots for different T. These figures show that the two methods agree with each other when $T < 50$ months for $\delta = -0.3$ and $T < 40$ months for $\delta = -0.5$.

3.2 Relation between δ and ρ

In order to explore the relation between δ and ρ, the Spearman's rank correlation coefficient, we simulated data according to (3.3) by varying (θ, γ) as $(-2, -3), (-3, -4), (-4, -5)$, δ as $(-0.5, -0.3, 0, 0.3, 0.5)$, and n as $(50, 100, 500)$. For each combination (45 in total), we generated 5000 repeated data sets and calculated the mean Spearman's correlation $\bar{\rho}$, which was found to change for different combinations of δ and n, but not for different (θ, γ). Table 3.1 presents the results. It shows that for each n, $\bar{\rho}$ increases or decreases as δ increases or decreases with the same sign. We simulated data using exponential distribution. Under this simple model, the dependence parameter δ has the same magnitude of the Spearman's rank correlation coefficient between T and C.

Figures 3.11, 3.12, and 3.13 present the scatter plot of one data set. The data set was chosen with $\rho \simeq \bar{\rho}$.

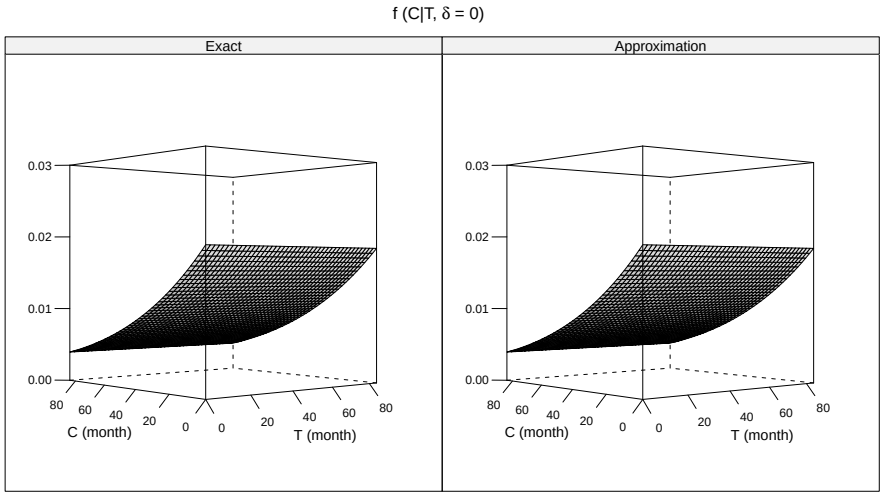

Figure 3.1: A 3D comparison of the conditional density for the exact method to that using Siannis' approximation; x axis, T(month), is the survival time, y axis, C(month), is the censoring time, z axis is the value of conditional density for $\delta = 0$.

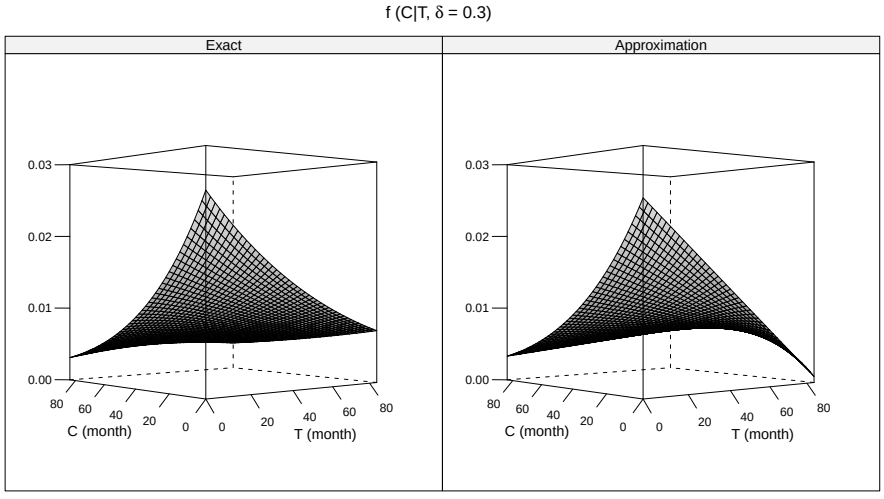

Figure 3.2: See note for Figure 3.1. This figure is for $\delta = 0.3$.

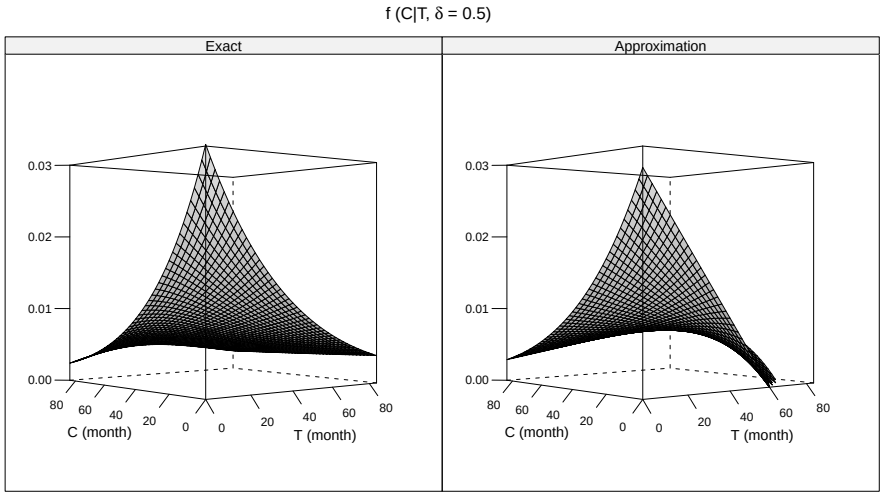

Figure 3.3: See note for Figure 3.1. This figure is for $\delta = 0.5$.

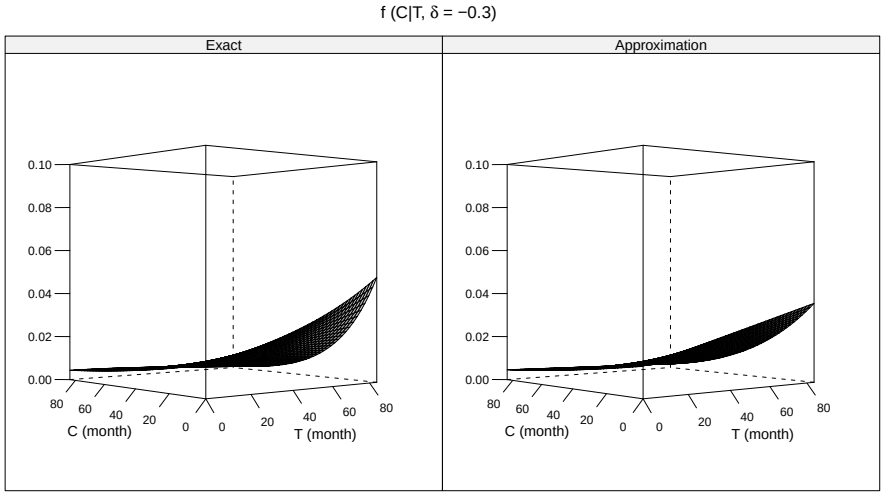

Figure 3.4: See note for Figure 3.1. This figure is for $\delta = -0.3$.

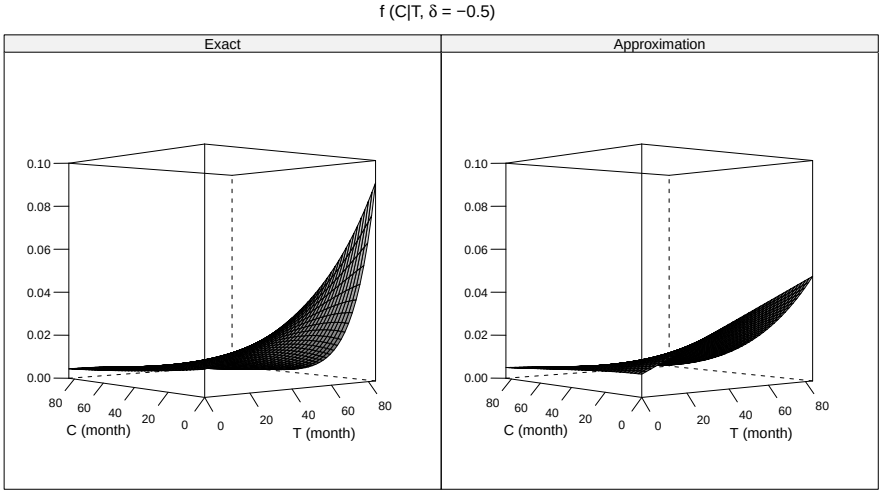

Figure 3.5: See note for Figure 3.1. This figure is for $\delta = -0.5$.

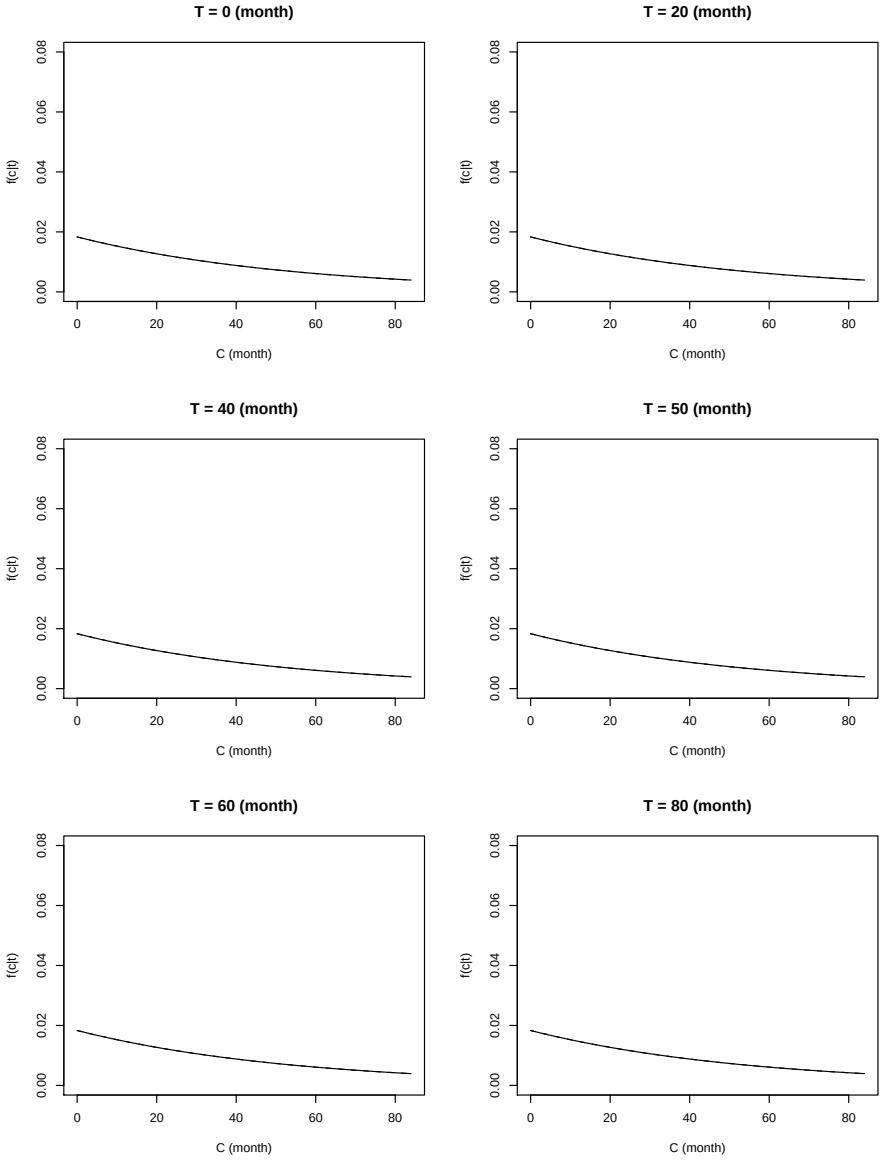

Figure 3.6: A 2D comparison of conditional density for the exact method to Siannis' approximation; x axis, C(month), is the censoring time, y axis is the value of conditional density for various survival time T.

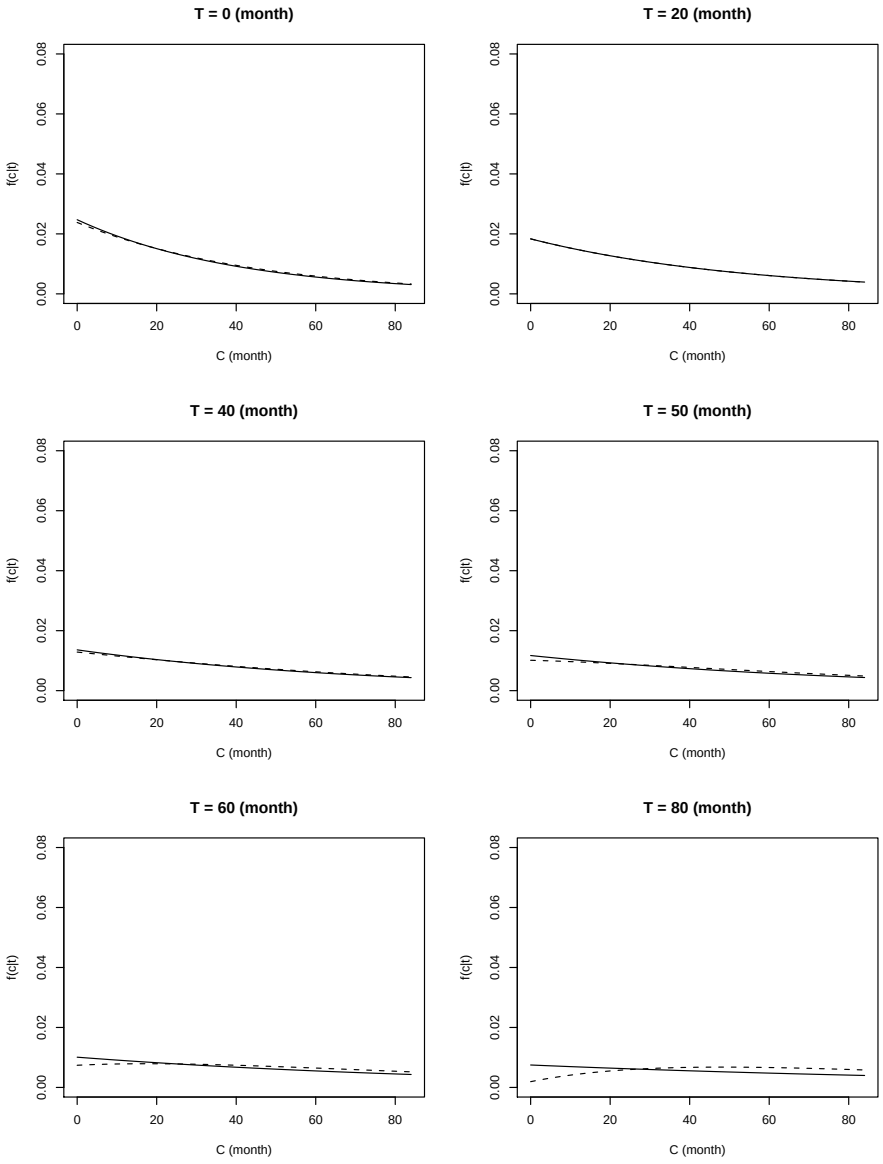

Figure 3.7: See note for Figure 3.6. This figure is for $\delta = 0.3$.

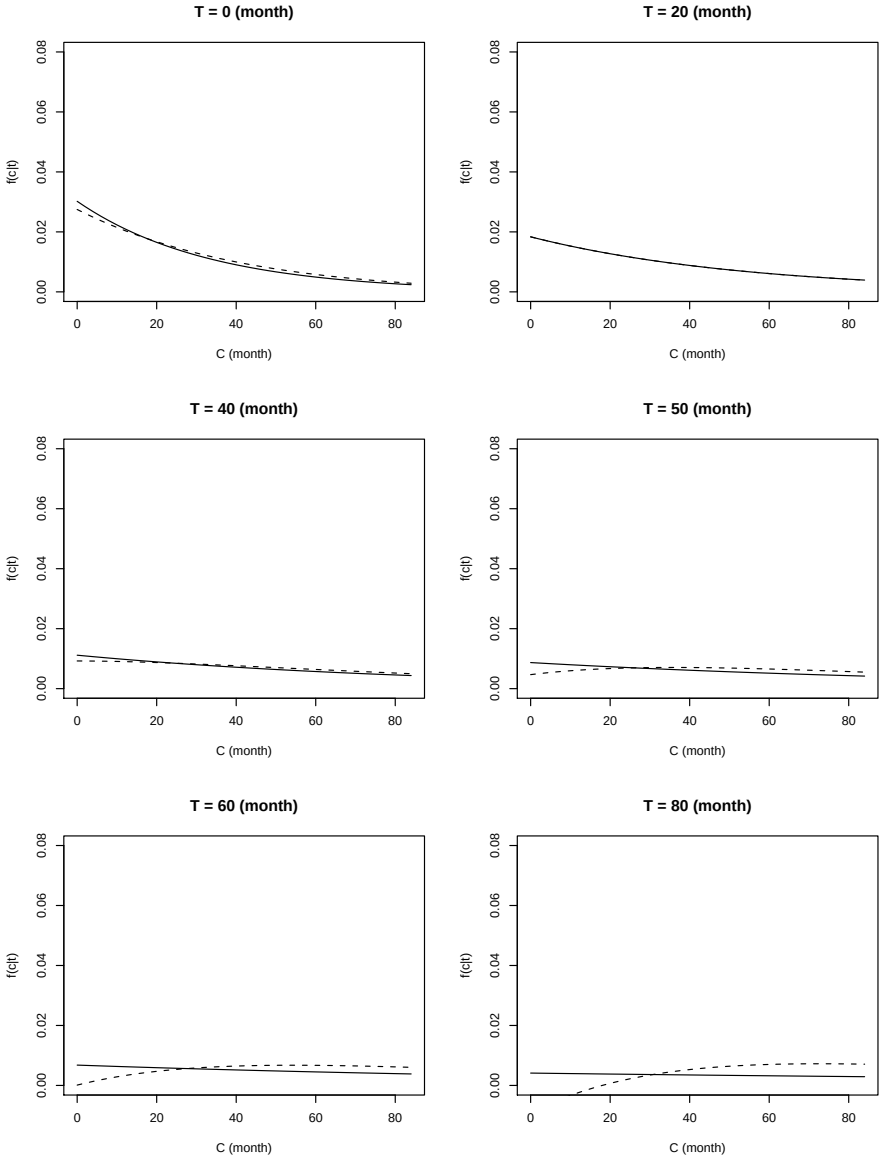

Figure 3.8: See note for Figure 3.6. This figure is for $\delta = 0.5$.

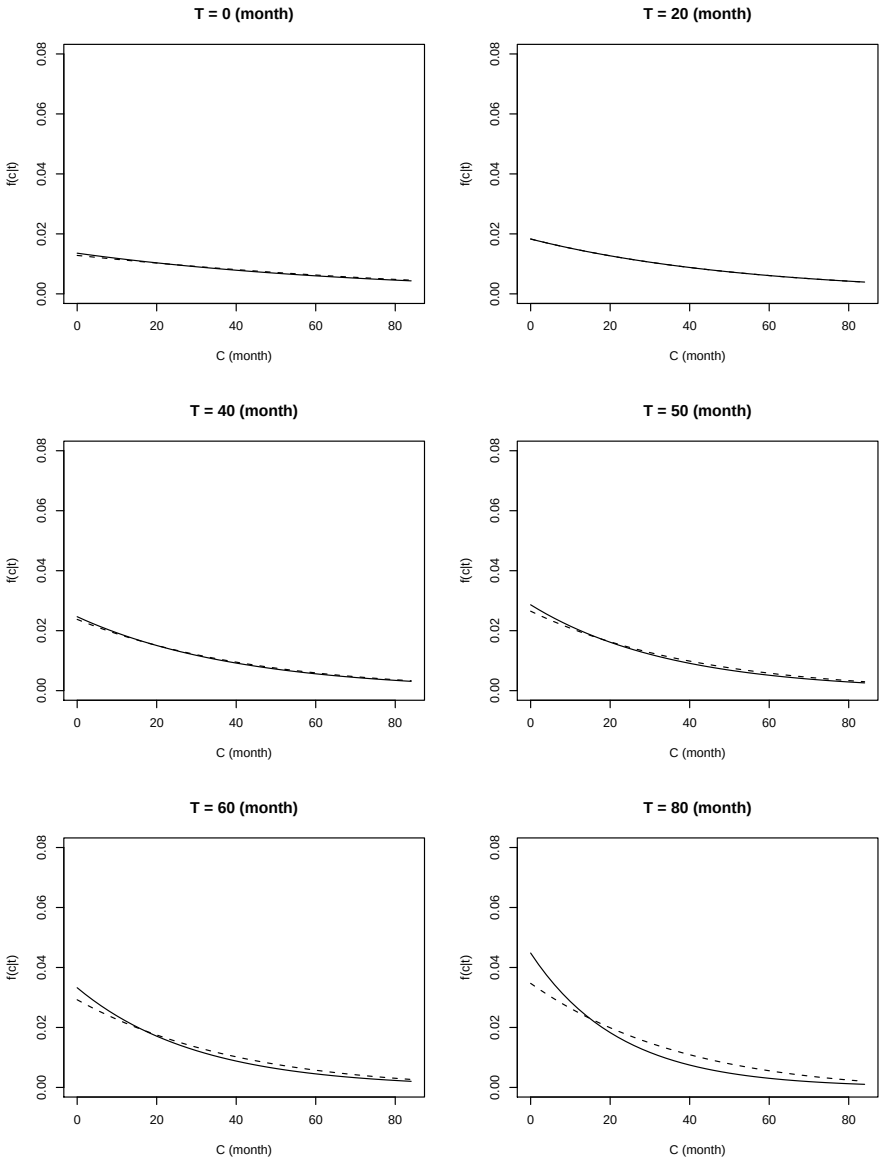

Figure 3.9: See note for Figure 3.6. This figure is for $\delta = -0.3$.

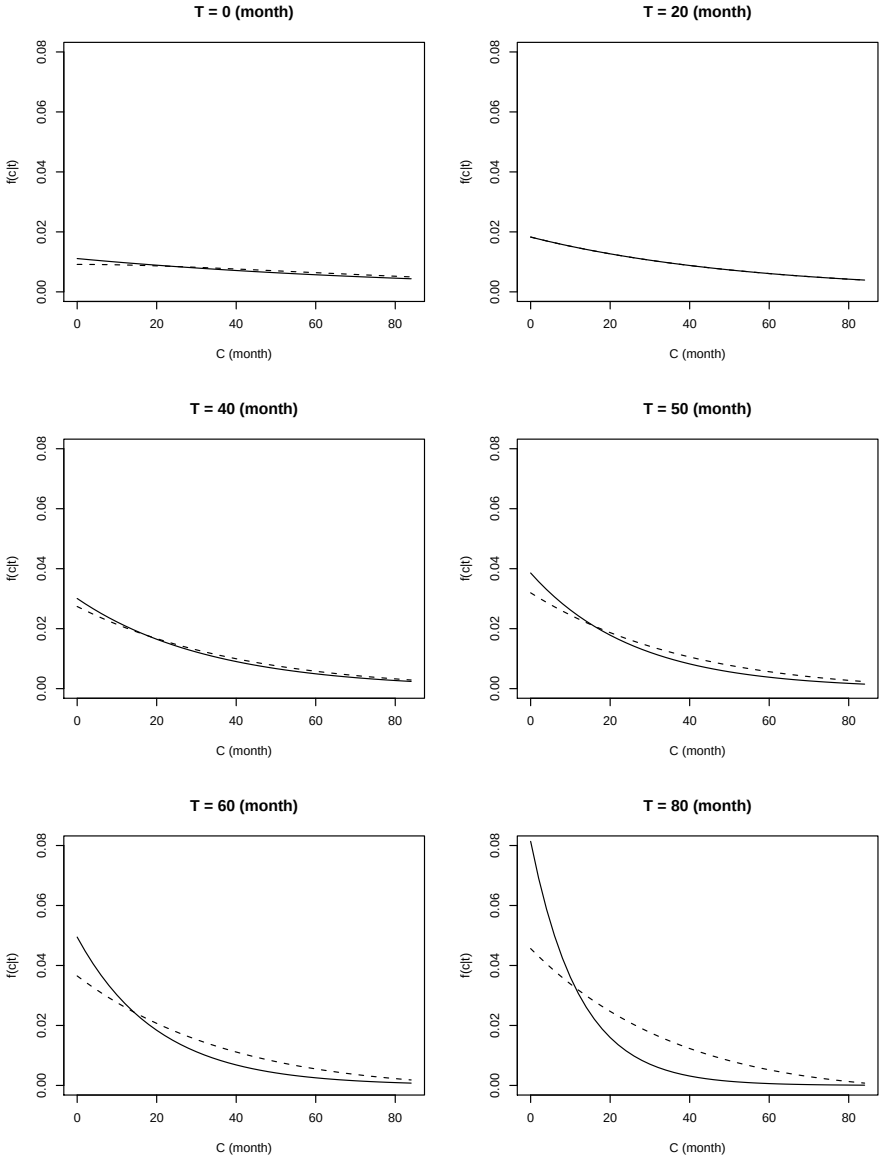

Figure 3.10: See note for Figure 3.6. This figure is for $\delta = -0.5$.

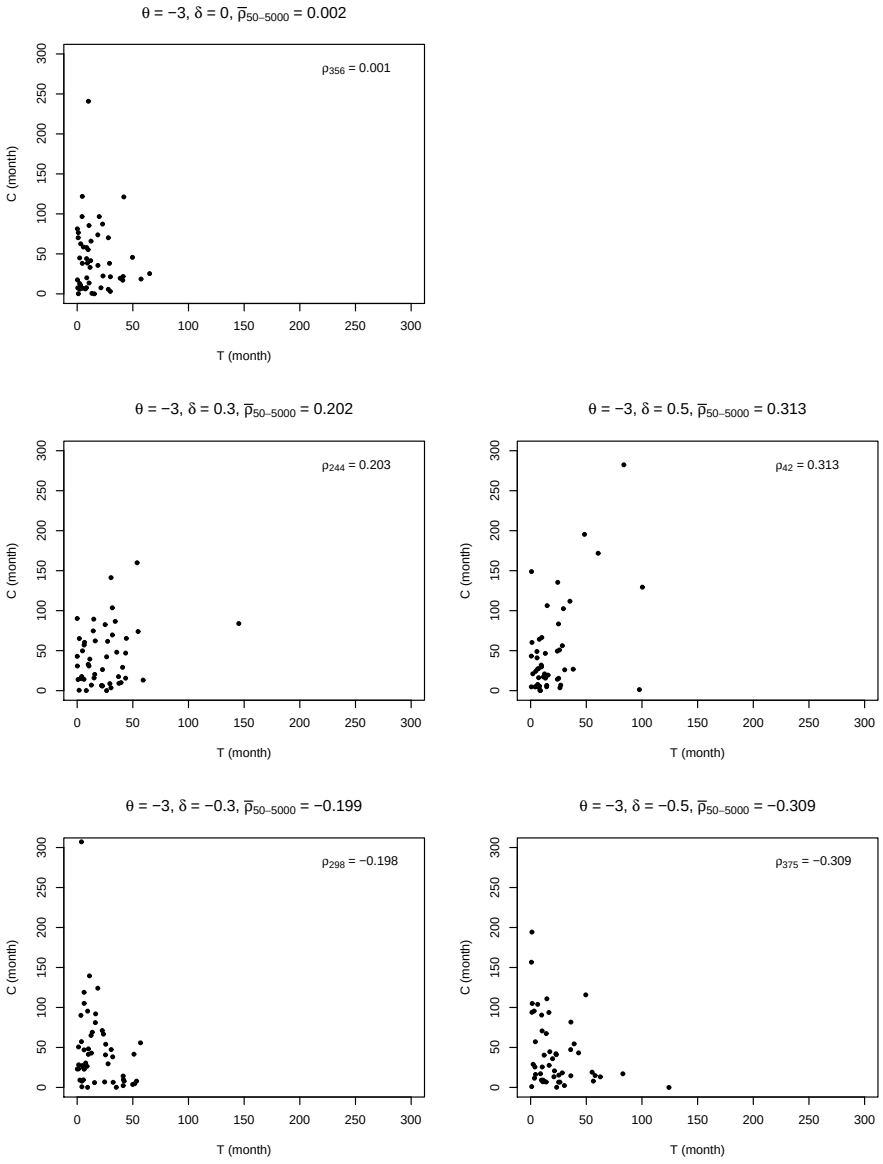

Figure 3.11: A scatter plot of C and T for one of the 5000 samples. ρ_i and $\bar{\rho}$ are the Spearman's correlation for the ith sample and the average of 5000 samples, respectively. This figure is for $n = 50$.

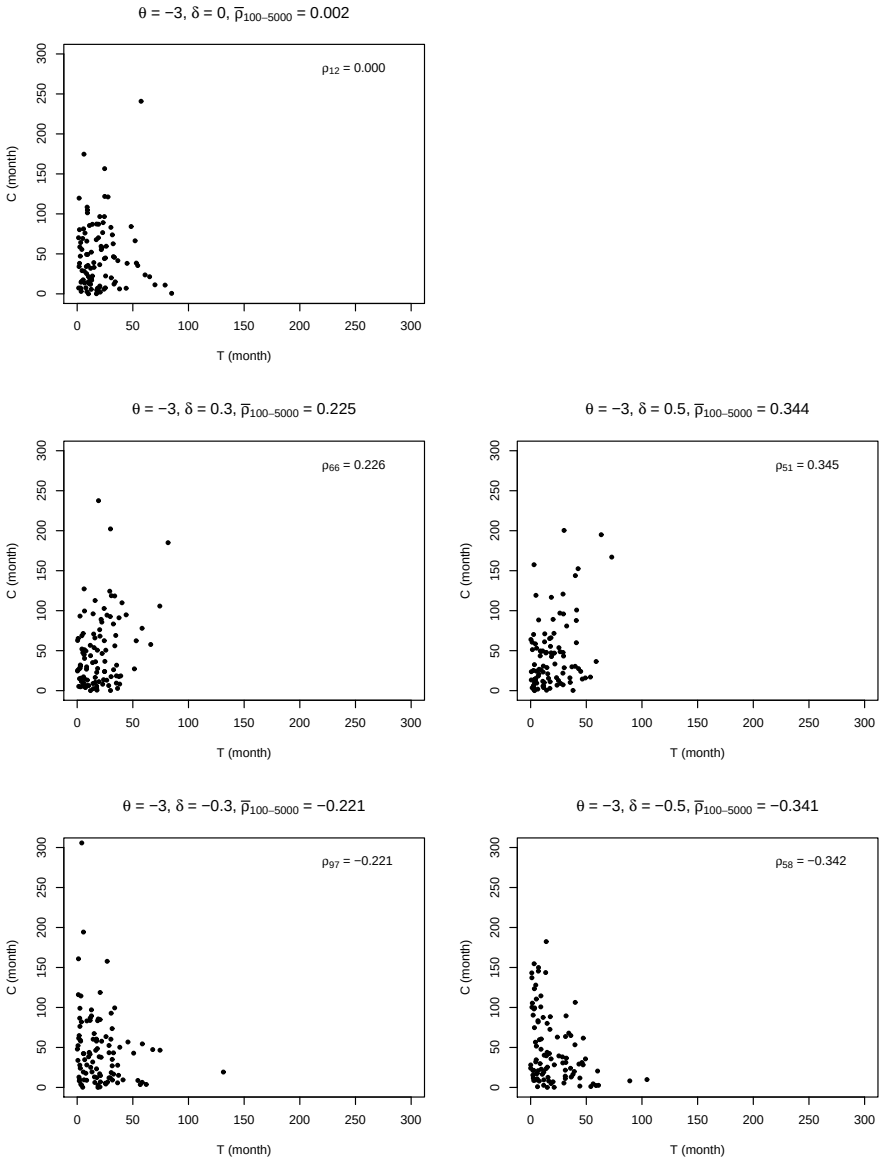

Figure 3.12: See note for Figure 3.11. This figure is for $n = 100$.

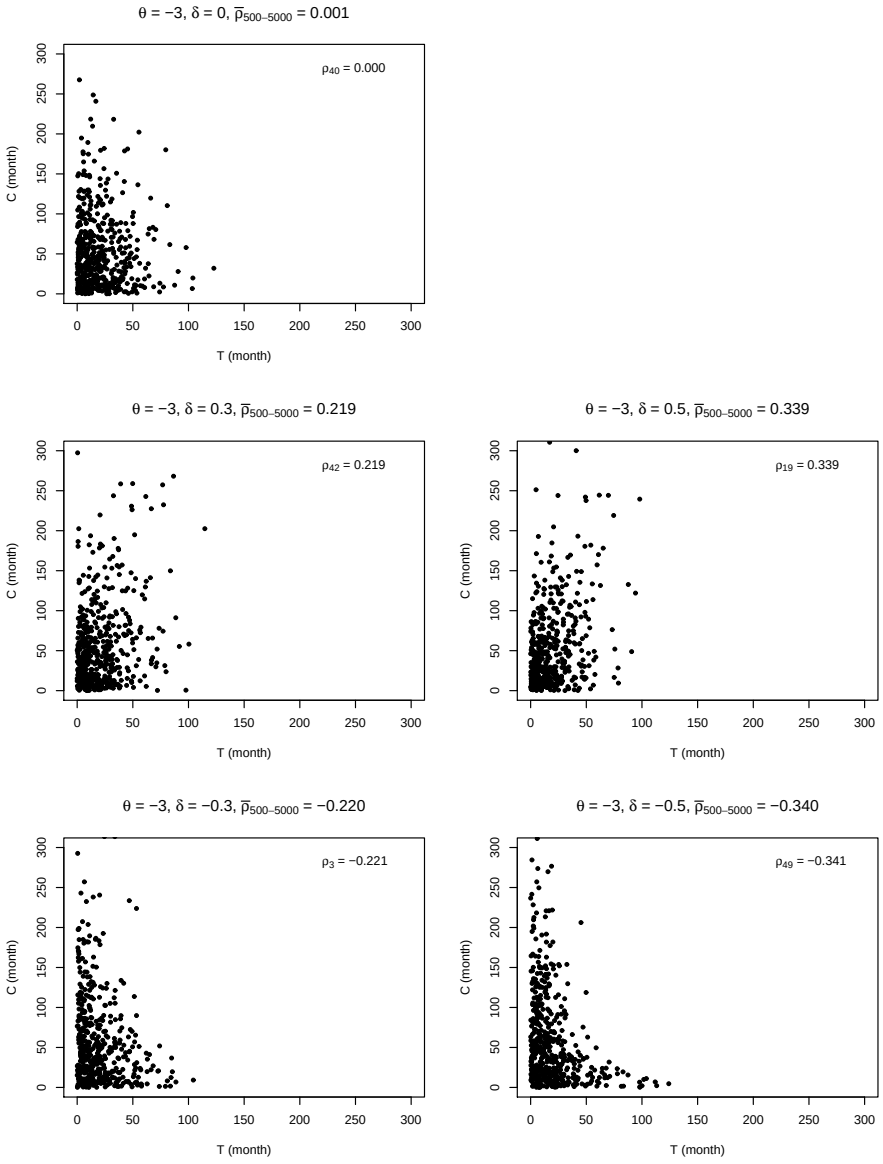

Figure 3.13: See note for Figure 3.11. This figure is for $n = 500$.

Table 3.1: Mean Spearman's rank correlation coefficient ($\bar{\rho}$) from 5000 repeated data with the standard error in parenthesis.

n\δ	-0.5	-0.3	0	0.3	0.5
50	-0.309	-0.199	0.002	0.202	0.313
	(0.018)	(0.020)	(0.021)	(0.020)	(0.019)
100	-0.341	-0.221	0.002	0.225	0.344
	(0.009)	(0.009)	(0.010)	(0.010)	(0.009)
500	-0.340	-0.220	-0.001	0.219	0.339
	(0.002)	(0.002)	(0.002)	(0.002)	(0.002)

Chapter 4

Simulation Study

4.1 Comparison of sensitivity analysis and MLE

We conducted a simulation study to evaluate the finite sample performance of Siannis et al.'s estimation and to compare it with the maximum likelihood estimation (MLE) method.

We generated data under the assumption that T is exponentially distributed with mean $e^{-\theta}$, where $\theta = -3$, and C given T is exponentially distributed with mean $e^{-\gamma-\delta i_\gamma^{-\frac{1}{2}} B(t,\theta)}$, where $\gamma = -4$, $i_\gamma^{-\frac{1}{2}} = 1$, and $B(t_i, \theta) = 1 - e^{\theta} t_i$. We varied δ in $(-0.5, -0.3, 0, 0.3, 0.5)$, and the sample size was 50, 100, or 200. We repeated each combination 5000 times to obtain the average of the estimator.

In this exponential distribution, the marginal models of T and C are as in (3.1), the score function $s_T(t, \theta)$ is (3.2), and the bias function is as suggested

by Siannis et al. (2005) and simplified as

$$\begin{aligned} B(t,\theta) &= i_\theta^{-\frac{1}{2}} s_T(t,\theta) \\ &= 1 - H_T(t,\theta). \end{aligned} \tag{4.1}$$

In order to obtain Siannis et al.'s estimates, note that

$$\begin{aligned} \mu(t,\theta) &= \frac{\int_t^\infty f_T(t,\theta)B(t,\theta)du}{S_T(t,\theta)} \\ &= \frac{\int_t^\infty [1-H_t(t,\theta)]f_T(u,\theta)du}{S_T(t,\theta)} \\ &= \frac{\int_t^\infty f_T(u,\theta)du}{S_T(t,\theta)} - \frac{\int_t^\infty H_t(t,\theta)f_T(u,\theta)du}{S_T(t,\theta)} \\ &= 1 - \frac{H_T(t,\theta)S_T(t,\theta) - \int_t^\infty h_T(t,\theta)S_T(t,\theta)du}{S_T(t,\theta)} \\ &= 1 - \frac{H_T(t,\theta)S_T(t,\theta) - \int_t^\infty f_T(t,\theta)du}{S_T(t,\theta)} \\ &= 1 - \frac{H_T(t,\theta)S_T(t,\theta) - S_T(t,\theta)}{S_T(t,\theta)} \\ &= -H_T(t,\theta), \end{aligned} \tag{4.2}$$

and from (2.9), (2.12),

$$\begin{aligned} \iota(\theta) &= -\frac{\partial^2 L_0(\theta,\gamma)}{\partial\theta^2} \\ &= \sum_{i=1}^n e^\theta H_T(t_i,\theta) \\ &= \sum_{i=1}^n H_T(t_i,\theta) \\ &= \sum_{i=1}^n e^\theta t_i. \end{aligned}$$

By plugging the above functions into (2.11), we can obtain

$$\begin{aligned}
\hat{\theta}_\delta - \hat{\theta}_0 &\simeq \delta i_\gamma^{-\frac{1}{2}} \iota(\theta)^{-1} \sum_{i=1}^{n} \Big\{ (1-I_i) \frac{\partial \mu(t_i,\theta)}{\partial \theta} s_C(t_i,\gamma) - I_i \frac{\partial B(t_i,\theta)}{\partial \theta} \frac{\partial H_C(t_i,\gamma)}{\partial \gamma} \Big\} \\
&= \delta \iota(\theta)^{-1} \sum_{i=1}^{n} \big\{ (1-I_i)(-H_T(t_i,\theta))(1-H_C(t_i,\gamma)) - I_i(-H_T(t_i,\theta))(H_C(t_i,\gamma)) \big\} \\
&= \delta \iota(\theta)^{-1} \sum_{i=1}^{n} \big\{ H_T(t_i,\theta) H_C(t_i,\gamma) - (1-I_i) H_T(t_i,\theta) \big\} \\
&= \delta \frac{1}{\sum_{i=1}^{n} e^\theta t_i} \sum_{i=1}^{n} \{ e^\theta t_i e^\gamma t_i - (1-I_i) e^\theta t_i \} \\
&= \delta \frac{\sum_{i=1}^{n} \{ e^\gamma t_i^2 - (1-I_i) t_i \}}{\sum_{i=1}^{n} t_i}. \qquad (4.3)
\end{aligned}$$

Siannis et al.'s estimation of $\hat{\theta}$, $\hat{\theta}_\delta$, is a function of δ, which is suggested to change from -0.3 to 0.3. The fourth column of Table 4.1 present the results obtained by using this method.

According to Siannis et al.'s method, the asymptotic confidence interval for θ is approximately

$$(\hat{\theta}_{\delta=-0.3} - z_\alpha \mathrm{SE}(\hat{\theta}_{\delta=0}), \hat{\theta}_{\delta=0.3} + z_\alpha \mathrm{SE}(\hat{\theta}_{\delta=0})), \qquad (4.4)$$

where z_α is the appropriate standard normal percentage point. In fact, this result is just the standard confidence interval for the ignorable model shifted by the bias correction.

To obtain MLE, we first write out the log-likelihood function under Siannis et al.'s assumption in the special case of exponential distribution. By plugging

(3.1) and (3.3) into (2.7), we have

$$\begin{aligned}
L(\theta,\gamma,\delta) &= \sum_{i=1}^{n} I_i log P(T=t_i \cap T<C) + (1-I_i) log P(C=t_i \cap C<T) \\
&= \sum_{i=1}^{n} I_i log \int_{t_i}^{\infty} f_T(t_i,\theta) f_C(c, \gamma+\delta\iota_\gamma^{-1/2} B(t_i,\theta)) dc \\
&\quad +(1-I_i) log \int_{t_i}^{\infty} f_T(t,\theta) f_C(ti, \gamma+\delta\iota_\gamma^{-1/2} B(t,\theta)) dt \\
&= \sum_{i=1}^{n} I_i log \int_{t_i}^{\infty} e^{\theta} e^{-e^{\theta} t_i} e^{\gamma+\delta(1-e^{\theta} t_i)} e^{-e^{\gamma+\delta(1-e^{\theta} t_i)} c} dc \\
&\quad +(1-I_i) log \int_{t_i}^{\infty} e^{\theta} e^{-e^{\theta} t} e^{\gamma+\delta(1-e^{\theta} t)} e^{-e^{\gamma+\delta(1-e^{\theta} t)} t_i} dt \\
&= \sum_{i=1}^{n} I_i [\theta - e^{\theta} t_i - e^{\gamma+\delta(1-e^{\theta} t_i)} t_i] \\
&\quad +(1-I_i)[\theta+\gamma+\delta+log \int_{t_i}^{\infty} e^{-(1+\delta)e^{\theta} t} e^{-e^{\gamma} t_i e^{\delta(1-e^{\theta} t)}} dt]. \quad (4.5)
\end{aligned}$$

$\hat{\theta}$, $\hat{\gamma}$, and $\hat{\delta}$ are the values at which the function $L(\theta,\gamma,\delta)$ is maximized. We obtained these estimates by using R programming. See the R code in the Appendix C. The last column of Table 4.1 presents the results of MLE.

Since we are dealing with an exponential family, under regularity conditions (Rao 1973, p364), the asymptotic covariance matrix is the inverse of the information matrix. SE denotes the standard error of the estimates. They are the square roots of the diagonal elements for the estimated covariance matrix, which equals the inverse observed information matrix. For example, if the information matrix is

$$\mathrm{IM} = - \begin{bmatrix} \frac{\partial^2 L}{\partial\theta^2} & \frac{\partial^2 L}{\partial\theta\partial\gamma} & \frac{\partial^2 L}{\partial\theta\partial\delta} \\ \frac{\partial^2 L}{\partial\gamma\partial\theta} & \frac{\partial^2 L}{\partial\gamma^2} & \frac{\partial^2 L}{\partial\gamma\partial\delta} \\ \frac{\partial^2 L}{\partial\delta\partial\theta} & \frac{\partial^2 L}{\partial\delta\partial\gamma} & \frac{\partial^2 L}{\partial\delta^2} \end{bmatrix},$$

then the estimated standard error of $\hat{\theta}$, $\hat{SE}_{\hat{\theta}}$, is $[(\mathrm{IM})^{-1}_{|\theta=\hat{\theta},\gamma=\hat{\gamma},\delta=\hat{\delta}}]^{\frac{1}{2}}_{11}$. Therefore,

by using MLE, the asymptotic confidence interval for θ is approximately

$$(\hat{\theta} - z_\alpha \hat{SE}_{\hat{\theta}}, \hat{\theta} + z_\alpha \hat{SE}_{\hat{\theta}}).$$

Figure 4.1 presents the confidence intervals with different sample sizes. The dashed lines represent Siannis et al.'s approximation, and the inner interval on the dashed lines is the range of $\hat{\theta}_{\delta=-0.3}$ and $\hat{\theta}_{\delta=+0.3}$. The solid lines represent the MLE. In each graph, the x axis represents the δ that we used to generate the data, and the y axis represents the estimates of θ. The horizontal line marks the true θ value that we used to generate the data. This figure reveals that, first, the confidence intervals are narrower as the sample size increases; second, the confidence interval obtained by using MLE is always narrower than that obtained by Siannis et al.'s method.

Note that although the $\hat{\theta}$ obtained by using MLE is an unbiased estimator, $\hat{\delta}$ is not accurate with a large mean square error.

4.2 Sensitivity Analysis and δ

Siannis et al. suggest changing δ from -0.3 to 0.3. We tested this suggestion using 5000 repeat simulation data to compare their approximation with the profile likelihood estimates of θ and γ given δ found from (4.5). Figures 4.2 and 4.3 present the results for sample size $n = 50$, and Figures 4.4 and 4.5 for $n = 100$. These figures show the exact (MLE, solid line) and the approximate (Siannis et al.'s method, dotdash line) estimates of θ for $|\delta| \leq 1$. The horizontal dotted line represents the standard estimate $\hat{\theta}_0$ and the two vertical dotted lines

mark the limits for δ that are suggested. These two lines are close within this range, but poor for large values of δ.

4.3 Accuracy and Validation

In Appendix C, we used function 'funSia' to estimate θ for Siannis' approximation. When $\delta = 0$, $\hat{\theta}$ is the same as what can be obtained from R's built-in function 'survreg', which assumes $\delta = 0$.

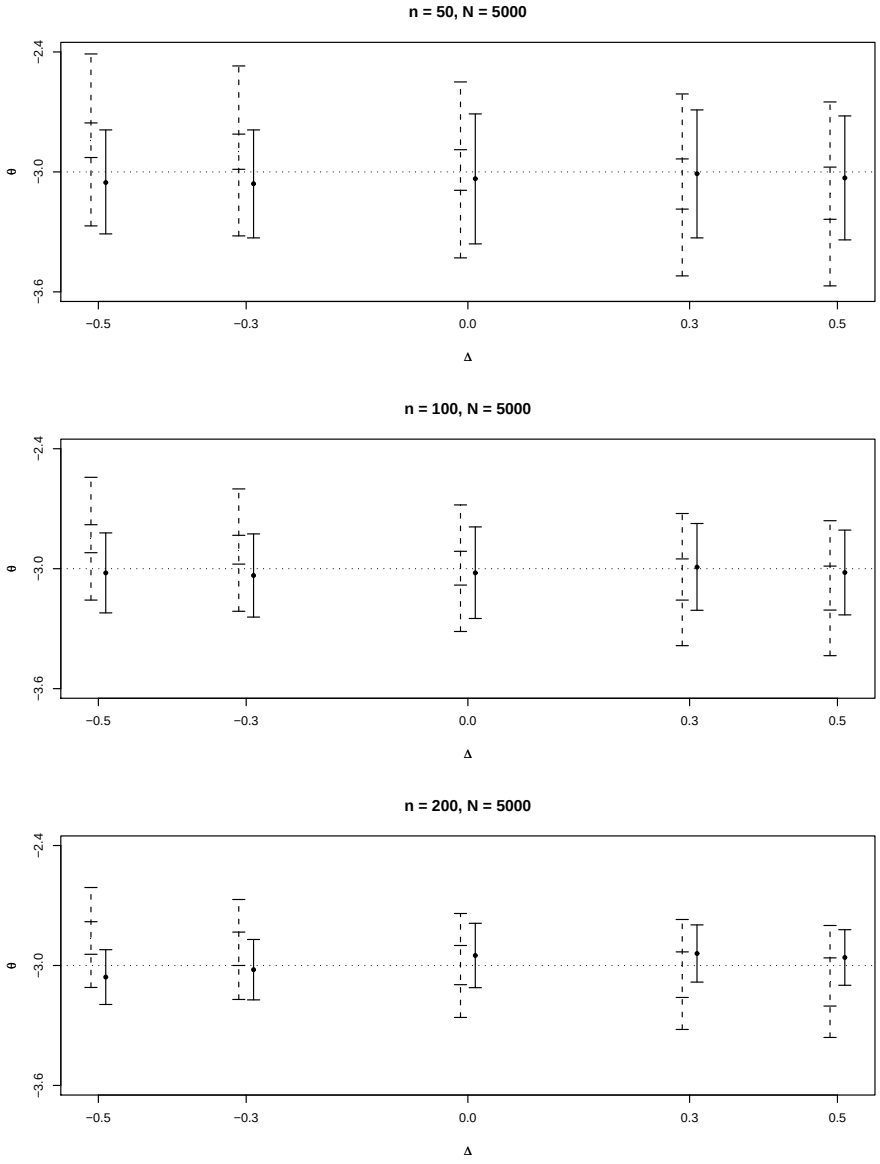

Figure 4.1: 95% confidence interval for θ. The dashed line represents the results obtained by using Siannis et al.'s method, and the solid line represents those obtained by using MLE. The inner interval on the dashed line is $(\hat{\theta}_{\delta=-.3}, \hat{\theta}_{\delta=+.3})$. The horizontal dotted line represents the true value of θ.

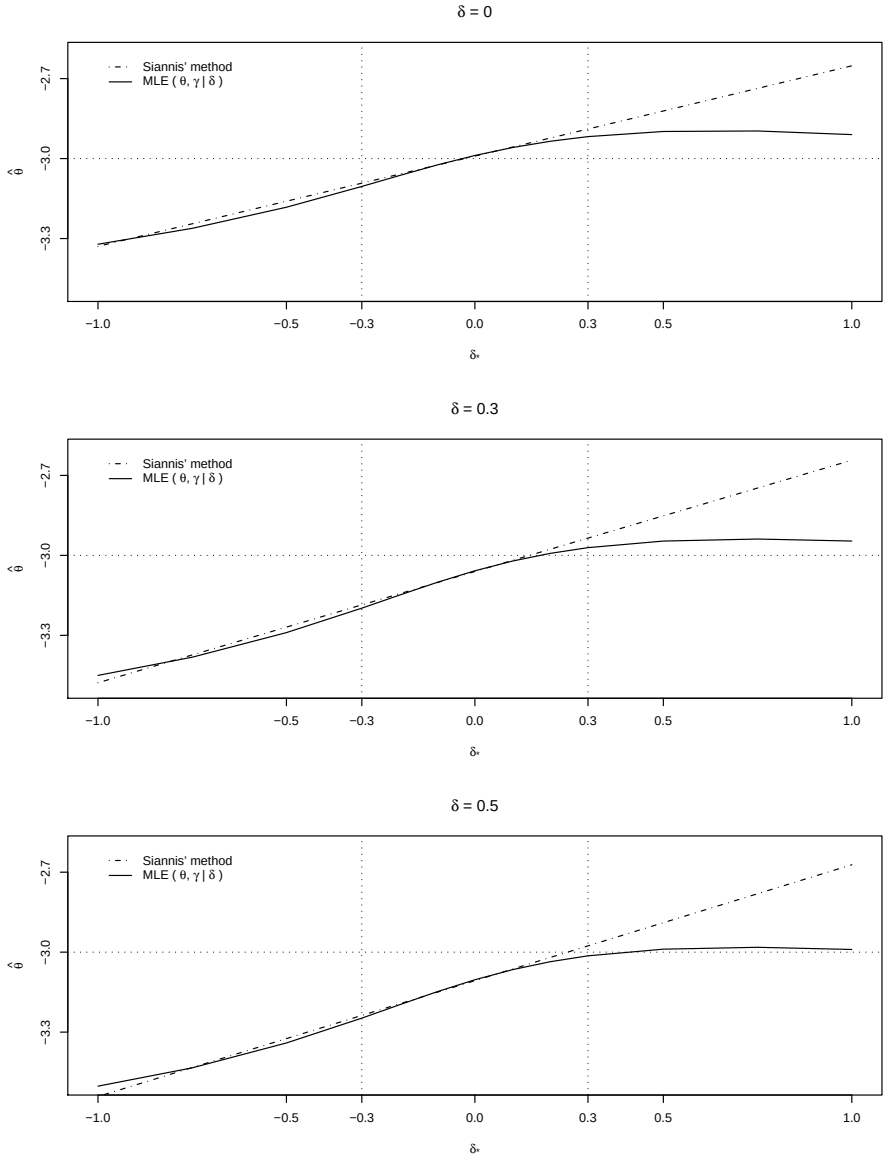

Figure 4.2: A sensitivity analysis of Siannis et al.'s sensitivity analysis versus the profile MLE for δ. The vertical dotted lines represent the cut-off points for $|\delta| = 0.3$. The horizontal dotted lines represent the true θ; x axis is the assumed δ. When n=50, $\delta \geq 0$, two methods nearly coincide when we assume $\delta \in [-0.3, 0.3]$.

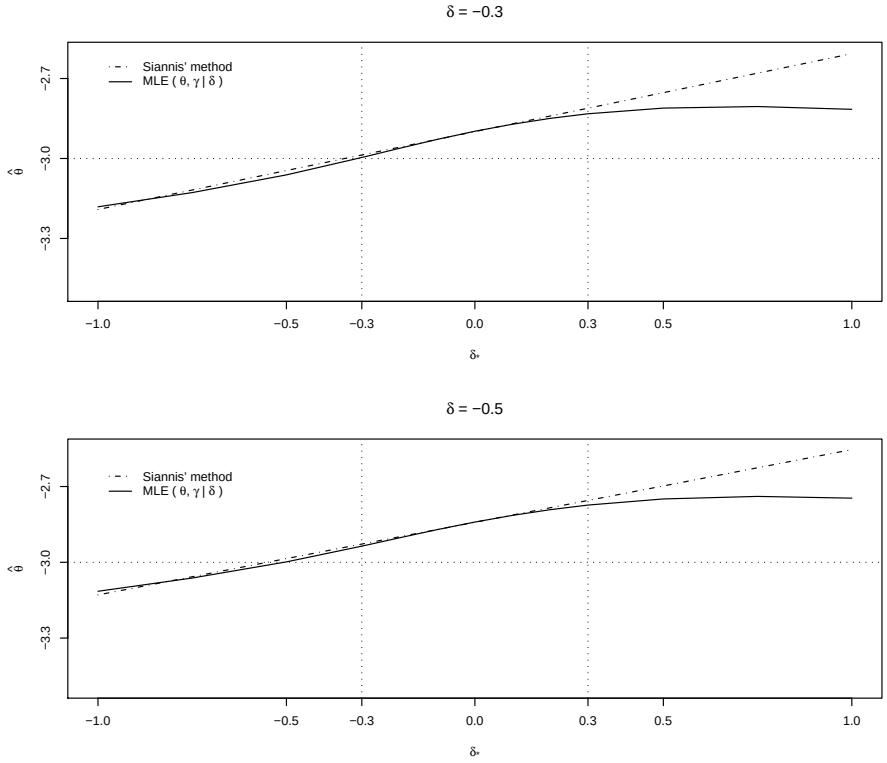

Figure 4.3: See note for Figure 4.2. When n=50, $\delta < 0$, two methods nearly coincide when we assume $\delta \in [-0.3, 0.3]$.

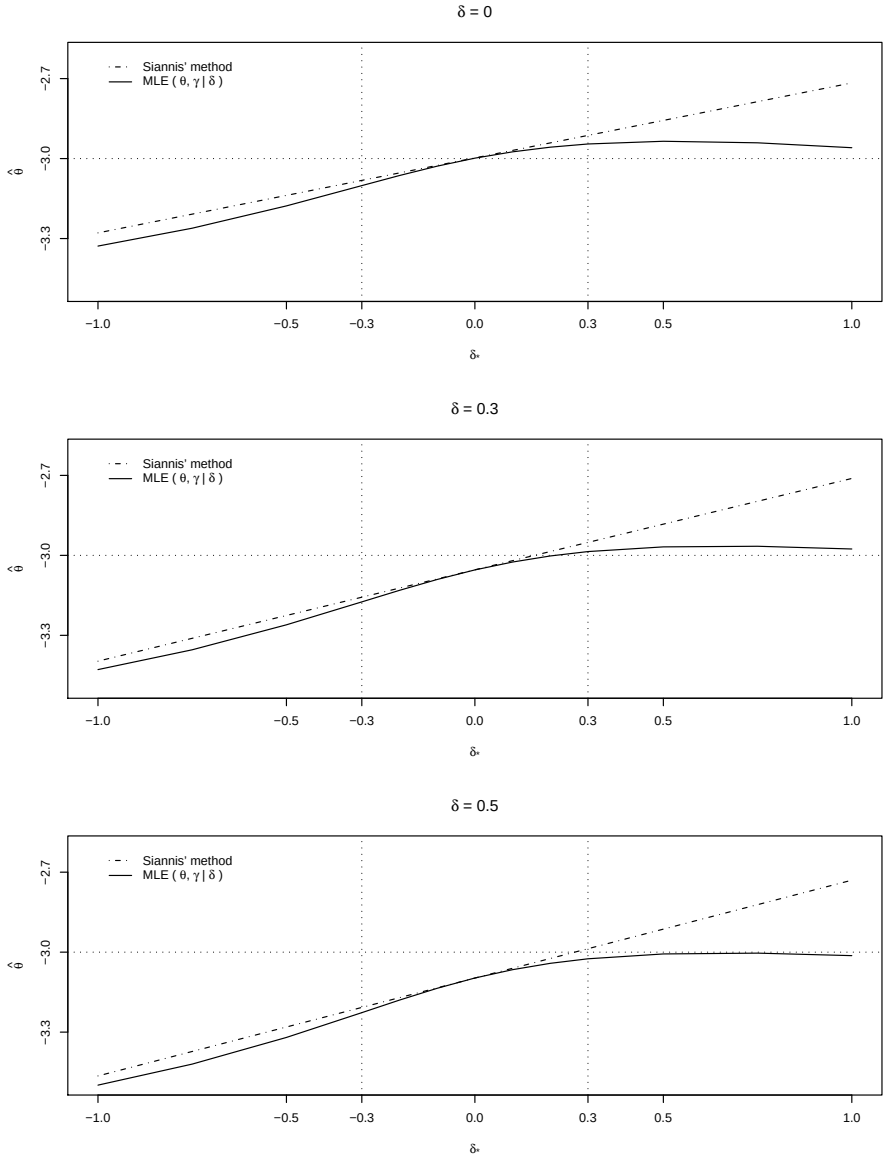

Figure 4.4: See note for Figure 4.2. When n=100, $\delta \geq 0$, two methods nearly coincide when we assume $\delta \in [-0.3, 0.3]$.

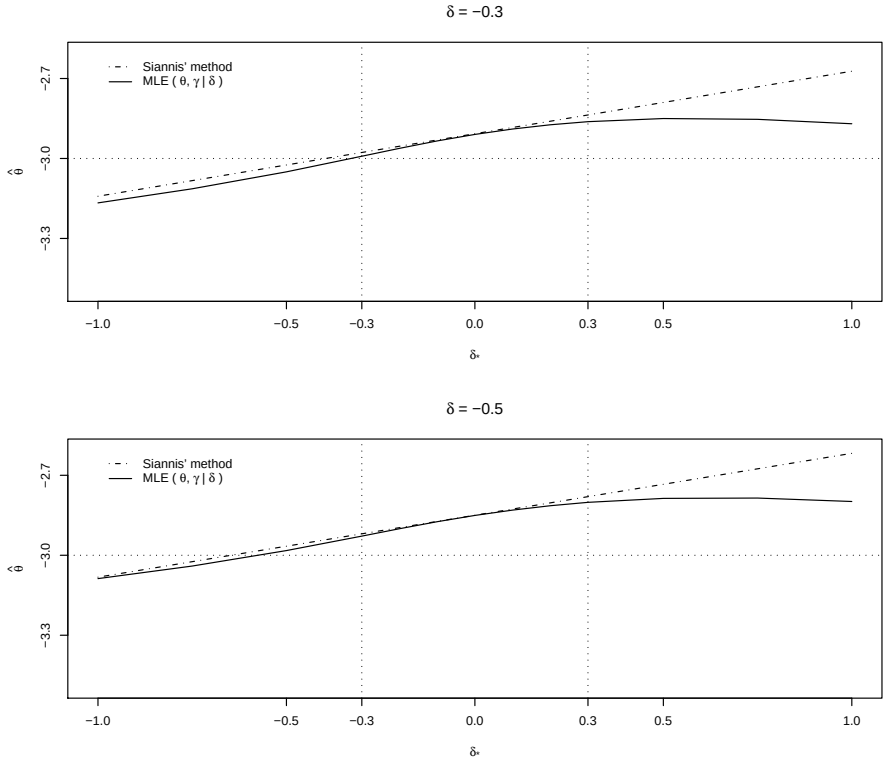

Figure 4.5: See note for Figure 4.2. When n=100, $\delta < 0$, two methods nearly coincide when we assume $\delta \in [-0.3, 0.3]$.

Table 4.1: $\bar{\hat{\theta}}$ from Siannis et al.'s approximation and MLE.

n	δ	Censored (%)	$\bar{\hat{\theta}}_{\delta=-.3}, \bar{\hat{\theta}}_{\delta=.3}$(SE)	$\bar{\hat{\theta}}_{\text{MLE}}$(SE)
50	-0.5	34.9	-2.93, -2.75(.176)	-3.05(.133)
	-0.3	34.0	-2.99, -2.81(.175)	-3.06(.138)
	0.0	32.2	-3.09, -2.89(.172)	-3.03(.164)
	0.3	29.9	-3.19, -2.94(.169)	-3.01(.165)
	0.5	29.7	-3.24, -2.98(.169)	-3.03(.159)
100	-0.5	31.8	-2.92, -2.78(.121)	-3.02(.102)
	-0.3	20.8	-2.98, -2.83(.120)	-3.03(.106)
	0.0	28.5	-3.08, -2.91(.118)	-3.02(.117)
	0.3	25.9	-3.16, -2.95(.116)	-2.99(.111)
	0.5	25.6	-3.21, -2.99(.116)	-3.02(.108)
200	-0.5	31.1	-2.94, -2.78(.085)	-3.06(.070)
	-0.3	30.0	-3.00, -2.83(.085)	-3.02(.077)
	0.0	27.7	-3.10, -2.90(.083)	-2.95(.082)
	0.3	25.3	-3.16, -2.93(.082)	-2.94(.073)
	0.5	25.0	-3.20, -2.96(.082)	-2.96(.071)

Chapter 5

Data Analysis

We will use two real-life data sets to compare Siannis et al.'s method and MLE.

5.1 Krall et al.'s Data

We first re-analyze the survival data originally discussed by Krall et al. (1975). In this study, 65 multiple myeloma patients, who were first diagnosed and then treated with alkylating agents at the West Virginia University Medical Center, were followed up to determine their survival time from diagnosis. Of these 65 patients, 48 were observed to have died during the time of the study, while 17 were lost to followup. The names of the variables are given in Table 5.1, and the entire original data set is provided in Appendix D.

To do a preliminary examination of this small data set, we first look at the Kaplan-Meier estimator presented in Table 5.2. From the output, we know that

the estimated median survival time is 18 months. However, the nonparametric Kaplan-Meier estimator of the marginal survival function is inconsistent unless censoring is noninformative (Scharfstein and Robins, 2002). We want to do a sensitivity analysis using parametric survival models to determine how robust the estimator is to the informative censoring. We use cumulative hazard plots, based on $\hat{H(t)}$, to assess the suitability of a particular parametric family and to explore T and C's survivor function by using the Kaplan-Meier estimator.

Figure 5.1 suggests that simple exponential survival models give a reasonable fit to the marginal distribution of T and C, as was also mentioned by Siannis et al. (2005). Therefore, we assume the marginal models for T and C are

$$f_T(t,\theta) = e^{\theta} e^{-e^{\theta} t}; \qquad f_C(c,\gamma) = e^{\gamma} e^{-e^{\gamma} c},$$

respectively.

Under the assumption of ignorable censoring, the standard maximum likelihood analysis gives an estimate of θ_0 and γ_0 as -3.48 and -4.52, respectively. The estimated median survival times for T and C are 22.6 months and 63.7 months, respectively. We can obtain these estimators by using standard software such as R.

If a small dependence between T and C is suspected, Siannis' method provides a sensitivity analysis of the estimate of the parameter of interest. We chose the bias function $B(t,\theta)$ as $1 - e^{\theta} t$, and the parameter δ changes between -0.3 to 0.3, $\hat{\theta}_\delta$ can be calculated by

$$\hat{\theta}_\delta \simeq \hat{\theta}_0 + \delta \frac{\sum_{i=1}^{65} \{e^{\hat{\gamma}_0} t_i^2 - (1 - I_i) t_i\}}{\sum_{i=1}^{65} t_i}.$$

The bias function was chosen to be a decreasing function of t to reflect that the nonignorable dependent information is more likely to be obtained for small values of t than for larger values. When the dependence parameter δ ranges between -0.3 to 0.3, it is equal to the ordinary correlation between T and C (Siannis et al., 2005) under proportional hazard assumption. Note that when $\delta = 0$, T and C are independent, and the censoring is ignorable. Therefore, if we assume a small dependence relation exists between T and C, we can see how sensitively $\hat{\theta}_\delta$ changes as we start to move δ away from zero. $\hat{\theta}_\delta$ is between 20.84 to 24.40 months, suggesting that even a small degree of dependence between the failure and censoring processes can have a noticeable effect on the analysis. On the other hand, tight bounds imply that inferences are robust and hence support the assumption of ignorable censoring.

Another way to estimate (θ, γ, δ) is by using MLE.

Table 5.3 presents the output obtained by using the Standard method, Siannis et al.'s approximation, and MLE. Because from MLE, we know $\hat{\delta} = -0.025$, this result indicates that the dependent relationship between T and C is not strong. Thus, not much difference exists between the approximation and MLE in terms of the confidence interval. This output provides researchers a way to compare and obtain information about how much they should be concerned about potential information censoring.

5.2 Freireich's Data

We re-analyze the survival data originally discussed by Freireich et al. (1963). In this study, 42 patients were randomly assigned to either of two study groups. The trial was designed to test the ability of the drug 6-MP to prolong the duration of a remission in treating acute leukemia. Therefore, we can consider the remission time as the survival time T. Table 5.4 presents the data.

The original analysis assumed that T and C were independent and stated that: "Overall survival was not significantly different for the two treatment programs" (p 713). However, the assumption of noninformative censoring is dubious, since of the 21 patients in the treatment group, 9 failed during the study period, and 12 were censored. In contrast, none of the data in the placebo group were censored; that is, all 21 patients in the placebo group went out of remission during the study period. If the censoring happened because of an inadequate treatment response, then withdrawal at time c may indicate that remission (T) was likely sooner than might have been expected otherwise, and that a positive dependent relation existed. By ignoring this dependent relation, the researchers tended to overestimate the treatment effect. On the other hand, if the censoring happened because the patients were feeling better and had no need to continue treatment, then withdrawal at time c may indicate that remission was likely longer than might have been expected otherwise, and that a negative dependent relation existed. By ignoring this dependent relation, the researchers tended to underestimate the treatment effect.

To do a preliminary examination of this small data set, we first look at

the cumulative hazard plots, which are based on $\hat{H}(t)$ and are frequently used to assess the suitability of a particular parametric family. We will explore T and C's survivor function by using the Kaplan-Meier estimator. Assume that, for a particular parametric family, the transformation $g_2(H(t))$ is linear in the transformed time scale $g_1(t)$; then a plot of $g_2(\hat{H}(t_i))$ versus $g_1(t_i)$, $i = 1, \cdots, n$ should be roughly linear if the proposed parametric family is an appropriate model for the data. For instance, the cumulative hazard of exponential distribution with mean $\frac{1}{\rho}$ is $H(t) = \rho t$. Therefore, if a plot of $\hat{H(t_i)}$ versus t_i appears to be roughly linear, this finding suggests that the exponential survival model gives a reasonable fit to the marginal distribution of T.

Figure 5.2 presents the plots of the cumulative hazard versus the failure time or censored time t. These plots suggest that simple exponential survival models give a reasonable fit to the marginal distribution of T and C. Therefore, we assume the marginal models for T in the treatment group, C in the treatment group, and P in the control group are

$$f_T(t, \theta) = e^{\theta} e^{-e^{\theta} t}; \quad f_C(c, \gamma) = e^{\gamma} e^{-e^{\gamma} c}; \quad f_P(p, \omega) = e^{\omega} e^{-e^{\omega} p}, \tag{5.1}$$

respectively.

Under the assumption of ignorable censoring, the standard maximum likelihood analysis gives estimates of θ_0, γ_0, and ω as -3.686, -3.398, and -2.159, respectively. The estimated median survival times for T, C, and P are 27.6,

20.7, and 6 weeks, respectively. The calculation is as below, from (2.9),

$$\because \quad L_0(\theta,\gamma) = \sum_{i=1}^{n}\{I_i log h_T(t_i,\theta) + (1-I_i) log h_C(t_i,\gamma) - H_T(t_i,\theta) - H_C(t_i,\gamma)\}$$

$$\therefore \quad \frac{\partial L_0(\theta,\gamma)}{\partial\theta} \overset{set}{=} 0 \Rightarrow \hat{\theta}_0 = log(\frac{\sum_{i=1}^{n} I_i}{\sum_{i=1}^{n} t_i}) = -3.686;$$

for the same reason,

$$\frac{\partial L_0(\theta,\gamma)}{\partial\gamma} \overset{set}{=} 0 \Rightarrow \hat{\gamma}_0 = log(\frac{\sum_{i=1}^{n}(1-I_i)}{\sum_{i=1}^{n} t_i}) = -3.398.$$

To obtain the median failure time,

$$S(t,\theta) = e^{-e^{\hat{\theta}_0} t} = 0.5 \quad \Rightarrow \quad t = 27.6;$$

$$S(c,\gamma) = e^{-e^{\hat{\gamma}_0} c} = 0.5 \quad \Rightarrow \quad c = 20.7;$$

$$S(p,\omega) = e^{-e^{\hat{\omega}_0} p} = 0.5 \quad \Rightarrow \quad p = 6.$$

We can also obtain these estimators by using standard software such as R.

If a small dependence between T and C is suspected, Siannis et al.'s method can be used to provide a sensitivity analysis of the estimate of the parameter of interest. We choose the bias function $B(t,\theta)$ as $1 - e^{\theta}t$, and the parameter δ changes between -0.3 to 0.3. By using (4.3), $\hat{\theta}_\delta$ is

$$\hat{\theta}_{-0.3} = -3.704, \qquad \hat{\theta}_{+0.3} = -3.668.$$

The 95% confidence interval is, by using (4.4),

$$\text{CI}_S = (-3.704 - 1.96 \times 0.33, -3.668 + 1.96 \times 0.33) = (-4.351, -3.021).$$

Based on Siannis et al.'s assumption that the conditional distribution of C given T has exactly the same parametric form as its marginal distribution

$f_C(c, \gamma)$, with the parameter allowed to depend on T, we use MLE to estimate (θ, γ, δ) all together, $\hat{\theta}, \hat{\delta}$ are -3.7124 and -0.2834, respectively. The corresponding standard errors are 0.2912 and 0.3579, respectively. Thus, the 95% confidence interval for θ is

$$\text{CI}_M = (-3.712 - 1.96 \times 0.2912, -3.712 + 1.96 \times 0.2912) = (-4.283, -3.142).$$

Therefore, in terms of 95% CI, by using Siannis et al.'s method, the median survival time for the treatment group is found to be $(14, 54)$ weeks; by using MLE, it is found to be $(16, 50)$ weeks. MLE provides narrower bounds. Table 5.5 presents the output.

To compare the survival experience of these two groups, we need to compare θ and $\omega = \theta + \beta$. If $\beta = 0$, then no difference in survival experience exists between the two groups.

The log-likelihood function for the observed data from the two study groups is given by

$$L(\theta, \gamma, \delta, \beta) = L_1(\theta, \gamma, \delta) + L_2(\theta, \beta),$$

where L_1 is the log-likelihood function from the treatment group with informative censoring, and L_2 is the log-likelihood function from the control group. In our case, L_1 equals (4.5), and L_2 equals (2.1). Therefore, the log-likelihood

function is

$$\begin{aligned} L(\theta,\gamma,\delta,\beta) &= L_1(\theta,\gamma,\delta) + L_2(\theta,\beta) \\ &= \sum_{i=1}^{n1} I_{i1}[\theta - e^{\theta}t_{i1} - e^{\gamma+\delta(1-e^{\theta}t_{i1})}t_{i1}] \\ &\quad +(1-I_{i1})[\theta+\gamma+\delta+log\int_{t_{i1}}^{\infty} e^{-(1+\delta)e^{\theta}t}e^{-e^{\gamma}t_{i1}e^{\delta(1-e^{\theta}t)}}dt] \\ &\quad +\sum_{i=1}^{n2}[I_{i2}(\theta+\beta) - e^{\theta+\beta}t_{i2}]. \end{aligned} \tag{5.2}$$

Since $\hat{\beta} = \hat{\omega} - \hat{\theta} = -2.1595 - (-3.7124) = 1.5529$, and the asymptotic $SD(\hat{\beta}) = 0.3639$ (see Appendix E for the R code used to produce these results), to test $\beta = 0$, the appropriate test statistic would be

$$\frac{\hat{\beta}-\beta}{SD(\hat{\beta})} = \frac{1.5529-0}{0.3639} = 4.27,$$

and the corresponding p-value is close to 0. Thus, we have strong evidence for rejecting the null hypothesis and concluding that a significantly reduced hazard occurs for the group of acute leukemia patients treated with the drug 6-MP.

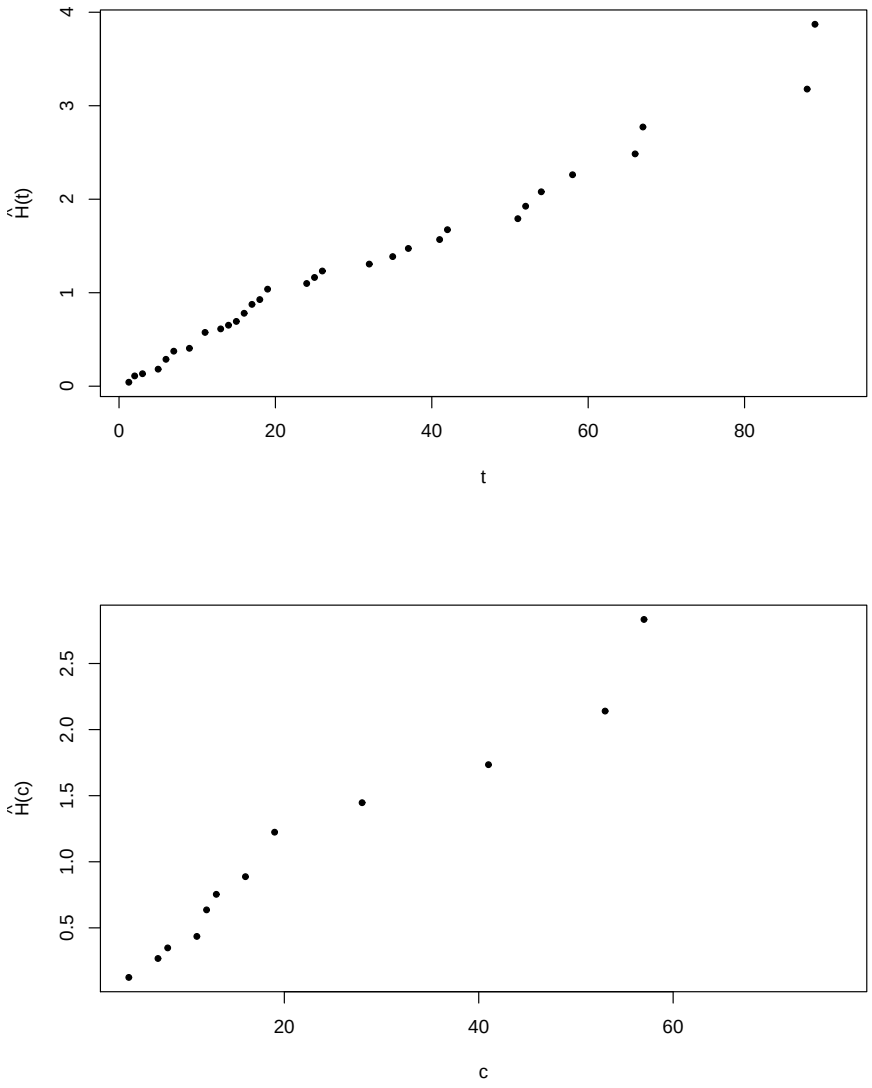

Figure 5.1: Plots of the cumulative hazard function ($\hat{H}(t)$) versus time (t) and the cumulative hazard function ($\hat{H}(c)$) versus time (c) for Krall et al.'s data.

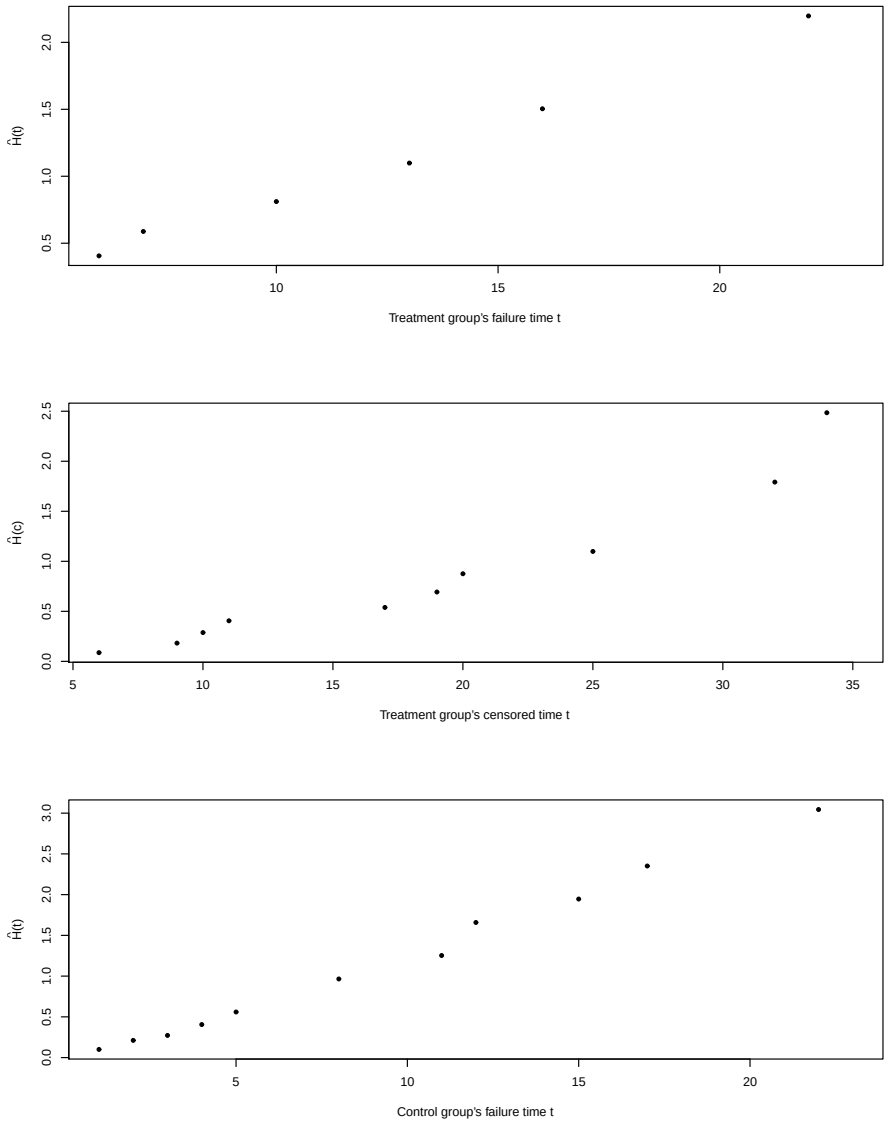

Figure 5.2: Plots of the cumulative hazard function ($\hat{H}(t)$) versus failure time or censored time t for Freireich et al.'s data.

Table 5.1: Description of variables for Krall et al.'s data.

Variable Name	Description of Variable
id	Patient's identification.
dur	Survival time from diagnosis to nearest month.
status	Censored status. 1 = event observed, 0 = lost to followup.

Table 5.2: Kaplan-Meier output for Krall et al.'s data

time	n.risk	n.event	survival	std.err	lower 95% CI	upper 95% CI
1.25	65	2	0.9692	0.0214	0.92814	1.000
2.00	63	3	0.9231	0.0331	0.86052	0.990
3.00	60	1	0.9077	0.0359	0.83998	0.981
5.00	57	2	0.8758	0.0411	0.79888	0.960
6.00	55	4	0.8121	0.0489	0.72171	0.914
7.00	51	3	0.7644	0.0533	0.66681	0.876
9.00	45	1	0.7474	0.0547	0.64749	0.863
11.00	44	5	0.6625	0.0603	0.55429	0.792
13.00	36	1	0.6441	0.0613	0.53441	0.776
14.00	34	1	0.6251	0.0624	0.51406	0.760
15.00	33	1	0.6062	0.0633	0.49398	0.744
16.00	32	2	0.5683	0.0648	0.45453	0.711
17.00	29	2	0.5291	0.0660	0.41439	0.676
18.00	27	1	0.5095	0.0664	0.39470	0.658
19.00	26	2	0.4703	0.0668	0.35603	0.621
24.00	22	1	0.4489	0.0671	0.33493	0.602
25.00	21	1	0.4275	0.0672	0.31417	0.582
26.00	20	1	0.4062	0.0672	0.29373	0.562
32.00	18	1	0.3836	0.0671	0.27224	0.541
35.00	17	1	0.3610	0.0669	0.25115	0.519
37.00	16	1	0.3385	0.0664	0.23046	0.497
41.00	15	1	0.3159	0.0657	0.21018	0.475
42.00	13	1	0.2916	0.0650	0.18844	0.451
51.00	12	1	0.2673	0.0639	0.16727	0.427
52.00	11	1	0.2430	0.0626	0.14671	0.403
54.00	9	1	0.2160	0.0612	0.12400	0.376
58.00	7	1	0.1851	0.0597	0.09841	0.348
66.00	6	1	0.1543	0.0572	0.07463	0.319
67.00	5	1	0.1234	0.0534	0.05285	0.288
88.00	3	1	0.0823	0.0490	0.02564	0.264
89.00	2	1	0.0411	0.0380	0.00673	0.252
92.00	1	1	0.0000	NA	NA	NA

Table 5.3: Comparison of Siannis et al.'s method to MLE for Krall et al.'s data.

		$\hat{\theta}$(SE)	$\text{CI}_{\hat{\theta}}$	$\text{CI}_{\text{Median}}$(Months)
Standard Method		-3.48(.144)	(-3.76, -3.20)	(17.0, 29.8)
Siannis' Method	$\delta = -.3$	-3.56(.144)	(-3.84, -3.12)	(15.7, 32.3)
	$\delta = +.3$	-3.40(.144)		
MLE		-3.49(.170)	(-3.82, -3.16)	(16.3, 31.6)

Table 5.4: Freireich et al.'s data.

Group	Remission times, in weeks						
6-MP	6+	6	6	6	7	9+	10+
	10	11+	13	16	17+	19+	20+
	22	23	25+	32+	32+	34+	35+
Placebo	1	1	2	2	3	4	4
	5	5	8	8	8	8	11
	11	12	12	15	17	22	23

Table 5.5: Comparison of Siannis et al.'s method to MLE for Freireich et al.'s data.

		$\hat{\theta}$(SE)	$\text{CI}_{\hat{\theta}}$	$\text{CI}_{\text{Median}}$(Weeks)
Standard Method		-3.69(.33)	(-4.34, -3.04)	(14.5, 53.2)
Siannis' Method	$\delta = -.3$	-3.70(.33)	(-4.35, -3.02)	(14.2, 53.7)
	$\delta = +.3$	-3.67(.33)		
MLE		-3.49(.29)	(-4.28, -3.14)	(16.0, 50.1)

Chapter 6

Concluding Remarks

In this thesis, we studied Siannis et al.'s approximation of sensitivity analysis and compare this method to the MLE method. Both methods are based on Siannis et al.'s assumption that the conditional distribution of C given T has exactly the same parametric form as its marginal distribution $f_C(,\gamma)$, but with the parameter allowed to depend on T.

It was found that Siannis et al.'s error of approximation was small only when δ, which was used to generate the simulation data, was between -0.3 to 0.5, and when the assumed δ was between -0.3 to 0.3 (Figures 4.5 - 4.8). With real-life data, we never know what the δ is, so deciding whether the approximation is trustworthy is difficult. But Siannis et al.'s approximation has easy calculations. On the other hand, MLE provides a narrower confidence interval than that of Siannis et al.'s approximation. Although $\hat{\delta}$, which measures the size of the dependent relation, was not accurate with a large standard error, $\hat{\theta}$,

the parameter in the distribution of the survival time $f_T(t, \theta)$ was unbiased for all δ. In practice, the purpose of most clinical trials is to estimate the marginal distribution $f_T(t, \theta)$, i.e., to consider data in which the risks of T and C act together, and to try to infer how T would act in isolation. Therefore, it is recommended that θ be estimated by maximizing the likelihood function under Siannis et al.'s assumption. Finally, to assist other researchers in computing the MLEs, a program was written in R.

In Chapters 3 and 4, we conducted graphical exploratory analysis and simulation studies for exponential distribution only. However, note that the approximation we considered is valid for any parametric models where the assumption that the conditional distribution of C given T has exactly the same parametric form as its marginal distribution, satisfying (2.4), where θ is the location parameter of the marginal density $f_T(t, \theta)$. A nonparametric approach to informative censoring problems has been discussed in Scharfstein and Robins (2002), although further research is needed.

For future work, we can, first, try using different bias functions $B(t, \theta)$, which measure the pattern of the dependent relationship. Second, as MLE is based on the assumption proposed by Siannis, we want to determine how we can check if a data set satisfied this assumption. Third, we can consider the case when θ is a vector of the parameters or a function of the covariates. Fourth, we can consider the ignorability of censoring at the end of the study. Fifth, the use of a nonparametric case is another approach that could be applied to deal with the censoring problem.

Bibliography

Allison, P. D. 1995. *Survival Analysis Using SAS: A Practical Guide*. SAS Press, Cary.

Crowder, M. J. 2001. *Classical Competing Risks*. Chapman Hall, Boca Raton.

Freireich, E. J., Gehan, E., Frei III, E., Schroeder, L. R., Wolman, I. J., Anbari, R., and Bergert, E. O. 1963. The effect of 6-mercaptopurine on the duration of steroid-induced remissions in acute leukemia: a model for evaluation of other potentially useful therapy. *Blood*, 21:699–717.

Kalbfleisch, J. D. and Prentice, R. L. 2002. *The Statistical Analysis of Failure Time Data*. Wiley, Hoboken, second edition.

Krall, J. M., Uthoff, V. A., and Harley, J. B. 1975. A step-up procedure for electiing variables associated with survival. *Biometrics*, 31:49–57.

Lawless, J. F. 2003. *Statistical Models and Methods for Lifetime Data*. Wiley, Hoboken, second edition.

Lee, E. T. and Wang, J. W. 2003. *Statistical Methods for Survival Data Analysis*. Wiley, Hoboken, third edition.

Peterson, A. V. 1976. Bounds for a joint distribution function with fixed sub-distributions functions: applications to competing risks. *proceedings of the National Academy of Sciences, USA*, 73:11–13.

Ruan, P. K. and Gray, R. J. 2008. Sensitivity analysis of progression-free survival with dependence withdrawal. *Statist. Med.*, 27:1180–1198.

Scharfstein, D. and Robins, J. M. 2002. Estimation of the failure time distribution in the presence of informative censoring. *Biometrika*, 89:617–634.

Scharfstein, D., Robins, J. M., Eddings, W., and Rotnitzky, A. 2001. Inference in randomized studies with informative censoring and discrete time-to-event endpoints. *Biometrics*, 57:404–413.

Siannis, F., Copas, J., and Lu, G. 2005. Sensitivity analysis for informative censoring in parameteric survival models. *Biostatistics*, 6:77–91.

Tsiatis, A. 1972. A nonidentifiability aspect of the problem of competing risks. *Proceedings of the National Academy of Sciences, USA*, 72:20–22.

Zhang, J. and Heitjan, D. F. 2005. Nonignorable censoring in randomized clinical trials. *Clinical Trials*, 2:488–196.

Appendix A

R code for Figures 3.1-3.10

This R code is to generate Figure 3.1 and 3.6. For Figure 3.2-3.5, 3.7-3.10, just change delta to the according value (0.3, 0.5, -0.3, -0.5).

```
library(lattice)
delta          <- 0
theta          <- -3
gamma          <- -4

fun2D             <- function(TT){
      xE <- topo.plt$cc[which(topo.plt$tt==TT&topo.plt$gr=="Exact")]
      yE <- topo.plt$y[which(topo.plt$tt==TT&topo.plt$gr=="Exact")]
      xA <- topo.plt$cc[which(topo.plt$tt==TT&topo.plt$gr!="Exact")]
      yA <- topo.plt$y[which(topo.plt$tt==TT&topo.plt$gr!="Exact")]
      plot(xE, yE, type="l", xlab="C (month)", ylab="f(c|t)",
      main=paste("T = ", TT, " (month)", sep=''), ylim=c(0, .08))
      lines(xA, yA, lty=2)
}

topo.mar     <- list(tt=seq(0, 85, 2), cc=seq(0, 85, 2),
                gr=c("Exact", "Approximation"))
topo.plt     <- expand.grid(topo.mar)
gr1Length    <- dim(topo.plt)[1]/2
tt1          <- topo.plt$tt[1:gr1Length]
cc1          <- topo.plt$cc[1:gr1Length]
tt2          <- topo.plt$tt[(gr1Length+1):(2*gr1Length)]
cc2          <- topo.plt$cc[(gr1Length+1):(2*gr1Length)]

topo.plt$y   <- c(exp(gamma+delta*(1-exp(theta)*tt1))*
                 exp(-exp(gamma+delta*(1-exp(theta)*tt1))*cc1),
```

```
                exp(gamma)*exp(-exp(gamma)*cc2)*(1+delta*
                (1-exp(gamma)*cc2)*(1-exp(theta)*tt2)))
fun         <- function(zz=40, xx=-90, yy=0)
                {wireframe(y ~ tt * cc|gr, data=topo.plt,
                main=expression(paste("f (C|T, ",
                delta, " = 0)", sep='')),
                scales = list(arrows = F, tick.number=4, tck=1.5),
                aspect = c(1, 1), drape = T, screen =
                list(z=zz, x=xx, y=yy),
                xlab="T (month)", ylab="C (month)", zlab="",
                colorkey=F, zlim=c(0, .03))}
postscript("thesis/Latex/output/3d0-3-4.eps")
fun()
dev.off()
postscript("thesis/Latex/output/2d0-3-4.eps", horizontal=F)
par(mfrow=c(3,2))
fun2D(0)
fun2D(20)
fun2D(40)
fun2D(50)
fun2D(60)
fun2D(80)
dev.off()
```

Appendix B

R code to generate Figures 3.11-3.13

This is the R code to generate Figure 3.11, to generate Figure 3.12 and 3.13,

just change n to 100 or 500.

```
n            <- 50
N            <- 5000

makeDat <- function(trueDelta=0, n=100, N=10, theta=-3, gamma0=-4){
       dat         <- data.frame(matrix(0, nrow=n*N, ncol = 6,
       dimnames=list(1:(n*N), c("id", "N", "Ii", "ti", "Ti", "Ci"))))
       for(j in 1:N){
          idN                  <- rep(j, n)
          Ii                   <- rep(1, n)
          set.seed(j)
          Ti                   <- rexp(n, exp(theta))
          lambda1              <- gamma0 + trueDelta*(1-(exp(theta))
                                   *Ti)
          Ci                   <- rep(0, n)
          eLambda1             <- exp(lambda1)
          for(i in 1:n) {
                       set.seed(N+i)
                       Ci[i]      <- rexp(1, eLambda1[i])
          }

          Ii[which(Ti>Ci)]   <- 0
          ti                    <- Ti
          ti[which(Ii==0)]   <- Ci[which(Ii==0)]
          dat[c((n*j-n+1):(n*j)), 2:6 ] <- data.frame(
```

```
                idN, Ii, ti, Ti, Ci)
        }
        dat
}

getCor       <- function(dat, N, n){
        corRet  <- rep(0, N)
        for (j in 1:N){
                Ti        <- dat[c((n*j-n+1):(n*j)), "Ti"]
                Ci        <- dat[c((n*j-n+1):(n*j)), "Ci"]
                corRet[j]  <- cor(Ti, Ci, method="spearman")
        }
        corRet
}

dat0          <- makeDat(trueDelta=0, n, N)
dat0$ti[which(dat0$ti>80)] = 80
dat.3         <- makeDat(trueDelta=0.3, n, N)
dat.3$ti[which(dat.3$ti>120)] = 120
dat.5         <- makeDat(trueDelta=0.5, n, N)
dat.5$ti[which(dat.5$ti>120)] = 120
datN.3        <- makeDat(trueDelta=-0.3, n, N)
datN.3$ti[which(datN.3$ti>70)] =70
datN.5        <- makeDat(trueDelta=-0.5, n, N)
datN.5$ti[which(datN.5$ti>60)] = 60

cor0          <- getCor(dat0, N, n)
cor.3         <- getCor(dat.3, N, n)
cor.5         <- getCor(dat.5, N, n)
corN.3        <- getCor(datN.3, N, n)
corN.5        <- getCor(datN.5, N, n)

(cor0_mean     <- mean(cor0))
(cor.3_mean    <- mean(cor.3))
(cor.5_mean    <- mean(cor.5))
(corN.3_mean   <- mean(corN.3))
(corN.5_mean   <- mean(corN.5))

which(cor0>(cor0_mean-.002) & cor0<(cor0_mean+.002))
which(cor.3>(cor.3_mean-.002) & cor.3<(cor.3_mean+.002))
which(cor.5>(cor.5_mean-.002) & cor.5<(cor.5_mean+.002))
which(corN.3>(corN.3_mean-.002) & corN.3<(corN.3_mean+.002))
which(corN.5>(corN.5_mean-.002) & corN.5<(corN.5_mean+.002))

postscript("thesis/Latex/output/corPlot_-3_-4_50_5000.eps",
            horizontal=F)
```

```
par(mfrow=c(3,2))
k <- 356
plot(dat0$Ti[(n*k-n+1):(n*k)], dat0$Ci[(n*k-n+1):(n*k)],
     xlab="T (month)", ylab="C (month)", main=expression
     (paste(theta, " = -3, ", delta, " = 0, ",
     bar(rho)[50-5000], " = 0.002", sep="")),
     xlim=c(0, 300), ylim=c(0, 300), pch=20)
legend(200, 300, expression(paste(rho[356], " = 0.001",
        sep="")), bty='n')
plot(dat0$Ti[(n*k-n+1):(n*k)], dat0$Ci[(n*k-n+1):(n*k)],
     xlab="", ylab="", type="n", axes=F)

k <- 244
plot(dat.3$Ti[(n*k-n+1):(n*k)], dat.3$Ci[(n*k-n+1):(n*k)],
     xlab="T (month)", ylab="C (month)", main=expression(
     paste(theta, " = -3, ", delta, " = 0.3, ",
     bar(rho)[50-5000], " = 0.202", sep="")),
     xlim=c(0, 300), ylim=c(0, 300), pch=20)
legend(200, 300, expression(paste(rho[244], " = 0.203",
        sep="")), bty='n')

k<- 42
plot(dat.5$Ti[(n*k-n+1):(n*k)], dat.5$Ci[(n*k-n+1):(n*k)],
     xlab="T (month)", ylab="C (month)", main=expression(
     paste(theta, " = -3, ", delta, " = 0.5, ",
     bar(rho)[50-5000], " = 0.313", sep="")),
     xlim=c(0, 300), ylim=c(0, 300), pch=20)
legend(200, 300, expression(paste(rho[42], " = 0.313",
        sep="")), bty='n')

k<- 298
plot(datN.3$Ti[(n*k-n+1):(n*k)], datN.3$Ci[(n*k-n+1):(n*k)],
     xlab="T (month)", ylab="C (month)", main=expression(
     paste(theta, " = -3, ", delta, " = -0.3, ",
     bar(rho)[50-5000], " = -0.199", sep="")),
     xlim=c(0, 300), ylim=c(0, 300), pch=20)
legend(200, 300, expression(paste(rho[298], " = -0.198",
    sep="")), bty='n')

k <- 375
plot(datN.5$Ti[(n*k-n+1):(n*k)], datN.5$Ci[(n*k-n+1):(n*k)],
     xlab="T (month)", ylab="C (month)", main=expression(
     paste(theta, " = -3, ", delta, " = -0.5, ",
     bar(rho)[50-5000], " = -0.309", sep="")),
     xlim=c(0, 300), ylim=c(0, 300), pch=20)
legend(200, 300, expression(paste(rho[375], " = -0.309",
```

```
    sep="")), bty='n')
dev.off()
```

Appendix C

R code to generate Table 4.1

This is the R code we generated 5000 repeated data with parameters $\theta = -3$, $\gamma = -4$, $\delta = 0$, and $n = 100$.

```
makeDat <- function(dat0, trueDelta=0, n=100, N=10, theta=-3,
             gamma0=-4){
       for(j in 1:N){
          idN                  <- rep(j, n)
          Ii                   <- rep(1, n)
          set.seed(j)
          Ti                   <- rexp(n, exp(theta))

          lambda1              <- gamma0 + trueDelta*(1-(exp
                                   (theta))*Ti)
          Ci                   <- rep(0, n)
          eLambda1             <- exp(lambda1)
          for(i in 1:n) {
                        set.seed(N+i)
                        Ci[i]      <- rexp(1, eLambda1[i])
          }

          Ii[which(Ti>Ci)]   <- 0
          ti                   <- Ti
          ti[which(Ii==0)]   <- Ci[which(Ii==0)]

          dat0[c((n*j-n+1):(n*j)), 2:6 ] <- data.frame(idN,
                Ii, ti, Ti, Ci)
       }
```

```
       dat0$ti[which(dat0$ti>80)]=80 #to avoid numerical problems
       dat0
}

theta        <- -3
trueDelta    <- 0
n            <- 100
N            <- 5000
dat0         <- data.frame(matrix(0, nrow=n*N, ncol = 6,
                    dimnames=list(1:(n*N),
                    c("id", "N", "Ii", "ti", "Ti", "Ci"))))
dat          <- makeDat(dat0, trueDelta, n, N)
dat$id       <- c(1:(n*N))
```

This is the R code that we obtained $\bar{\hat{\theta}}_\delta$ and $\bar{\hat{\theta}}_{\text{MLE}}$.

```
del        <- c(-1.0, -0.75, -0.5, -0.3, -0.2, -0.1,
                0.0,  0.1,  0.2,  0.3,  0.5,  0.75,  1.0)
parMLE     <- data.frame(matrix(0, nrow=3, ncol = N,
                dimnames=list(c("thetaHat", "gammaHat",
                "deltaHat"), c(1:N))))
thetaSia   <- data.frame(matrix(0, nrow=length(del),
                ncol = N, dimnames=list(del, c(1:N))))
outCome    <- data.frame(matrix(0, nrow=1, ncol = 14,
                dimnames=list("0", c("SiaN.3", "Sia0",
                "SiaP.3", "SE_Sia", "Sia_lower", "Sia_upper",
                "MLE", "SE_MLE", "MLE_lower", "MLE_upper",
                "delta", "SE_delta", "delta_lower", "delta_upper"))))

funSia <- function(delta=0) log(sum(Ii)/sum(ti)) + delta*sum(
            (exp(log(sum(1-Ii)/sum(ti))))*((ti)^2)-(1-Ii)
            *(ti))/sum(ti)
for (j in 1:N) {
      Ii    <- dat[c((n*j-n+1):(n*j)), "Ii"]
      ti    <- dat[c((n*j-n+1):(n*j)), "ti"]
      b     <- c(-3.1, -4.1, 0)
      fun   <- function(b){
            fun_A <- function(){
                  xx   <- function(tx) exp((-1-delta)*
                     exp(theta)*tx)*
                     exp(-exp(Gamma)*tti*
                     exp(delta*(1-exp(theta)*tx)))
                  tmp  <- rep(0, n)
                  for(i in 1:n) {
                        tti <- ti[i]
                        tmp[i]<-integrate(xx,
```

```
                                    lower=tti, upper=180,
                                    subdivisions=30)$value
                     }
                     tmp
            }
                    theta           = b[1]
                    Gamma           = b[2]
                    delta           = b[3]
                    intA            = fun_A()
                    L               = -sum(Ii*(theta-
                            exp(theta)*ti-exp(Gamma+
                            delta*(1-exp(theta)*ti))*ti)+(1-Ii)*
                            (theta+Gamma+delta+log(intA)))
      }
       parMLE[,j]   <- optim(b, fn=fun, method = "L-BFGS-B",
       lower=c(-3.5, -4.5, -1), upper=c(-2.5, -3.5, 1))$par
       thetaSia[,j] <- sapply(del, funSia)
}
parMLE_mean      <- apply(parMLE, 1, mean)
thetaSia_mean    <- apply(thetaSia, 1, mean)
```

Appendix D

Krall et al.'s data

id	dur	status	tPi
1	1.25	1	40
2	1.25	1	51
3	2	1	21
4	2	1	54
5	2	1	41
6	3	1	29
7	5	1	19
8	5	1	29
9	6	1	90
10	6	1	85
11	6	1	14
12	6	1	25
13	7	1	93
14	7	1	61
15	7	1	92
16	9	1	89
17	11	1	68
18	11	1	15
19	11	1	35
20	11	1	78
21	11	1	41
22	13	1	93
23	14	1	28
24	15	1	39
25	16	1	62
26	16	1	93
27	17	1	79
28	17	1	60
29	18	1	49
30	19	1	44

```
31 19 1 52
32 24 1 39
33 25 1 56
34 26 1 34
35 32 1 48
36 35 1 66
37 37 1 137
38 41 1 58
39 42 1 73
40 51 1 61
41 52 1 57
42 54 1 94
43 58 1 74
44 66 1 74
45 67 1 97
46 88 1 90
47 89 1 121
48 92 1 111
49 4 0 4
50 4 0 4
51 7 0 7
52 7 0 7
53 8 0 8
54 12 0 12
55 11 0 11
56 12 0 12
57 13 0 13
58 16 0 16
59 19 0 19
60 19 0 19
61 28 0 28
62 41 0 41
63 53 0 53
64 57 0 57
65 77 0 77
```

Appendix E

R code to generate Figure 5.1.

```
library(survival)
      #Siannis' data
dat                 <- read.table("data/Siannis.txt", h=T)
fit                 <- survfit(Surv(dur, status)~1, data=dat)
summary(fit)
fit1                 <- survfit(Surv(dur, status)~1, data=dat,
                        subset=(status==1))
fit2                 <- survfit(Surv(dur, 1-status)~1, data=dat,
                        subset=(status==0))
par(mfrow=c(2,1))
plot(fit1$time, -log(fit1$surv), pch=20, ylab=expression(
    paste(hat(H), "(t)")), xlab="t")
plot(fit2$time, -log(fit2$surv), pch=20, ylab=expression(
    paste(hat(H), "(c)")), xlab="c")
```

Appendix F

R code to generate Table 5.2 and Table 5.3.

```
library(splines)
library(survival)
funSia <- function(delta=0) log(sum(Ii)/sum(ti)) + delta*sum(
            (exp(log(sum(1-Ii)/sum(ti))))*((ti)^2)-
            (1-Ii)*(ti))/sum(ti)
fun3       <- function(b){
    fun_A          <- function(){
        xx  <- function(tx) exp((-1-delta)*exp(theta)*tx)*
                exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*tx)))
        tmp <- rep(0, n)
        for(i in 1:n) {
             tti <- ti[i]
             tmp[i]<-integrate(xx, lower=tti, upper=500,
                       subdivisions=500)$value}
             tmp
    }

    theta           = b[1]
    Gamma           = b[2]
    delta           = b[3]
    intA            = fun_A()
    L               = -sum(Ii*(theta-exp(theta)*ti-exp(Gamma+
                        delta*(1-exp(theta)*ti))*ti)+(1-Ii)*
                        (theta+Gamma+delta+log(intA)))
}

getStErr3     <- function(theta, Gamma, delta){
fun_L_theta_2       <- function(){
```

```
     xx    <- function(x) (-1-delta)*exp(theta)*x*exp((-1-delta)*
     exp(theta)*x)*exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*x)))
     +exp((-1-delta)*exp(theta)*x)*exp(Gamma)*tti*delta*exp(theta)*x
     *exp(delta*(1-exp(theta)*x))*exp(-exp(Gamma)*tti
     *exp(delta*(1-exp(theta)*x)))
     tmp   <- rep(0, n)
     for(i in 1:n) {tti <- ti[i]; tmp[i]<-integrate(xx, lower=tti,
     upper=180, subdivisions=30)$value}; tmp}
fun_A                <- function(){
     xx    <- function(t) exp((-1-delta)*exp(theta)*t)*exp(-exp(Gamma)
     *tti*exp(delta*(1-exp(theta)*t)))
     tmp   <- rep(0, n)
     for(i in 1:n) {tti <- ti[i]; tmp[i]<-integrate(xx, lower=tti,
     upper=180, subdivisions=30)$value}; tmp}
fun_L_gamma_2        <- function(){
     xx    <- function(t) -exp((-1-delta)*exp(theta)*t)*exp(Gamma)*tti
     *exp(delta*(1-exp(theta)*t))*exp(-exp(Gamma)*tti
     *exp(delta*(1-exp(theta)*t)))
     tmp   <- rep(0, n)
     for(i in 1:n) {tti <- ti[i]; tmp[i]<-integrate(xx, lower=tti,
     upper=180, subdivisions=30)$value}; tmp}
fun_L_delta_2        <- function(){
     xx    <- function(t) -exp(theta)*t*exp((-1-delta)*exp(theta)*t)
     *exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))-exp((-1-delta)
     *exp(theta)*t)*exp(Gamma)*tti*(1-exp(theta)*t)*exp(delta
     *(1-exp(theta)*t))*exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))
     tmp   <- rep(0, n)
     for(i in 1:n) {tti <- ti[i]; tmp[i]<-integrate(xx, lower=tti,
     upper=180, subdivisions=30)$value}; tmp}
fun_L_theta_2_t      <- function(){
     xx    <- function(t) (-1-delta)*exp(theta)*t*exp((-1-delta)
     *exp(theta)*t)*exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))
     +(-1-delta)^2*exp(theta)^2*t^2*exp((-1-delta)*exp(theta)*t)
     *exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))+2*(-1-delta)
     *exp(theta)^2*t^2*exp((-1-delta)*exp(theta)*t)*exp(Gamma)*tti
     *delta*exp(delta*(1-exp(theta)*t))*exp(-exp(Gamma)*tti
     *exp(delta*(1-exp(theta)*t)))+exp((-1-delta)*exp(theta)*t)
     *exp(Gamma)*tti*delta*exp(theta)*t*exp(delta*(1-exp(theta)*t))
     *exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))-exp((-1-delta)
     *exp(theta)*t)*exp(Gamma)*tti*delta^2*exp(theta)^2*t^2
     *exp(delta*(1-exp(theta)*t))*exp(-exp(Gamma)*tti
     *exp(delta*(1-exp(theta)*t)))+exp((-1-delta)*exp(theta)*t)
     *exp(Gamma)^2*tti^2*delta^2*exp(theta)^2*t^2*exp(delta
     *(1-exp(theta)*t))^2*exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))
     tmp   <- rep(0, n)
     for(i in 1:n) {tti <- ti[i]; tmp[i]<-integrate(xx, lower=tti,
```

```
     upper=180, subdivisions=30)$value}; tmp}
fun_A_t              <- function(){
     xx    <- function(t) (-1-delta)*exp(theta)*t*exp((-1-delta)
     *exp(theta)*t)*exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))
     +exp((-1-delta)*exp(theta)*t)*exp(Gamma)*tti*delta*exp(theta)*t
     *exp(delta*(1-exp(theta)*t))*exp(-exp(Gamma)*tti
     *exp(delta*(1-exp(theta)*t)))
     tmp   <- rep(0, n)
     for(i in 1:n) {tti <- ti[i]; tmp[i]<-integrate(xx, lower=tti,
     upper=180, subdivisions=30)$value}; tmp}
fun_L_theta_2_g      <- function(){
     xx    <- function(t) -(-1-delta)*exp(theta)*t*exp((-1-delta)
     *exp(theta)*t)*exp(Gamma)*tti*exp(delta*(1-exp(theta)*t))
     *exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))+exp((-1-delta)
     *exp(theta)*t)*exp(Gamma)*tti*delta*exp(theta)*t*exp(delta
     *(1-exp(theta)*t))*exp(-exp(Gamma)*tti*exp(delta
     *(1-exp(theta)*t)))-exp((-1-delta)*exp(theta)*t)*exp(Gamma)^2
     *tti^2*exp(delta*(1-exp(theta)*t))^2*delta*exp(theta)*t
     *exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))
     tmp   <- rep(0, n)
     for(i in 1:n) {tti <- ti[i]; tmp[i]<-integrate(xx, lower=tti,
     upper=180, subdivisions=30)$value}; tmp}
fun_A_g              <- function(){
     xx    <- function(t) -exp((-1-delta)*exp(theta)*t)*exp(Gamma)*tti
     *exp(delta*(1-exp(theta)*t))*exp(-exp(Gamma)*tti
     *exp(delta*(1-exp(theta)*t)))
     tmp   <- rep(0, n)
     for(i in 1:n) {tti <- ti[i]; tmp[i]<-integrate(xx, lower=tti,
     upper=180, subdivisions=30)$value}; tmp}
fun_L_gamma_2_t      <- function(){
     xx    <- function(t) -(-1-delta)*exp(theta)*t*exp((-1-delta)
     *exp(theta)*t)*exp(Gamma)*tti*exp(delta*(1-exp(theta)*t))
     *exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))+exp((-1-delta)
     *exp(theta)*t)*exp(Gamma)*tti*delta*exp(theta)*t*exp(delta
     *(1-exp(theta)*t))*exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))
     -exp((-1-delta)*exp(theta)*t)*exp(Gamma)^2*tti^2*exp(delta
     *(1-exp(theta)*t))^2*delta*exp(theta)*t*exp(-exp(Gamma)*tti
     *exp(delta*(1-exp(theta)*t)))
     tmp   <- rep(0, n)
     for(i in 1:n) {tti <- ti[i]; tmp[i]<-integrate(xx, lower=tti,
     upper=180, subdivisions=30)$value}; tmp}
fun_L_gamma_2_g      <- function(){
     xx    <- function(t) -exp((-1-delta)*exp(theta)*t)*exp(Gamma)*tti
     *exp(delta*(1-exp(theta)*t))*exp(-exp(Gamma)*tti*exp(delta
     *(1-exp(theta)*t)))+exp((-1-delta)*exp(theta)*t)*exp(Gamma)^2
     *tti^2*exp(delta*(1-exp(theta)*t))^2*exp(-exp(Gamma)*tti
```

```
     *exp(delta*(1-exp(theta)*t)))
     tmp   <- rep(0, n)
     for(i in 1:n) {tti <- ti[i]; tmp[i]<-integrate(xx, lower=tti,
     upper=180, subdivisions=30)$value}; tmp}
fun_L_theta_2_d      <- function(){
     xx    <- function(t) -exp(theta)*t*exp((-1-delta)*exp(theta)*t)
     *exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))-(-1-delta)
     *exp(theta)^2*t^2*exp((-1-delta)*exp(theta)*t)*exp(-exp(Gamma)
     *tti*exp(delta*(1-exp(theta)*t)))-(-1-delta)*exp(theta)*t
     *exp((-1-delta)*exp(theta)*t)*exp(Gamma)*tti*(1-exp(theta)*t)
     *exp(delta*(1-exp(theta)*t))*exp(-exp(Gamma)*tti*exp(delta
     *(1-exp(theta)*t)))-exp(theta)^2*t^2*exp((-1-delta)*exp(theta)*t)
     *exp(Gamma)*tti*delta*exp(delta*(1-exp(theta)*t))*exp(-exp(Gamma)
     *tti*exp(delta*(1-exp(theta)*t)))+exp((-1-delta)*exp(theta)*t)
     *exp(Gamma)*tti*exp(theta)*t*exp(delta*(1-exp(theta)*t))
     *exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))+exp((-1-delta)
     *exp(theta)*t)*exp(Gamma)*tti*delta*exp(theta)*t*(1-exp(theta)*t)
     *exp(delta*(1-exp(theta)*t))*exp(-exp(Gamma)*tti*exp(delta
     *(1-exp(theta)*t)))-exp((-1-delta)*exp(theta)*t)*exp(Gamma)^2
     *tti^2*delta*exp(theta)*t*exp(delta*(1-exp(theta)*t))^2
     *(1-exp(theta)*t)*exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))
     tmp   <- rep(0, n)
     for(i in 1:n) {tti <- ti[i]; tmp[i]<-integrate(xx, lower=tti,
     upper=180, subdivisions=30)$value}; tmp}
fun_A_d              <- function(){
     xx    <- function(t) -exp(theta)*t*exp((-1-delta)*exp(theta)*t)
     *exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))-exp((-1-delta)
     *exp(theta)*t)*exp(Gamma)*tti*(1-exp(theta)*t)*exp(delta
     *(1-exp(theta)*t))*exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))
     tmp   <- rep(0, n)
     for(i in 1:n) {tti <- ti[i]; tmp[i]<-integrate(xx, lower=tti,
     upper=180, subdivisions=30)$value}; tmp}
fun_L_gamma_2_d      <- function(){
     xx    <- function(t) exp((-1-delta)*exp(theta)*t)*exp(Gamma)*tti
     *exp(theta)*t*exp(delta*(1-exp(theta)*t))*exp(-exp(Gamma)*tti
     *exp(delta*(1-exp(theta)*t)))-exp((-1-delta)*exp(theta)*t)
     *exp(Gamma)*tti*(1-exp(theta)*t)*exp(delta*(1-exp(theta)*t))
     *exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))+exp((-1-delta)
     *exp(theta)*t)*exp(Gamma)^2*tti^2*exp(delta*(1-exp(theta)*t))^2
     *(1-exp(theta)*t)*exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))
     tmp   <- rep(0, n)
     for(i in 1:n) {tti <- ti[i]; tmp[i]<-integrate(xx, lower=tti,
     upper=180, subdivisions=30)$value}; tmp}
fun_L_delta_2_t      <- function(){
     xx    <- function(t) -exp(theta)*t*exp((-1-delta)*exp(theta)*t)
     *exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))-(-1-delta)
```

```
     *exp(theta)^2*t^2*exp((-1-delta)*exp(theta)*t)*exp(-exp(Gamma)
     *tti*exp(delta*(1-exp(theta)*t)))-(-1-delta)*exp(theta)*t
     *exp((-1-delta)*exp(theta)*t)*exp(Gamma)*tti*(1-exp(theta)*t)
     *exp(delta*(1-exp(theta)*t))*exp(-exp(Gamma)*tti*exp(delta
     *(1-exp(theta)*t)))-exp(theta)^2*t^2*exp((-1-delta)*exp(theta)*t)
     *exp(Gamma)*tti*delta*exp(delta*(1-exp(theta)*t))*exp(-exp(Gamma)
     *tti*exp(delta*(1-exp(theta)*t)))+exp((-1-delta)*exp(theta)*t)
     *exp(Gamma)*tti*exp(theta)*t*exp(delta*(1-exp(theta)*t))
     *exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))+exp((-1-delta)
     *exp(theta)*t)*exp(Gamma)*tti*delta*exp(theta)*t*(1-exp(theta)*t)
     *exp(delta*(1-exp(theta)*t))*exp(-exp(Gamma)*tti*exp(delta
     *(1-exp(theta)*t)))-exp((-1-delta)*exp(theta)*t)*exp(Gamma)^2
     *tti^2*delta*exp(theta)*t*exp(delta*(1-exp(theta)*t))^2
     *(1-exp(theta)*t)*exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))
     tmp   <- rep(0, n)
     for(i in 1:n) {tti <- ti[i]; tmp[i]<-integrate(xx, lower=tti,
     upper=180, subdivisions=30)$value}; tmp}
fun_L_delta_2_g      <- function(){
     xx    <- function(t) exp((-1-delta)*exp(theta)*t)*exp(Gamma)*tti
     *exp(theta)*t*exp(delta*(1-exp(theta)*t))*exp(-exp(Gamma)*tti
     *exp(delta*(1-exp(theta)*t)))-exp((-1-delta)*exp(theta)*t)
     *exp(Gamma)*tti*(1-exp(theta)*t)*exp(delta*(1-exp(theta)*t))
     *exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))+exp((-1-delta)
     *exp(theta)*t)*exp(Gamma)^2*tti^2*exp(delta*(1-exp(theta)*t))^2
     *(1-exp(theta)*t)*exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))
     tmp   <- rep(0, n)
     for(i in 1:n) {tti <- ti[i]; tmp[i]<-integrate(xx, lower=tti,
     upper=180, subdivisions=30)$value}; tmp}
fun_L_delta_2_d      <- function(){
     xx    <- function(t) exp(theta)^2*t^2*exp((-1-delta)*exp(theta)*t)
     *exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))+2*exp(theta)*t
     *exp((-1-delta)*exp(theta)*t)*exp(Gamma)*tti*(1-exp(theta)*t)
     *exp(delta*(1-exp(theta)*t))*exp(-exp(Gamma)*tti*exp(delta
     *(1-exp(theta)*t)))-exp((-1-delta)*exp(theta)*t)*exp(Gamma)*tti
     *(1-exp(theta)*t)^2*exp(delta*(1-exp(theta)*t))*exp(-exp(Gamma)
     *tti*exp(delta*(1-exp(theta)*t)))+exp((-1-delta)*exp(theta)*t)
     *exp(Gamma)^2*tti^2*(1-exp(theta)*t)^2*exp(delta*(1-exp(theta)
     *t))^2*exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))
     tmp   <- rep(0, n)
     for(i in 1:n) {tti <- ti[i]; tmp[i]<-integrate(xx, lower=tti,
     upper=180, subdivisions=30)$value}; tmp}

L_theta_1       = Ii*(1-exp(theta)*ti+delta*exp(theta)*ti^2*exp(Gamma
+delta*(1-exp(theta)*ti)))
intL_theta_2    = fun_L_theta_2()
intA            = fun_A()
```

```
L_gamma_1       = -Ii*exp(Gamma+delta*(1-exp(theta)*ti))*ti
intL_gamma_2    = fun_L_gamma_2()
L_delta_1       = -Ii*(1-exp(theta)*ti)*exp(Gamma+delta*(1-exp(theta)*ti))*ti
intL_delta_2    = fun_L_delta_2()
L_theta_1_t     = Ii*(-exp(theta)*ti+delta*exp(theta)*ti^2*exp(Gamma+delta
                *(1-exp(theta)*ti))-delta^2*exp(theta)^2*ti^3
                *exp(Gamma+delta*(1-exp(theta)*ti)))
intA_t          = fun_A_t()
L_theta_1_g     = Ii*delta*exp(theta)*ti^2*exp(Gamma+delta*(1-exp(theta)*ti))
intA_g          = fun_A_g()
L_gamma_1_t     = Ii*delta*exp(theta)*ti^2*exp(Gamma+delta*(1-exp(theta)*ti))
L_gamma_1_g     =   -Ii*exp(Gamma+delta*(1-exp(theta)*ti))*ti

L_theta_1_d     = Ii*(exp(theta)*ti^2*exp(Gamma+delta*(1-exp(theta)*ti))
                +delta*exp(theta)*ti^2*(1-exp(theta)*ti)
                *exp(Gamma+delta*(1-exp(theta)*ti)))
intA_d          = fun_A_d()
L_gamma_1_d     = -Ii*(1-exp(theta)*ti)*exp(Gamma+delta
                *(1-exp(theta)*ti))*ti
L_delta_1_t     = Ii*exp(theta)*ti^2*exp(Gamma+delta*(1-exp(theta)*ti))
                +Ii*(1-exp(theta)*ti)*delta*exp(theta)*ti^2
                *exp(Gamma+delta*(1-exp(theta)*ti))
L_delta_1_g     = -Ii*(1-exp(theta)*ti)*exp(Gamma+delta*(1-exp(theta)*ti))*ti
L_delta_1_d     = -Ii*(1-exp(theta)*ti)^2*exp(Gamma+delta*(1-exp(theta)*ti))*ti

  Lt    = sum(L_theta_1 + (1-Ii) * (1 + intL_theta_2/intA), na.rm=T)
  Lg    = sum(L_gamma_1 + (1-Ii) * (1 + intL_gamma_2/intA), na.rm=T)
  Ld    = sum(L_delta_1 + (1-Ii) * (1 + intL_delta_2/intA), na.rm=T)
  Ltt   = sum(L_theta_1_t + (1-Ii) * (fun_L_theta_2_t() * intA
        - intL_theta_2 * intA_t)/intA^2, na.rm=T)
  Ltg   = sum(L_theta_1_g + (1-Ii) * (fun_L_theta_2_g() * intA
        - intL_theta_2 * intA_g)/intA^2, na.rm=T)
  Lgt   = sum(L_gamma_1_t + (1-Ii) * (fun_L_gamma_2_t() * intA
        - intL_gamma_2 * intA_t)/intA^2, na.rm=T)
  Lgg   = sum(L_gamma_1_g + (1-Ii) * (fun_L_gamma_2_g() * intA
        - intL_gamma_2 * intA_g)/intA^2, na.rm=T)

  Ltd   = sum(L_theta_1_d + (1-Ii) * (fun_L_theta_2_d() * intA
        - intL_theta_2 * intA_d)/intA^2, na.rm=T)
  Lgd   = sum(L_gamma_1_d + (1-Ii) * (fun_L_gamma_2_d() * intA
        - intL_gamma_2 * intA_d)/intA^2, na.rm=T)
  Ldt   = sum(L_delta_1_t + (1-Ii) * (fun_L_delta_2_t() * intA
        - intL_delta_2 * intA_t)/intA^2, na.rm=T)
  Ldg   = sum(L_delta_1_g + (1-Ii) * (fun_L_delta_2_g() * intA
        - intL_delta_2 * intA_g)/intA^2, na.rm=T)
  Ldd   = sum(L_delta_1_d + (1-Ii) * (fun_L_delta_2_d() * intA
```

```
        - intL_delta_2 * intA_d)/intA^2, na.rm=T)
  H     = cbind(c(Ltt, Lgt, Ldt), c(Ltg, Lgg, Ldg), c(Ltd, Lgd, Ldd))
  diag(solve(-H, tol=1e-1440))^.5
}

################## different result with notes#####################33
dat                 <- read.table("data/Siannis.txt", h=T)
Ii                  <- dat$status
ti                  <- dat$dur

############## KM estimator ###################################
fit1      <- survfit(Surv(ti, Ii)~1, subset=(Ii==1))
fit2      <- survfit(Surv(ti, 1-Ii)~1, subset=(Ii==0))

postscript("output/KrallKMPlot.eps", horizontal=F)
par(mfrow=c(2,1))
plot(fit1$ti, -log(fit1$surv), pch=20, ylab=expression(paste
    (hat(H), "(t)")), xlab="t")
plot(fit2$ti, -log(fit2$surv), pch=20, ylab=expression(paste
    (hat(H), "(c)")), xlab="c")
dev.off()
#################################################################

fittheta0  <- survreg(Surv(ti, Ii)~1, dist="exponential")
fitGamma0  <- survreg(Surv(ti, 1-Ii)~1, dist="exponential")

theta0     <- -fittheta0$coeff
Gamma0     <- -fitGamma0$coeff
theta0SE   <- fittheta0$var^.5

n          <- length(ti)
del        <- c(-0.3, 0, 0.3)##seq(-1, 1, .01)
thetaSia      <- sapply(del, funSia)

######### MLE of (theta, Gamma, Delta)
b3         <- c(theta0, Gamma0, 0)
parMLE     <- optim(b3, fun3)$par
getStErr3(parMLE[[1]], parMLE[[2]], parMLE[[3]])
```

Appendix G

R code to generate Table 5.5.

```
library(splines)
library(survival)
funSia <- function(delta=0) log(sum(Ii)/sum(ti)) + delta*sum(
            (exp(log(sum(1-Ii)/sum(ti))))*((ti)^2)-(1-Ii)
            *(ti))/sum(ti)
fun3       <- function(b){
    fun_A          <- function(){
        xx  <- function(tx) exp((-1-delta)*exp(theta)*tx)*
                exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)
                *tx)))
        tmp <- rep(0, n)
        for(i in 1:n) {
             tti <- ti[i]
             tmp[i]<-integrate(xx, lower=tti, upper=500,
             subdivisions=500)$value}
             tmp
    }

    theta           = b[1]
    Gamma           = b[2]
    delta           = b[3]
    intA            = fun_A()
    L               = -sum(Ii*(theta-exp(theta)*ti-exp(Gamma+
                        delta*(1-exp(theta)*ti))*ti)+(1-Ii)*
                        (theta+Gamma+delta +log(intA)))
}

getStErr4     <- function(theta, Gamma, delta, bbeta){
```

```
fun_L_theta_2       <- function(){
     xx    <- function(x) (-1-delta)*exp(theta)*x*exp((-1-delta)*
     exp(theta)*x)*exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*x)))
     +exp((-1-delta)*exp(theta)*x)*exp(Gamma)*tti*delta*exp(theta)*x
     *exp(delta*(1-exp(theta)*x))*exp(-exp(Gamma)*tti
     *exp(delta*(1-exp(theta)*x)))
     tmp   <- rep(0, n)
     for(i in 1:n) {tti <- ti[i]; tmp[i]<-integrate(xx, lower=tti,
     upper=180, subdivisions=30)$value}; tmp}
fun_A                <- function(){
     xx    <- function(t) exp((-1-delta)*exp(theta)*t)*exp(-exp(Gamma)
     *tti*exp(delta*(1-exp(theta)*t)))
     tmp   <- rep(0, n)
     for(i in 1:n) {tti <- ti[i]; tmp[i]<-integrate(xx, lower=tti,
     upper=180, subdivisions=30)$value}; tmp}
fun_L_gamma_2        <- function(){
     xx    <- function(t) -exp((-1-delta)*exp(theta)*t)*exp(Gamma)*tti
     *exp(delta*(1-exp(theta)*t))*exp(-exp(Gamma)*tti
     *exp(delta*(1-exp(theta)*t)))
     tmp   <- rep(0, n)
     for(i in 1:n) {tti <- ti[i]; tmp[i]<-integrate(xx, lower=tti,
     upper=180, subdivisions=30)$value}; tmp}
fun_L_delta_2        <- function(){
     xx    <- function(t) -exp(theta)*t*exp((-1-delta)*exp(theta)*t)
     *exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))-exp((-1-delta)
     *exp(theta)*t)*exp(Gamma)*tti*(1-exp(theta)*t)*exp(delta
     *(1-exp(theta)*t))*exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))
     tmp   <- rep(0, n)
     for(i in 1:n) {tti <- ti[i]; tmp[i]<-integrate(xx, lower=tti,
     upper=180, subdivisions=30)$value}; tmp}
fun_L_theta_2_t      <- function(){
     xx    <- function(t) (-1-delta)*exp(theta)*t*exp((-1-delta)
     *exp(theta)*t)*exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))
     +(-1-delta)^2*exp(theta)^2*t^2*exp((-1-delta)*exp(theta)*t)
     *exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))+2*(-1-delta)
     *exp(theta)^2*t^2*exp((-1-delta)*exp(theta)*t)*exp(Gamma)*tti
     *delta*exp(delta*(1-exp(theta)*t))*exp(-exp(Gamma)*tti
     *exp(delta*(1-exp(theta)*t)))+exp((-1-delta)*exp(theta)*t)
     *exp(Gamma)*tti*delta*exp(theta)*t*exp(delta*(1-exp(theta)*t))
     *exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))-exp((-1-delta)
     *exp(theta)*t)*exp(Gamma)*tti*delta^2*exp(theta)^2*t^2
     *exp(delta*(1-exp(theta)*t))*exp(-exp(Gamma)*tti
     *exp(delta*(1-exp(theta)*t)))+exp((-1-delta)*exp(theta)*t)
     *exp(Gamma)^2*tti^2*delta^2*exp(theta)^2*t^2*exp(delta
     *(1-exp(theta)*t))^2*exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))
     tmp   <- rep(0, n)
```

```
     for(i in 1:n) {tti <- ti[i]; tmp[i]<-integrate(xx, lower=tti,
     upper=180, subdivisions=30)$value}; tmp}
fun_A_t              <- function(){
     xx    <- function(t) (-1-delta)*exp(theta)*t*exp((-1-delta)
     *exp(theta)*t)*exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))
     +exp((-1-delta)*exp(theta)*t)*exp(Gamma)*tti*delta*exp(theta)*t
     *exp(delta*(1-exp(theta)*t))*exp(-exp(Gamma)*tti
     *exp(delta*(1-exp(theta)*t)))
     tmp   <- rep(0, n)
     for(i in 1:n) {tti <- ti[i]; tmp[i]<-integrate(xx, lower=tti,
     upper=180, subdivisions=30)$value}; tmp}
fun_L_theta_2_g      <- function(){
     xx    <- function(t) -(-1-delta)*exp(theta)*t*exp((-1-delta)
     *exp(theta)*t)*exp(Gamma)*tti*exp(delta*(1-exp(theta)*t))
     *exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))+exp((-1-delta)
     *exp(theta)*t)*exp(Gamma)*tti*delta*exp(theta)*t*exp(delta
     *(1-exp(theta)*t))*exp(-exp(Gamma)*tti*exp(delta
     *(1-exp(theta)*t)))-exp((-1-delta)*exp(theta)*t)*exp(Gamma)^2
     *tti^2*exp(delta*(1-exp(theta)*t))^2*delta*exp(theta)*t
     *exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))
     tmp   <- rep(0, n)
     for(i in 1:n) {tti <- ti[i]; tmp[i]<-integrate(xx, lower=tti,
     upper=180, subdivisions=30)$value}; tmp}
fun_A_g              <- function(){
     xx    <- function(t) -exp((-1-delta)*exp(theta)*t)*exp(Gamma)*tti
     *exp(delta*(1-exp(theta)*t))*exp(-exp(Gamma)*tti
     *exp(delta*(1-exp(theta)*t)))
     tmp   <- rep(0, n)
     for(i in 1:n) {tti <- ti[i]; tmp[i]<-integrate(xx, lower=tti,
     upper=180, subdivisions=30)$value}; tmp}
fun_L_gamma_2_t      <- function(){
     xx    <- function(t) -(-1-delta)*exp(theta)*t*exp((-1-delta)
     *exp(theta)*t)*exp(Gamma)*tti*exp(delta*(1-exp(theta)*t))
     *exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))+exp((-1-delta)
     *exp(theta)*t)*exp(Gamma)*tti*delta*exp(theta)*t*exp(delta
     *(1-exp(theta)*t))*exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))
     -exp((-1-delta)*exp(theta)*t)*exp(Gamma)^2*tti^2*exp(delta
     *(1-exp(theta)*t))^2*delta*exp(theta)*t*exp(-exp(Gamma)*tti
     *exp(delta*(1-exp(theta)*t)))
     tmp   <- rep(0, n)
     for(i in 1:n) {tti <- ti[i]; tmp[i]<-integrate(xx, lower=tti,
     upper=180, subdivisions=30)$value}; tmp}
fun_L_gamma_2_g      <- function(){
     xx    <- function(t) -exp((-1-delta)*exp(theta)*t)*exp(Gamma)*tti
     *exp(delta*(1-exp(theta)*t))*exp(-exp(Gamma)*tti*exp(delta
     *(1-exp(theta)*t)))+exp((-1-delta)*exp(theta)*t)*exp(Gamma)^2
```

```
     *tti^2*exp(delta*(1-exp(theta)*t))^2*exp(-exp(Gamma)*tti
     *exp(delta*(1-exp(theta)*t)))
     tmp   <- rep(0, n)
     for(i in 1:n) {tti <- ti[i]; tmp[i]<-integrate(xx, lower=tti,
     upper=180, subdivisions=30)$value}; tmp}
fun_L_theta_2_d      <- function(){
     xx    <- function(t) -exp(theta)*t*exp((-1-delta)*exp(theta)*t)
     *exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))-(-1-delta)
     *exp(theta)^2*t^2*exp((-1-delta)*exp(theta)*t)*exp(-exp(Gamma)
     *tti*exp(delta*(1-exp(theta)*t)))-(-1-delta)*exp(theta)*t
     *exp((-1-delta)*exp(theta)*t)*exp(Gamma)*tti*(1-exp(theta)*t)
     *exp(delta*(1-exp(theta)*t))*exp(-exp(Gamma)*tti*exp(delta
     *(1-exp(theta)*t)))-exp(theta)^2*t^2*exp((-1-delta)*exp(theta)*t)
     *exp(Gamma)*tti*delta*exp(delta*(1-exp(theta)*t))*exp(-exp(Gamma)
     *tti*exp(delta*(1-exp(theta)*t)))+exp((-1-delta)*exp(theta)*t)
     *exp(Gamma)*tti*exp(theta)*t*exp(delta*(1-exp(theta)*t))
     *exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))+exp((-1-delta)
     *exp(theta)*t)*exp(Gamma)*tti*delta*exp(theta)*t*(1-exp(theta)*t)
     *exp(delta*(1-exp(theta)*t))*exp(-exp(Gamma)*tti*exp(delta
     *(1-exp(theta)*t)))-exp((-1-delta)*exp(theta)*t)*exp(Gamma)^2
     *tti^2*delta*exp(theta)*t*exp(delta*(1-exp(theta)*t))^2
     *(1-exp(theta)*t)*exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))
     tmp   <- rep(0, n)
     for(i in 1:n) {tti <- ti[i]; tmp[i]<-integrate(xx, lower=tti,
     upper=180, subdivisions=30)$value}; tmp}
fun_A_d              <- function(){
     xx    <- function(t) -exp(theta)*t*exp((-1-delta)*exp(theta)*t)
     *exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))-exp((-1-delta)
     *exp(theta)*t)*exp(Gamma)*tti*(1-exp(theta)*t)*exp(delta
     *(1-exp(theta)*t))*exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))
     tmp   <- rep(0, n)
     for(i in 1:n) {tti <- ti[i]; tmp[i]<-integrate(xx, lower=tti,
     upper=180, subdivisions=30)$value}; tmp}
fun_L_gamma_2_d      <- function(){
     xx    <- function(t) exp((-1-delta)*exp(theta)*t)*exp(Gamma)*tti
     *exp(theta)*t*exp(delta*(1-exp(theta)*t))*exp(-exp(Gamma)*tti
     *exp(delta*(1-exp(theta)*t)))-exp((-1-delta)*exp(theta)*t)
     *exp(Gamma)*tti*(1-exp(theta)*t)*exp(delta*(1-exp(theta)*t))
     *exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))+exp((-1-delta)
     *exp(theta)*t)*exp(Gamma)^2*tti^2*exp(delta*(1-exp(theta)*t))^2
     *(1-exp(theta)*t)*exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))
     tmp   <- rep(0, n)
     for(i in 1:n) {tti <- ti[i]; tmp[i]<-integrate(xx, lower=tti,
     upper=180, subdivisions=30)$value}; tmp}
fun_L_delta_2_t      <- function(){
     xx    <- function(t) -exp(theta)*t*exp((-1-delta)*exp(theta)*t)
```

```
     *exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))-(-1-delta)
     *exp(theta)^2*t^2*exp((-1-delta)*exp(theta)*t)*exp(-exp(Gamma)
     *tti*exp(delta*(1-exp(theta)*t)))-(-1-delta)*exp(theta)*t
     *exp((-1-delta)*exp(theta)*t)*exp(Gamma)*tti*(1-exp(theta)*t)
     *exp(delta*(1-exp(theta)*t))*exp(-exp(Gamma)*tti*exp(delta
     *(1-exp(theta)*t)))-exp(theta)^2*t^2*exp((-1-delta)*exp(theta)*t)
     *exp(Gamma)*tti*delta*exp(delta*(1-exp(theta)*t))*exp(-exp(Gamma)
     *tti*exp(delta*(1-exp(theta)*t)))+exp((-1-delta)*exp(theta)*t)
     *exp(Gamma)*tti*exp(theta)*t*exp(delta*(1-exp(theta)*t))
     *exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))+exp((-1-delta)
     *exp(theta)*t)*exp(Gamma)*tti*delta*exp(theta)*t*(1-exp(theta)*t)
     *exp(delta*(1-exp(theta)*t))*exp(-exp(Gamma)*tti*exp(delta
     *(1-exp(theta)*t)))-exp((-1-delta)*exp(theta)*t)*exp(Gamma)^2
     *tti^2*delta*exp(theta)*t*exp(delta*(1-exp(theta)*t))^2
     *(1-exp(theta)*t)*exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))
     tmp   <- rep(0, n)
     for(i in 1:n) {tti <- ti[i]; tmp[i]<-integrate(xx, lower=tti,
     upper=180, subdivisions=30)$value}; tmp}
fun_L_delta_2_g      <- function(){
     xx    <- function(t) exp((-1-delta)*exp(theta)*t)*exp(Gamma)*tti
     *exp(theta)*t*exp(delta*(1-exp(theta)*t))*exp(-exp(Gamma)*tti
     *exp(delta*(1-exp(theta)*t)))-exp((-1-delta)*exp(theta)*t)
     *exp(Gamma)*tti*(1-exp(theta)*t)*exp(delta*(1-exp(theta)*t))
     *exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))+exp((-1-delta)
     *exp(theta)*t)*exp(Gamma)^2*tti^2*exp(delta*(1-exp(theta)*t))^2
     *(1-exp(theta)*t)*exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))
     tmp   <- rep(0, n)
     for(i in 1:n) {tti <- ti[i]; tmp[i]<-integrate(xx, lower=tti,
     upper=180, subdivisions=30)$value}; tmp}
fun_L_delta_2_d      <- function(){
     xx    <- function(t) exp(theta)^2*t^2*exp((-1-delta)*exp(theta)*t)
     *exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))+2*exp(theta)*t
     *exp((-1-delta)*exp(theta)*t)*exp(Gamma)*tti*(1-exp(theta)*t)
     *exp(delta*(1-exp(theta)*t))*exp(-exp(Gamma)*tti*exp(delta
     *(1-exp(theta)*t)))-exp((-1-delta)*exp(theta)*t)*exp(Gamma)*tti
     *(1-exp(theta)*t)^2*exp(delta*(1-exp(theta)*t))*exp(-exp(Gamma)
     *tti*exp(delta*(1-exp(theta)*t)))+exp((-1-delta)*exp(theta)*t)
     *exp(Gamma)^2*tti^2*(1-exp(theta)*t)^2*exp(delta*(1-exp(theta)
     *t))^2*exp(-exp(Gamma)*tti*exp(delta*(1-exp(theta)*t)))
     tmp   <- rep(0, n)
     for(i in 1:n) {tti <- ti[i]; tmp[i]<-integrate(xx, lower=tti,
     upper=180, subdivisions=30)$value}; tmp}

L_theta_1       = Ii*(1-exp(theta)*ti+delta*exp(theta)*ti^2*exp(Gamma
+delta*(1-exp(theta)*ti)))
intL_theta_2    = fun_L_theta_2()
```

```
intA            = fun_A()
L_gamma_1       = -Ii*exp(Gamma+delta*(1-exp(theta)*ti))*ti
intL_gamma_2    = fun_L_gamma_2()
L_delta_1       = -Ii*(1-exp(theta)*ti)*exp(Gamma+delta*(1-exp(theta)
                    *ti))*ti
intL_delta_2    = fun_L_delta_2()
L_theta_1_t     = Ii*(-exp(theta)*ti+delta*exp(theta)*ti^2*exp(Gamma
                +delta
                *(1-exp(theta)*ti))-delta^2*exp(theta)^2*ti^3
                *exp(Gamma+delta*(1-exp(theta)*ti)))
intA_t          = fun_A_t()
L_theta_1_g     = Ii*delta*exp(theta)*ti^2*exp(Gamma+delta*(1-
                exp(theta)*ti))
intA_g          = fun_A_g()
L_gamma_1_t     = Ii*delta*exp(theta)*ti^2*exp(Gamma+delta*(1-
                exp(theta)*ti))
L_gamma_1_g     =   -Ii*exp(Gamma+delta*(1-exp(theta)*ti))*ti

L_theta_1_d     = Ii*(exp(theta)*ti^2*exp(Gamma+delta*(1-
                exp(theta)*ti))
                +delta*exp(theta)*ti^2*(1-exp(theta)*ti)
                *exp(Gamma+delta*(1-exp(theta)*ti)))
intA_d          = fun_A_d()
L_gamma_1_d     = -Ii*(1-exp(theta)*ti)*exp(Gamma+delta
                *(1-exp(theta)*ti))*ti
L_delta_1_t     = Ii*exp(theta)*ti^2*exp(Gamma+delta*(1-exp(theta)*ti))
                +Ii*(1-exp(theta)*ti)*delta*exp(theta)*ti^2
                *exp(Gamma+delta*(1-exp(theta)*ti))
L_delta_1_g     = -Ii*(1-exp(theta)*ti)*exp(Gamma+delta*(1-
                exp(theta)*ti))*ti
L_delta_1_d     = -Ii*(1-exp(theta)*ti)^2*exp(Gamma+delta*(1-
                exp(theta)*ti))*ti

  Lt    = sum(L_theta_1 + (1-Ii) * (1 + intL_theta_2/intA), na.rm=T)
  Lg    = sum(L_gamma_1 + (1-Ii) * (1 + intL_gamma_2/intA), na.rm=T)
  Ld    = sum(L_delta_1 + (1-Ii) * (1 + intL_delta_2/intA), na.rm=T)
  Ltt   = sum(L_theta_1_t + (1-Ii) * (fun_L_theta_2_t() * intA
        - intL_theta_2 * intA_t)/intA^2, na.rm=T)
  Ltg   = sum(L_theta_1_g + (1-Ii) * (fun_L_theta_2_g() * intA
        - intL_theta_2 * intA_g)/intA^2, na.rm=T)
  Lgt   = sum(L_gamma_1_t + (1-Ii) * (fun_L_gamma_2_t() * intA
        - intL_gamma_2 * intA_t)/intA^2, na.rm=T)
  Lgg   = sum(L_gamma_1_g + (1-Ii) * (fun_L_gamma_2_g() * intA
        - intL_gamma_2 * intA_g)/intA^2, na.rm=T)

  Ltd   = sum(L_theta_1_d + (1-Ii) * (fun_L_theta_2_d() * intA
```

```
        - intL_theta_2 * intA_d)/intA^2, na.rm=T)
  Lgd   = sum(L_gamma_1_d + (1-Ii) * (fun_L_gamma_2_d() * intA
        - intL_gamma_2 * intA_d)/intA^2, na.rm=T)
  Ldt   = sum(L_delta_1_t + (1-Ii) * (fun_L_delta_2_t() * intA
        - intL_delta_2 * intA_t)/intA^2, na.rm=T)
  Ldg   = sum(L_delta_1_g + (1-Ii) * (fun_L_delta_2_g() * intA
        - intL_delta_2 * intA_g)/intA^2, na.rm=T)
  Ldd   = sum(L_delta_1_d + (1-Ii) * (fun_L_delta_2_d() * intA
        - intL_delta_2 * intA_d)/intA^2, na.rm=T)
  Ltb   = L2tt
  Lgb   = Ldb = Lbg = Lbd = 0
  Lbt   = L2tt
  Lbb   = L2tt
  H     = cbind(c(Ltt, Lgt, Ldt, Lbt), c(Ltg, Lgg, Ldg, Lbg),
          c(Ltd, Lgd, Ldd, Lbd), c(Ltb, Lgb, Ldb, Lbb))
  diag(solve(-H, tol=1e-1440))^.5
}

ti     <- c(6, 6, 6, 6, 7, 9, 10, 10, 11, 13, 16, 17, 19, 20, 22,
           23, 25, 32, 32, 34, 35)
Ii     <- c(0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0,
           0, 0, 0)
tiP    <- c(1, 1, 2, 2, 3, 4, 4, 5, 5, 8, 8, 8, 8, 11, 11, 12, 12,
           15, 17, 22, 23)
IiP    <- rep(1, length(tiP))
#dat    <- data.frame(ti, Ii)


############# KM estimator ###################################
fit1      <- survfit(Surv(ti, Ii)~1, subset=(Ii==1))
fit2      <- survfit(Surv(ti, 1-Ii)~1, subset=(Ii==0))
fitP      <- survfit(Surv(tiP, IiP)~1)
postscript("output/exampleBKM.eps", horizontal=F)
par(mfrow=c(3,1))
plot(fit1$ti, -log(fit1$surv), pch=20, ylab=expression(paste
    (hat(H), "(t)")), xlab="Treatment group's failure time t")
plot(fit2$ti, -log(fit2$surv), pch=20, ylab=expression(paste
    (hat(H), "(c)")), xlab="Treatment group's censored time t")
plot(fitP$ti, -log(fitP$surv), pch=20, ylab=expression(paste
    (hat(H), "(t)")),
    xlab="Control group's failure time t")
dev.off()
#################################################################


fittheta0  <- survreg(Surv(ti, Ii)~1, dist="exponential")
```

```
fitGamma0  <- survreg(Surv(ti, 1-Ii)~1, dist="exponential")
fitOmega0  <- survreg(Surv(tiP, IiP)~1, dist="exponential")
theta0     <- -fittheta0$coeff
Gamma0     <- -fitGamma0$coeff
Omega0     <- -fitOmega0$coeff
theta0SE   <- fittheta0$var^.5


n          <- length(ti)
n2         <- length(tiP)
del        <- c(-0.3, 0, 0.3)##seq(-1, 1, .01)


thetaSia      <- sapply(del, funSia)

######### MLE of (theta, Gamma, Delta)
b3         <- c(theta0, Gamma0, 0)
parMLE     <- optim(b3, fun3)$par
getStErr4(parMLE[[1]], parMLE[[2]], parMLE[[3]], Omega0-parMLE[[1]])
```