Universidade do Minho
Escola de Ciências

Ana Cristina Machado Lopes Moreira

Methods for analysis of Multi-state survival data

May 2014

Universidade do Minho
Escola de Ciências

Ana Cristina Machado Lopes Moreira

Methods for analysis of Multi-state survival data

Tese de Doutoramento em Ciências
Especialidade de Matemática

Trabalho efectuado sob a orientação do
Professor Doutor Luís Filipe Meira Machado

May 2014

Acknowledgments

Firstly I want to express my extreme gratitude to my supervisor, Professor Luís Machado, for his encouragement, support and guidance during these last years. My thanks are directed to Professor Jacobo de Uña Álvarez from University of Vigo for receiving me in his department and helping me in solving theoretical questions.

My thanks to the Professors of the Department of Mathematics and Applications, University of Minho, Conceição during these last years and my colleagues for help and support.

I express my thanks to my parents, brother, my friends and my all family for all support. I would like to thank Pedro for his patience, questions and encouragement.

I would like to acknowledge the financial support from the Portuguese Ministry of Science, Technology and Higher Education in the form of grant SFRH/BD/62284/2009 and by Programa Operacional Factores de Competitividade COMPETE and by research Centre of Mathematics of the University of Minho through FCT - Fundação para a Ciência e a Tecnologia, within the Project Est-C/MAT/UI0013/2011 and CMAT.

POPH
PROGRAMA OPERACIONAL POTENCIAL HUMANO

QREN
QUADRO DE REFERÊNCIA ESTRATÉGICO NACIONAL
PORTUGAL 2007.2013

UNIÃO EUROPEIA
Fundo Social Europeu

Abstract

This thesis is concerned with multi-state survival analysis. In this context, we propose methods for the analysis of multi-state survival data. The methods developed in this thesis are motivated by the applications to the medical sciences. However, they can also be applied to economics, astronomy, and engineering, among other fields. This is an exciting and full potential area of research, with many interesting problems.

Survival Analysis is concerned with studying inter-event times. In a classical setup, the focus is on the elapsed time between two well-defined events: the starting event ("alive"), and the terminating event ("death"). Multi-state models can be considered as a generalization of the survival process where "death" is the ultimate outcome, but where intermediate states are identified. If the events are of the same nature, this is usually referred as recurrent events, whereas if they represent different states they are usually modelled through their intensity functions. When analyzing recurrent event data, the inter-event times are referred to as the gap times, and they are of course determined by the times at which the recurrences take place (i.e. the recurrence times). The statistical analysis of consecutive gap times is an issue of much importance. Most of the times, one will be interested in describing not only the marginal distribution of the gap times but also the bivariate distribution of the joint gap times. This will be considered in Chapter 2. Specifically, we propose methods for estimate the bivariate distribution under right censoring and conditional bivariate distribution given a quantitative covariate.

Alternatively, we may think the gap times as arising from a particular multi-state model such as the progressive three-state model or the progressive k-state model.

A multi-state model is a model for a stochastic process, which is characterized by a set of states and the possible transitions among them. The states represent different stages of the disease course along a follow-up. Several multi-state models that have been widely used in biomedical applications but the three-state progressive model and the illness-death model are certainly the most common. The illness-death model is a generalization of the three-state progressive model in which a direct transition from the "alive" state to the final, absorbing "dead" state is possible. In this model one of the major goals is the estimation of the so-called transition probabilities. Traditionally, this estimation is performed under a Markov assumption, which leads to the so-called Aalen-Johansen estimator. Unfortunately, the variance of this estimator may be large in heavily censored scenarios. The possibility of improving this estimator via presmoothing is explored in Chapter 3.

For the practical application of the methods presented in Chapters 2 and 3, we developed several functions in R (R Development Core Team, 2013). Some of these functions were used to build an R package for the estimation of the bivariate distribution function. Details about this and other packages for multi-state modelling are given in Chapter 4.

All methods are illustrated by means of its application to real biomedical datasets.

Resumo

Esta tese está focada na análise de sobrevivência multiestado. Neste contexto, propusemos métodos para a análise de dados de sobrevivência multiestado. Os métodos desenvolvidos nesta tese foram motivados pelas aplicações na medicina. No entanto, estes podem ser aplicados à economia, astronomia e engenharia entre outros campos. É uma área excitante e de grande potencial de investigação, com muitos problemas interessantes.

A análise de sobrevivência preocupa-se com o estudo de tempos entre eventos. Numa versão clássica, o foco é sobre o tempo decorrido entre dois eventos bem definidos: o evento inicial ("vivo"), e o evento final ("morte"). Os modelos multiestado podem ser considerados como uma generalização de um processo de sobrevivência onde "morte" é o resultado final, mas onde estados intermédios são identificados. Se os eventos são da mesma natureza, estamos no contexto de eventos recorrentes; se os estados representam diferentes eventos então eles são habitualmente modelados através de funções de intensidade. Na análise de dados de eventos recorrentes, os tempos entre eventos são usualmente referidos como "*gap times*", e são determinados pelos tempos onde as recorrências ocorrem (ou seja, tempos de recorrência). A análise estatística de "*gap times*" consecutivos é um tema que tem recebido muita atenção nos últimos anos. Na maioria das vezes, não estão só interessados em descrever a distribuição marginal dos "*gap times*", mas também a distribuição bivariada conjunta dos mesmos. Isto será considerado no Capítulo 2. Especificamente, propusemos métodos para estimar a distribuição bivariada na presença de censura e a distribuição bivariada condicional, dada uma covariável quantitativa.

Alternativamente, pensamos nos "*gap times*" como resultado de um modelo multiestado particular tal como modelo progressivo de três estados ou modelo progressivo de k-estados. Um modelo multiestado é um modelo para um processo estocástico, que é caracterizado por um conjunto de estados e possíveis transições entre eles. Os estados representam diferentes etapas do percurso da doença ao longo de um acompanhamento "*follow up*". Vários modelos multiestados têm sido amplamente utilizados em aplicações biomédicas, mas o modelo progressivo de três estados e o modelo doença-morte são os mais comuns. O modelo doença-morte é a generalização do modelo progressivo de três estados em que uma transição direta do estado "vivo" para o final, estado absorvente "morte" é possível. Neste modelo um dos principais objetivos é a estimativa das probabilidades de transição. Tradicionalmente, esta estimativa é calculada sob o pressuposto de Markov, que tradicionalmente recorre ao estimador de Aalen-Johansen. Infelizmente, a variância deste estimador pode ser elevada em cenários com elevadas taxas de censura. A possibilidade de melhorar este estimador com pré-suavização é explorada no Capítulo 3.

Para aplicações práticas dos métodos presentes no Capítulo 2 e 3, desenvolvemos várias funções em R (R Development Core Team, 2013). Algumas dessas funções foram usadas para construir um package no R para estimar a função distribuição bivariada. Detalhes sobre este package e outros packages para modelação multiestado são dados no Capítulo 4.

Todos os métodos serão ilustrados por meio da sua aplicação em dados reais em medicina.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Survival Analysis

Survival analysis is a branch of statistics devoted to the analysis of the elapsed time from a starting point until the occurrence of a given event of interest. Survival analysis or time-to-event data analysis is prominently used in the biomedical sciences where the interest is in observing time to death either of patients or of laboratory animals. This time is therefore called the “lifetime” or the “survival time”. In engineering sciences, it is also known “reliability analysis” or “failure time analysis” and the main focus is in modelling the time it takes for machines or electronic components to break down. Other applications include the economics, astronomy, social sciences, and psychology among other fields. There exists an extensive literature on Survival Analysis. Main contributions include books by Kalbfleisch and Prentice (2002), Cox and Oakes (1984), Klein and Moeschberger (1997) and Hougaard (2000).

In survival analysis the variable of interest or response variable is time. Let T be a random non-negative variable representing the individual survival time from a homogeneous population. Assume T is a continuous variable with probability density function $f(t)$ and distribution function $F(t) = P(T \leqslant t)$. The survivor function, $S(t)$, is defined to be the probability that the survival time is greater than t

$$S(t) = P(T > t) = 1 - F(t) = \int_t^\infty f(x)dx$$

$S(t)$ is a non-increasing left continuous function with $S(0) = 1$ and $lim_{x \to \infty} S(t) = 0$. The hazard function is the probability that an individual "dies" at some time t, conditional that he survived until that time. Thus, the hazard function represents the instantaneous probability that the event will occur at a given time t and can be written as

$$h(t) = lim_{dt \to 0} \frac{P(t \leqslant T < t + dt | T \geqslant t)}{dt} = \frac{f(t)}{S(t)}$$

For example, $h(t)$ is the probability that an individual, who is alive on day t, dies in the following day, the survival time is measured in days. The function $H(t)$ is called cumulative hazard or cumulative risk and is defined by

$$H(t) = \int_0^t h(x)dx$$

We can obtain some useful relationships between each one of these functions:

$$\begin{aligned} S(t) &= exp(-\int_0^t h(x)dx) \\ S(t) &= exp(-H(t)) \\ h(t) &= \frac{f(t)}{S(t)} \end{aligned}$$

The distinguishable feature of survival analysis is censoring. A censored lifetime occurs when we have some information about individual survival time, but we do not know the survival time exactly. Right censoring take place when for some individuals the event of interest has not been observed until the end of the study and therefore we do not know the exact waiting time. Right censoring can occur because the event of interest has not yet occurred but also due to loss of follow-up. Sometimes the survival time is less than some specified time t, in other words, the observed time is bigger than the time where the event of interest occur and the observation is called to be left-censored. The interval-censoring is when individuals are known to have experienced an event within an interval of time. Censored observations can not be ignored since they carry important information about the survival. Because of censoring standard statistical methods such as regression analysis or student's t-test are not valid.

A basic task in survival analysis is the estimation of survival in the presence of censoring. Suppose first that we have a sample of dimension n with observed survival times, $t_1, t_2, ..., t_n$, where none of the observations are censored. Then, survival can be estimated nonparametrically using the empirical estimator, given by

$$\hat{S}(t) = \frac{1}{n} \sum_{i=1}^{n} I(t_i > t)$$

which is the ratio of the total number of individuals alive at time t to the total number of the individuals in the study and I is the indicator function.

The Kaplan-Meier estimator (Kaplan and Meier, 1958) is a nonparametric estimator which may be used to estimate the survival distribution function from censored data. The estimator is also referred to as the Product-Limit estimator can be seen as a generalization of the empirical estimator for censored data. Let $t_{(1)} < t_{(2)} < ... < t_{(m)}$ denote the distinct ordered times of death (not counting censoring times). Let d_i be the number of deaths or individuals who experienced the event at $t_{(i)}$, and let n_i be the number of individuals at risk who were alive and uncensored just before $t_{(i)}$. Then the Kaplan-Meier estimates of the survivor function is

$$\hat{S}(t) = \prod_{i:t_{(i)} \leqslant t} \left(1 - \frac{d_i}{n_i} \right)$$

This estimator is a step function which steps down at each event time (only), with $\hat{S}(t) = 1$ for $t < t_{(1)}$. When there is no censoring this estimator matches with the empirical estimator. The Kaplan-Meier estimator can also be expressed in terms of Kaplan-Meier weights.

$$\hat{S}_{KM}(t) = \sum_{i=1}^{n} W_i I(t_i > t)$$

where $W_i = \frac{\delta_i}{n-i+1} \prod_{j=1}^{i-1} (1 - \frac{\delta_j}{n-j+1})$ are the Kaplan-Meier weights. Here δ_i is 1 if the event occur and 0 otherwise. We introduce an estimator based on inverse probability of censoring weighted (Satten and Datta, 2001)

$$\hat{S}_{ipcw}(t) = \frac{1}{n}\sum_{i=1}^{n}\frac{I(t_i > t)\delta_i}{1 - \hat{G}(t_i^-)}$$

where $\hat{G}(t)$ is the censoring distribution function.

One major goal in survival analysis is to study the relationship between the different covariates and survival time. A classical model relating the hazard function and a certain number of covariates is given by the proportional hazards model, called Cox model (Cox, 1972). The Cox proportional hazards model that is a semiparametric model can be written as

$$h_i(t|Z_i) = h_0(t)e^{\beta^t Z_i}$$

where $h_0(t)$ is a non-negative baseline hazard function, $Z_i = (Z_{i1}, Z_{i2}, ..., Z_{ip})$ a vector of p covariates, and $\beta^t = (\beta_1, \beta_2, ..., \beta_p)$ the associated vector of unknown regression parameters. Using the Cox model to evaluate the impact of a set of covariates on the hazard function implies two important assumptions: the effect of covariates do not vary over the time (proportional hazards assumption) and the effect of covariates acts linearly on the logarithm of the hazard ratio. In clinical studies, individuals are observed and individual data and covariate information are collected at many occasions through a follow-up study. In many medical studies covariate data are collected longitudinally. In many instances there are covariates that change their values over time and their analysis is most often modelled using the time-dependent Cox proportional hazards model.

$$h(t|Z_i) = h_0 e^{(\beta^t Z_i(t))}$$

The introduction of these covariates in the survival process can make the patients risk change from on time point to the next as the values of the covariates change. Time dependent covariates might represent either a qualitative change in patient's conditions or individual continuous information. Further details about the time-dependent

Cox model can be seen in the monographs of Kalbfleisch and Prentice (2002) and Hougaard (2000).

1.2 Multi-state Models

Multi-state models (Andersen et al., 1993; Meira-Machado et al., 2009) are models for a stochastic process, which at any time occupies one of a set of discrete states. A change of state is called a transition, or an event. States can be transient or absorbing. An absorbing state is a state from which one can ever leave once it enters. These models can be successfully used for describing complicated event history data, for example describing stages in the disease progression of a patient. In contrast to traditional survival methods (e.g. the Cox model and the Kaplan-Meier estimator of survival), in these (longitudinal) survival studies, besides overall survival, more than one endpoint can be observed. For example, in cancer studies, other endpoints such as locoregional recurrence, distant metastasis and dead are observed.

The state structure of a multi-state model (MSM) identifies the states and also the transitions allowed between states (Hougaard, 2000). The complexity of a MSM greatly depends on the number of states defined and also on the transitions allowed between these states. The simplest form of a MSM is the mortality model (Figure 1.1) (with states "alive" and "dead" and a single transition allowed between them) for survival analysis.

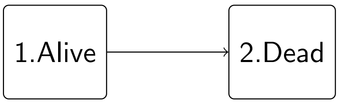

Figure 1.1: Mortality model

By splitting the "Alive" state from the simple mortality model for survival data into two transient states, we therefore obtain the simplest progressive three-state model (see Figure 1.2).

This model is suitable in the presence of an intermediate event (e.g. a recurrence) which may influence the survival prognosis. A more general model is the k-state

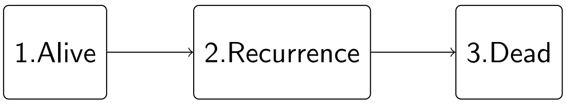

Figure 1.2: Progressive three-state model

progressive model with $(k-1)$ transient states and an absorbing state. If the events of concern are of the same nature (e.g. cancer patients may experience several recurrent disease episodes) this are usually referred as recurrent event data. See Cook and Lawless (2007) for an up-to-date review of statistical methods for recurrent event data. Another possible MSM to describe the disease progression is the illness-death model (Figure 1.3). This model, also known as disability model, is probably the most used model in literature. The illness-death model is fully characterized by three states and three transition intensities ($1 \to 2$, $1 \to 3$ and $2 \to 3$) each one describing the instantaneous hazard of moving out of one state into another. This model can be used to study the incidence of the disease as well as death. In particular, one may evaluate if previously diseased subjects have the same risk of death as those who have been healthy all their lives.

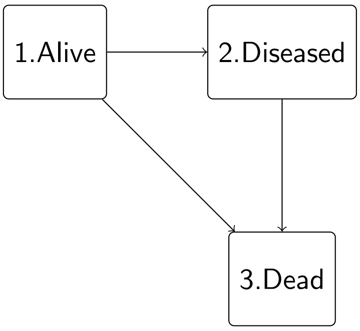

Figure 1.3: Illness-death model

Other common models in literature include the competing risks model and the bivariate model in Figure 1.4 (for bivariate failure times, e.g. survival of twins). The competing risk model extends the simple mortality model for survival data by considering that each individual may "die" due to any of several causes (Hougaard, 1999).

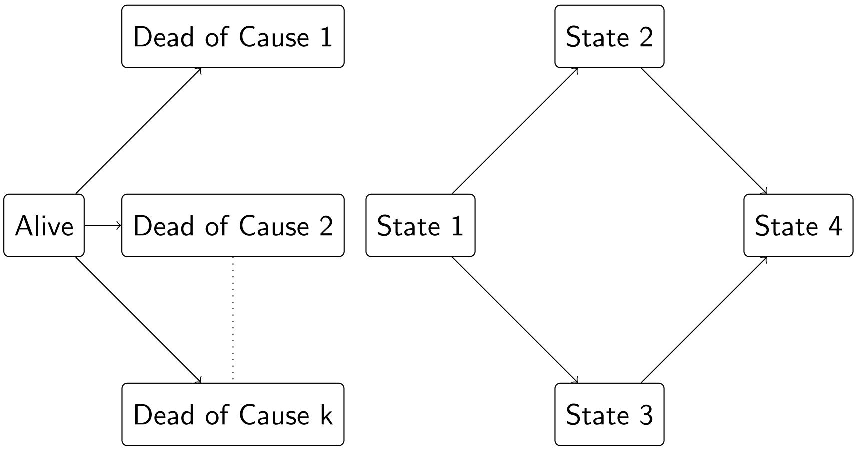

Figure 1.4: Competing risks model and bivariate model

1.3 Research Significance and Objectives

In many medical studies, patients may experience several events. The analysis in such studies is often performed using multi-state models. These models are very useful for describing event history data offering a better understanding of the process of the illness, and leading to a better knowledge of the evolution of the disease over time. Issues of interest include the estimation of progression rates (state occupation probabilities, transition probabilities), assessing the effects of individual risk factors, survival rates or prognostic forecasting. Other interests include the estimation of the cumulative incidence functions, the waiting time distributions, the bivariate distribution function for sequentially ordered events (gap times), etc.

In longitudinal studies of disease typical multi-state models include the illness-death model and the progressive three-state model for which we aim to derive new estimators for the transition probabilities and for the bivariate distribution function for censored gap times. Main objectives of this thesis include:

- Development of new methods for estimating several quantities of interest such as the transition probabilities and the bivariate distribution function;

- Validation of the new methodologies through theoretical results and simulation studies;

- Development of programs (open-source software packages in R) to implement the new methods and promote these programs among biomedical researches;

- Application to real survival datasets.

1.4 Real Data

The methods proposed in this thesis are illustrated by means of its application to real biomedical datasets. For illustrating our methods in the three-state progressive and illness-death models we have used the following public and widely used medical databases.

Bladder Cancer Study

Data coming from a Bladder cancer study (Byar, 1980) conducted by the Veterans Administration Cooperative Urological Research Group are used to illustrate the new estimators for the bivariate distribution function in the context of the progressive three-state model in Chapter 4. In this study, patients had superficial bladder tumors that were removed by transurethral resection. Many patients had multiple recurrences (up to a maximum of 9) of tumors during the study, and new tumors were removed at each visit. For illustration purposes we re-analyze data from 85 individuals in the placebo and thiotepa treatment groups. From the total of 85 patients, 47 relapsed at least once and among these, 29 experienced a new recurrence. Here, only the first two recurrence times (in months) are considered. These dataset is available as part of the *R* **survival** package (dataset bladder) and the **survivalBIV** package (see Section 4.5 in Chapter 4).

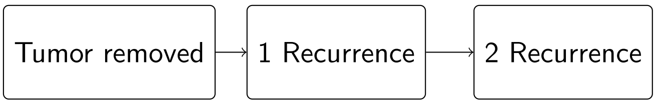

Figure 1.5: Progressive three-state model for Bladder cancer study

Colon Cancer Data

Due to large number of people affected by cancer of Colon, there is much demand for information on this disease. In a large percentage of the patients, the diagnosis is made at a sufficiently early stage when all apparent disease tissue can be surgically removed. Unfortunately, some of these patients have residual cancer, which leads to recurrence of disease and death (in some cases). Cancer patients who have experienced a recurrence are known to be at a substantially higher risk of mortality. In the context of multi-state modelling, we may consider the "recurrence" as an associated state of risk, and then use the progressive illness-death model with states "Alive and disease-free", "Alive with Recurrence" and "Dead". In Chapter 4 we analyzed data from one of the first successful trials of adjuvant chemotherapy for Colon cancer. In this trial one main goal is to compare three therapies (Levamisole, a low-toxicity compound; 5-FU is a moderately toxic chemotherapy agent; and Observation). For each individual, an indicator of its final vital status (censored or not), the survival times (time to recurrence, time to death) from the entry of the patient in the study (in days), and a vector of covariates including rx (treatment), sex, age, among others. In the original format of the database, there are two records per person, one for recurrence and one for death (see variable `etype`). From the total of 929 patients, 468 developed recurrence and among these 414 died. In the database, there are 7 individuals that relapsed and their time from recurrence to death (i.e. zero). This observations were eliminate from our study. This database is available on the **survival** package of the `R` statistical software; We used this dataset to show the available `R` packages for the multi-state models in Chapter 4.

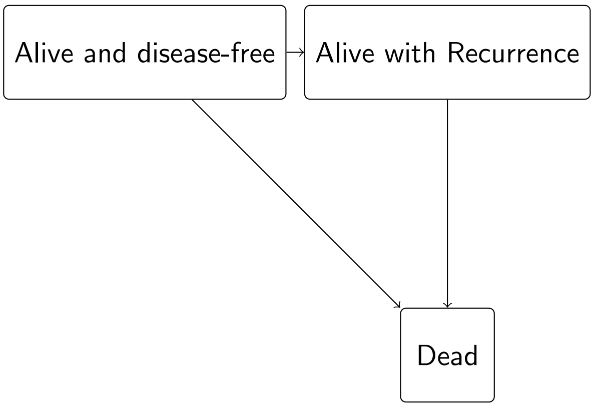

Figure 1.6: Illness-death model for Colon cancer data

German Breast Cancer Data

The German Breast cancer study is available as part of the book by Hosmer and Lemeshow (2008). In this dataset we have a total of 686 woman with primary node positive Breast cancer that were recruited in the period between 1984 and 1989. From this total 299 developed a recurrence and among these 171 died. For each patient, the two gap times (time to recurrence and time from recurrence to death) and the corresponding indicator status is recorded. A vector of covariates including age at acceptance were also recorded. The covariate recurrence is the only time-dependent covariate, while the other covariates included are fixed. Recurrence can be considered as an intermediate transient state and modelled using a progressive three-state model with states "Alive and disease-free", "Alive with Recurrence" and "Dead". This dataset was used to illustrate the methods developed in Chapter 2 (estimation of the bivariate distribution under right censoring and conditional bivariate distribution given a quantitative covariate).

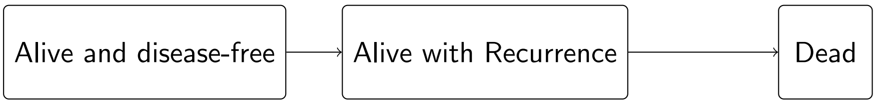

Figure 1.7: Progressive three-state model for German Breast cancer data

Stanford Heart Transplant Data

In Chapter 3 we analyze data from the Stanford Heart Transplant study. This well-known and widely used dataset is used in Chapter 3 to illustrate the new (semi-parametric) estimator for the transition probabilities based on presmoothing. This dataset is available as part of the *R* survival package, and it is also reported in the book by Crowley and Hu (1977). This study covers the period from October 1967 to April 1974. It includes 103 patients enrolled in the Stanford Heart transplant program, from which 69 received a heart transplant and among these 45 died. The total number of deaths was 75 (30 without transplantation); the remaining 28 patients contributed with censored survival times. The transplant can be considered as an associated state of risk, and we may use the so-called illness-death model with states "Own Heart", "New Heart" (or transplant) and "Dead".

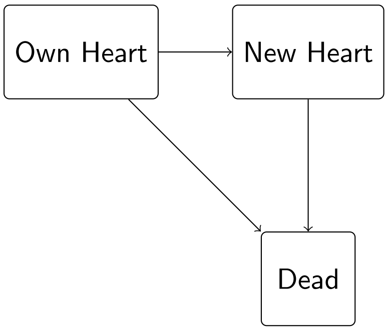

Figure 1.8: Illness-death model for Stanford Heart Transplant data

1.5 Outline of the thesis

The thesis is organized as follows. Chapter 2 is devoted to the study of the bivariate distribution function for censored gap times. In this section, we review some recent proposals and introduce new estimators that account with the influence of covariates (conditional transition probabilities) (Section 2.2). Simulation studies are performed to investigate the finite sample proprieties of the estimators (Section 2.3). The real data illustration with the German Breast cancer data example is given in Section 2.4.

In Chapter 3 we propose a modification of the Aalen-Johansen estimator (typically assumed in for estimating the transition probabilities in Markov processes) in the illness-death model based on presmoothing. The idea of presmoothing involves replacing the censoring indicators by some smooth fit before the Kaplan-Meier formula is applied. This preliminary smoothing may be based on a certain parametric family such as the logistic (thus leading to a semiparametric estimator). The properties of the estimator are investigated both theoretically (Section 3.5) and through simulations (Section 3.3). Section 3.4 is devoted to the illustration of the proposed methods using the Stanford heart transplant data example.

Chapter 4 focus to the available R packages for the analysis of multi-state survival data and describes the R **survivalBIV** package. Section 4.2 contains a detailed description of the existing software for implementing multi-state models using R. The **survivalBIV** package is described in Section 4.3. Section 4.4 is devoted to the generation of bivariate survival data. A real data illustration with the Bladder cancer data is given in Section 4.5.

We conclude with some final remarks and possible directions for future research in Chapter 5.

Chapter 2

Estimators for censored gap times

2.1 Introduction

In many medical studies individuals can experience several events across a follow-up study. The events of concern can be of the same nature (e.g., cancer patients can experience recurrent disease episodes) or represent different states in the disease process (e.g., alive and disease-free, alive with recurrence and dead). If the events are of the same nature, this is usually referred as recurrent events (Cook and Lawless, 2007), whereas if they represent different states they are usually modelled through their intensity functions (Andersen et al., 1993; Meira-Machado et al., 2009). In both cases, it is important to study the inter-event times, also known as the gap times. In these studies, often some events are not completely experienced before the end of a study. This leads to (right) censored gap times and conventional methods are usually no longer applicable. Several issues of interest arise from censored gap times: (a) bivariate distribution function; (b) marginal distribution; (c) conditional distribution of the second gap time; and (d) correlation between gap times. The aim of this chapter is therefore two fold. Firstly we focus on the estimation of the bivariate distribution function under right censoring. Secondly, two competing nonparametric regression estimators of the conditional bivariate distribution are also introduced.

The estimation of the bivariate distribution function is a issue that has received much attention recently. Among others, it was investigated by Campbell (1981), Tsai et al. (1986), Burke (1988), Dabrowska (1988), Prentice and Cai (1992), Lin and Ying (1993), Van Der Laan (1996), Wang and Wells (1997), Lin et al. (1999), Akritas and Keilegom (2003), Prentice et al. (2004). More recent contributions were made by de Uña-Álvarez and Meira-Machado (2008) and de Uña-Álvarez and Amorim (2011).

In this chapter we present four methods (estimators) for the bivariate distribution function of the gap times. One simple estimator is based on the conditional probability and Kaplan-Meier survival function. This estimator is related to that proposed in Lin et al. (1999) and with estimators proposed by de Uña-Álvarez (de Uña-Álvarez and Meira-Machado, 2008; de Uña-Álvarez and Amorim, 2011) since all use (in different ways) the Kaplan-Meier estimator (Kaplan and Meier, 1958). The estimator proposed by Lin in 1999 uses inverse probability of censoring weighted based on the Kaplan-Meier estimator. On the other hand, the idea behind both estimators proposed by de Uña-Álvarez is the use of the Kaplan-Meier estimator pertaining to the distribution of the total time to weight the bivariate data. The difference between these two methods is that the more recent paper uses a presmoothed version of the Kaplan-Meier estimator (Dikta, 1998). Without smoothing, the estimator described in de Uña-Álvarez and Amorim (2011) reduces to that in the de Uña-Álvarez and Meira-Machado (2008).

The estimator proposed by Lin in 1999 uses weights based on inverse probability to estimate the bivariate distribution function. However, the proposed estimator may induce negative probability mass and therefore do not satisfy the monotonicity requirements of a distribution function. The estimators proposed by de Uña-Álvarez and Meira-Machado (2008) make use of the Kaplan-Meier estimator pertaining to the distribution of the total time to weight the bivariate data. This estimator provides a monotone distribution function and can also be written as a sum of weights based on inverse probability of censoring. Most of the proposed estimators are based on the assumption that the vector of gap times and censoring times are independent. In a number of practical situations, this is however not an acceptable assumption. In this chapter two competing nonparametric regression estimators of the conditional bivariate distribution are also introduced. These estimators are based on inverse probability of censoring weighting and will account for the influence of covariates while handling for dependent censoring. In both estimators, local smoothing is done by introducing regression kernel weights that are either based on a local constant (i.e. Nadaraya-Watson) or a local linear regression.

Our methods are motivated by data on Breast cancer which is available in the book by Hosmer and Lemeshow (2008). These data can be viewed as arising from a progressive three-state model where "alive with recurrence" can be modelled as an intermediate state and "dead" the absorbing dead state (see Figure 1.7). We will use this data set to illustrate the estimators for the bivariate distribution function. In addition, we will use the two competing estimators of the conditional bivariate distribution to study the effect of age on the bivariate distribution function. Extensive simulation studies are provided to compare the performance of all methods in different scenarios.

The chapter is organized as follows. In the next section, we introduce the formal notations and the estimators. Section 2.3 describes the simulation setup and the finding of a number of simulation experiments. In Section 2.4 we use data from the German Breast cancer study to illustrate the proposed methods.

2.2 Estimators

2.2.1 Notation

The topic of this chapter is encountered in many medical studies (e.g., recurrences in cancer studies; relapse episodes in schizophrenic disease) where the first gap time is the time from some initial stage of the disease (e.g. healthy, disease-free, etc), to some intermediate stage of the disease or event, and finally the second gap time is the time from that state to a subsequent episode (recurrence or relapse). This means that one individual cannot experience the final event of interest without experiencing the intermediate event.

Consider n independent and identically distributed pairs of successive failure (gap) times (T_{1i}, T_{2i}), $1 \leq i \leq n$ with joint distribution function $F_{12}(x, y)$. These pairs of gap times are subject to univariate right-censoring at times C_i with distribution function $G(t) = P(C \leq t)$ and which is usually assumed to be independent of (T_{1i}, T_{2i}). Because of this we only observe $(\widetilde{T}_{1i}, \widetilde{T}_{2i}, \Delta_{1i}, \Delta_{2i})$ where $\widetilde{T}_{1i} = \min(T_{1i}, C_i)$, $\Delta_{1i} = I(T_{1i} \leq C_i)$, $\widetilde{T}_{2i} = \min(T_{2i}, C_{2i})$, $\Delta_{2i} = I(T_{2i} \leq C_{2i})$ where $C_{2i} = (C_i - T_{1i})I(T_{1i} \leq C_i)$. Let $T = T_1 + T_2$ be the total time and introduce $\widetilde{T} = \min(T, C)$. If the censoring time is assumed to be independent of the process, the marginal distribution of the first gap time T_1, say F_1 may be consistently estimated by the Kaplan-Meier estimator based on the pairs $(\widetilde{T}_{1i}, \Delta_{1i})$'s. Similarly, the distribution of the total time may be consistently estimated by the Kaplan-Meier estimator based on the $(\widetilde{T}_i, \Delta_{2i})$'s. The Kaplan-Meier estimator of the second gap time cannot be used here, since the independence of T_2 and C_2 can not be assumed.

Below we will introduce new estimators for the bivariate distribution assuming that C is independent of (T_1, T) given Z, where Z denotes a quantitative covariate. Note that this assumption does not exclude the possibility of dependent censoring (i.e., C conditionally dependent on (T_1, T)).

2.2.2 Bivariate Distribution Function

Several methods have been proposed to estimate the bivariate distribution function $F_{12}(x, y) = P(T_1 \leq x, T_2 \leq y)$ in the presence of right censoring. Almost all using the Kaplan-Meier estimator of survival. Some related problems such as estimation of the marginal distribution of the second gap time will also be discussed.

Conditional Kaplan-Meier estimator

A simple estimator for the bivariate distribution function of the gap times is based on Bayes' theorem and Kaplan-Meier survival function (conditional Kaplan-Meier, CKM). Since $F_{12}(x, y) = P(T_1 \leq x, T_2 \leq y) = P(T_2 \leq y|T_1 \leq x)P(T_1 \leq x)$ one simple estimator for the bivariate distribution is given by

$$\widehat{F}_{12}(x, y) = \widehat{F}_1(x)\widehat{F}_{KM}(y|T_1 \leq x, \Delta_1 = 1) \tag{2.1}$$

where $\widehat{F}_1(x)$ is the Kaplan-Meier product-limit estimator based on the pairs $(\widetilde{T}_{1i}, \Delta_{1i})$'s and $\widehat{F}_{KM}(y|T_1 \leq x, \Delta_1 = 1)$ is the Kaplan-Meier estimator based on the pairs $(\widetilde{T}_{2i}, \Delta_{2i})$'s. The $\widehat{F}_{KM}(y|T_1 \leq x, \Delta_1 = 1)$ is the conditional distribution function for the subset of $T_1 \leq x$ and $\Delta_1 = 1$ (the Kaplan-Meier estimator based on the pairs $(\widetilde{T}_{2i}, \Delta_{2i})$'s such that $\widetilde{T}_{1i} \leq x$ and $\Delta_{1i} = 1$).

Since the independence between T_2 and C_2 can not be assumed in general, the CKM estimator may be inconsistent. The consistency of this estimator can only be ensured when $P(\Delta_1|T_1 \leq x) = 1$. These features can be seen in our simulation results presented in Section 2.3.1. Even so, this estimator still can be used in variety of statistical problems, for example, to the study the relation between a variable of interest T and some covariate.

Kaplan-Meier weighted estimator

Another simple estimator was recently proposed by de Uña-Álvarez and Meira-Machado (2008). The idea behind their estimator is to use the Kaplan-Meier estimator pertaining to the distribution of the total time to weight the bivariate data. The proposed estimator (Kaplan-Meier Weighted Estimator, KMW) is given by

$$\widetilde{F}_{12}(x, y) = \sum_{i=1}^{n} W_i I(\widetilde{T}_{1i} \leq x, \widetilde{T}_{2i} \leq y) \tag{2.2}$$

where $W_i = \frac{\Delta_{2i}}{n-R_i+1} \prod_{j=1}^{i-1} \left[1 - \frac{\Delta_{2j}}{n-R_j+1}\right]$ is the Kaplan-Meier weight attached to $\widetilde{T}_i$ when estimating the marginal distribution of T from $(\widetilde{T}_i, \Delta_{2i})$'s, and for which the ranks of the censored $\widetilde{T}_i$'s, R_i, are higher than those for uncensored values in the case of ties.

An alternative estimator can be proposed using inverse probability of censoring weights:

$$\widetilde{F}_{12}(x,y) = \frac{1}{n}\sum_{i=1}^{n} \frac{I\left(\widetilde{T}_{1i} \leq x, \widetilde{T}_{2i} \leq y\right)\Delta_{2i}}{1-\widehat{G}(\widetilde{T}_i)}. \tag{2.3}$$

where $\hat{G}(.)$ is the Kaplan-Meier estimator of the censoring.

Kaplan-Meier presmooth weighted estimator

Recently, de Uña-Álvarez and Amorim (2011) propose a modification of estimator (2.2) based on presmoothing (Dikta, 1998), which allows for a variance reduction in the presence of censoring. Basically, this method uses a presmoothed version of the Kaplan-Meier estimator (see e.g. Dikta (1998) and references therein) pertaining to the distribution of the total time to weight the bivariate data. This is obtained by replacing the censoring indicator variables in the expression of the Kaplan-Meier weights by a smooth fit of a binary regression. This estimator (Kaplan-Meier Presmooth Weighted Estimator, KMPW) is expressed as

$$\breve{F}_{12}(x,y) = \sum_{i=1}^{n} W_i^{\star} I(\widetilde{T}_{1i} \leq x, \widetilde{T}_{2i} \leq y) \tag{2.4}$$

where $W_i^{\star} = \frac{m(\widetilde{T}_{1i},\widetilde{T}_i)}{n-R_i+1}\prod_{j=1}^{i-1}\left[1-\frac{m(\widetilde{T}_{1j},\widetilde{T}_j)}{n-R_j+1}\right]$ are the presmoothed Kaplan-Meier weights. Here, $m(x,y) = P(\Delta_2 = 1|\widetilde{T}_1 = x, \widetilde{T} = y, \Delta_1 = 1)$, belongs to a parametric (smooth) family of binary regression curves, e.g. logistic. In practice, we assume that $m(x,y) = m(x,y;\beta)$ where β is a vector of parameters which typically will be computed by maximizing the conditional likelihood of the Δ_2's given $(\widetilde{T}_1, \widetilde{T}_2)$ for those with $\Delta_1 = 1$.

Note that, unlike (2.2), the KMPW can attach positive mass to pair of gap times with censored second gap time. However, both estimators (2.2) and (2.4) attach a zero weight to pairs of gap times with censored first gap time. In the limit case of no presmoothing, the estimator (2.4) reduces to (2.2). Conditions under which both estimators are consistent are fully discussed in papers by de Uña-Álvarez and Meira-Machado (2008) and de Uña-Álvarez and Amorim (2011). In the latter paper the authors compare the performance of the presmoothed (semiparametric) estimator with the purely nonparametric estimator (without presmoothing) and concluded that the presmoothed estimator improves efficiency in the multivariate setup of gap times.

Inverse probability of censoring weighted estimator

Another estimator for the bivariate distribution function was proposed by Lin et al. (1999). This estimator is based on inverse probability of censoring weighted (Lin). The rationale behind Lin is that each subject that is observed at time u is representative (on average) of $\frac{1}{G(u)}$ individuals that might have been observed if there was no censoring. Lin's estimator is expressed as

$$\overline{F}_{12}(x,y) = \overline{H}(x,0) - \overline{H}(x,y) \tag{2.5}$$

where

$$\overline{H}(x,y) = \frac{1}{n}\sum_{i=1}^{n} \frac{I(\widetilde{T}_{1i} \leq x, \widetilde{T}_{2i} > y)}{1-\widehat{G}(\widetilde{T}_{1i}+y)}$$

The censoring distribution function G is typically unknown and needs to be replaced by an estimate. This can be obtained by reversing the role of T and C, using a Kaplan-Meier estimate $\hat{G}$ of the censoring distribution function, i.e., using an estimate based on the $(\widetilde{T}_{1i}, 1-\Delta_{1i})$'s (for the first term in the right-hand side of equation (2.5)) or $(\widetilde{T}_{i}, 1-\Delta_{2i})$'s (for the second term in the right-hand side of equation (2.5)). This is the simplest choice and was assumed by Lin et al. (1999). Other procedures for estimation of G are appropriate, for example the approach used in Gerds and Schumacher (2006). Without ties (between event times and censoring times) the two procedures (Lin et al., 1999; Gerds and Schumacher, 2006) provide the same result.

Estimator (2.5) is also written as

$$\overline{F}_{12}(x,y)=\frac{1}{n}\sum_{i=1}^{n}\frac{I\left(\widetilde{T}_{1i}\le x\right)\Delta_{1i}}{1-\widehat{G}(\widetilde{T}_{1i})}-\frac{1}{n}\sum_{i=1}^{n}\frac{I\left(\widetilde{T}_{1i}\le x,\widetilde{T}_{2i}>y\right)}{1-\widehat{G}(\widetilde{T}_{1i}+y)} \tag{2.6}$$

Note that consistency of estimators (2.2), (2.4) and (2.5) is only guaranteed whenever $x+y$ is smaller than the upper bound of the support of the censoring time. As mentioned before, the CKM estimator may be inconsistent in the presence of censoring of the first gap time. In addition, monotonicity of this estimator is not guaranteed. The monotonicity problem can be explained by the fact that, as the conditioning set $T_1\le x$ changes, the redistribution to the right of the probability mass associated with censored observations also changes. In contrast to the other two methods, the estimators based on Kaplan-Meier weights (KMW and KMPW) are monotonic (distribution) functions, in the sense that they attach positive mass to each observation.

Other estimators were proposed to estimate the bivariate distribution function. A valid estimator of the bivariate distribution function, was provided by Van Keilegom (2004) which is based on Akritas (1994). However, this approach has some limitations since some smoothing is required. Recently, alternative estimators for these quantities were also given in Van Keilegom et al. (2011). This methodology assumes that the vector of gap times (T_1,T_2) satisfies the nonparametric location-scale regression model, allowing for the transfer of tail information from lightly censored areas to heavily ones.

One alternative approach is based on the conditional distribution of T_2 given T_1. The expectation $E\left[I(T_1\le x,T_2\le y)\right]$ can be estimated by

$$\widehat{F}^{*}_{12}(x,y)=\widehat{P}(T_1\le x,T_2\le y)=\int_{(0,x)}\widehat{P}(T_2\le y\mid u-h<T_1<u+h)d\hat{F}_1(u). \tag{2.7}$$

where $\hat{F}_1(u)$ is an estimator of the distribution of the first gap time, for example, the Kaplan-Meier estimator based on the pairs $(\widetilde{T}_{1i},\Delta_{1i})$'s, and h is a sequence of positive constants tending to zero as n tends to infinity, called a bandwidth sequence.

Under random right censorship, the nonparametric estimator of the conditional distribution (Beran, 1981) may be used to estimate the bivariate distribution function. This estimator can also be adjusted for dependent censoring following Satten et al. (2001). These authors suggest to estimate the conditional probabilities as follow:

$$\widehat{P}(T_2 \leq y \mid u-h < T_1 < u+h) \;=\; 1 - \prod_{(0,y]}\left[1 - \frac{d\hat{N}(dv, u-h < T_1 < u+h)}{\hat{Y}(v, u-h < T_1 < u+h)}\right]$$

where

$$\hat{N}(v, u-h < T_1 < u+h) \;=\; \sum_{i=1}^{n} \frac{I(\widetilde{T}_{2i} \leq v, u-h < \widetilde{T}_{1i} < u+h)\Delta_{2i}}{1-\widehat{G}(\widetilde{T}_i)}$$

and (method 1-condBIV_1)

$$\hat{Y}(v, u-h < T_1 < u+h) \;=\; \sum_{i=1}^{n} \frac{I(\widetilde{T}_{2i} \geq v, u-h < \widetilde{T}_{1i} < u+h)\Delta_{2i}}{1-\widehat{G}(\widetilde{T}_i)}$$

alternatively, $\hat{Y}$ can be estimated (method 2-condBIV_2) by

$$\hat{Y}(v, u-h < T_1 < u+h) \;=\; \sum_{i=1}^{n} \frac{I(\widetilde{T}_{2i} \geq v, u-h < \widetilde{T}_{1i} < u+h)}{1-\widehat{G}(\widetilde{T}_{1i}+v)}$$

This method cannot be directly applied for real and simulated data without considering the problem of the choice of an optimal bandwidth. A large number of methods for automatic bandwidth selection exist being the least squares cross-validation one of the most common approach. However, proposed methods are still scarce to deal the problem of censoring. Below we suggest a method for the bandwidth selection. We propose the following procedure to obtain the bandwidth h used to obtain the Beran-type estimator:

Step 1. First for $b = 1$ to B (e.g. B=1000) simulate the random sample

$$S^b = \left\{ \widetilde{T}_{1i}^{\bullet b}, \widetilde{T}_{2i}^{\bullet b}, \Delta_{1i}^{\bullet b}, \Delta_{2i}^{\bullet b} \right\}_{i=1}^{n}$$

by randomly sampling the n items from the original data set $\left\{ \widetilde{T}_{1i}, \widetilde{T}_{2i}, \Delta_{1i}, \Delta_{2i} \right\}_{i=1}^{n}$ with replacement.

Step 2. Then, the bandwidth h is automatically selected by minimizing the following error criterion:

$$C(h) = \sum\nolimits_{b=1}^{B} \sum\nolimits_{i=1}^{n} \left(\widehat{F}_{12}(\widetilde{T}_{1i}^{\bullet b}, \widetilde{T}_{2i}^{\bullet b}) - \widehat{F}_{12}^{*}(\widetilde{T}_{1i}^{\bullet b}, \widetilde{T}_{2i}^{\bullet b}) \right)^2,$$

where $\widehat{F}_{12}$ is the estimate obtained from the sample S^b, using estimator (2.2) and $\widehat{F}_{12}^{*}$ is the estimate obtained from the Beran-type estimator (2.7) based on the same sample.

From (2.1), (2.2), (2.4) and (2.5) we may obtain an estimator for the marginal distribution of the second gap time, $F_2(y) = P(T_2 \leq y)$, namely

$$\widehat{F}_2(y) = \widehat{F}_{12}(+\infty, y) = \widehat{F}_1(+\infty)\widehat{F}_{KM}(y|\Delta_1 = 1) \tag{2.8}$$

$$\widetilde{F}_2(y) = \widetilde{F}_{12}(+\infty, y) = \sum_{i=1}^{n} W_i I(\widetilde{T}_{2i} \leq y) \tag{2.9}$$

Note that if $\widehat{F}_1(+\infty) = 1$, then (2.8) is the Kaplan-Meier estimator based on $(\widetilde{T}_{2i}, \Delta_{2i})$'s such that $\Delta_1 = 1$ (i.e., for which the first gap time is uncensored). Estimator (2.9) is different because the Kaplan-Meier weights W_i in this estimator are based on the $\widetilde{T}_i$-ranks rather than on the $\widetilde{T}_{2i}$-ranks. In fact, since T_2 and C_2 are expected to be dependent, the ordinary Kaplan-Meier estimator of F_2 (estimator (2.8)) will be generally inconsistent. The corresponding estimator for (2.4) is obtained using the same ideas as for (2.9) by replacing the weights W_i by the presmoothed Kaplan-Meier weight $W_i^{\star}$ previously defined. Similarly, from Lin's estimator (2.5) one can obtain an estimator for the marginal distribution of the second gap time. Again, note that such estimator does not guarantee monotonicity.

Below we will provide two competing nonparametric regression estimators which are adapted from estimators (2.2) and (2.6) to handle the influence of covariates on the bivariate distribution function.

2.2.3 Conditional Bivariate Distribution Function

In this section we will introduce two estimators for the conditional distribution function, $F_{12}(x, y \mid Z)$ where Z denotes a quantitative covariate. Both methods are based on inverse probability of censoring weighting. This can be done via estimating the general conditional expectation of type $E\left[\varphi\left(T_1, T_2\right) \mid Z = z\right]$. To estimate these quantities we may use kernel smoothing techniques by calculating a local average of the $\varphi(T_1, T_2)$. This can be written as $\sum_{i=1}^n W_{1i}(x)\varphi(T_{1i}, T_{2i})$ where $W_{1i}(x)$ is a weight function which can be estimated using Nadaraya-Watson (Nadaraya, 1965; Watson, 1964) or local linear estimators. In our case, we have to estimate $E\left[\varphi_{x,y}\left(T_1, T_2\right) \mid Z = z\right]$, $E\left[\widetilde{\varphi}_{x,y}\left(T_1, T_2\right) \mid Z = z\right]$ and $E\left[\xi_x\left(T_1\right) \mid Z = z\right]$, where $\varphi_{x,y}\left(u, v\right) = I(u \leq x, v > y)$, $\widetilde{\varphi}_{x,y}(u, v) = I(u \leq x, v \leq y)$ and $\xi_x(u) = I(u \leq x)$.

To estimate these quantities, we need to estimate the d.f. of C given Z, G_Z. Let G_{Z_i} denote the conditional distribution function of $C \mid Z = Z_i$ and let $\widehat{G}_{Z_i}$ stand for its estimator. The estimation of the conditional distribution function of the response, given the covariate under random censoring has been considered in many papers. This topic was introduced by Beran (1981) and was further studied by several authors (see e.g. papers by Dabrowska (1987, 1988, 1989a,b); Akritas (1994); Van Keilegom et al. (2001) and Van Keilegom (2004)). Their proposals can also be used to estimate the conditional distribution function of $C \mid Z$, say $\widehat{G}_Z$. This can be done using the Kaplan-Meier estimator introduced by Beran (1981),

$$1 - \widehat{G}_z(y) = \prod_{\tilde{T}_i \leq y, \Delta_{2i} = 0} \left[1 - \frac{W_{0i}(z, a_n)}{\sum_{j=1}^n I(\tilde{T}_j \geq \tilde{T}_i) W_{0j}(z, a_n)}\right] \tag{2.10}$$

with

$$W_{0i}(z,a_n)=\frac{K_0\left((z-Z_i)/a_n\right)}{\sum_{j=1}^{n}K_0\left((z-Z_j)/a_n\right)}$$

where $W_{0i}(z,a_n)$ are the Nadaraya-Watson weights (NW), K_0 is a known probability density function (kernel) and a_n a sequence of bandwidths. This estimator reduces to the so-known Kaplan-Meier (Kaplan and Meier, 1958) estimator when all weights are equal. To cope with left-truncated data one can also use the estimator of the conditional distribution, proposed by Iglesias Pérez and González Manteiga (2003).

In order to introduce our estimators note that, assuming that the support of conditional distribution of T is contained in that of $C|Z$, we have $E[\varphi(T_1,T_2)\mid Z]=E[\varphi(\widetilde{T}_1,\widetilde{T}_2)\Delta/1-G_Z(\widetilde{T})\mid Z)]$. We propose to plug-in Beran's estimator $\hat{G}_Z$ and use the local linear estimator (LL) or a Nadaraya-Watson estimator (NW), to introduce Inverse Probability Censoring Weighted estimators (IPCW) for the conditional bivariate distribution:

$$\widehat{F}_{12}(z;x,y)=\sum_{i=1}^{n}W_{1i}(z,b_n)\frac{\varphi_{x,y}(\widetilde{T}_{1i},\widetilde{T}_{2i})\Delta_{2i}}{1-\widehat{G}_{Z_i}(\widetilde{T}_i)}=\sum_{i=1}^{n}W_{1i}(z,b_n)\frac{I(\widetilde{T}_{1i}\leq x,\widetilde{T}_{2i}\leq y)\Delta_{2i}}{1-\widehat{G}_{Z_i}(\widetilde{T}_i)}$$

where $W_{1i}(z,b_n)$ are Nadaraya-Watson weights or local linear weights,

$$W_{1i}(z,b_n)=\frac{K_1\left((z-Z_i)/b_n\right)\left[S_{n,2}(z)-(z-Z_i)S_{n,1}(z)\right]}{\sum_{j=1}^{n}K_1\left((z-Z_j)/b_n\right)\left[S_{n,2}-(z-Z_j)S_{n,1}(z)\right]}$$

with $S_{n,l}=\sum_{i=1}^{n}K_1((z-Z_i)/b_n)(z-Z_i)^l$, $l=0,1,2$ and where b_n is a sequence of bandwidths and K_1 is a known kernel function.

Note that since $E[\varphi_{x,y}(T_1,T_2)\mid Z]=E[\xi_x(T_1)\mid Z]-E[\widetilde{\varphi}_{x,y}(T_1,T_2)\mid Z]$. Thus, $E[\varphi_{x,y}(T_1,T_2)\mid Z]=E[I(T_1\leq x)\mid Z]-E[I(T_1\leq x,T_2>y)\mid Z]=E[I(\tilde{T}_1\leq x)I(C>T_1)/1-G_Z^0(\tilde{T}_1)\mid Z]-E[I(\tilde{T}_1\leq x,\tilde{T}_2>y)I(C>T_1+y)/1-G_Z(\tilde{T}_1+y)\mid Z]$. Then, alternative estimator can be given for the conditional probabilities (Lin et al., 1999). In this case LIN-based estimators are given by

$$\begin{aligned}\widetilde{F}_{12}(z;x,y) &= \sum_{i=1}^{n} W_{1i}(z,b_n)\frac{\bar{\varphi}_x(\widetilde{T}_1)\Delta_{1i}}{1-\widehat{G}^0_{Z_i}(\widetilde{T}_{1i})} - \sum_{i=1}^{n} W_{1i}(z,b_n)\frac{\widetilde{\varphi}_{x,y}(\widetilde{T}_1,\widetilde{T}_2)}{1-\widehat{G}_{Z_i}(\widetilde{T}_{1i}+y)}\\ &= \sum_{i=1}^{n} W_{1i}(z,b_n)\frac{I(\widetilde{T}_{1i}\leq x)\Delta_{1i}}{1-\widehat{G}^0_{Z_i}(\widetilde{T}_{1i})} - \sum_{i=1}^{n} W_{1i}(z,b_n)\frac{I(\widetilde{T}_{1i}\leq x,\widetilde{T}_{2i}>y)}{1-\widehat{G}_{Z_i}(\widetilde{T}_{1i}+y)}\end{aligned}$$

where G^0_Z stands for an estimator of the conditional distribution $C \mid Z = Z_i$, for example, the based on the $(\widetilde{T}_{1i}, 1-\Delta_{1i})$'s.

In the Section 2.3.2 we will study the finite sample performance of IPCW and LIN-based estimators.

2.3 Simulation Studies

2.3.1 Bivariate Distribution Function

In this section, we compare by simulations the four estimators 2.1 to 2.5, for the bivariate distribution function. We consider two simulated scenarios, the first scenario is the same as that described in Lin's paper (see their Section 3). In this scenario, the gap times were generated from Gumbel's bivariate distribution function, the so-called Fairlie-Gumbel-Morgenstern families of bivariate cdf's

$$F(x,y) = F_1(x)F_2(y)[1+\delta(1-F_1(x))(1-F_2(y))]$$

where $|\delta| \leqslant 1$ for a bivariate density to exist. The marginal distributions, F_1 and F_2 are exponential with rate parameter 1. The case of independence is obtained for $\delta = 0$ while the maximum of correlation (between T_1 and T_2) for the bivariate exponential distribution is obtained for $\delta = 1$ with bound equal to 0.25. As in Lin's paper, for this scenario, the uniform censoring time C was generated according to models $U[0,4]$ and $U[0,3]$. The first model ($U[0,4]$) resulted in 25% of censoring of the first gap time, and 46% of censoring in the second gap time. In the second model ($U[0,3]$) we have censoring levels of 32% and 60% for the corresponding gap times. One limitation of the so-called Fairlie-Gumbel-Morgenstern families of bivariate cdf's, is that the correlation of T_1 and T_2 can never exceed $1/3$ (0.25 in the bivariate exponential distribution). One potential category of bivariate distributions is the family of bivariate weibull distributions. This distribution clearly allows for a larger correlation between the two gap times, making it superior than the bivariate exponential. For this reason, in our second scenario we consider the bivariate weibull distribution with two-parameter marginal distributions. Its survival function is given by

$$S(x,y) = P(T_1 > x, T_2 > y) = exp\left[- \left[\left(\frac{x}{\theta_1}\right)^{\frac{\beta_1}{\delta}} + \left(\frac{y}{\theta_2}\right)^{\frac{\beta_2}{\delta}} \right]^{\delta} \right]$$

where $0 < \delta \leqslant 1$, and each marginal distribution has shape parameter β_i and a scale parameter θ_i, $i = 1, 2$. The correlation between the two gap times can be obtained though is a complicated function of the shape and scale parameters and of δ. For our simulation we consider $\delta = 0.6$, $\theta_1 = \theta_2 = 7$ and shape parameters $\beta_1 = \beta_2 = 2$, for which we obtained about 54% of correlation.

For each scenario we have considered two sample sizes, $n = 50$ and $n = 100$ and for each simulation, 1000 samples were generated. For each setting we computed the mean and standard deviations for the bivariate estimators at pairs of time points (x, y), where x and y takes values corresponding to: marginal survival probabilities of 0.8, 0.6, 0.4, 0.2 and 0.05 for the bivariate exponential scenario; and to marginal survival probabilities of 0.8, 0.6, 0.4, 0.2 and 0.1 for the bivariate weibull scenario. The true values of $F_{12}(x, y)$ are reported in Tables 2.1 and 2.2.

Table 2.1: True values of the bivariate exponential distribution of the gap times.

	$\delta = 0$					$\delta = 1$				
y	0.2231	0.5108	0.9163	1.6094	2.9990	0.2231	0.5108	0.9163	1.6094	2.9990
x										
0.2231	0.0400	0.0800	0.1200	0.1600	0.1900	0.0656	0.1184	0.1584	0.1856	0.1976
0.5108	0.0800	0.1600	0.2400	0.3200	0.3801	0.1184	0.2176	0.2976	0.3584	0.3914
0.9163	0.1200	0.2400	0.3600	0.4800	0.5701	0.1584	0.2976	0.4176	0.5184	0.5815
1.6094	0.1600	0.3200	0.4800	0.6400	0.7601	0.1856	0.3584	0.5184	0.6656	0.7677
2.9990	0.1900	0.3801	0.5701	0.7601	0.9028	0.1976	0.3914	0.5815	0.7677	0.9051

Table 2.2: True values of the bivariate weibull distribution of the gap times.

y	3.3067	5.0030	6.7006	8.8805	10.622
x					
3.3067	0.1130	0.1574	0.1800	0.1930	0.1972
5.0030	0.1574	0.2610	0.3294	0.3741	0.3895
6.7006	0.1800	0.3294	0.4494	0.5406	0.5751
8.8805	0.1930	0.3741	0.5406	0.6872	0.7500
10.622	0.1972	0.3895	0.5751	0.7500	0.8305

Let $\hat{F}_{12}(x, y)$ denote the estimated bivariate distribution, for each (x, y) we computed estimates of the bias as: $bias(\hat{F}_{12}(x, y)) = F_{12}(x, y) - \overline{\hat{F}_{12}(x, y)}$

Results reveal that, in general, the bias increases for higher censoring levels ($C \sim U[0, 3]$) and decreases with the increasing of the sample size. Tables 2.3 to 2.7 report the mean estimate along with the corresponding standard deviation for estimators 2.1 to 2.5.

As it can be seen, in all estimators the bias of the bivariate distribution achieved reasonable levels. In all cases the variance increases at the right tail of the bivariate distribution, where the censoring effects are stronger. From these tables we can see that

a) the CKM estimator has larger bias for higher values of x, the first gap time,

Table 2.3: Mean values and standard deviation of $\hat{F}_{12}(x, y)$ for the bivariate exponential scenario. Sample size of $n = 50$, uniform censoring $C \sim U[0, 3]$.

			$\delta = 0$					$\delta = 1$				
		y	0.2231	0.5108	0.9163	1.6094	2.9990	0.2231	0.5108	0.9163	1.6094	2.9990
	x											
	0.2231		0.0674	0.1201	0.1613	0.1788	0.1870	0.0652	0.1151	0.1584	0.1794	0.1897
			(0.0374)	(0.0468)	(0.0555)	(0.0595)	(0.0590)	(0.0370)	(0.0477)	(0.0535)	(0.0568)	(0.0598)
	0.5108		0.1191	0.2191	0.2962	0.3556	0.3727	0.1192	0.2207	0.2994	0.3661	0.3742
			(0.0499)	(0.0604)	(0.0679)	(0.0781)	(0.0805)	(0.0461)	(0.0615)	(0.0690)	(0.0763)	(0.0798)
CKM	0.9163		0.1592	0.3014	0.4228	0.5175	0.5557	0.1611	0.3051	0.4237	0.5211	0.5562
			(0.0554)	(0.0714)	(0.0798)	(0.0855)	(0.0891)	(0.0554)	(0.0741)	(0.0828)	(0.0852)	(0.0879)
	1.6094		0.1955	0.3789	0.5356	0.6799	0.7325	0.1954	0.3784	0.5324	0.6812	0.7217
			(0.0649)	(0.0819)	(0.0899)	(0.0937)	(0.0969)	(0.0623)	(0.0824)	(0.0904)	(0.0975)	(0.0996)
	2.9990		0.2186	0.4268	0.6089	0.7815	0.8372	0.2229	0.4198	0.6085	0.7772	0.8350
			(0.0681)	(0.0926)	(0.1033)	(0.1036)	(0.1038)	(0.0746)	(0.0897)	(0.0959)	(0.1031)	(0.1042)
	0.2231		0.0384	0.0802	0.1205	0.1587	0.1966	0.0640	0.1191	0.1571	0.1829	0.1977
			(0.0327)	(0.0457)	(0.0533)	(0.0618)	(0.0585)	(0.0393)	(0.0504)	(0.0558)	(0.0559)	(0.0541)
	0.5108		0.0798	0.1611	0.2404	0.3185	0.3965	0.1180	0.2183	0.2967	0.3604	0.4017
			(0.0472)	(0.0619)	(0.0716)	(0.0843)	(0.0738)	(0.0519)	(0.0697)	(0.0750)	(0.0791)	(0.0724)
Lin	0.9163		0.1225	0.2448	0.3606	0.4833	0.6015	0.1564	0.3146	0.4189	0.5065	0.5945
			(0.0562)	(0.0772)	(0.0925)	(0.1077)	(0.0760)	(0.0588)	(0.0729)	(0.0821)	(0.0941)	(0.0831)
	1.6094		0.1601	0.3247	0.4779	0.6610	0.8003	0.1874	0.3582	0.5287	0.6820	0.7964
			(0.0691)	(0.0901)	(0.1132)	(0.1236)	(0.0722)	(0.0714)	(0.0885)	(0.1169)	(0.1481)	(0.0710)
	2.9990		0.1894	0.3812	0.5792	0.7846	0.9269	0.1997	0.4328	0.6130	0.8103	0.9321
			(0.1069)	(0.1271)	(0.1315)	(0.1318)	(0.0683)	(0.0999)	(0.1264)	(0.1446)	(0.1138)	(0.0637)
	0.2231		0.0400	0.0803	0.1202	0.1612	0.1883	0.0655	0.1191	0.1572	0.1861	0.1938
			(0.0283)	(0.0394)	(0.0523)	(0.0631)	(0.0802)	(0.0367)	(0.0503)	(0.0563)	(0.0645)	(0.0712)
	0.5108		0.0798	0.1609	0.2410	0.3208	0.3698	0.1196	0.2158	0.2975	0.3591	0.3834
			(0.0396)	(0.0564)	(0.0690)	(0.0815)	(0.1018)	(0.0496)	(0.0625)	(0.0741)	(0.0831)	(0.0909)
KMW	0.9163		0.1213	0.2412	0.3600	0.4800	0.5481	0.1575	0.3001	0.4170	0.5195	0.5613
			(0.0520)	(0.0699)	(0.0830)	(0.0970)	(0.1186)	(0.0563)	(0.0795)	(0.0853)	(0.0945)	(0.1121)
	1.6094		0.1597	0.3200	0.4839	0.6331	0.7036	0.1844	0.3594	0.5159	0.6608	0.6972
			(0.0622)	(0.0818)	(0.0964)	(0.1120)	(0.1265)	(0.0624)	(0.0822)	(0.0954)	(0.1136)	(0.1272)
	2.9990		0.1850	0.3637	0.5411	0.7086	0.7708	0.1953	0.3828	0.5610	0.7020	0.7543
			(0.0788)	(0.0985)	(0.1154)	(0.1263)	(0.1328)	(0.0719)	(0.0970)	(0.1109)	(0.1259)	(0.1310)
	0.2231		0.0416	0.0832	0.1232	0.1615	0.1896	0.0657	0.1202	0.1586	0.1847	0.1939
			(0.0283)	(0.0368)	(0.0478)	(0.0574)	(0.0696)	(0.0348)	(0.0468)	(0.0528)	(0.0600)	(0.0654)
	0.5108		0.0826	0.1651	0.2440	0.3185	0.3732	0.1210	0.2175	0.2999	0.3574	0.3845
			(0.0372)	(0.0518)	(0.0647)	(0.0741)	(0.0922)	(0.0474)	(0.0594)	(0.0703)	(0.0792)	(0.0852)
KMPW	0.9163		0.1239	0.2440	0.3581	0.4728	0.5513	0.1589	0.3015	0.4163	0.5157	0.5679
			(0.0477)	(0.0642)	(0.0766)	(0.0899)	(0.1068)	(0.0529)	(0.0743)	(0.0798)	(0.0875)	(0.1005)
	1.6094		0.1613	0.3185	0.4750	0.6247	0.7120	0.1868	0.3577	0.5072	0.6497	0.7038
			(0.0558)	(0.0768)	(0.0883)	(0.1015)	(0.1112)	(0.0581)	(0.0762)	(0.0884)	(0.1024)	(0.1118)
	2.9990		0.1899	0.3726	0.5482	0.7107	0.7920	0.1999	0.3862	0.5565	0.7003	0.7677
			(0.0680)	(0.0926)	(0.1035)	(0.1095)	(0.1080)	(0.0641)	(0.0853)	(0.0968)	(0.1095)	(0.1083)

but in general is one of the estimators with less variance;

b) the KMW estimator has less bias than its smooth version, KMPW. However as expected the later obtained less variance;

c) the KMW and Lin estimator are almost unbiased but the last one obtains higher

Table 2.4: Mean values and standard deviation of $\hat{F}_{12}(x, y)$ for the bivariate exponential scenario. Sample size of $n = 100$, uniform censoring $C \sim U[0, 3]$.

			$\delta = 0$					$\delta = 1$				
		y	0.2231	0.5108	0.9163	1.6094	2.9990	0.2231	0.5108	0.9163	1.6094	2.9990
	x											
	0.2231		0.0652	0.1194	0.1565	0.1845	0.1927	0.0652	0.1175	0.1585	0.1854	0.1931
			(0.0253)	(0.0360)	(0.0389)	(0.0433)	(0.0401)	(0.0258)	(0.0350)	(0.0387)	(0.0419)	(0.0401)
	0.5108		0.1203	0.2186	0.2993	0.3596	0.3783	0.1194	0.2163	0.3002	0.3592	0.3808
			(0.0349)	(0.0436)	(0.0500)	(0.0522)	(0.0536)	(0.0346)	(0.0425)	(0.0503)	(0.0557)	(0.0544)
CKM	0.9163		0.1635	0.2996	0.4259	0.5247	0.5587	0.1639	0.3023	0.4217	0.5222	0.5595
			(0.0403)	(0.0522)	(0.0603)	(0.0613)	(0.0639)	(0.0399)	(0.0512)	(0.0567)	(0.0626)	(0.0640)
	1.6094		0.1969	0.3764	0.5369	0.6776	0.7369	0.1985	0.3786	0.5405	0.6817	0.7419
			(0.0467)	(0.0582)	(0.0638)	(0.0702)	(0.0738)	(0.0469)	(0.0613)	(0.0655)	(0.0682)	(0.0706)
	2.9990		0.2187	0.4291	0.6180	0.7897	0.8644	0.2243	0.4275	0.6169	0.7907	0.8678
			(0.0476)	(0.0641)	(0.0694)	(0.0758)	(0.0802)	(0.0514)	(0.0637)	(0.0713)	(0.0781)	(0.0799)
	0.2231		0.0405	0.0783	0.1223	0.1603	0.1979	0.0709	0.1174	0.1585	0.1830	0.1961
			(0.0235)	(0.0294)	(0.0370)	(0.0431)	(0.0422)	(0.0276)	(0.0369)	(0.0420)	(0.0417)	(0.0417)
	0.5108		0.0811	0.1576	0.2369	0.3222	0.3994	0.1181	0.2178	0.3090	0.3575	0.4031
			(0.0317)	(0.0447)	(0.0503)	(0.0613)	(0.0515)	(0.0338)	(0.0444)	(0.0583)	(0.0545)	(0.0490)
Lin	0.9163		0.1201	0.2404	0.3592	0.4816	0.5993	0.1570	0.2981	0.4205	0.5237	0.5951
			(0.0401)	(0.0512)	(0.0624)	(0.0724)	(0.0568)	(0.0417)	(0.0579)	(0.0655)	(0.0741)	(0.0516)
	1.6094		0.1599	0.3165	0.4811	0.6570	0.7986	0.1864	0.3681	0.5289	0.6938	0.8041
			(0.0497)	(0.0645)	(0.0802)	(0.0884)	(0.0521)	(0.0562)	(0.0648)	(0.0776)	(0.0812)	(0.0501)
	2.9990		0.1930	0.3905	0.5875	0.7900	0.9358	0.1954	0.4167	0.6024	0.8309	0.9303
			(0.0856)	(0.0993)	(0.1019)	(0.0973)	(0.0501)	(0.0779)	(0.1033)	(0.1064)	(0.1030)	(0.0451)
	0.2231		0.0411	0.0810	0.1187	0.1589	0.1900	0.0652	0.1176	0.1589	0.1852	0.1939
			(0.0202)	(0.0285)	(0.0368)	(0.0428)	(0.0571)	(0.0257)	(0.0348)	(0.0406)	(0.0453)	(0.0478)
	0.5108		0.0807	0.1581	0.2388	0.3168	0.3708	0.1174	0.2202	0.2949	0.3602	0.3833
			(0.0294)	(0.0399)	(0.0487)	(0.0579)	(0.0756)	(0.0328)	(0.0461)	(0.0501)	(0.0591)	(0.0686)
KMW	0.9163		0.1206	0.2419	0.3608	0.4805	0.5427	0.1563	0.2993	0.4210	0.5200	0.5666
			(0.0358)	(0.0505)	(0.0586)	(0.0688)	(0.0877)	(0.0408)	(0.0528)	(0.0598)	(0.0693)	(0.0838)
	1.6094		0.1627	0.3208	0.4816	0.6380	0.7064	0.1900	0.3584	0.5162	0.6600	0.7123
			(0.0424)	(0.0575)	(0.0680)	(0.0815)	(0.0962)	(0.0461)	(0.0565)	(0.0627)	(0.0824)	(0.0966)
	2.9990		0.1867	0.3689	0.5498	0.7082	0.7807	0.1969	0.3906	0.5678	0.7118	0.7620
			(0.0565)	(0.0723)	(0.0875)	(0.0921)	(0.1033)	(0.0500)	(0.0703)	(0.0828)	(0.0950)	(0.1033)
	0.2231		0.0427	0.0840	0.1218	0.1594	0.1951	0.0659	0.1184	0.1593	0.1838	0.1947
			(0.0187)	(0.0264)	(0.0326)	(0.0372)	(0.0486)	(0.0245)	(0.0327)	(0.0383)	(0.0417)	(0.0447)
	0.5108		0.0835	0.1632	0.2425	0.3172	0.3791	0.1193	0.2217	0.2953	0.3590	0.3920
			(0.0267)	(0.0368)	(0.0448)	(0.0536)	(0.0697)	(0.0313)	(0.0438)	(0.0478)	(0.0551)	(0.0661)
KMPW	0.9163		0.1232	0.2451	0.3609	0.4739	0.5545	0.1590	0.3013	0.4203	0.5161	0.5783
			(0.0327)	(0.0459)	(0.0540)	(0.0622)	(0.0806)	(0.0381)	(0.0493)	(0.0557)	(0.0632)	(0.0752)
	1.6094		0.1626	0.3193	0.4745	0.6327	0.7266	0.1926	0.3594	0.5093	0.6542	0.7293
			(0.0375)	(0.0533)	(0.0629)	(0.0724)	(0.0826)	(0.0423)	(0.0513)	(0.0567)	(0.0733)	(0.0835)
	2.9990		0.1933	0.3810	0.5619	0.7291	0.8247	0.2030	0.3961	0.5700	0.7190	0.7922
			(0.0500)	(0.0686)	(0.0789)	(0.0792)	(0.0759)	(0.0426)	(0.0643)	(0.0731)	(0.0802)	(0.0846)

levels of variance for small values of the second gap time, y.

In Table 2.7 we can see that for larger values of y, the Lin obtains less variance than both KMW and KMPW.

Table 2.5: Mean values and standard deviation of $\hat{F}_{12}(x, y)$ for the bivariate exponential scenario. Sample size of $n = 50$, uniform censoring $C \sim U[0, 4]$.

			$\delta = 0$					$\delta = 1$				
		y	0.2231	0.5108	0.9163	1.6094	2.9990	0.2231	0.5108	0.9163	1.6094	2.9990
	x											
CKM	0.2231		0.0651	0.1196	0.1603	0.1832	0.1932	0.0645	0.1207	0.1549	0.1847	0.1932
			(0.0363)	(0.0491)	(0.0547)	(0.0576)	(0.0577)	(0.0361)	(0.0479)	(0.0526)	(0.0590)	(0.0584)
	0.5108		0.1201	0.2171	0.2999	0.3555	0.3822	0.1192	0.2172	0.3010	0.3549	0.3807
			(0.0492)	(0.0637)	(0.0728)	(0.0747)	(0.0737)	(0.0458)	(0.0632)	(0.0700)	(0.0742)	(0.0760)
	0.9163		0.1605	0.3003	0.4179	0.5225	0.5691	0.1592	0.2971	0.4213	0.5184	0.5708
			(0.0519)	(0.0721)	(0.0777)	(0.0798)	(0.0821)	(0.0557)	(0.0693)	(0.0807)	(0.0780)	(0.0803)
	1.6094		0.1943	0.3699	0.5310	0.6739	0.7583	0.1937	0.3736	0.5298	0.6764	0.7539
			(0.0595)	(0.0784)	(0.0842)	(0.0845)	(0.0805)	(0.0606)	(0.0786)	(0.0835)	(0.0833)	(0.0838)
	2.9990		0.2186	0.4268	0.6089	0.7815	0.8372	0.2229	0.4198	0.6085	0.7772	0.8350
			(0.0663)	(0.0845)	(0.0874)	(0.0843)	(0.0782)	(0.0677)	(0.0832)	(0.0891)	(0.0852)	(0.0788)
Lin	0.2231		0.0400	0.0793	0.1205	0.1605	0.1916	0.0657	0.1188	0.1610	0.1846	0.1987
			(0.0306)	(0.0444)	(0.0519)	(0.0585)	(0.0616)	(0.0380)	(0.0473)	(0.0564)	(0.0560)	(0.0581)
	0.5108		0.0816	0.1621	0.2392	0.3170	0.3776	0.1195	0.2165	0.2951	0.3587	0.3967
			(0.0452)	(0.0597)	(0.0709)	(0.0782)	(0.0804)	(0.0508)	(0.0657)	(0.0718)	(0.0766)	(0.0769)
	0.9163		0.1198	0.2433	0.3638	0.4814	0.5780	0.1571	0.2960	0.4154	0.5161	0.5884
			(0.0542)	(0.0719)	(0.0861)	(0.0862)	(0.0448)	(0.0576)	(0.0740)	(0.0854)	(0.0882)	(0.0832)
	1.6094		0.1609	0.3195	0.4804	0.6420	0.7785	0.1847	0.3520	0.5189	0.6684	0.7883
			(0.0650)	(0.0823)	(0.0928)	(0.1015)	(0.0814)	(0.0670)	(0.0813)	(0.0880)	(0.0977)	(0.0802)
	2.9990		0.1818	0.3795	0.5772	0.7696	0.9222	0.1918	0.3919	0.5864	0.7880	0.9330
			(0.0838)	(0.1015)	(0.1112)	(0.1031)	(0.0747)	(0.0903)	(0.0994)	(0.1199)	(0.1092)	(0.0645)
KMW	0.2231		0.0394	0.0801	0.1207	0.1586	0.1911	0.0654	0.1180	0.1559	0.1826	0.1978
			(0.0288)	(0.0386)	(0.0510)	(0.0583)	(0.0697)	(0.0360)	(0.0465)	(0.0553)	(0.0604)	(0.0647)
	0.5108		0.0806	0.1576	0.2387	0.3235	0.3775	0.1176	0.2174	0.2989	0.3543	0.3928
			(0.0386)	(0.0544)	(0.0702)	(0.0757)	(0.0877)	(0.0488)	(0.0619)	(0.0724)	(0.0799)	(0.0822)
	0.9163		0.1179	0.2435	0.3603	0.4797	0.5694	0.1570	0.2976	0.4151	0.5180	0.5855
			(0.0480)	(0.0640)	(0.0773)	(0.0866)	(0.0998)	(0.0542)	(0.0687)	(0.0792)	(0.0856)	(0.0967)
	1.6094		0.1550	0.3230	0.4815	0.6372	0.7468	0.1843	0.3600	0.5161	0.6628	0.7495
			(0.0561)	(0.0759)	(0.0907)	(0.0942)	(0.1034)	(0.0602)	(0.0763)	(0.0857)	(0.0919)	(0.1030)
	2.9990		0.1883	0.3716	0.5735	0.7451	0.8613	0.1956	0.3857	0.5738	0.7520	0.8495
			(0.0677)	(0.0904)	(0.0959)	(0.1030)	(0.0982)	(0.0644)	(0.0850)	(0.0937)	(0.1007)	(0.1045)
KMPW	0.2231		0.0405	0.0816	0.1225	0.1591	0.1897	0.0656	0.1171	0.1550	0.1807	0.1948
			(0.0266)	(0.0368)	(0.0476)	(0.0528)	(0.0624)	(0.0339)	(0.0450)	(0.0531)	(0.0575)	(0.0601)
	0.5108		0.0820	0.1596	0.2417	0.3228	0.3768	0.1178	0.2169	0.2990	0.3532	0.3918
			(0.0365)	(0.0510)	(0.0658)	(0.0727)	(0.0824)	(0.0461)	(0.0590)	(0.0694)	(0.0771)	(0.0789)
	0.9163		0.1193	0.2452	0.3619	0.4768	0.5660	0.1576	0.2988	0.4177	0.5163	0.5847
			(0.0459)	(0.0610)	(0.0729)	(0.0806)	(0.0936)	(0.0510)	(0.0652)	(0.0764)	(0.0810)	(0.0895)
	1.6094		0.1560	0.3229	0.4798	0.6302	0.7400	0.1860	0.3603	0.5150	0.6581	0.7482
			(0.0524)	(0.0705)	(0.0862)	(0.0860)	(0.0931)	(0.0568)	(0.0735)	(0.0790)	(0.0827)	(0.0905)
	2.9990		0.1882	0.3711	0.5697	0.7416	0.8595	0.1979	0.3866	0.5700	0.7404	0.8421
			(0.0622)	(0.0846)	(0.0900)	(0.0928)	(0.0833)	(0.0597)	(0.0792)	(0.0863)	(0.0895)	(0.0887)

The problem that we consider in this part is more simpler since we are assuming that the first gap time is uncensored. Here we consider a (X, T) be a random vector where the response variable T denotes a lifetime, which is subject to random right

Table 2.6: Mean values and standard deviation of $\hat{F}_{12}(x, y)$ for the bivariate exponential scenario. Sample size of $n = 100$, uniform censoring $C \sim U[0, 4]$.

		$\delta = 0$					$\delta = 1$				
	y	0.2231	0.5108	0.9163	1.6094	2.9990	0.2231	0.5108	0.9163	1.6094	2.9990
	x										
	0.2231	0.0651	0.1174	0.1588	0.1861	0.1955	0.0667	0.1209	0.1606	0.1868	0.1946
		(0.0255)	(0.0348)	(0.0384)	(0.0411)	(0.0416)	(0.0254)	(0.0342)	(0.0379)	(0.0405)	(0.0427)
	0.5108	0.1191	0.2156	0.2997	0.3592	0.3879	0.1210	0.2170	0.2954	0.3607	0.3892
		(0.0342)	(0.0416)	(0.0488)	(0.0527)	(0.0541)	(0.0340)	(0.0436)	(0.0459)	(0.0501)	(0.0542)
CKM	0.9163	0.1629	0.3013	0.4211	0.5195	0.5816	0.1615	0.3016	0.4211	0.5223	0.5742
		(0.0388)	(0.0493)	(0.0534)	(0.0579)	(0.0576)	(0.0390)	(0.0495)	(0.0547)	(0.0568)	(0.0586)
	1.6094	0.1958	0.3732	0.5328	0.6752	0.7630	0.1942	0.3706	0.5281	0.6744	0.7647
		(0.0447)	(0.0578)	(0.0608)	(0.0590)	(0.0569)	(0.0435)	(0.0553)	(0.0589)	(0.0561)	(0.0579)
	2.9990	0.2161	0.4238	0.6155	0.7924	0.9003	0.2156	0.4194	0.6125	0.7908	0.9025
		(0.0469)	(0.0591)	(0.0597)	(0.0632)	(0.0575)	(0.0461)	(0.0599)	(0.0621)	(0.0621)	(0.0561)
	0.2231	0.0397	0.0806	0.1201	0.1602	0.1878	0.0663	0.1179	0.1571	0.1867	0.1991
		(0.0212)	(0.0315)	(0.0363)	(0.0392)	(0.0444)	(0.0266)	(0.0358)	(0.0385)	(0.0417)	(0.0411)
	0.5108	0.0787	0.1585	0.2396	0.3206	0.3784	0.1193	0.2192	0.2976	0.3603	0.3927
		(0.0301)	(0.0387)	(0.0478)	(0.0550)	(0.0606)	(0.0345)	(0.0458)	(0.0505)	(0.0528)	(0.0556)
Lin	0.9163	0.1190	0.2396	0.3572	0.4810	0.5760	0.1559	0.2992	0.4178	0.5185	0.5817
		(0.0376)	(0.0502)	(0.0571)	(0.0603)	(0.0665)	(0.0406)	(0.0526)	(0.0572)	(0.0606)	(0.0626)
	1.6094	0.1614	0.3223	0.4846	0.6417	0.7728	0.1853	0.3579	0.5183	0.6632	0.7823
		(0.0440)	(0.0584)	(0.0646)	(0.0696)	(0.0639)	(0.0461)	(0.0576)	(0.0640)	(0.0675)	(0.0610)
	2.9990	0.1909	0.3801	0.5671	0.7706	0.9236	0.1981	0.3889	0.5833	0.7878	0.9306
		(0.0601)	(0.0706)	(0.0805)	(0.0725)	(0.0584)	(0.0575)	(0.0717)	(0.0846)	(0.0812)	(0.0530)
	0.2231	0.0396	0.0798	0.1193	0.1590	0.1892	0.0638	0.1185	0.1585	0.1875	0.1967
		(0.0202)	(0.0288)	(0.0361)	(0.0404)	(0.0487)	(0.0251)	(0.0347)	(0.0391)	(0.0416)	(0.0463)
	0.5108	0.0794	0.1580	0.2395	0.3182	0.3814	0.1167	0.2189	0.2995	0.3604	0.3956
		(0.0282)	(0.0380)	(0.0475)	(0.0553)	(0.0629)	(0.0332)	(0.0451)	(0.0506)	(0.0535)	(0.0621)
KMW	0.9163	0.1217	0.2390	0.3593	0.4802	0.5694	0.1569	0.3007	0.4176	0.5198	0.5809
		(0.0344)	(0.0482)	(0.0542)	(0.0628)	(0.0738)	(0.0388)	(0.0508)	(0.0558)	(0.0592)	(0.0710)
	1.6094	0.1586	0.3194	0.4810	0.6432	0.7544	0.1839	0.3615	0.5157	0.6670	0.7591
		(0.0417)	(0.0529)	(0.0621)	(0.0619)	(0.0705)	(0.0423)	(0.0541)	(0.0602)	(0.0626)	(0.0734)
	2.9990	0.1904	0.3802	0.5707	0.7505	0.8716	0.1955	0.3918	0.5784	0.7595	0.8578
		(0.0491)	(0.0646)	(0.0711)	(0.0733)	(0.0717)	(0.0445)	(0.0617)	(0.0684)	(0.0757)	(0.0768)
	0.2231	0.0410	0.0813	0.1219	0.1603	0.1895	0.0635	0.1179	0.1571	0.1846	0.1939
		(0.0190)	(0.0265)	(0.0334)	(0.0369)	(0.0435)	(0.0237)	(0.0329)	(0.0374)	(0.0395)	(0.0431)
	0.5108	0.0817	0.1615	0.2420	0.3198	0.3811	0.1176	0.2186	0.2989	0.3581	0.3945
		(0.0261)	(0.0361)	(0.0447)	(0.0513)	(0.0587)	(0.0318)	(0.0435)	(0.0482)	(0.0510)	(0.0580)
KMPW	0.9163	0.1237	0.2425	0.3613	0.4775	0.5680	0.1583	0.3021	0.4187	0.5184	0.5826
		(0.0320)	(0.0446)	(0.0516)	(0.0589)	(0.0675)	(0.0373)	(0.0491)	(0.0529)	(0.0568)	(0.0666)
	1.6094	0.1605	0.3205	0.4797	0.6377	0.7523	0.1870	0.3637	0.5151	0.6613	0.7618
		(0.0380)	(0.0488)	(0.0581)	(0.0570)	(0.0650)	(0.0393)	(0.0512)	(0.0577)	(0.0573)	(0.0635)
	2.9990	0.1905	0.3798	0.5697	0.7524	0.8803	0.2003	0.3956	0.5743	0.7493	0.8607
		(0.0429)	(0.0600)	(0.0664)	(0.0658)	(0.0590)	(0.0404)	(0.0562)	(0.0630)	(0.0634)	(0.0583)

censoring, and X denotes a covariate. Note that the CKM estimator is consistent in this case.

Now we compare by simulations the 6 estimators (CKM, KMW, KMPW, Lin, condBIV_1 and condBIV_2), for the bivariate distribution function of (X, T). In this

Table 2.7: Mean values and standard deviation of $\hat{F}_{12}(x, y)$ for the bivariate Weibull scenario. Sample size $n = 50$ and $n = 100$.

			$n = 50$					$n = 100$				
		y	3.3067	5.0030	6.7006	8.8805	10.622	3.3067	5.0030	6.7006	8.8805	10.622
	x											
CKM	3.3067		0.1155	0.1575	0.1821	0.1876	0.1918	0.1164	0.1608	0.1830	0.1944	0.1958
			(0.0531)	(0.0586)	(0.0612)	(0.0633)	(0.0634)	(0.0356)	(0.0423)	(0.0455)	(0.0446)	(0.0447)
	5.0030		0.1699	0.2672	0.3334	0.3743	0.3833	0.1658	0.2661	0.3333	0.3726	0.3899
			(0.0663)	(0.0780)	(0.0856)	(0.0822)	(0.0832)	(0.0457)	(0.0532)	(0.0597)	(0.0585)	(0.0584)
	6.7006		0.2008	0.3491	0.4647	0.5448	0.5695	0.1979	0.3496	0.4614	0.5431	0.5714
			(0.0707)	(0.0885)	(0.0935)	(0.0940)	(0.0920)	(0.0508)	(0.0613)	(0.0659)	(0.0646)	(0.0648)
	8.8805		0.2231	0.4081	0.5657	0.7014	0.7549	0.2222	0.4100	0.5606	0.7029	0.7558
			(0.0804)	(0.0998)	(0.1017)	(0.0991)	(0.0900)	(0.0574)	(0.0704)	(0.0715)	(0.0645)	(0.0639)
	10.622		0.2343	0.4379	0.6145	0.7719	0.8356	0.2347	0.4341	0.6139	0.7704	0.8390
			(0.0826)	(0.1019)	(0.1073)	(0.0970)	(0.0860)	(0.0599)	(0.0724)	(0.0762)	(0.0653)	(0.0607)
Lin	3.3067		0.1137	0.1572	0.1814	0.1917	0.1958	0.1127	0.1565	0.1793	0.1933	0.1982
			(0.0574)	(0.0595)	(0.0647)	(0.0624)	(0.0639)	(0.0384)	(0.0444)	(0.0447)	(0.0441)	(0.0454)
	5.0030		0.1550	0.2611	0.3267	0.3757	0.3863	0.1555	0.2616	0.3315	0.3700	0.3922
			(0.0678)	(0.0815)	(0.0847)	(0.0851)	(0.0821)	(0.0520)	(0.0586)	(0.0632)	(0.0585)	(0.0588)
	6.7006		0.1774	0.3266	0.4551	0.5412	0.5845	0.1830	0.3306	0.4506	0.5377	0.5757
			(0.0786)	(0.1007)	(0.0996)	(0.0967)	(0.0914)	(0.0580)	(0.0674)	(0.0724)	(0.0642)	(0.0657)
	8.8805		0.1954	0.3777	0.5412	0.6906	0.7476	0.1931	0.3717	0.5407	0.6913	0.7521
			(0.0860)	(0.1092)	(0.1112)	(0.1101)	(0.0994)	(0.0607)	(0.0767)	(0.0809)	(0.0732)	(0.0659)
	10.622		0.1991	0.3902	0.5742	0.7585	0.8322	0.1988	0.3991	0.5763	0.7525	0.8304
			(0.0950)	(0.1111)	(0.1203)	(0.1071)	(0.0985)	(0.0646)	(0.0779)	(0.0819)	(0.0797)	(0.0694)
KMW	3.3067		0.1141	0.1599	0.1792	0.1947	0.1987	0.1138	0.1577	0.1810	0.1939	0.1971
			(0.0546)	(0.0623)	(0.0686)	(0.0751)	(0.0732)	(0.0378)	(0.0464)	(0.0480)	(0.0500)	(0.0533)
	5.0030		0.1573	0.2601	0.3307	0.3816	0.3895	0.1578	0.2654	0.3300	0.3754	0.3899
			(0.0656)	(0.0810)	(0.0924)	(0.0984)	(0.0990)	(0.0432)	(0.0574)	(0.0625)	(0.0661)	(0.0714)
	6.7006		0.1765	0.3243	0.4504	0.5480	0.5776	0.1799	0.3302	0.4521	0.5431	0.5775
			(0.0692)	(0.0906)	(0.1016)	(0.1113)	(0.1149)	(0.0508)	(0.0643)	(0.0722)	(0.0797)	(0.0803)
	8.8805		0.1898	0.3788	0.5461	0.6830	0.7559	0.1957	0.3734	0.5397	0.6835	0.7540
			(0.0713)	(0.0980)	(0.1098)	(0.1174)	(0.1086)	(0.0521)	(0.0689)	(0.0797)	(0.0796)	(0.0777)
	10.622		0.1970	0.3890	0.5769	0.7544	0.8319	0.1945	0.3894	0.5776	0.7517	0.8279
			(0.0735)	(0.0983)	(0.1144)	(0.1117)	(0.1038)	(0.0520)	(0.0727)	(0.0803)	(0.0781)	(0.0722)
KMPW	3.3067		0.1052	0.1502	0.1713	0.1861	0.1915	0.1036	0.1465	0.1704	0.1848	0.1894
			(0.0508)	(0.0599)	(0.0665)	(0.0719)	(0.0708)	(0.0346)	(0.0445)	(0.0459)	(0.0487)	(0.0514)
	5.0030		0.1533	0.2541	0.3259	0.3803	0.3913	0.1543	0.2578	0.3250	0.3744	0.3912
			(0.0612)	(0.0799)	(0.0906)	(0.0963)	(0.0978)	(0.0405)	(0.0550)	(0.0618)	(0.0651)	(0.0697)
	6.7006		0.1779	0.3217	0.4487	0.5522	0.5838	0.1813	0.3253	0.4520	0.5484	0.5859
			(0.0656)	(0.0874)	(0.0974)	(0.1062)	(0.1105)	(0.0464)	(0.0599)	(0.0696)	(0.0763)	(0.0767)
	8.8805		0.1929	0.3722	0.5398	0.6833	0.7591	0.2007	0.3689	0.5346	0.6860	0.7605
			(0.0653)	(0.0909)	(0.1044)	(0.1096)	(0.1017)	(0.0482)	(0.0644)	(0.0751)	(0.0753)	(0.0729)
	10.622		0.2006	0.3807	0.5634	0.7473	0.8303	0.2014	0.3841	0.5666	0.7440	0.8282
			(0.0671)	(0.0901)	(0.1084)	(0.1041)	(0.0935)	(0.0471)	(0.0662)	(0.0759)	(0.0745)	(0.0670)

scenario a covariate quantitative X was generated according $U[0, 1]$. The survival time of individuals T is exponential with parameter $1+X$. The exponential censoring time C was generated according to model $exp(1.471972)$. We show the results for $n = 200$ based on 1000 generated samples. We can see in Table 2.8 that the CKM and condBIV_2 are the ones with less MSE. The method based on presmoothing leads to good results for some quantiles. For the largest values of x and y, we observe that the values of MSE decreased. The values decrease when the sample size increases, results not shown here.

Table 2.8: Mean Square Error of bivariate distribution function with sample size $n = 200$

	y	0.1002	0.3424	0.6603	0.8161	1.3009	1.5719
	x						
	0.1	0.000050	0.000180	0.000362	0.000463	0.000624	0.000673
	0.25	0.000143	0.000451	0.000819	0.001026	0.001435	0.001610
CKM	0.5	0.000305	0.000865	0.001539	0.001762	0.002363	0.002603
	0.75	0.000480	0.001270	0.001925	0.002079	0.002658	0.002980
	0.9	0.000576	0.001479	0.002136	0.002315	0.002765	0.002949
	0.1	0.000218	0.000802	0.001046	0.000989	0.000730	0.000689
	0.25	0.000815	0.003402	0.003976	0.003460	0.002000	0.001717
Lin	0.5	0.001211	0.007811	0.008508	0.007059	0.003744	0.003129
	0.75	0.000901	0.007411	0.008536	0.007097	0.004137	0.003664
	0.9	0.004424	0.004182	0.006566	0.005944	0.004326	0.003876
	0.1	0.000045	0.000153	0.000360	0.000512	0.001009	0.001264
	0.25	0.000132	0.000503	0.001537	0.002313	0.004757	0.005974
KMW	0.5	0.000306	0.001098	0.003812	0.005675	0.011650	0.014526
	0.75	0.000609	0.001314	0.003181	0.004734	0.010109	0.012743
	0.9	0.001062	0.002100	0.002347	0.002548	0.004242	0.005294
	0.1	0.000022	0.000096	0.000283	0.000418	0.000890	0.001106
	0.25	0.000069	0.000384	0.001403	0.002143	0.004550	0.005659
KMPW	0.5	0.000176	0.000947	0.003709	0.005559	0.011510	0.014261
	0.75	0.000378	0.001054	0.002877	0.004358	0.009547	0.012039
	0.9	0.000786	0.001782	0.001739	0.001876	0.003287	0.004219
	0.1	0.000192	0.000643	0.000929	0.000944	0.000851	0.000774
	0.25	0.000718	0.002363	0.003173	0.003110	0.002572	0.002185
condBIV_1	0.5	0.001681	0.005580	0.007034	0.006724	0.005137	0.004065
	0.75	0.002477	0.007963	0.010011	0.009049	0.006572	0.005103
	0.9	0.002767	0.008921	0.010925	0.010242	0.007402	0.005439
	0.1	0.000037	0.000140	0.000274	0.000357	0.000496	0.000545
	0.25	0.000131	0.000423	0.000751	0.000942	0.001273	0.001379
condBIV_2	0.5	0.000290	0.000844	0.001494	0.001709	0.002203	0.002337
	0.75	0.000461	0.001229	0.001918	0.002058	0.002517	0.002877
	0.9	0.000562	0.001444	0.002109	0.002272	0.002640	0.003041

2.3.2 Conditional Bivariate Distribution

In this section, we carry out some simulations to investigate the behavior of the proposed estimators for finite sample sizes. More specifically, the Beran-type estimator with estimators by Lin et al. (1999) and de Uña-Álvarez and Meira-Machado (2008). The two competing nonparametric regression estimators $\hat{F}_{12}(z;x,y)$ (IPCW) and $\tilde{F}_{12}(z;x,y)$ (LIN-based) introduced in Section 2.2.3 are compared them to each other.

To simulate the data we follow the work described by Amorim et al. (2011), but including covariate effects. In summary, the simulation procedure is as follows:

(1) $V_1 \sim U(0,1)$; $V_2 \sim U(0,1)$ and $Z \sim U(0,1)$ are independently generated;

(2) $U_1 = V_1$; $A = (2U_1 - 1) - 1$; $B = (1 - (2U_1 - 1))^2 + 4V_2(2U_1 - 1)$

(3) $U_2 = \frac{2V_2}{\sqrt{B} - A}$

(4) $T_1 = \ln(\frac{1}{1-U_1})$; $T_2 = \ln(\frac{1}{1-U_2})$

(5) $\lambda(Z) = 0.6Z + 0.4$; $T_1(Z) = \frac{T_1}{\lambda(Z)}$; $T_2(Z) = \frac{T_2}{\lambda(Z)}$ and $T = T_1(Z) + T_2(Z)$

Note that the transformation of the Z and T in item (5) introduce some dependency of the covariate on the gap times. For the censoring variable we considered that $C|Z = z$ is generated from an exponential distribution with rate $\lambda(z) = 0.15 + 0.35z$, this scenario provides dependent censoring.

The goal of this simulation study is to investigate the performance of the two proposed estimators for the conditional bivariate distribution (LIN-based and IPCW) and to compare them to each other. For measuring the estimates' performance we computed the integrated mean square errors (IMSE) of the estimates. For each simulated scenario we derived the analytic expression of $F_{12}(z;x,y)$ so that the MSE of the estimator could be computed. $M = 1000$ Monte Carlo trials were generated with four different sample sizes $n = 50$, 100, 150 and 200 (only results for sample sizes $n = 100$ and $n = 200$ are shown). Let $\hat{F}^k_{12}(z;x,y)$ denote the estimated conditional bivariate distribution based on the kth generated data set. For each fixed triple $(z;x,y)$ we computed the pointwise estimates of the MSE as:

$$\widehat{MSE}(\hat{F}_{12}(z;x,y)) = \frac{1}{M}\sum_{k=1}^{M}[\hat{F}_{12}^{k}(z;x,y) - F_{12}(z;x,y)]^2$$

To summarize the results we fixed the values of (x, y) using several quantiles (the same pairs as used in the paper by Lin et al. (1999)) and calculated the IMSE as

$$\widehat{IMSE} = \sum_{z} \widehat{MSE}(\hat{F}_{12}(z;x,y)) \times \delta$$

The results displayed in Tables 2.9 and 2.10 were obtained by numerical integration on the interval of Z, taking a grid of step $\delta = 0.025$. To compute the conditional bivariate distribution we have used a common bandwidth selector and Gaussian kernels. To this end we have used the `dpik` function from the `R` **KernSmooth** package. This is the data based bandwidth selector of Wand and Jones (1995). For the computation of $W_1(z, b_n)$ we have used Local Linear and Nadaraya-Watson weights.

Results shown in Tables 2.9 and 2.10 support the idea that the IPCW method leads to better results for the conditional bivariate distribution. As expected, the IMSE decreases with an increase in the sample size. The IMSE increase with x and with y.

Figure 2.1 depicts the averaged estimates for the bivariate distribution function along 1000 Monte Carlo trials of size 100. Results obtained for the two methods by fixing $x = 0.2231$ and $y = 0.9163$, and varying Z, reveal that both methods are almost unbiased.

In Figure 2.2 we present plots for the conditional bivariate distribution, based on simulated data, by fixing $x = 0.2231$ and considering two possible values for the covariate information (first and third quartile). The results obtained for the two methods, based on a single Monte Carlo sample with size 1000, show that the estimates of the bivariate distribution greatly depends on covariate information.

Table 2.9: Integrated Mean Square Error (x1000) of the estimated bivariate distribution $\widehat{F}_{12}(z;x,y)$ along 1,000 trials for different sample sizes; Results for IPCW and LIN-based methods using **Nadaraya-Watson** weights.

			y	0.2231	0.5108	0.9163	1.6094
		x					
n=100	IPCW	0.2231		0.9741	1.9338	2.6740	3.4968
	LIN-based			1.0411	1.9978	2.6976	3.2134
	IPCW	0.5108		1.8276	3.4581	4.7382	6.1404
	LIN-based			2.0400	3.7950	5.0302	5.9663
	IPCW	0.9163		2.6538	4.9718	6.6870	8.4243
	LIN-based			3.0459	5.5695	7.3580	8.4556
	IPCW	1.6094		3.4150	6.2015	8.4886	10.0234
	LIN-based			4.0485	7.0899	9.5805	10.7170
n=200	IPCW	0.2231		0.6707	1.2795	1.8296	2.3932
	LIN-based			0.7421	1.3731	1.8603	2.1953
	IPCW	0.5108		1.2427	2.3095	3.2838	4.2425
	LIN-based			1.4330	2.6043	3.4980	4.1050
	IPCW	0.9163		1.7842	3.2508	4.5633	5.8255
	LIN-based			2.0612	3.6827	4.9211	5.8276
	IPCW	1.6094		2.3310	4.1534	5.9006	7.1868
	LIN-based			2.8839	4.9110	6.6283	7.7565

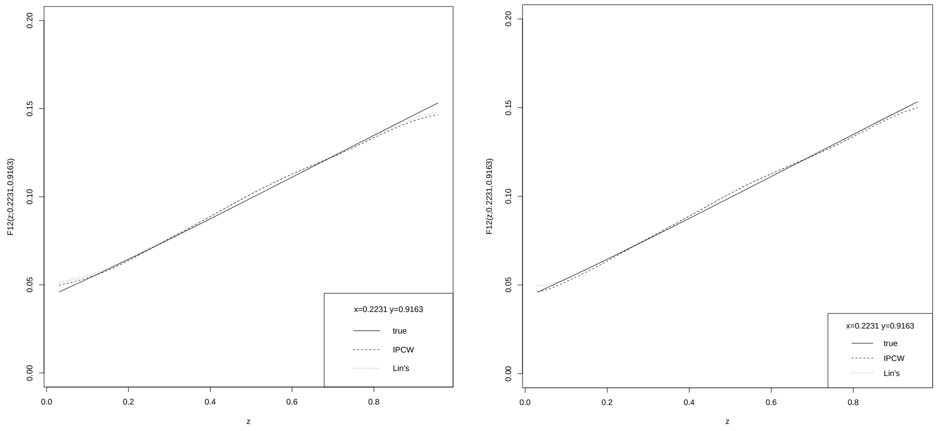

Figure 2.1: Conditional bivariate distribution $\hat{F}_{12}(z;0.2231,0.9163)$. Nadaraya-Watson (left hand-side) and Local Linear (right hand-side) Weights.

Table 2.10: Integrated Mean Square Error (x1000) of the estimated bivariate distribution $\widehat{F}_{12}(z;x,y)$ along 1,000 trials for different sample sizes; Results for IPCW and LIN-based methods using **Local Linear** weights.

			y	0.2231	0.5108	0.9163	1.6094
		x					
n=100	IPCW	0.2231		1.0292	2.0531	2.8389	3.7110
	LIN-based			1.1007	2.1078	2.8478	3.3922
	IPCW	0.5108		1.9425	3.6930	5.0470	6.5451
	LIN-based			2.1729	4.0381	5.3553	6.3150
	IPCW	0.9163		2.8169	5.2762	7.0903	8.9606
	LIN-based			3.2403	5.9146	7.7989	8.9281
	IPCW	1.6094		3.6130	6.5878	9.0045	10.6468
	LIN-based			4.2828	7.5701	10.1949	11.3425
n=200	IPCW	0.2231		0.7451	1.4050	2.0038	2.6408
	LIN-based			0.8221	1.5046	2.0397	2.4183
	IPCW	0.5108		1.3808	2.5382	3.6222	4.6908
	LIN-based			1.5821	2.8487	3.8488	4.5094
	IPCW	0.9163		1.9809	3.5798	5.0529	6.4570
	LIN-based			2.2903	4.0397	5.4514	6.4123
	IPCW	1.6094		2.5959	4.6106	6.5681	8.0202
	LIN-based			3.2042	5.4484	7.4278	8.6701

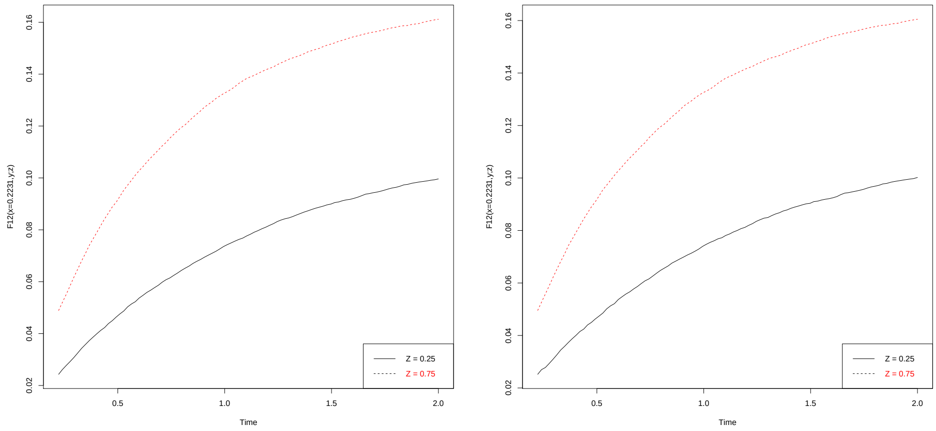

Figure 2.2: Conditional bivariate distribution $\hat{F}_{12}(z;x,y)$ based on simulated data. IPCW method (left hand-side) and Lin-based method (right hand-side).

2.4 Real Data Illustration

To illustrate our methods use data from a German Breast cancer (described in detail in Chapter 1). From the total of 686 women, 299 developed a recurrence and among these 171 died. A vector of covariates including age at acceptance were also recorded. Recurrence can be considered as an intermediate transient state and modeled using a progressive three-state model with states "Alive and disease-free", "Alive with Recurrence" and "Dead".

In this section we will provide results for the bivariate distribution function (CKM, KMW, KMPW and Lin) and for the conditional bivariate distribution (Lin-based and IPCW). All methods will be illustrated using several plots and tables.

To illustrate the estimators for the bivariate distribution function we present in Table 2.11 some estimates for several specific values for all four estimators introduced in Section 2.2.2. We can see that all four methods provide similar values for all pairs of values. The performance of the four methods can be seen in Figure 2.3 by fixing $x = 567$. The graph reveals that the values are similar. The Lin estimator provides non-monotone curves which can be considered a serious problem. The good behaviour of the CKM estimator in Figure 2.3, is explain by the low proportion of censoring for the subset of data $(T_1 \leq 567)$.

Figure 2.4 depicts the estimates along the covariate *age* together with 95$\%$ pointwise confidence bands based on simple bootstrap. In both plots it is seen that these curves are not constant; the effects of *age* depicted in these plots, which are purely nonparametric indicate some influence of this covariate in the bivariate distribution function. Both plots, based on different methods, suggest that the bivariate distribution function decreases with age. A visual inspection suggest that patients near thirty years old have higher values for the bivariate distribution than those in near seventy years old.

We plot in Figure 2.5 the conditional bivariate distribution for patients with 35 years old and patients with 65 years old. A particular problem of LIN-based method is appreciated in these figure, because the displayed curves for $F_{12}(z; x, y)$ are not

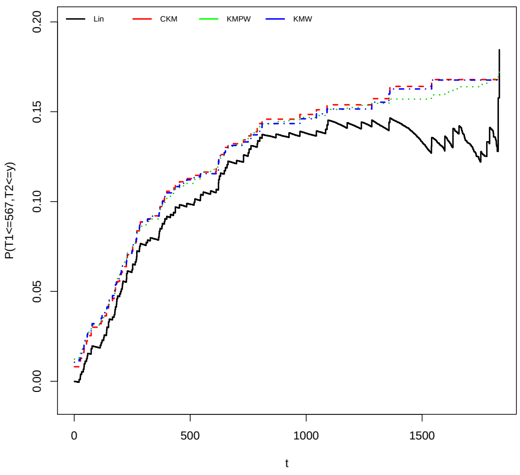

Figure 2.3: Evolution of the bivariate distribution $\hat{F}_{12}(567, y)$. Breast cancer data.

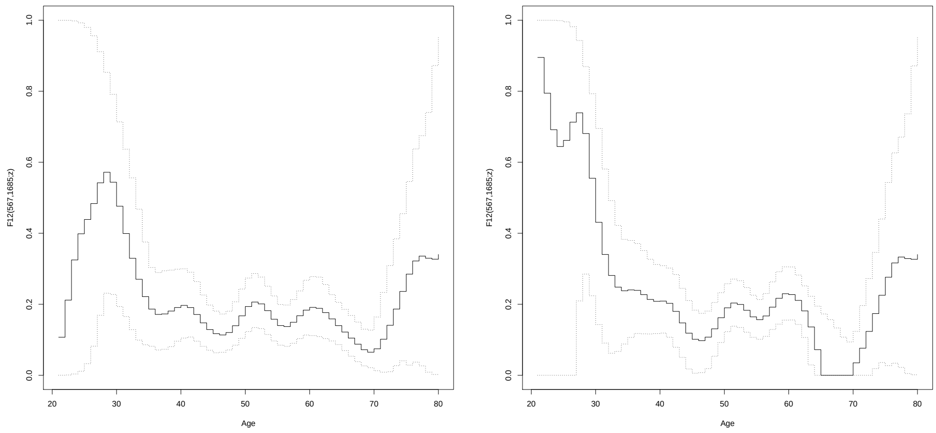

Figure 2.4: Evolution of the bivariate distribution $\hat{F}_{12}(567, 1685)$ along the covariate age. IPCW method on the left hand-side and LIN-based method on right hand-side. Breast cancer data.

monotone increasing in y. This is a consequence of the specific reweighting of the data which is used in this approach. This reweighting is the explanation why the LIN-based methods has several jump point in contrast to the IPCW method. The Table 2.12 show the estimates for the conditional bivariate distribution for patients with 35 and 65 years old.

Table 2.11: Estimates for the bivariate distribution function for several quantiles. Breast cancer study.

		y	567	1084	1685
	x				
CKM	567		0.1164	0.1511	0.1679
Lin			0.1049	0.1403	0.1345
KMPW			0.1166	0.1490	0.1639
KMW			0.1155	0.1488	0.1676
CKM	1084		0.1875	0.2553	0.2944
Lin			0.1602	0.2305	0.2745
KMPW			0.1885	0.2525	0.2852
KMW			0.1874	0.2547	0.2921
CKM	1685		0.2401	0.3326	0.3912
Lin			0.2017	0.2871	0.3770
KMPW			0.2419	0.3256	0.3993
KMW			0.2324	0.3052	0.3426
CKM	2195		0.2868	0.4025	0.4739
Lin			0.2094	0.3374	0.4273
KMPW			0.2813	0.3747	0.4485
KMW			0.2724	0.3838	0.4212

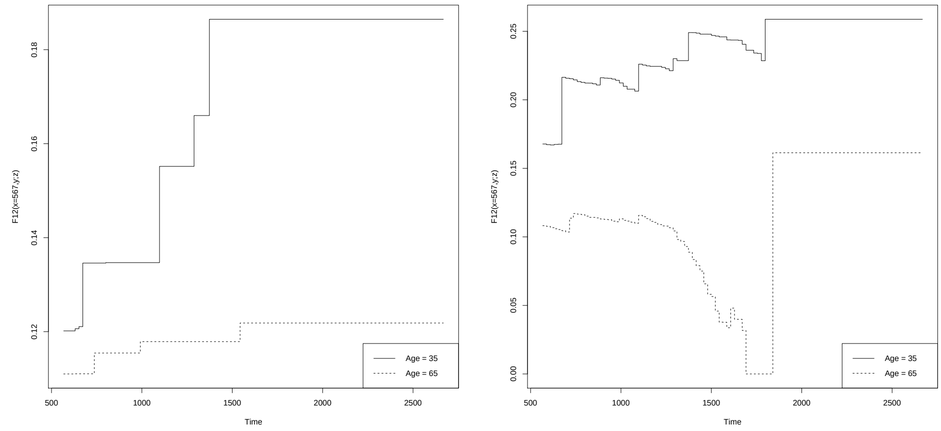

Figure 2.5: Conditional bivariate distribution for the Breast cancer data (IPCW method - left hand-side and LIN-based method - right hand-side) for $age = 35$ and $age = 65$.

Table 2.12: Estimates of the conditional bivariate distribution. Breast cancer study.

		y	567	1084	1685
		x			
z=35	IPCW	567	0.1202	0.1347	0.1865
			(0.0324,0.2236)	(0.0465,0.2381)	(0.0791,0.3026)
	LIN-based		0.1678	0.2060	0.2405
			(0.0530,0.2974)	(0.0783,0.3630)	(0.1083,0.3847)
	IPCW	1084	0.1619	0.2513	0.3031
			(0.0518,0.2893)	(0.1169,0.4030)	(0.1571,0.4517)
	LIN-based		0.1898	0.2944	0.3867
			(0.0428,0.3404)	(0.1316,0.4629)	(0.2313,0.5382)
	IPCW	1685	0.2052	0.2946	0.3464
			(0.0769,0.3568)	(0.1431,0.4628)	(0.1812,0.52140)
	LIN-based		0.0703	0.4781	0.5704
			(0.0000,0.2906)	(0.2719,0.6691)	(0.3939,0.7303)
	IPCW	2195	0.2056	0.2950	0.3468
			(0.0773,0.3568)	(0.1435,0.4665)	(0.1812,0.5173)
	LIN-based		0.0848	0.4927	0.5850
			(0.0000,0.4400)	(0.2918,0.8929)	(0.4080,0.9695)
z=65	IPCW	567	0.1110	0.1179	0.1218
			(0.0595,0.1729)	(0.0651,0.1800)	(0.0681,0.1842)
	LIN-based		0.1082	0.1104	0.0000
			(0.0551,0.1732)	(0.0507,0.1769)	(0.0000,0.1751)
	IPCW	1084	0.2234	0.2458	0.2806
			(0.1497,0.3043)	(0.1706,0.3303)	(0.1888,0.3829)
	LIN-based		0.1904	0.2198	0.1924
			(0.1094,0.2740)	(0.1040,0.3382)	(0.0000,0.3963)
	IPCW	1685	0.2710	0.2934	0.3282
			(0.1809,0.3623)	(0.2033,0.3842)	(0.2246,0.4381)
	LIN-based		0.2998	0.2305	0.3673
			(0.1835,0.4041)	(0.0000,0.4866)	(0.1124,0.5759)
	IPCW	2195	0.2768	0.2992	0.3340
			(0.1885,0.3651)	(0.2096,0.3895)	(0.2305,0.4457)
	LIN-based		0.3072	0.2378	0.3747
			(0.1909,0.4129)	(0.0000,0.4935)	(0.1182,0.5841)

Chapter 3

Presmoothing the transition probabilities in the illness-death model

3.1 Introduction

Multi-state models (Andersen et al., 1993; Meira-Machado et al., 2009) are the most common models used for the description of longitudinal survival data. A multi-state model is a stochastic process $(X(t), t \in \mathcal{T})$ with a finite state space, where $X(t)$ represents the state occupied by the process at time $t \geq 0$. For two states i,j and $s < t$, introduce the so-called transition probabilities

$$p_{ij}(s,t) = P\left(X(t) = j | X(s) = i\right).$$

Estimating these quantities is interesting, since they allow for long-term predictions of the process. The inference in multi-state models is traditionally performed under the Markov assumption, which states that past and future are independent given the present state. Aalen and Johansen (1978) introduced a nonparametric estimator of $p_{ij}(s,t)$ for non-homogeneous Markov models. Their estimation method extends the time-honored Kaplan-Meier estimator (Kaplan and Meier, 1958) to Markov chains. As for the Kaplan-Meier, the standard error of the Aalen-Johansen estimator may be large when censoring is heavy, particularly with a small sample size. Interestingly, the variance of the Kaplan-Meier estimator may be reduced by presmoothing. The idea of presmoothing (Dikta, 1998) involves replacing the censoring indicators by some smooth fit before the Kaplan-Meier formula is applied. This preliminary smoothing may be based on a certain parametric family such as the logistic (thus leading to a semiparametric estimator), or on a nonparametric estimator of the binary regression curve. Successful applications of presmoothed estimators include nonparametric curve estimation (Cao and Jácome, 2004), regression analysis (de Uña Álvarez and Campos-Rodríguez, 2004; Yuan, 2005), and the estimation of the bivariate distribution of censored gap times (de Uña-Álvarez and Amorim, 2011). The main goal of the present chapter is to propose a presmoothed version of the Aalen-Johansen estimator for the transition matrix of a Markov illness-death model, and to investigate its statistical properties. The proposed estimator is different to that in Amorim et al. (2011), who considered presmoothed transition probabilities for possibly non-Markov models. In general, Markov and non-Markov approaches lead to completely different estimators, so markovian estimators can not be obtained as particular cases of non-markovian estimators, and vice-versa.

The rest of the chapter is organized as follows. In Section 3.2 we introduce the new estimator and we formally establish its consistency. In Section 3.3 we compare by simulations the proposed estimator to the original Aalen-Johansen curve. In Section 3.4 we illustrate the proposed method using data from the Stanford Heart Transplant study, previously presented in Chapter 1. Technical proofs are deferred to Section 3.5.

3.2 The estimator: Main Results

In this chapter we consider the (progressive) illness-death model depicted in Figure 1.3. We assume that all subjects are in State 1 ("healthy") at time $t = 0$, and that they may either visit State 2 ("diseased") at some time point; or not, going directly to the absorbing ("dead") state. Given two time points $s < t$, there are in essence three different transition probabilities to estimate: $p_{11}(s,t)$, $p_{12}(s,t)$, and $p_{22}(s,t)$. The two other transition probabilities ($p_{13}(s,t)$ and $p_{23}(s,t)$) can be obtained from $p_{13}(s,t) = 1 - p_{11}(s,t) - p_{12}(s,t)$ and $p_{23}(s,t) = 1 - p_{22}(s,t)$.

The irreversible illness-death model is fully characterized by three transitions represented by the arrows in Figure 1.3. Let T_{ij} denote the potential transition time from State i to State j. In this model we have two competing transitions $1 \to 2$ and $1 \to 3$. Therefore, we denote by $\rho = I(T_{12} \leq T_{13})$ the indicator of visiting state 2 at some time, and introduce $Z = T_{12} \wedge T_{13}$, the sojourn time in state 1. A subject visiting State 2 will arrive at the absorbing "dead" state at time $T_{12} + T_{23}$, while this time will be T_{13} for those not visiting State 2 (i.e. $\rho = 0$). Finally, let $T = Z + \rho T_{23}$ denote the total survival time of the process. However, because of follow-up limitations, lost cases and so on, rather than (Z, T, ρ) one observes $(\widetilde{Z}, \widetilde{T}, \Delta_1, \Delta, \Delta_1\rho)$ where $\widetilde{Z} = Z \wedge C$, $\widetilde{T} = T \wedge C$, $\Delta_1 = I(Z \leq C)$ and $\Delta = I(T \leq C)$. Here C denotes the potential censoring time, which we assume to be independent of the process (that is, C and (Z, T) are assumed to be independent). Under continuity, the information provided by $\Delta_1\rho$ is superfluous since we have $\Delta_1\rho = I(\widetilde{Z} < \widetilde{T})$. With this notation, the transition probabilities are written as

$$\begin{aligned} p_{11}(s,t) &= \frac{P(Z > t)}{P(Z > s)}, \qquad p_{12}(s,t) = \frac{P(s < Z \leq t, T > t)}{P(Z > s)}, \\ p_{22}(s,t) &= \frac{P(Z \leq s, T > t)}{P(Z \leq s, T > s)}. \end{aligned}$$

Under the Markov assumption, all these quantities are estimated nonparametrically using Aalen-Johansen estimators. Explicit formulae of the Aalen-Johansen estimator for the illness-death model are available (Borgan, 1998). Here we give alternative expressions for this estimator suitable to motivate our method of presmoothing below. Assume that we have a sample of n individuals from the population under study.

Let $(\widetilde{Z}_i, \widetilde{T}_i, \Delta_{1i}, \Delta_i, \Delta_{1i}\rho_i)$, $i = 1, ..., n$ be the corresponding sampling information. The Aalen-Johansen estimate of the transition probability $p_{11}(s,t)$ is the Kaplan-Meier estimator

$$\widehat{p}_{11}^{AJ}(s,t) = \prod_{s<\widetilde{Z}_i\le t} \left[1 - \frac{\Delta_{1i}}{n\widetilde{M}_{0n}(\widetilde{Z}_i)}\right] \tag{3.1}$$

where

$$\widetilde{M}_{0n}(y) = \frac{1}{n}\sum_{j=1}^{n} I(\widetilde{Z}_j \ge y).$$

Then, Aalen-Johansen estimate of the transition probability $p_{22}(s,t)$ is the Kaplan-Meier estimator

$$\widehat{p}_{22}^{AJ}(s,t) = \prod_{s<\widetilde{T}_i\le t, \widetilde{Z}_i<\widetilde{T}_i} \left[1 - \frac{\Delta_i}{n\widetilde{M}_{1n}(\widetilde{T}_i)}\right] \tag{3.2}$$

where

$$\widetilde{M}_{1n}(y) = \frac{1}{n}\sum_{j=1}^{n} I(\widetilde{Z}_j < y \le \widetilde{T}_j).$$

Finally, the estimator for $p_{12}(s,t)$ is given by

$$\widehat{p}_{12}^{AJ}(s,t) = \frac{1}{n}\sum_{i=1}^{n} \frac{\widehat{p}_{11}^{AJ}(s,\widetilde{Z}_i^-)\widehat{p}_{22}^{AJ}(\widetilde{Z}_i,t) I(s<\widetilde{Z}_i\le t, \widetilde{Z}_i<\widetilde{T}_i)}{n\widetilde{M}_{0n}(\widetilde{Z}_i)} \tag{3.3}$$

where

$$\widehat{p}_{11}^{AJ}(s,t^-) = lim_{u\uparrow t}\widehat{p}_{11}^{AJ}(s,u)$$

Now, we discuss how to introduce modified estimators based on presmoothing. Presmoothing the Aalen-Johansen (AJ) involves replacing the censoring indicators (in the transition probabilities $p_{11}(s,t)$ and $p_{22}(s,t)$) by a smooth fit. The presmoothed version of $p_{11}(s,t)$ is obtained by replacing the Δ_{1i}'s in (3.1) by some smooth fit to the binary regression function $m_0(z) = P\left(\Delta_1 = 1 | \widetilde{Z} = z\right)$ (see e.g. Dikta (1998)). Then, the corresponding presmoothed Aalen-Johansen (P-AJ) estimator is given by

$$\widetilde{p}_{11}^{PAJ}(s,t) = \prod_{s<\widetilde{Z}_i\leq t}\left[1 - \frac{m_{0n}(\widetilde{Z}_i)}{n\widetilde{M}_{0n}(\widetilde{Z}_i)}\right] \tag{3.4}$$

where $m_{0n}(z)$ stands for an estimator of the binary regression function $m_0(z)$. Then, $m_0(\widetilde{Z})$ is the conditional probability of the event $\Delta_1 = 1$ given $\widetilde{Z}$. Since the pair $\widetilde{Z}, \Delta_1$ is observable, the function $m_0(z)$ can be estimated by standard methods. For example, logistic regression may be performed. Consider now the presmoothed version of (3.2) given by

$$\widetilde{p}_{22}^{PAJ}(s,t) = \prod_{s<\widetilde{T}_i\leq t, \widetilde{Z}_i<\widetilde{T}_i}\left[1 - \frac{m_{1n}(\widetilde{Z}_i, \widetilde{T}_i)}{n\widetilde{M}_{1n}(\widetilde{T}_i)}\right] \tag{3.5}$$

where $m_{1n}(z,t)$ stands for an estimator of the binary regression function $m_1(z,t) = P\left(\Delta = 1 | \widetilde{Z} = z, \widetilde{T} = t, \Delta_1\rho = 1\right)$. Then, $m_1(\widetilde{Z}, \widetilde{T})$ is the conditional probability of the event $\Delta = 1$ given $(\widetilde{Z}, \widetilde{T})$ and given that transition $1 \to 2$ is observed $(\Delta_1\rho = 1)$. Amorim et al. (2011) discussed the role of the function $m_1(z,t)$ as a suitable presmoothing strategy for $p_{22}(s,t)$; although these authors considered a different context in which the Markov assumption may not hold, their discussion on the presmoothing issue remains valid here. As before, $\widetilde{Z}, \widetilde{T}, \Delta$ and $\Delta_1\rho$ are observable, allowing the function $m_1(z,t)$ to be estimated by standard methods. Finally the transition probability $p_{12}(s,t)$ can be estimated by plugging (3.4) and (3.5) into equation (3.3).

The estimator $m_{0n}(z)$ is based on the whole sample, while $m_{1n}(z,t)$ is based on the subsample $i : \Delta_{1i}\rho_i = 1$. We assume that these two empirical functions approximate well their targets in a uniform sense; more specifically, set

$$U_1: \ \sup_z |m_{0n}(z) - m_0(z)| \to 0 \qquad \text{w. p. 1,}$$

and

$$U_2: \ \sup_{z,t} |m_{1n}(z,t) - m_1(z,t)| \to 0 \qquad \text{w. p. 1.}$$

Conditions under which U_1 and U_2 can be fulfilled were investigated in a number of papers, including Dikta (1998, 2000), Devroye (1978a,b), Mack and Silverman (1982) and Hardle and Luckhaus (1984). The uniform consistency of $\widehat{p}_{11}^{PAJ}(s,t)$ will hold on $0 \leq s < t \leq \tau$, where τ is strictly smaller than the upper bound of the support of $\widetilde{Z}$. Put $\widetilde{M_1}(y) = P(\widetilde{Z} < y \leq \widetilde{T})$. For the uniform consistency of $\widehat{p}_{22}^{PAJ}(s,t)$ and $\widehat{p}_{12}^{PAJ}(s,t)$ we will refer to the following assumption:

$$M: \ \ \widetilde{M_1} \text{ is bounded from below on } [\tau_0, \tau_1].$$

This condition allows to handle some denominators which appear in the proofs. It can be interpreted as a "non empty risk set" assumption for the transition from State 2 to State 3. By force, $\tau_0 > 0$, while τ_1 is (similarly as for τ) strictly smaller than the upper bound of the support of $\widetilde{T}$. We have the following result and the respectively proof.

Theorem 1. (a) Under U_1 we have w. p. 1

$$\sup_{0\leq s<t\leq\tau} \left|\widehat{p}_{11}^{PAJ}(s,t) - p_{11}(s,t)\right| \to 0.$$

(b) Besides, under U_2 and M, we have w. p. 1

$$\sup_{\tau_0\leq s<t\leq\tau_1} \left|\widehat{p}_{22}^{PAJ}(s,t) - p_{22}(s,t)\right| \to 0.$$

(c) Finally, under U_1, U_2 and M we have w. p. 1

$$\sup_{\tau_0\leq s<t\leq\tau} \left|\widehat{p}_{12}^{PAJ}(s,t) - p_{12}(s,t)\right| \to 0.$$

3.3 Simulation Study

In this section, we compare by simulations the presmoothed Aalen-Johansen estimator for the transition probabilities to the original Aalen-Johansen estimator. More specifically, the AJ and P-AJ type estimators $\widehat{p}_{11}(s,t)$, $\widehat{p}_{12}(s,t)$ and $\widehat{p}_{22}(s,t)$ introduced in Section 3.2 are considered. As presmoothing function we always take a parametric (logistic) family, so we actually have a semiparametric Aalen-Johansen estimator.

To simulate the data in the illness-death model, we followed the work of Amorim et al. (2011). We assume that all individuals are in State 1 ("healthy") at time $t = 0$. Therefore, the patient's history (or course) may be divided into two groups according to whether the disease occurred (that is, passing through State 2) $(1 \to 2 \to 3)$ or not $(1 \to 3)$. We separately consider these two possible subgroups of individuals. For the first subgroup of individuals $(\rho = 1)$, the successive gap times $(Z, T - Z)$ are simulated according to the bivariate distribution

$$F_{12}(x,y) = F_1(x)F_2(y)\left[1 + \theta\left\{1 - F_1(x)\right\}\left\{1 - F_2(y)\right\}\right]$$

with unit exponential margins. The parameter θ controls for the amount of dependency between the gap times $(Z, T - Z)$ and was set to 0 and 1, corresponding to 0 and 0.25 correlation between Z and $T - Z$. For the second subgroup of individuals $(\rho = 0)$, the value of Z is simulated according to an exponential with rate parameter 1. In summary the simulation procedure is as follows:

Step 1 Draw $\rho \sim Ber(p)$ where p is the proportion of subjects passing through State 2.

Step 2 If $\rho = 1$ then:

a) $V_1 \sim U(0,1), V_2 \sim U(0,1)$ are independently generated;

b) $U_1 = V_1, A = \theta(2U_1 - 1) - 1, B = (1 - \theta(2U_1 - 1))^2 + 4\theta V_2(2U_1 - 1)$

c) $U_2 = \frac{2V_2}{\sqrt{B} - A}$

d) $Z = log(\frac{1}{1-U_1}), T = log(\frac{1}{1-U_2}) + Z$

Step 3 If $\rho = 0$ then:

a) $Z = log(\frac{1}{1-U(0,1)})$.

In our simulation we consider that 70% of the individuals were in the first group. The follow-up time was subjected to right censoring, C, according to uniform models $U\,[0, 4]$ and $U\,[0, 3]$. The first model results in 24% of censoring on the first gap time Z, and in 47% of censoring on the second gap time $T - Z$, for those individuals with $\rho = 1$. The second model increases these censoring levels to 32% and about 57%, respectively.

After some algebra, it is seen that the function $m_1(z, t) = P\left(\Delta = 1|\widetilde{Z} = z, \widetilde{T} = t, \Delta_1\rho = 1\right)$ is written as

$$m_1(z, t) = \frac{1}{1 + \eta_1(z, t)}, \qquad \text{where} \;\; \eta_1(z, t) = \frac{\lambda_G(t)}{\lambda^1_{T|Z=z}(t|z)}$$

and where $\lambda_G(.)$ and $\lambda^1_{T|Z=z}(.|z)$ stand respectively for the hazard rate of the censoring variable and the hazard rate of T given $Z = z$ under restriction $\rho = 1$. Note that $\lambda_G(t) = 1/(\tau_G - t)$ when $C \sim U\,[0, \tau_G]$ and that $\lambda^1_{T|Z=z}(t|z)$ is given by

$$\lambda^1_{T|Z=z}(t|z) = \frac{2 + 4\exp(-t) - 2\exp(-z) - 2\exp(-t + z)}{2 + 2\exp(-t) - 2\exp(-z) - \exp(-t + z)} \qquad \text{if } \theta = 1,$$

being 1 when $\theta = 0$. The function $m_1(z, t)$ belongs to the logistic family with some preliminary transformation of the conditioning variables, namely we have (for $\beta_0 = 0$ and $\beta_1 = 1$)

$$m_1(z, t; \beta) = \frac{1}{1 + \exp(\beta_0 + \beta_1 \ln(\eta_1(z, t)))}.$$

This is the parametric model we fit to $m_1(z, t)$ in the simulations. For $m_0(z) = P\left(\Delta_1 = 1|\widetilde{Z} = z\right)$, we have

$$m_0(z) = \frac{1}{1+\eta_0(z)}, \qquad \text{where } \eta_0(z) = \frac{\lambda_G(z)}{\lambda_Z(z)}$$

and where $\lambda_Z(z)$ stands for the hazard function of Z.

Similarly as above, we also perform logistic presmoothing for the function $m_0(z)$, with the variable $\widetilde{Z}$ transformed by $-ln(\tau_G - \widetilde{Z})$. This function belongs to the logistic family with some preliminary transformation. To estimate the function $m_0(z)$ in the simulations, we fit the logistic model

$$m_0(z;\gamma) = \frac{1}{1+\exp(\gamma_0 + \gamma_1 \ln(\eta_0(z)))}$$

which contains the true presmoothing function m_0 as a special case ($\gamma_0 = 0$,$\gamma_1 = 1$).

The β parameter in model $m_1(.;\beta)$ is estimated via maximization of the conditional likelihood of the Δ_i's given the $(\widetilde{Z}_i, \widetilde{T}_i)$'s, for those subjects with $\Delta_1\rho = 1$ (see Dikta (1998, 2000)). Similarly, the γ parameter in model $m_0(.;\gamma)$ is estimated via maximization of the conditional likelihood of the Δ_{1i}'s given the $\widetilde{Z}_i$'s. Note that the β parameter is needed for estimating $p_{22}(s,t)$ and $p_{12}(s,t)$, while γ enters the estimation of $p_{11}(s,t)$ and (again) $p_{12}(s,t)$. The aim of this simulation study is to compare the Aalen-Johansen estimator (1978) and the new estimator based on presmoothing (P-AJ). Again, for measuring the estimates' relative performance, we followed the work of Amorim et al. (2011). As in Amorim et al. (2011), we computed the integrated absolute bias, integrated variance and the integrated MSE of the estimates. For each simulated setting ($\theta = 0$ and $\theta = 1$) we derived the analytic expression of $p_{ij}(s,t)$ so that the bias and the MSE of the estimator could be examined. K = 1000 data sets were generated, with three different sample sizes $n = 50$, $n = 100$ and $n = 200$.

Let $\hat{p}^k_{ij}(s,t)$ denote the estimated transition probability based on the k^{th} generated data set. For each fixed (s,t) we obtained the mean for all generated data sets, $\overline{\hat{p}_{ij}(s,t)} = \frac{1}{K}\sum_{k=1}^{K} \hat{p}^k_{ij}(s,t)$. We then computed the pointwise estimates of the bias, variance, MSE and L1 distance as:

$$\widehat{bias}(s,t) = p_{ij}(s,t) - \overline{\hat{p}_{ij}(s,t)}$$

$$\widehat{var}(\hat{p}_{ij}(s,t)) = \frac{1}{K-1}\sum_{k=1}^{K}[\hat{p}_{ij}^{k}(s,t) - \overline{\hat{p}_{ij}(s,t)}]^2$$

$$\widehat{MSE}(\hat{p}_{ij}(s,t)) = \frac{1}{K}\sum_{k=1}^{K}[\hat{p}_{ij}^{k}(s,t) - p_{ij}(s,t)]^2$$

$$\widehat{L1}(\hat{p}_{ij}(s,t)) = \frac{1}{K}\sum_{k=1}^{K}|\hat{p}_{ij}^{k}(s,t) - p_{ij}(s,t)|$$

To summarize the results we also calculated the integrated absolute bias (BIAS), integrated variance (VAR), the integrated MSE (IMSE) and the integrated L1 distance (L1), defined in Table 3.1. We fixed the values of s using the quantiles 0.25, 0.5 and 0.75 of the exponential distribution with rate 1. The results given in Tables 3.2 to 3.5 were obtained by numerical integration on the interval $[s, t_1]$ with $t_1 = 4$, taking a grid of step $\delta = 0.05$.

Table 3.1: Summary statistics measuring bias, variance, Mean Square Error and L1 distance.

Statistic	Definition	Estimator
Integrated Absolute Bias	$\int_s^{t_1} \lvert bias(s,t)\rvert dt$	$\sum_{t=s}^{t_1} \lvert\widehat{bias}(s,t)\rvert\delta$
Integrated Variance	$\int_s^{t_1} var(\hat{p}_{ij}(s,t))dt$	$\sum_{t=s}^{t_1} \widehat{var}(\hat{p}_{ij}(s,t))\delta$
Integrated MSE	$\int_s^{t_1} MSE(\hat{p}_{ij}(s,t))dt$	$\sum_{t=s}^{t_1} \widehat{MSE}(\hat{p}_{ij}(s,t))\delta$
Integrated L1	$\int_s^{t_1} L1(\hat{p}_{ij}(s,t))dt$	$\sum_{t=s}^{t_1} \widehat{L1}(\hat{p}_{ij}(s,t))\delta$

In Tables 3.2 to 3.5 we report the results for the summary statistics attained by the proposed estimator when based on several presmoothing functions (P-AJ), for all scenarios. In all tables, the row labeled with m corresponds to presmoothing with the true function which is unrealistic in practice, because this function will be typically unknown. However, this row represents a “gold standard” the other methods can be compared to. The row labeled with $m(.;\beta,\gamma)$ corresponds to a semiparametric estimator which is obtained using a presmoothing based on a parametric family which contains the true m. Specifically, we consider a logistic model with the preliminary transformation of the conditioning variables $\widetilde{Z} = z, \widetilde{T} = t$ shown before. In order to investigate the robustness of the proposed estimator with respect to misspecifications of the binary regression family, we considered also presmoothing via standard logistic models, without any preliminary transformation of the gap times. This is labeled with $m(.,\xi)$. Note that the true m does not belong to this parametric family. Finally, we also report the results pertaining to the Aalen-Johansen estimator, which corresponds to the situation with no presmoothing at all. This is labeled in the Tables as AJ.

It is obvious from the analysis of Tables 3.2 to 3.5, that presmoothing leads to estimators with smaller variance and thus attaining better results with regard to the integrated MSE. As expected, the (integrated) MSE, bias, L1 norm and variance of the estimated transition probabilities always decrease with an increasing sample size, while they increase with the censoring degree. The estimator which makes use of the true m is the one with the best performance. However, this estimator is unrealistic since in practice one has to estimate the function m. In general, the lowest errors among the realistic versions of the estimators correspond to the estimator based on the correctly specified parametric family, $m(.;\beta,\gamma)$. However, the presmoothed estimator based on the wrong parametric model $m(.;\xi)$ is still (much) better than AJ. This means that it is worthwhile doing some presmoothing even when we are not completely sure about the parametric family.

Results shown in the Tables 3.2 to 3.5 support the idea that presmoothing leads to variance improvement. When compared to the estimators based on presmoothing, the relative efficiency (defined as the quotient between the two integrated MSEs) of the Aalen-Johansen estimator is always below 1. For higher values of s, where the censoring effects are stronger, the relative efficiency can drop below 50%. These findings agree with the results obtained by Amorim et al. (2011) and support the intuition that the use of presmoothing for the estimation of transition probabilities will be more clearly seen in the presence of large censoring degrees.

In general, presmoothing introduces some bias in estimation, while reducing the variance. This bias component is larger when there is some misspecification in the chosen parametric model. Our simulation results serve to illustrate this issue too. Indeed, it is seen that, despite of offering a smaller IMSE, the bias associated to the semiparametric Aalen-Johansen estimator is sometimes larger than that of the original Aalen-Johansen.

Tables 3.2 and 3.3 show a systematic bias for all estimators of the transition probabilities $p_{12}(s,t)$ and $p_{22}(s,t)$. This is because these tables report the results attained when generating data from a dependency scenario and therefore reflects the failure of the Markov assumption. To illustrate these features we present in Figures 3.1 to 3.6 the graphical average results for the two methods (AJ and P-AJ corresponding to presmoothing via standard logistic models, $m(.,\xi)$). These figures plot the data generating functions and pointwise 95% oscillation limits of the estimates $p_{22}(s,t)$, for sample sizes of $n = 200$ with percentages of censored data obtained using $C \sim U[0,3]$. The good performance of the resulting estimates (for both methods) is evident for independent gap times ($\theta = 0$), recovering the functional forms of the corresponding true curves very successfully. However, a systematic bias of $p_{12}(s,t)$ and $p_{22}(s,t)$ in the dependent scenario ($\theta = 1$) is also clear, see Figure 3.3 and 3.5. This bias is much more evident when s is large, in agreement with the amount of false information introduced by the Markov condition (which increases with s). In all scenarios, the use of the presmoothing yields estimators with less variability.

To better understand the finite sample performance of these estimators we illustrate in Figures 3.7 to 3.11 the behavior of the MSE, variance and efficiency over a variety of scenarios. We have considered two simulation scenarios (dependent and independent gap times) with four sample sizes (50,100, 150 and 200) and two censoring levels ($U[0,3]$ and $U[0,4]$). Figures 3.7 and 3.8 show the behavior of the MSE for the dependency scenario with uniform censoring $U[0,3]$.

Table 3.2: Integrated absolute bias, integrated variance and the integrated Mean Square Error of $\hat{p}_{ij}(s,.)$ along 1,000 trials, case $\theta = 1$ and $C \backsim U[0,4]$.

	n	50				100				200			
$P_{ij}(s,t)$	$Method$	$IMSE$	$BIAS$	VAR	$L1$	$IMSE$	$BIAS$	VAR	$L1$	$IMSE$	$BIAS$	VAR	$L1$
$P_{11}(0.2877,t)$	$m(.;\beta,\gamma)$	0.01864	0.04079	0.01769	0.20299	0.00878	0.01909	0.00855	0.14024	0.00452	0.01357	0.00443	0.10110
	$m(.;\xi)$	0.01878	0.04246	0.01800	0.20297	0.00883	0.02126	0.00868	0.14011	0.00460	0.01582	0.00452	0.10166
	AJ	0.02123	0.02158	0.02092	0.22117	0.01028	0.00955	0.01022	0.15440	0.00537	0.00800	0.00533	0.11131
	m	0.01312	0.02146	0.01280	0.16731	0.00665	0.01079	0.00656	0.11945	0.00344	0.00671	0.00342	0.08669
$P_{12}(0.2877,t)$	$m(.;\beta,\gamma)$	0.02174	0.03026	0.02141	0.22326	0.01121	0.02500	0.01100	0.16200	0.00612	0.02916	0.00584	0.11929
	$m(.;\xi)$	0.02269	0.02669	0.02243	0.22802	0.01170	0.02092	0.01153	0.16527	0.00632	0.02470	0.00609	0.12118
	AJ	0.02702	0.02891	0.02677	0.24970	0.01393	0.02727	0.01370	0.17924	0.00732	0.03171	0.00701	0.12949
	m	0.01881	0.02859	0.01857	0.20834	0.00994	0.02612	0.00972	0.15167	0.00547	0.03169	0.00516	0.11322
$P_{22}(0.2877,t)$	$m(.;\beta,\gamma)$	0.04065	0.18028	0.03067	0.27812	0.02499	0.18808	0.01403	0.23513	0.01759	0.18551	0.00678	0.20948
	$m(.;\xi)$	0.04094	0.17961	0.03104	0.27923	0.02509	0.18810	0.01419	0.23574	0.01752	0.18515	0.00682	0.20937
	AJ	0.04237	0.16216	0.03398	0.27813	0.02599	0.18096	0.01567	0.23554	0.01812	0.18317	0.00752	0.20998
	m	0.03502	0.16667	0.02628	0.25642	0.02215	0.18047	0.01192	0.22462	0.01635	0.18258	0.00577	0.20446
$P_{11}(0.6931,t)$	$m(.;\beta,\gamma)$	0.03168	0.05996	0.02947	0.24945	0.01404	0.02708	0.01352	0.16863	0.00734	0.01871	0.00713	0.12204
	$m(.;\xi)$	0.03197	0.06149	0.03016	0.24970	0.01416	0.03022	0.01455	0.16873	0.00747	0.01962	0.00734	0.12294
	AJ	0.03750	0.03148	0.03677	0.27727	0.01738	0.01321	0.01724	0.19121	0.00907	0.01069	0.00898	0.13696
	m	0.02099	0.03053	0.02026	0.20057	0.01061	0.01558	0.01040	0.14329	0.00540	0.00855	0.00534	0.10299
$P_{12}(0.6931,t)$	$m(.;\beta,\gamma)$	0.03353	0.05172	0.03256	0.26512	0.01739	0.05133	0.01644	0.19139	0.00994	0.05500	0.00882	0.14398
	$m(.;\xi)$	0.03502	0.04245	0.03435	0.27045	0.01803	0.04047	0.01740	0.19428	0.01003	0.04502	0.00926	0.14453
	AJ	0.04290	0.05486	0.04186	0.29885	0.02212	0.05527	0.02104	0.21282	0.01204	0.05855	0.01080	0.15687
	m	0.02989	0.05345	0.02886	0.25128	0.01623	0.05223	0.01526	0.18415	0.00934	0.05884	0.00810	0.10007
$P_{22}(0.6931,t)$	$m(.;\beta,\gamma)$	0.04377	0.16461	0.03463	0.28657	0.02471	0.15916	0.01617	0.22272	0.01634	0.15786	0.00791	0.18980
	$m(.;\xi)$	0.04482	0.16395	0.03568	0.29072	0.02515	0.15866	0.01656	0.22495	0.01652	0.15726	0.00806	0.19097
	AJ	0.05003	0.14281	0.04313	0.30182	0.02702	0.14921	0.01949	0.23006	0.01738	0.15153	0.00956	0.19308
	m	0.03403	0.14646	0.02685	0.25163	0.02029	0.15018	0.01264	0.20434	0.01438	0.15295	0.00641	0.17982
$P_{11}(1.3863,t)$	$m(.;\beta,\gamma)$	0.07510	0.10977	0.06691	0.33918	0.03363	0.05112	0.03160	0.23071	0.01740	0.03539	0.01659	0.16655
	$m(.;\xi)$	0.07165	0.09970	0.06577	0.33124	0.03213	0.04383	0.03119	0.22548	0.01680	0.02807	0.01647	0.16456
	AJ	0.09922	0.06458	0.09584	0.39775	0.04451	0.02572	0.04387	0.27073	0.02321	0.01962	0.02288	0.19487
	m	0.04581	0.06145	0.04256	0.26275	0.02268	0.03088	0.02176	0.18758	0.01152	0.01697	0.01126	0.13406
$P_{12}(1.3863,t)$	$m(.;\beta,\gamma)$	0.06659	0.07348	0.06401	0.33574	0.03530	0.08320	0.03225	0.24506	0.02043	0.08684	0.01714	0.18471
	$m(.;\xi)$	0.06926	0.06745	0.06722	0.34113	0.03643	0.07357	0.03415	0.24764	0.02048	0.07735	0.01789	0.18434
	AJ	0.08594	0.08094	0.08282	0.38015	0.04449	0.08388	0.04140	0.27086	0.02468	0.08903	0.02121	0.20031
	m	0.06411	0.07731	0.06128	0.32803	0.03538	0.07969	0.03259	0.24512	0.02058	0.08970	0.01706	0.18585
$P_{22}(1.3863,t)$	$m(.;\beta,\gamma)$	0.07104	0.15190	0.05960	0.32833	0.03372	0.12455	0.02667	0.23020	0.01881	0.11085	0.01328	0.17492
	$m(.;\xi)$	0.07763	0.16072	0.06520	0.34277	0.03687	0.13391	0.02867	0.24064	0.02084	0.12039	0.01417	0.18441
	AJ	0.09115	0.11812	0.08482	0.37625	0.04292	0.10587	0.03798	0.26149	0.02227	0.09872	0.01794	0.19022
	m	0.04746	0.11902	0.04076	0.26792	0.02412	0.10993	0.01875	0.19666	0.01413	0.09979	0.00972	0.15288

Table 3.3: Integrated absolute bias, integrated variance and the integrated Mean Square Error of $\hat{p}_{ij}(s,.)$ along 1,000 trials, case $\theta = 1$ and $C \backsim U[0,3]$.

	n	50				100				200			
$P_{ij}(s,t)$	$Method$	$IMSE$	$BIAS$	VAR	$L1$	$IMSE$	$BIAS$	VAR	$L1$	$IMSE$	$BIAS$	VAR	$L1$
$P_{11}(0.2877,t)$	$m(.;\beta,\gamma)$	0.02953	0.10624	0.02315	0.25453	0.01473	0.07644	0.01102	0.18122	0.00789	0.05496	0.00581	0.13110
	$m(.;\xi)$	0.02632	0.09326	0.02210	0.24119	0.01188	0.05811	0.01025	0.16371	0.00571	0.03641	0.00514	0.11291
	AJ	0.03275	0.07520	0.02880	0.27554	0.01738	0.05731	0.01481	0.20077	0.00960	0.04389	0.00799	0.14793
	m	0.01576	0.06984	0.01220	0.19366	0.00829	0.05236	0.00603	0.14179	0.00476	0.04450	0.00316	0.10727
$P_{12}(0.2877,t)$	$m(.;\beta,\gamma)$	0.03195	0.07673	0.02826	0.26335	0.01770	0.06396	0.01549	0.19799	0.01073	0.05867	0.00875	0.15037
	$m(.;\xi)$	0.03225	0.06543	0.02951	0.26496	0.01670	0.04196	0.01565	0.19267	0.00923	0.03519	0.00859	0.14224
	AJ	0.04214	0.07597	0.03878	0.30293	0.02353	0.06286	0.02163	0.22571	0.01424	0.05894	0.01241	0.17096
	m	0.02367	0.06837	0.02104	0.23150	0.01404	0.06356	0.01204	0.17697	0.00913	0.06058	0.00727	0.13999
$P_{22}(0.2877,t)$	$m(.;\beta,\gamma)$	0.05085	0.23753	0.03434	0.32685	0.02842	0.21001	0.01528	0.26141	0.02056	0.20695	0.00784	0.23521
	$m(.;\xi)$	0.05044	0.23121	0.03479	0.32366	0.02757	0.20100	0.01533	0.25427	0.01920	0.19292	0.00777	0.22262
	AJ	0.05321	0.20781	0.04065	0.32557	0.02933	0.19422	0.01801	0.25910	0.02112	0.19866	0.00934	0.23384
	m	0.03757	0.20957	0.02484	0.28746	0.02325	0.19957	0.01140	0.24403	0.01729	0.19522	0.00594	0.22072
$P_{11}(0.6931,t)$	$m(.;\beta,\gamma)$	0.05636	0.16091	0.04151	0.32809	0.02877	0.11555	0.02023	0.23633	0.01530	0.08120	0.01059	0.17049
	$m(.;\xi)$	0.04874	0.14156	0.03894	0.30725	0.02217	0.08628	0.01845	0.20967	0.01026	0.05163	0.00902	0.14262
	AJ	0.06414	0.11352	0.05495	0.36089	0.03437	0.08654	0.02848	0.26476	0.01884	0.06642	0.01514	0.19472
	m	0.02725	0.10585	0.01898	0.24243	0.01502	0.07888	0.00985	0.18085	0.00876	0.06510	0.00515	0.13766
$P_{12}(0.6931,t)$	$m(.;\beta,\gamma)$	0.04722	0.08383	0.04321	0.30800	0.02693	0.07881	0.02411	0.23421	0.01647	0.07472	0.01388	0.17838
	$m(.;\xi)$	0.04795	0.06495	0.04518	0.31015	0.02577	0.05767	0.02435	0.22921	0.01470	0.05209	0.01351	0.17128
	AJ	0.06564	0.08976	0.06141	0.35886	0.03744	0.08014	0.03469	0.26963	0.02259	0.07569	0.02003	0.20522
	m	0.03907	0.08358	0.03571	0.28252	0.02342	0.08059	0.02059	0.21697	0.01528	0.07817	0.01260	0.17220
$P_{22}(0.6931,t)$	$m(.;\beta,\gamma)$	0.07295	0.25545	0.04772	0.37608	0.04069	0.22121	0.02272	0.28957	0.02646	0.20766	0.01088	0.24299
	$m(.;\xi)$	0.07299	0.24931	0.04976	0.37558	0.03866	0.21119	0.02291	0.28075	0.02316	0.18830	0.01075	0.22489
	AJ	0.07732	0.20713	0.06053	0.39031	0.04333	0.19808	0.02917	0.29899	0.02816	0.19499	0.01456	0.24824
	m	0.04427	0.20789	0.02782	0.30883	0.02715	0.19887	0.01286	0.25162	0.01935	0.18761	0.00665	0.21849
$P_{11}(1.3863,t)$	$m(.;\beta,\gamma)$	0.15828	0.29155	0.10488	0.48913	0.08415	0.21623	0.05218	0.35770	0.04715	0.15201	0.02922	0.26206
	$m(.;\xi)$	0.11915	0.22087	0.08857	0.42771	0.05278	0.12627	0.04157	0.28755	0.02464	0.06960	0.02130	0.19581
	AJ	0.20944	0.23542	0.16998	0.57574	0.11095	0.17679	0.08648	0.41795	0.06199	0.13227	0.04708	0.31014
	m	0.07598	0.21746	0.04090	0.35990	0.04199	0.15815	0.02099	0.26782	0.02568	0.12626	0.01136	0.20766
$P_{12}(1.3863,t)$	$m(.;\beta,\gamma)$	0.08819	0.07494	0.08539	0.38556	0.05252	0.07908	0.04934	0.29742	0.03167	0.07506	0.02885	0.22588
	$m(.;\xi)$	0.08883	0.07613	0.08580	0.38667	0.05214	0.09063	0.04745	0.29747	0.03181	0.09885	0.02603	0.23114
	AJ	0.12562	0.07176	0.12305	0.45163	0.07381	0.08038	0.07056	0.34183	0.04413	0.07355	0.04143	0.26023
	m	0.09009	0.07455	0.08731	0.38742	0.05502	0.07825	0.05194	0.29976	0.03535	0.07559	0.03251	0.23500
$P_{22}(1.3863,t)$	$m(.;\beta,\gamma)$	0.16509	0.35887	0.09329	0.50415	0.08891	0.25952	0.04873	0.36668	0.05575	0.23567	0.02312	0.29257
	$m(.;\xi)$	0.16617	0.34178	0.10439	0.50378	0.07845	0.23345	0.04957	0.34449	0.04181	0.19140	0.02324	0.25453
	AJ	0.20923	0.29425	0.15625	0.57797	0.10352	0.21472	0.07417	0.40258	0.06376	0.21087	0.03698	0.31641
	m	0.08661	0.28003	0.03956	0.38979	0.04982	0.22075	0.01907	0.29484	0.03531	0.20314	0.00957	0.24717

Table 3.4: Integrated absolute bias, integrated variance and the integrated Mean Square Error of $\hat{p}_{ij}(s, .)$ along 1,000 trials, case $\theta = 0$ and $C \backsim U[0, 4]$.

	n	50				100				200			
$P_{ij}(s,t)$	$Method$	$IMSE$	$BIAS$	VAR	$L1$	$IMSE$	$BIAS$	VAR	$L1$	$IMSE$	$BIAS$	VAR	$L1$
$P_{11}(0.1438410, t)$	$m(.;\beta,\gamma)$	0.00838	0.02707	0.00809	0.11934	0.00402	0.01428	0.00393	0.08222	0.00199	0.00884	0.00196	0.05754
	$m(.;\xi)$	0.00834	0.02676	0.00807	0.11862	0.00400	0.01280	0.00393	0.08148	0.00198	0.00754	0.00196	0.05688
	AJ	0.00919	0.01602	0.00910	0.12256	0.00442	0.00933	0.00438	0.08603	0.00219	0.00603	0.00217	0.06073
	m	0.00712	0.01665	0.00701	0.10465	0.00360	0.00924	0.00357	0.07483	0.00178	0.00589	0.00177	0.05274
$P_{12}(0.1438410, t)$	$m(.;\beta,\gamma)$	0.01373	0.02980	0.01327	0.17988	0.00695	0.01800	0.00681	0.12696	0.00332	0.00827	0.00327	0.08855
	$m(.;\xi)$	0.01388	0.02675	0.01353	0.18053	0.00705	0.01811	0.00695	0.12779	0.00338	0.00883	0.00335	0.08933
	AJ	0.01509	0.01858	0.01494	0.19063	0.00771	0.01066	0.00766	0.13600	0.00375	0.00444	0.00374	0.09527
	m	0.01096	0.01962	0.01079	0.15451	0.00587	0.01081	0.00581	0.11415	0.00291	0.00480	0.00290	0.08099
$P_{22}(0.1438410, t)$	$m(.;\beta,\gamma)$	0.03485	0.03926	0.03406	0.27344	0.01570	0.02428	0.01547	0.18287	0.00816	0.01279	0.00808	0.13284
	$m(.;\xi)$	0.03524	0.03541	0.03460	0.27422	0.01587	0.02268	0.01570	0.18357	0.00821	0.01549	0.00815	0.13319
	AJ	0.03825	0.02212	0.03803	0.28866	0.01761	0.01110	0.01756	0.19630	0.00907	0.00601	0.00906	0.14168
	m	0.02648	0.02187	0.02625	0.22917	0.01260	0.01044	0.01254	0.15853	0.00666	0.00686	0.00663	0.11623
$P_{11}(0.3465736, t)$	$m(.;\beta,\gamma)$	0.01361	0.04000	0.01295	0.15009	0.00651	0.02182	0.00631	0.10365	0.00315	0.01291	0.00309	0.07204
	$m(.;\xi)$	0.01354	0.03944	0.01292	0.14909	0.00648	0.01945	0.00631	0.10260	0.00314	0.01082	0.00309	0.07110
	AJ	0.01526	0.02288	0.01505	0.15467	0.00724	0.01392	0.00716	0.10915	0.00355	0.00833	0.00352	0.07679
	m	0.01121	0.02422	0.01095	0.12959	0.00572	0.01372	0.00564	0.09296	0.00279	0.00816	0.00276	0.06547
$P_{12}(0.3465736, t)$	$m(.;\beta,\gamma)$	0.01897	0.03421	0.01836	0.20509	0.00941	0.01940	0.00923	0.14380	0.00452	0.00872	0.00446	0.10082
	$m(.;\xi)$	0.01926	0.02938	0.01881	0.20612	0.00959	0.02009	0.00945	0.14495	0.00461	0.00891	0.00458	0.10179
	AJ	0.02111	0.02110	0.02090	0.21863	0.01053	0.01259	0.01045	0.15469	0.00513	0.00495	0.00512	0.10868
	m	0.01563	0.02167	0.01540	0.17913	0.00815	0.01290	0.00807	0.13082	0.00404	0.00487	0.00402	0.09301
$P_{22}(0.3465736, t)$	$m(.;\beta,\gamma)$	0.03453	0.04611	0.03336	0.27081	0.01648	0.02898	0.01612	0.18627	0.00836	0.01563	0.00824	0.13367
	$m(.;\xi)$	0.03496	0.04375	0.03398	0.27229	0.01673	0.02831	0.01644	0.18740	0.00848	0.01709	0.00838	0.13457
	AJ	0.03879	0.02603	0.03845	0.28994	0.01874	0.01389	0.01865	0.20166	0.00959	0.00743	0.00956	0.14502
	m	0.02506	0.02703	0.02468	0.22237	0.01251	0.01212	0.01241	0.15849	0.00659	0.00781	0.00655	0.11577
$P_{11}(0.6931472, t)$	$m(.;\beta,\gamma)$	0.03292	0.07945	0.03021	0.22659	0.01543	0.04093	0.01460	0.15586	0.00703	0.02497	0.00676	0.10566
	$m(.;\xi)$	0.03237	0.07819	0.02985	0.22420	0.01521	0.03796	0.01453	0.15345	0.00694	0.02117	0.00675	0.10358
	AJ	0.03878	0.04530	0.03786	0.23450	0.01758	0.02712	0.01724	0.16521	0.00823	0.01612	0.00812	0.11438
	m	0.02502	0.04954	0.02393	0.18901	0.01280	0.02740	0.01244	0.13609	0.00605	0.01610	0.00594	0.09438
$P_{12}(0.6931472, t)$	$m(.;\beta,\gamma)$	0.03348	0.04312	0.03259	0.25681	0.01716	0.02217	0.01688	0.18327	0.00796	0.00992	0.00788	0.12704
	$m(.;\xi)$	0.03406	0.03520	0.03345	0.25817	0.01751	0.02291	0.01733	0.18470	0.00814	0.00865	0.00812	0.12820
	AJ	0.03699	0.03089	0.03663	0.27351	0.01911	0.01463	0.01900	0.19729	0.00905	0.00719	0.00903	0.13718
	m	0.02800	0.03097	0.02760	0.22821	0.01559	0.01603	0.01545	0.17015	0.00736	0.00645	0.00733	0.11954
$P_{22}0.6931472, t)$	$m(.;\beta,\gamma)$	0.04656	0.06709	0.04407	0.30324	0.02035	0.03591	0.01967	0.20076	0.01041	0.02166	0.01016	0.14425
	$m(.;\xi)$	0.04775	0.06316	0.04563	0.30664	0.02092	0.03790	0.02037	0.20326	0.01078	0.02540	0.01056	0.14672
	AJ	0.05389	0.03560	0.05318	0.32949	0.02449	0.01699	0.02431	0.22332	0.01249	0.01136	0.01242	0.15989
	m	0.03105	0.03742	0.03025	0.24026	0.01475	0.01626	0.01455	0.16716	0.00791	0.01171	0.00783	0.12323

Table 3.5: Integrated absolute bias, integrated variance and the integrated Mean Square Error of $\hat{p}_{ij}(s,.)$ along 1,000 trials, case $\theta = 0$ and $C \backsim U[0,3]$.

	n	50				100				200			
$P_{ij}(s,t)$	$Method$	$IMSE$	$BIAS$	VAR	$L1$	$IMSE$	$BIAS$	VAR	$L1$	$IMSE$	$BIAS$	VAR	$L1$
$P_{11}(0.1438410,t)$	$m(.;\beta,\gamma)$	0.01011	0.05216	0.00903	0.13916	0.00465	0.02690	0.00431	0.09372	0.00232	0.01836	0.00218	0.06636
	$m(.;\xi)$	0.00987	0.05049	0.00890	0.13673	0.00448	0.02401	0.00422	0.09088	0.00222	0.01329	0.00214	0.06305
	AJ	0.01114	0.03199	0.01072	0.13939	0.00523	0.01876	0.00506	0.09742	0.00264	0.01474	0.00255	0.07044
	m	0.00721	0.03439	0.00671	0.11514	0.00358	0.01954	0.00339	0.08055	0.00188	0.01396	0.00178	0.05859
$P_{12}(0.14384410,t)$	$m(.;\beta,\gamma)$	0.01794	0.06344	0.01539	0.20468	0.01053	0.05551	0.00851	0.15756	0.00562	0.03775	0.00458	0.11471
	$m(.;\xi)$	0.01721	0.05431	0.01536	0.20061	0.00933	0.04606	0.00819	0.14862	0.00465	0.02610	0.00429	0.10501
	AJ	0.02001	0.04733	0.01853	0.22012	0.01182	0.04105	0.01052	0.17007	0.00653	0.03120	0.00571	0.12539
	m	0.01256	0.05168	0.01079	0.17184	0.00712	0.03777	0.00591	0.13161	0.00401	0.03306	0.00312	0.09939
$P_{22}(0.14384410,t)$	$m(.;\beta,\gamma)$	0.04573	0.09080	0.04120	0.31646	0.02252	0.07677	0.01925	0.22833	0.01074	0.04930	0.00916	0.15877
	$m(.;\xi)$	0.04442	0.08009	0.04103	0.31062	0.02068	0.06389	0.01878	0.21654	0.00936	0.03394	0.00880	0.14621
	AJ	0.05071	0.05839	0.04834	0.33645	0.02515	0.05380	0.02319	0.24467	0.01271	0.04037	0.01150	0.17462
	m	0.03039	0.06363	0.02755	0.25734	0.01440	0.05175	0.01255	0.18305	0.00727	0.03926	0.00602	0.13198
$P_{11}(0.3465736,t)$	$m(.;\beta,\gamma)$	0.01766	0.07802	0.01519	0.18081	0.00813	0.04089	0.00733	0.12200	0.00388	0.02710	0.00354	0.08509
	$m(.;\xi)$	0.01716	0.07535	0.01495	0.17731	0.00777	0.03661	0.00716	0.11783	0.00364	0.01933	0.00346	0.08020
	AJ	0.01946	0.04699	0.01853	0.18033	0.00938	0.02837	0.00898	0.12761	0.00450	0.02088	0.00430	0.09097
	m	0.01185	0.05129	0.01071	0.14709	0.00601	0.02955	0.00557	0.10369	0.00309	0.02069	0.00288	0.07501
$P_{12}(0.3465736,t)$	$m(.;\beta,\gamma)$	0.02448	0.07199	0.02112	0.23327	0.01451	0.06295	0.01180	0.17975	0.00767	0.04143	0.00628	0.13062
	$m(.;\xi)$	0.02359	0.05989	0.02117	0.22932	0.01296	0.05208	0.01142	0.16969	0.00634	0.02957	0.00587	0.11954
	AJ	0.02802	0.05509	0.02599	0.25259	0.01651	0.04770	0.01472	0.19459	0.00898	0.03519	0.00786	0.14334
	m	0.01787	0.05939	0.01548	0.19936	0.01006	0.04377	0.00840	0.15132	0.00558	0.03673	0.00437	0.11391
$P_{22}(0.3465736,t)$	$m(.;\beta,\gamma)$	0.04739	0.10555	0.04069	0.31873	0.02563	0.09178	0.02073	0.23817	0.01244	0.05873	0.01007	0.16668
	$m(.;\xi)$	0.04573	0.09499	0.04068	0.31241	0.02300	0.07892	0.02004	0.22490	0.01041	0.04130	0.00953	0.15216
	AJ	0.05282	0.07153	0.04933	0.34248	0.02863	0.06451	0.02571	0.25701	0.01447	0.04638	0.01268	0.18318
	m	0.02883	0.07802	0.02464	0.25234	0.01509	0.05921	0.01235	0.18604	0.00787	0.04693	0.00599	0.13592
$P_{11}(0.6931472,t)$	$m(.;\beta,\gamma)$	0.05074	0.16039	0.04031	0.29141	0.02131	0.08110	0.01808	0.19131	0.00968	0.05098	0.00834	0.13232
	$m(.;\xi)$	0.04802	0.15437	0.03876	0.28290	0.01959	0.07031	0.01720	0.18189	0.00862	0.03673	0.00789	0.12190
	AJ	0.05762	0.10097	0.05343	0.29049	0.02562	0.05671	0.02394	0.20188	0.01161	0.03948	0.01081	0.14210
	m	0.02894	0.10975	0.02389	0.22840	0.01380	0.05862	0.01200	0.15725	0.00720	0.03949	0.00637	0.11458
$P_{12}(0.6931472,t)$	$m(.;\beta,\gamma)$	0.04507	0.08466	0.04044	0.29743	0.02650	0.07508	0.02238	0.22817	0.01397	0.04836	0.01182	0.16650
	$m(.;\xi)$	0.04357	0.06638	0.04038	0.29273	0.02376	0.05956	0.02157	0.21536	0.01158	0.03460	0.01090	0.15219
	AJ	0.05378	0.06815	0.05080	0.32565	0.03093	0.05686	0.02813	0.24811	0.01627	0.04390	0.01452	0.18248
	m	0.03633	0.07501	0.03270	0.26505	0.01927	0.05530	0.01666	0.19652	0.01094	0.04572	0.00899	0.14931
$P_{22}(0.6931472,t)$	$m(.;\beta,\gamma)$	0.06945	0.15333	0.05520	0.36666	0.03880	0.12909	0.02881	0.27527	0.01897	0.08283	0.01421	0.19413
	$m(.;\xi)$	0.06660	0.13952	0.05570	0.35867	0.03408	0.11298	0.02785	0.25864	0.01517	0.06032	0.01330	0.17557
	AJ	0.07825	0.10236	0.07088	0.39785	0.04380	0.09094	0.03791	0.30107	0.02278	0.06504	0.01923	0.21803
	m	0.03859	0.10977	0.02990	0.28212	0.02029	0.08309	0.01481	0.20801	0.01138	0.06728	0.00761	0.15700

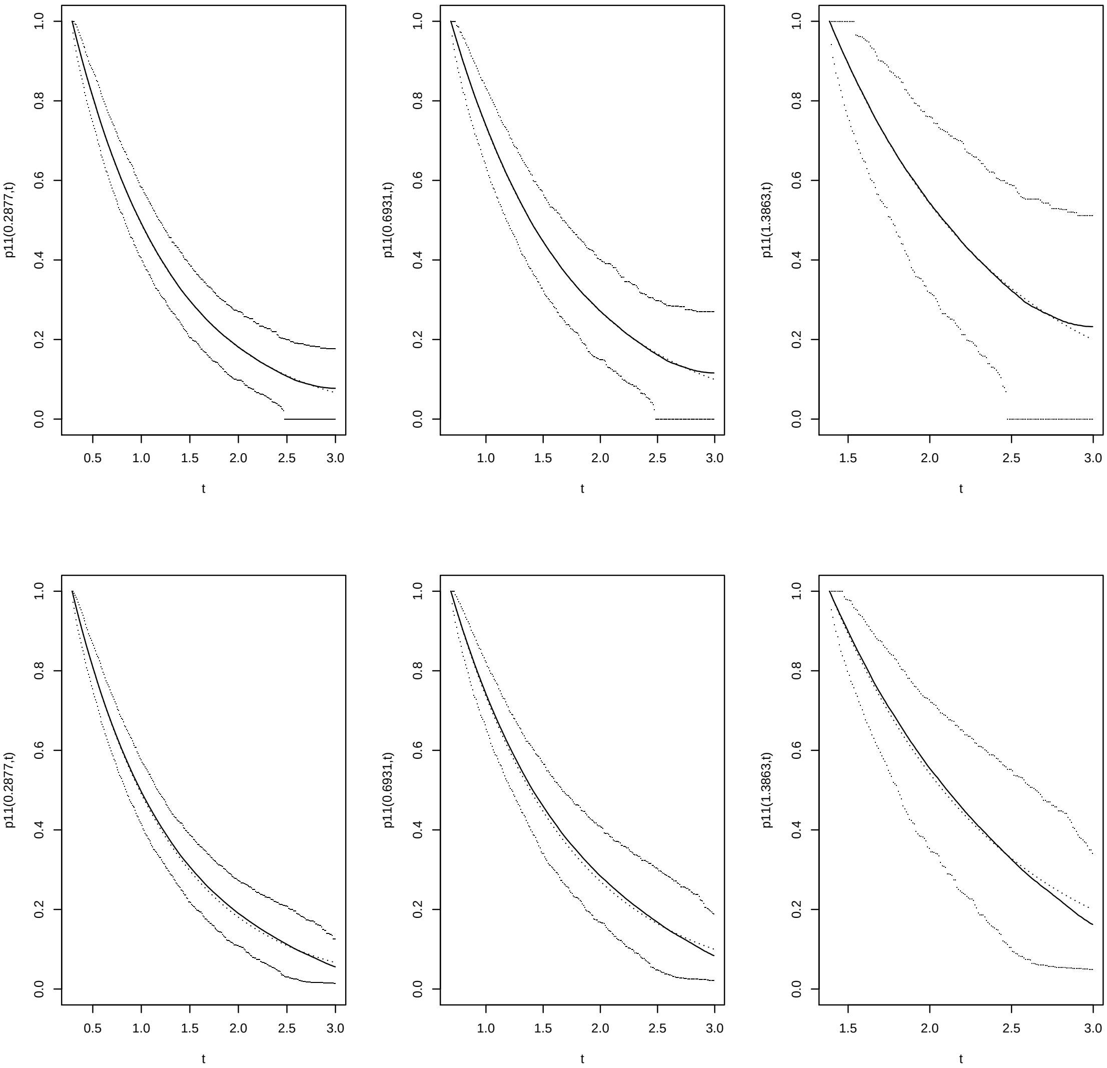

Figure 3.1: True $p_{11}(s,t)$ (dotted line), average estimator (solid line), and 95% oscillation limits of the AJ estimates (first row) and P-AJ (second row) for $s = 0.2877$, $s = 0.6931$ and $s = 1.3863$. Estimates with $n = 200$ and $U[0,3]$ censoring. Dependency scenario.

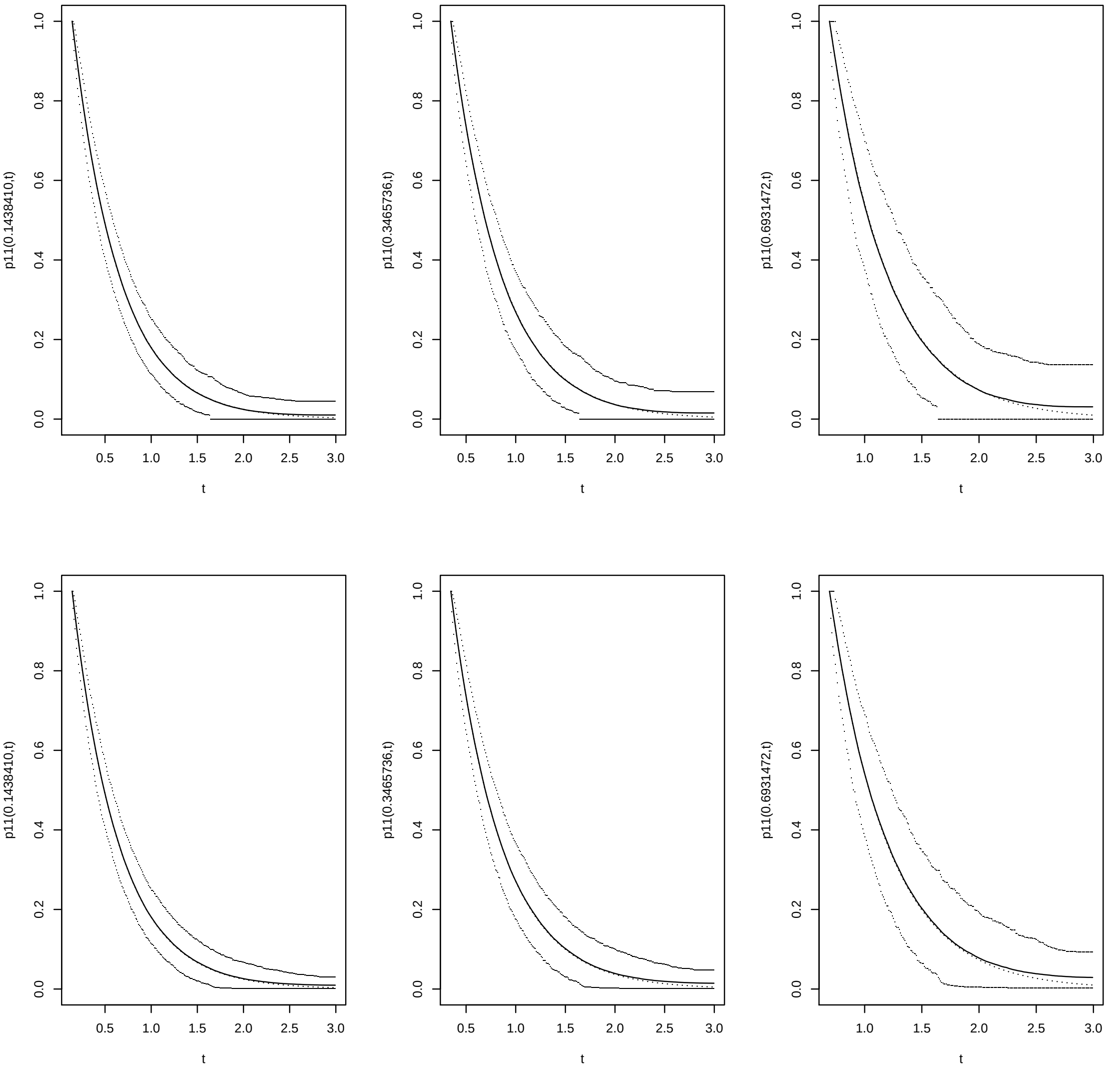

Figure 3.2: True $p_{11}(s,t)$ (dotted line), average estimator (solid line), and 95% oscillation limits of the AJ estimates (first row) and P-AJ (second row) for $s = 0.1438$, $s = 0.3466$ and $s = 0.6931$. Estimates with $n = 200$ and $U[0,3]$ censoring. Independency scenario.

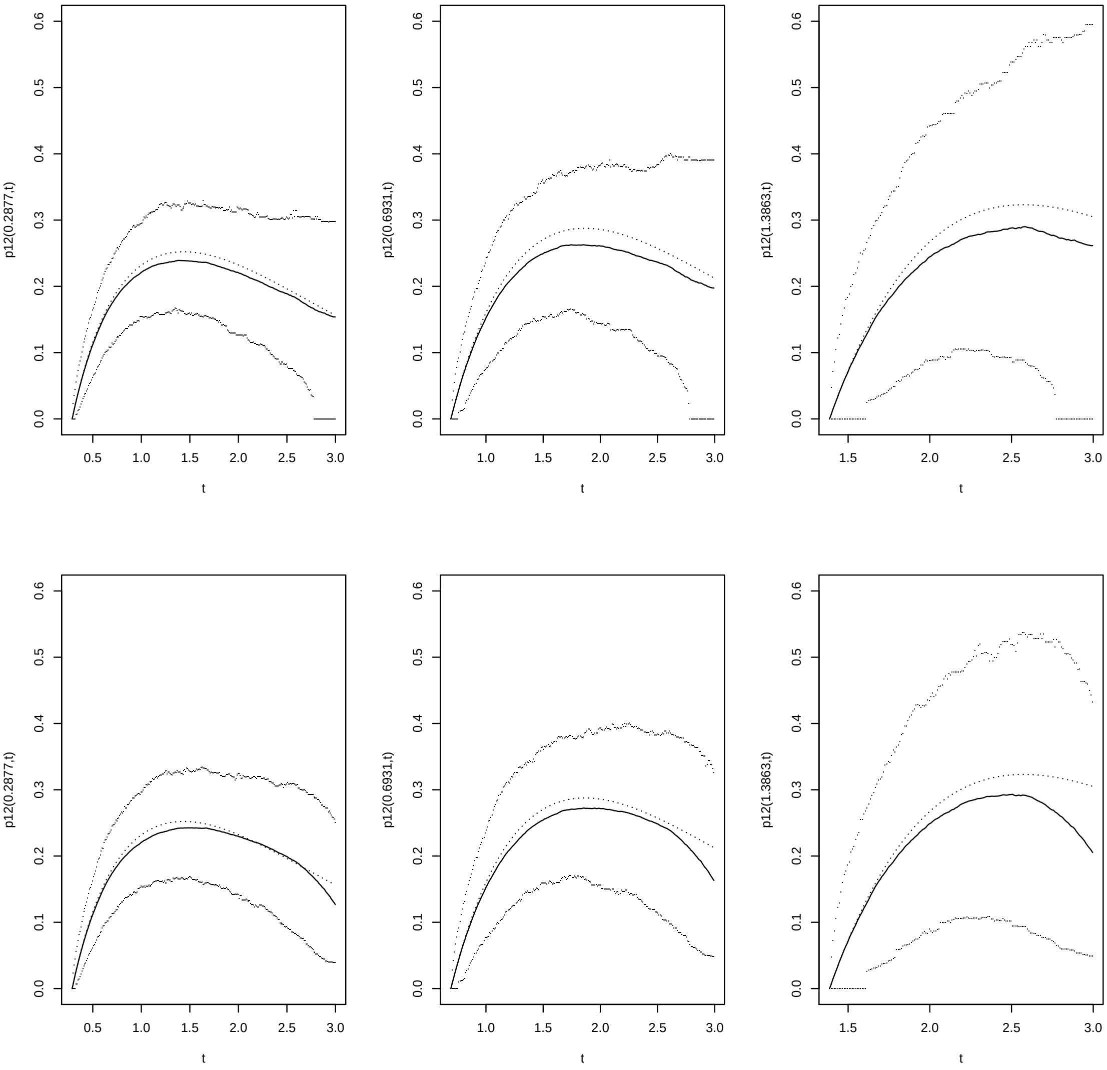

Figure 3.3: True $p_{12}(s,t)$ (dotted line), average estimator (solid line), and 95% oscillation limits of the AJ estimates (first row) and P-AJ (second row) for $s = 0.2877$, $s = 0.6931$ and $s = 1.3863$. Estimates with $n = 200$ and $U[0,3]$ censoring. Dependency scenario.

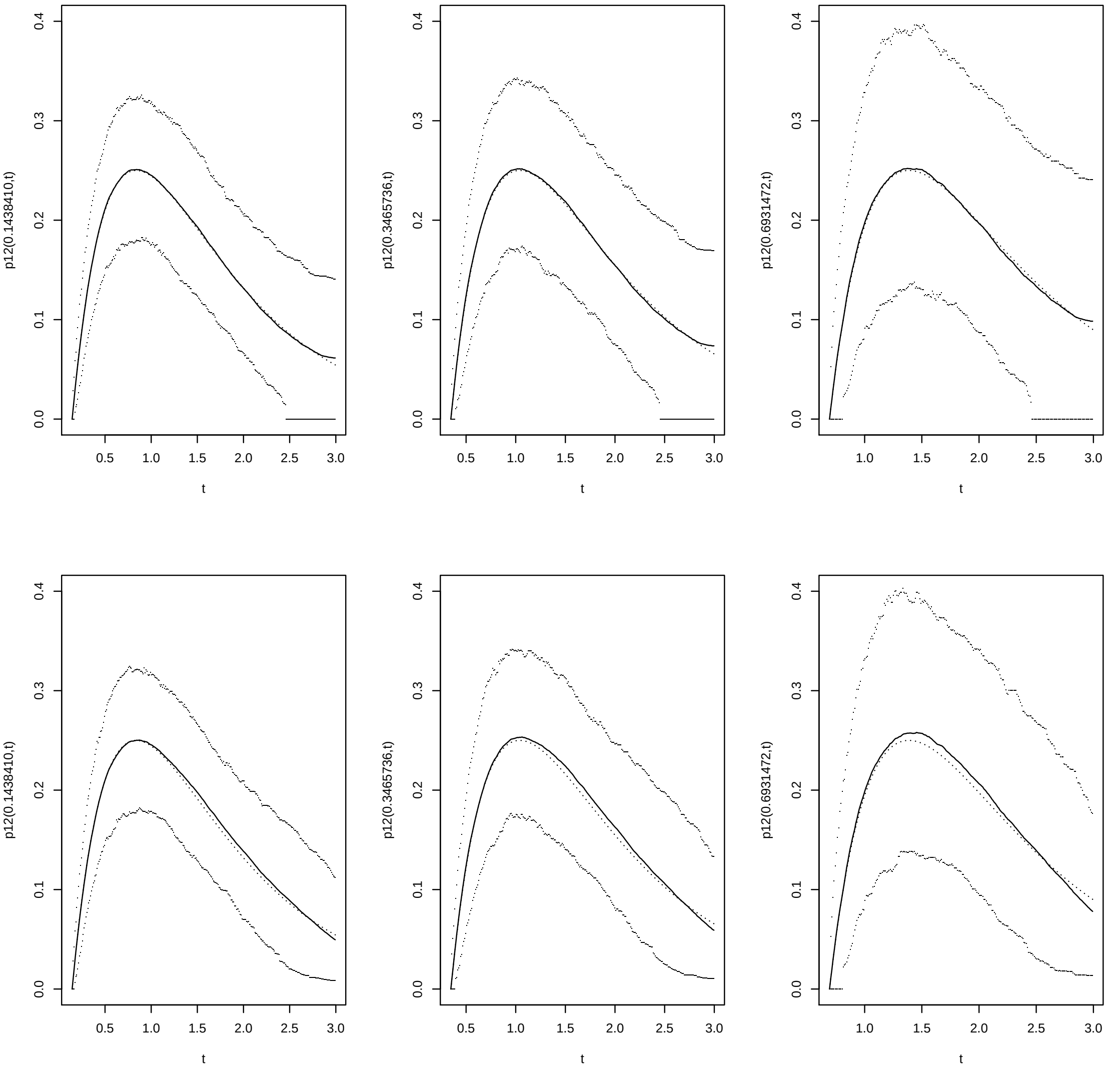

Figure 3.4: True $p_{12}(s,t)$ (dotted line), average estimator (solid line), and 95% oscillation limits of the AJ estimates (first row) and P-AJ (second row) for $s = 0.1438$, $s = 0.3466$ and $s = 0.6931$. Estimates with $n = 200$ and $U[0,3]$ censoring. Independency scenario.

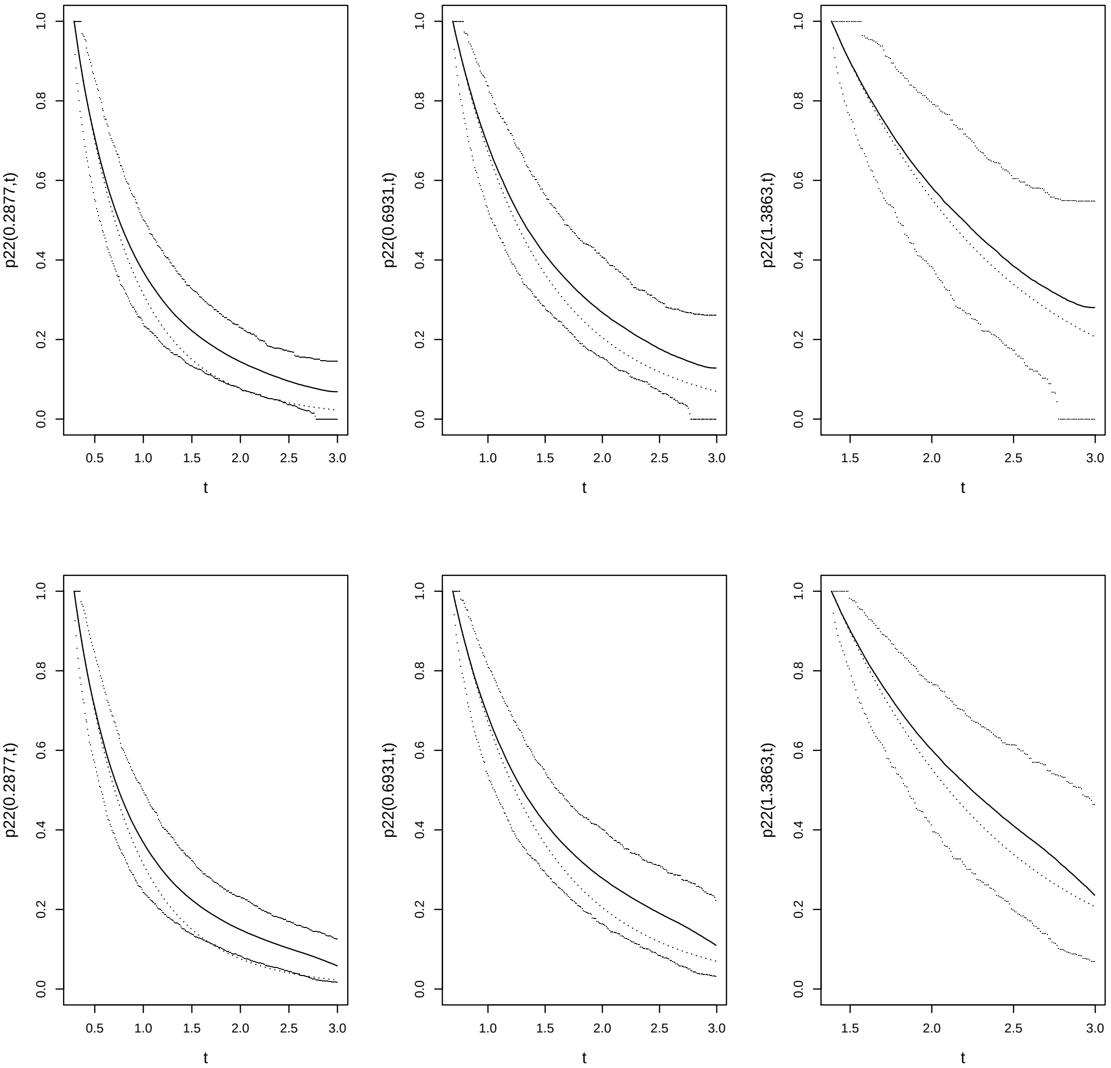

Figure 3.5: True $p_{22}(s,t)$ (dotted line), average estimator (solid line), and 95% oscillation limits of the AJ estimates (first row) and P-AJ (second row) for $s = 0.2877$, $s = 0.6931$ and $s = 1.3863$. Estimates with $n = 200$ and $U[0,3]$ censoring. Dependency scenario.

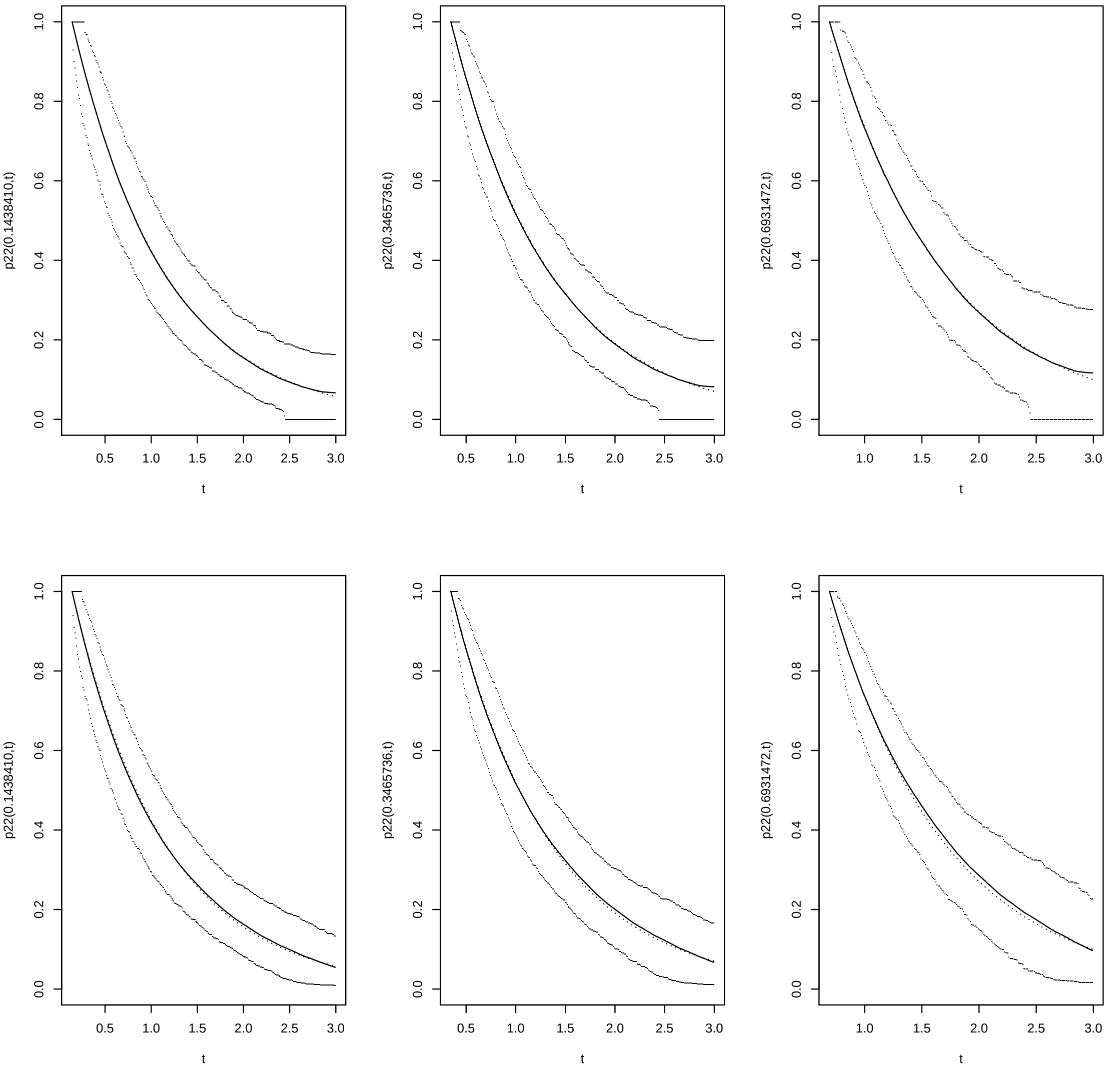

Figure 3.6: True $p_{22}(s,t)$ (dotted line), average estimator (solid line), and 95% oscillation limits of the AJ estimates (first row) and P-AJ (second row) for $s = 0.1438$, $s = 0.3466$ and $s = 0.6931$. Estimates with $n = 200$ and $U[0,3]$ censoring. Independency scenario.

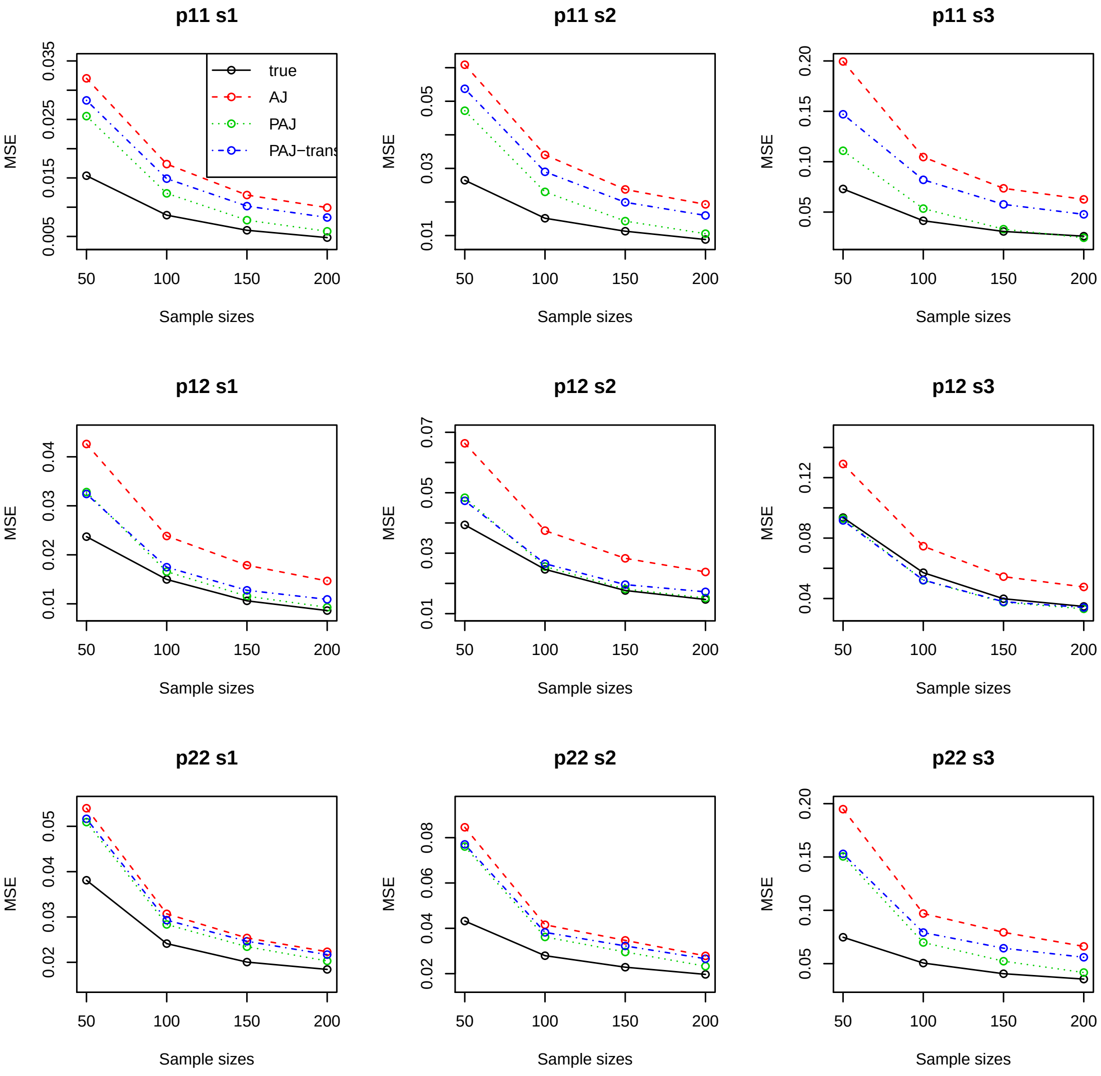

Figure 3.7: Mean Square Error of transition probabilities for dependency scenario.

These plots show that the Aalen-Johansen estimator (labeled in the tables as AJ) is the one with higher values of MSE while the presmoothed estimators (labeled in the tables as $m(.;\beta,\gamma)$ and $m(.;\xi)$) show lower values. The estimator with the "true presmoothing function" (labeled in the tables as m) gets better performance but this function is unrealistic in practice. We see that the MSE goes down with an increasing sample size.

Results for uniform censoring $U[0,3]$ are in general worse than those for $U[0,4]$, this is true for all situations. The advantages between the estimators are more clear for higher proportions of censorship. For example in the case of p_{12} of dependent $U[0,3]$ it is easy to see that there is a big difference between the presmoothed estimators and Aalen-Johansen estimator (see Figure 3.9). Figure 3.10 presents the corresponding plots for variance for all sample sizes with censoring $U[0,3]$. The variance decreases with an increase in the sample size. In all cases, the variance of the Aalen-Johansen estimator is larger. Variance tends to be a bit larger when introducing some correlation between the gap times (case $\delta = 1$), although some exceptions are found.

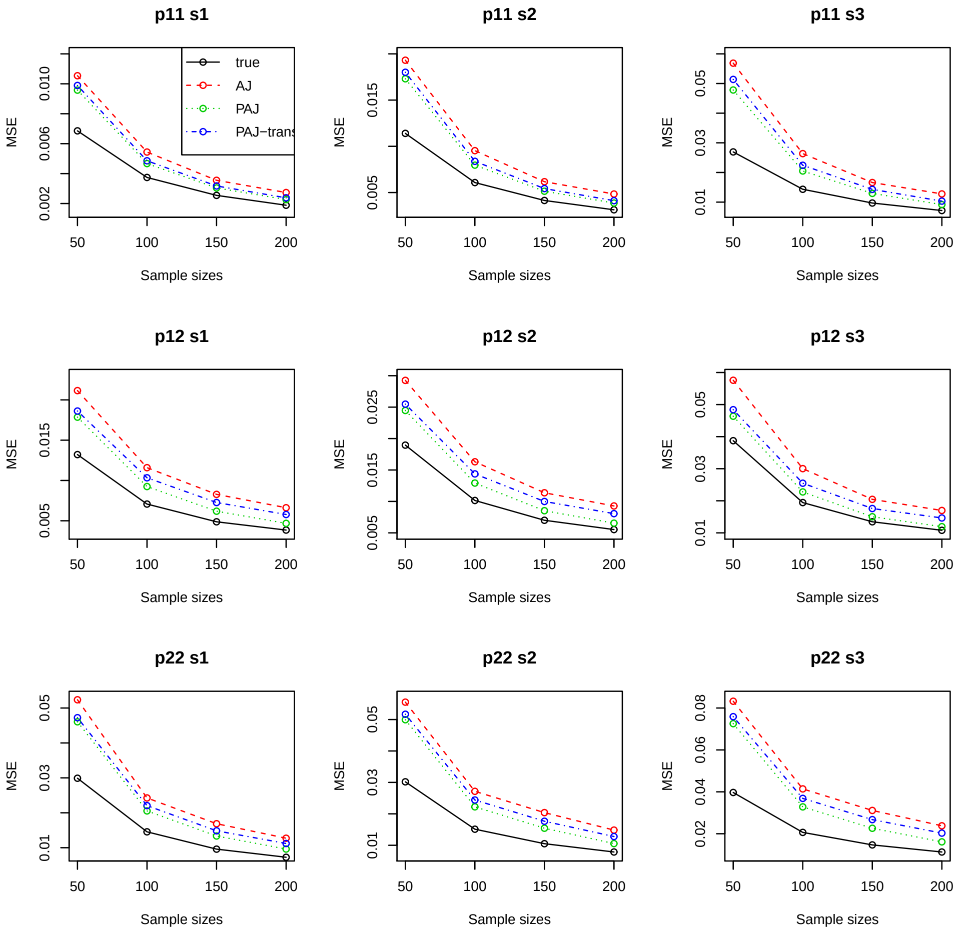

Figure 3.8: Mean Square Error of transition probabilities for independency scenario.

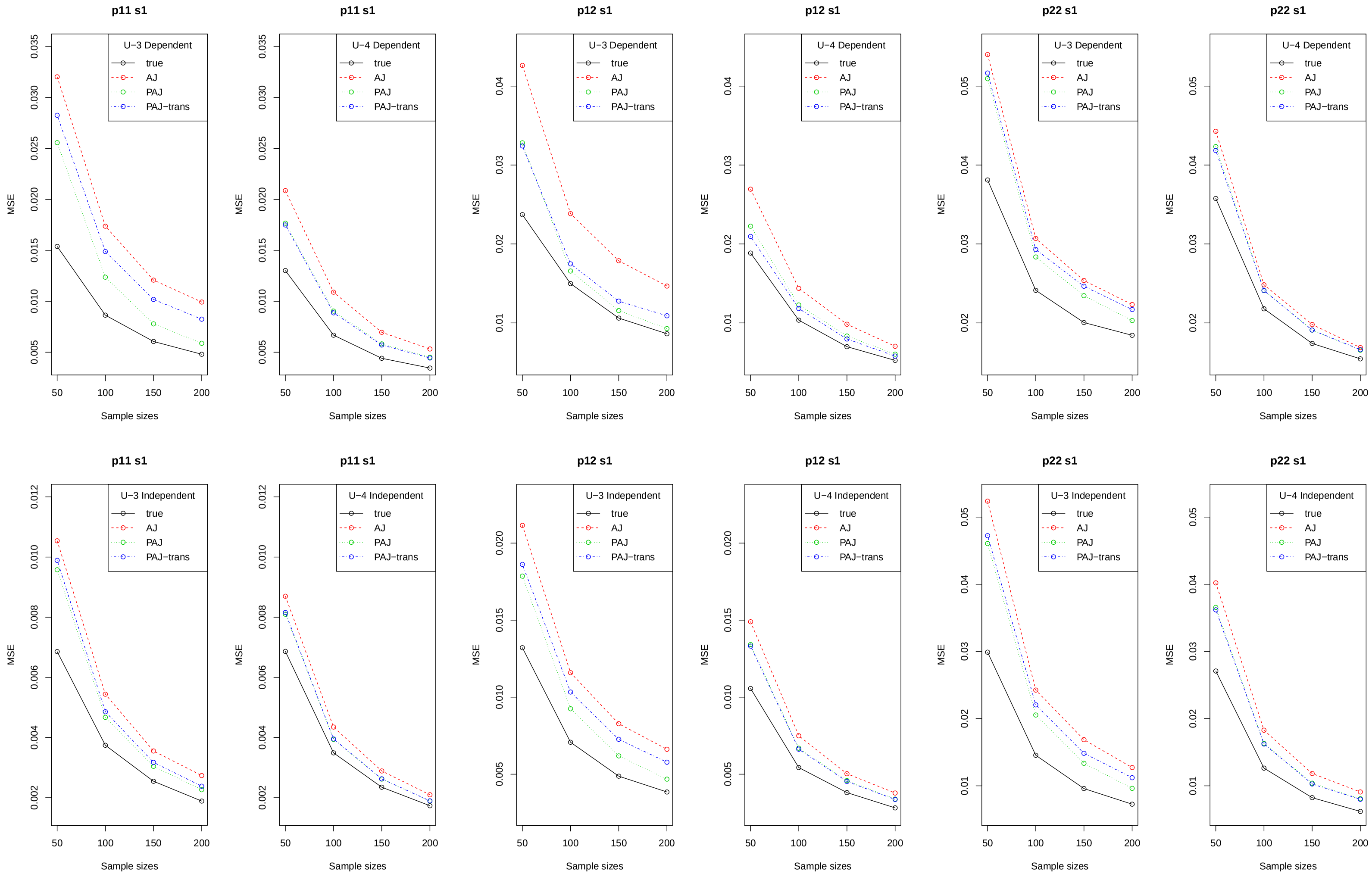

Figure 3.9: Mean Square Error of transition probabilities for different scenarios of censoring.

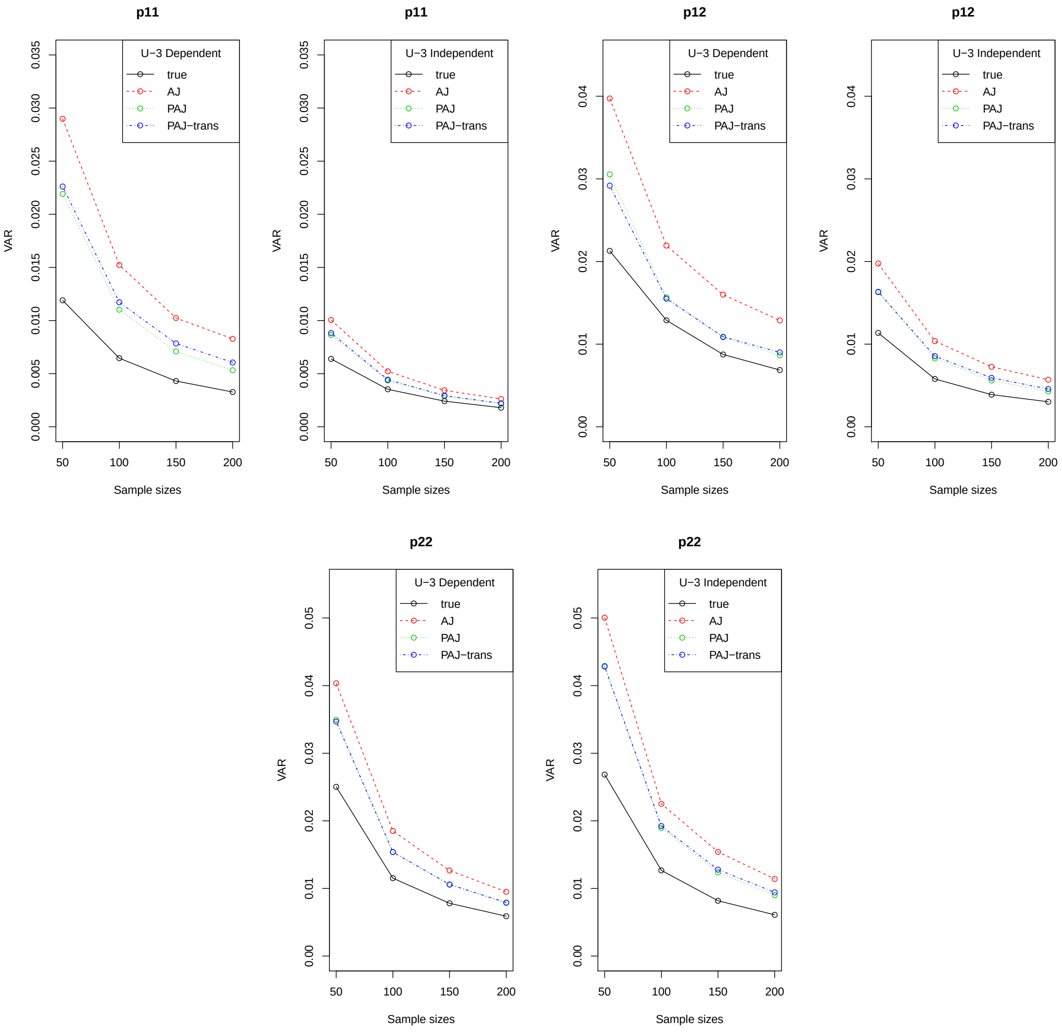

Figure 3.10: Variance for all sample sizes with censoring $U[0, 3]$

To compare the efficiency of the Aalen-Johansen estimator with the presmoothed Aalen-Johansen estimator we calculated the ratios between MSE(AJ) and MSE(PAJ) for the two scenario (dependency and independency) with uniform censoring $U[0, 3]$. Values greater than 1, shown in Figure 3.11, reveal that the PAJ is more efficient than AJ. These differences can also be seen for the four sample sizes.

In our simulations we have also considered different scenarios with different proportions of individuals passing through state 2. A larger value of $p = P(\rho = 1)$ is favourable for the estimation of $p_{22}(s, t)$ (lower values for IMSE, BIAS, L1 norm and variance), whereas a smaller value of p lead to better estimates for $p_{12}(s, t)$. When comparing the two methods (with and without presmoothing) similar conclusions were obtained and therefore they are not reported here.

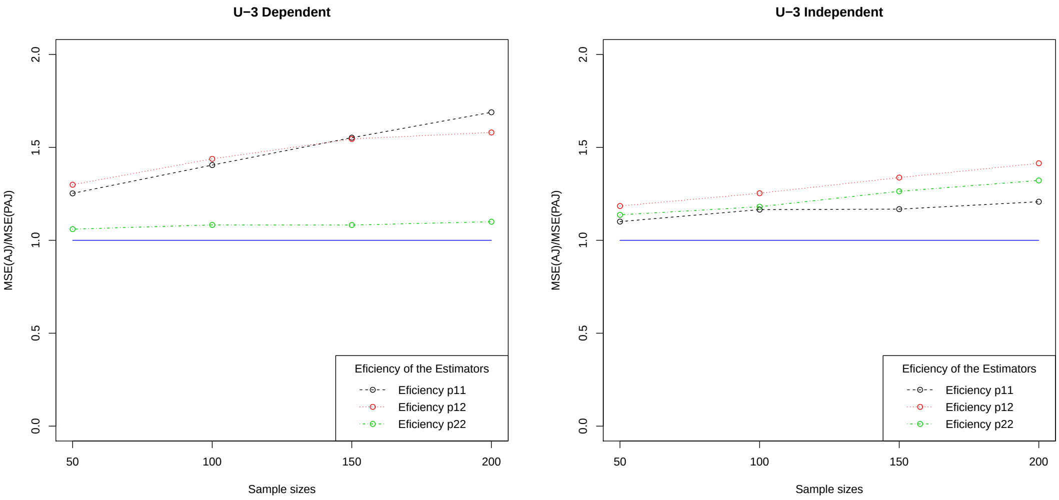

Figure 3.11: Efficiency of the estimators.

3.4 Real Data Illustration

For illustration purposes, we apply the proposed methods of Section 3.2 to data from the Stanford Heart Transplant study, previously presented in Chapter 1. It includes 103 patients enrolled in the Stanford Heart Transplant program, from which 69 received a Heart Transplant and among these 45 died. We may use the so-called illness-death model with states “Own heart”, “New heart” (or transplant) and “Dead”. In most applications, a Markov model is often assumed for the multi-state model. A Cox model (Cox, 1972) can be used to test this assumption (Hougaard, 1999; Andersen et al., 2000). This is usually performed by including covariates depending on the history, such as the time of transition to the current state or the time since entry into the current state. This assumption was verified for the Stanford Transplant study, e.g. by Hougaard (1999), which conclude that there is no effect of time since transplant on mortality, and thus that the Markov model is satisfactory. This is important, because otherwise, the consistency of the Aalen-Johansen estimator and the new estimator based on presmoothing cannot be ensured. On the other hand, if markovianity is fulfilled, the use of these methods is a wise choice. To deal with ties, a re-definition of the empiricals $M_{0n}(y)$ and $M_{1n}(y)$ is needed. Put $\widetilde{Z}_{i:n}$ for the *i*-th ordered Z-statistics. Similarly, put $\widetilde{T}_{i:n}$ for the *i*-th ordered T-statistics. For $y = \widetilde{Z}_{k:n}$ we define $\widetilde{M}_{0n}(y) = \frac{1}{n}\sum_{i=k}^{n} I(\widetilde{Z}_{i:n} \geq y)$ while for $y = \widetilde{T}_{k:n}$ we define $\widetilde{M}_{1n}(y) = \frac{1}{n}\sum_{i=k}^{n} I(\widetilde{Z}_{[i:n]} < y \leq \widetilde{T}_{i:n})$ where $\widetilde{Z}_{[i:n]}$ is the *i*-th concomitant (i.e. the Z-value attached to $\widetilde{T}_{i:n}$). When there are no ties, these empiricals reduce to those introduced in section 3.2.

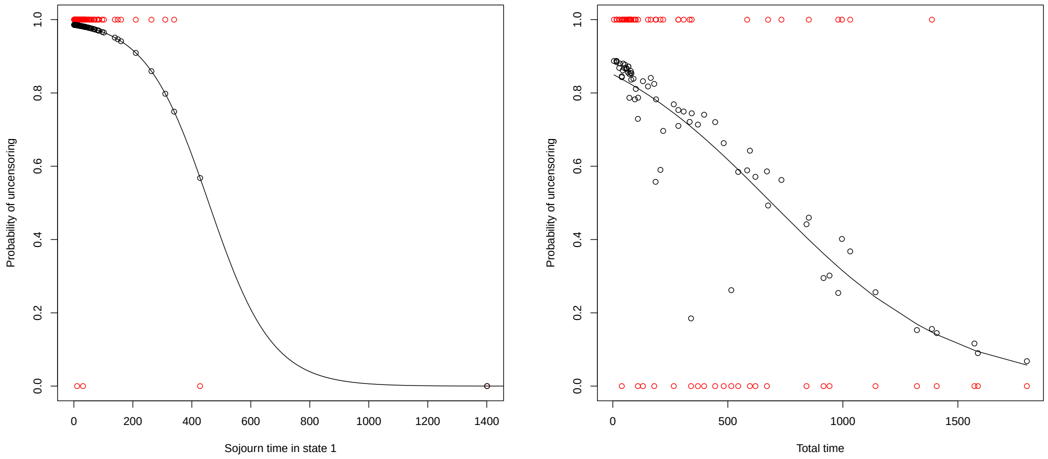

Figure 3.12: Presmoothing functions m_0 (left) and m_1 (right) estimated by logistic models. Stanford Heart Transplant data.

Our aim with this application is to illustrate the differences between the estimated transition probabilities from Aalen-Johansen estimator (AJ) and the semiparametric estimator based on presmoothing (P-AJ). The semiparametric estimator was obtained using standard logistic regression for $m_{0n}(z) = \widehat{P}(\Delta_1 = 1|\widetilde{Z} = z)$ and $m_{1n}(z,t) = \widehat{P}(\Delta = 1|\widetilde{Z} = z, \widetilde{T} = t, \Delta_1\rho = 1)$. Figure 3.12 displays these functions for the Stanford Heart data. The noise around displayed line comes from the fact that the variable z is omitted in the plot while it is present in the model. In Table 3.6 we present the summary (coefficients, standard errors between brackets and p-value) of the two presmoothing functions. In this case the influence of $\widetilde{Z}$ is not statistically significant on $m_1(z,t)$. The goodness-of-fit test that we used for testing the parametric presmoothing functions is an application of the Kolmogorov-Smirnov type version of the model-based bootstrap approach described in Dikta et al. (2006). The Kolmogorov-Sminorv test was used for testing the parametric logistic presmoothing functions $m_{0n}(z)$, $m_{1n}(z,t)$. In both cases the test was not able to reject the logistic model (respectively p-values of 0.638 and 0.237). We also show the goodness-of-fit test proposed by Hosmer and Lemeshow (2008) was used for testing the parametric logistic presmoothing functions $m_{0n}(z)$, $m_{1n}(z,t)$. In both cases the test was not able to reject the logistic model (without reaching statistical significance, p-value=0.218 and p-value=0.566).

Table 3.6: Summary of the two presmoothing functions m_{0n} and m_{1n} based on logistic models.

Presmoothing functions	Estimated coefficients	p-value
$m_{0n}(z) = (1 + exp(\widehat{\gamma}_0 + \widehat{\gamma}_1 z))^{-1}$	$\widehat{\gamma}_0 = 4.2605(0.8310)$	2.94e-07
	$\widehat{\gamma}_1 = -0.0093(0.0042)$	0.0283
$m_{1n}(z,t) = (1 + exp(\widehat{\beta}_0 + \widehat{\beta}_1 z + \widehat{\beta}_2 t))^{-1}$	$\widehat{\beta}_0 = 2.1148(0.5052)$	2.83e-05
	$\widehat{\beta}_1 = -0.0089(0.0058)$	0.1281
	$\widehat{\beta}_2 = -0.0025(0.0007)$	0.0006

Figures 3.13 and 3.14 plot, for the two methods, the estimated transition probabilities $p_{ij}(s,t)$, $1 \leq i \leq j \leq 3$ together with pointwise confidence bands based on the bootstrap. The bootstrap estimates were obtained for $B = 1000$ replicates, by randomly sampling the n items from the original data set with replacement. The bootstrap estimates were used to obtain the 95% limits for the confidence interval of $p_{11}(s,t)$, $p_{12}(s,t)$ and $p_{22}(s,t)$. The value s was chosen to be the percentile 25 and 50 of the total time ($s = 32$ and $s = 90$ days). As expected, the P-AJ estimator has less variability than the AJ estimator, which has fewer jump points as t increases. For example, the extra jump points of the presmoothed AJ estimator of $p_{22}(s,t)$ cor-

respond to transplanted patients with censored values of the total time. However, both methods provide similar point estimates for all values of time. In sum, the new approach provides more reliable curves with less variability and accordingly narrower pointwise confidence bands.

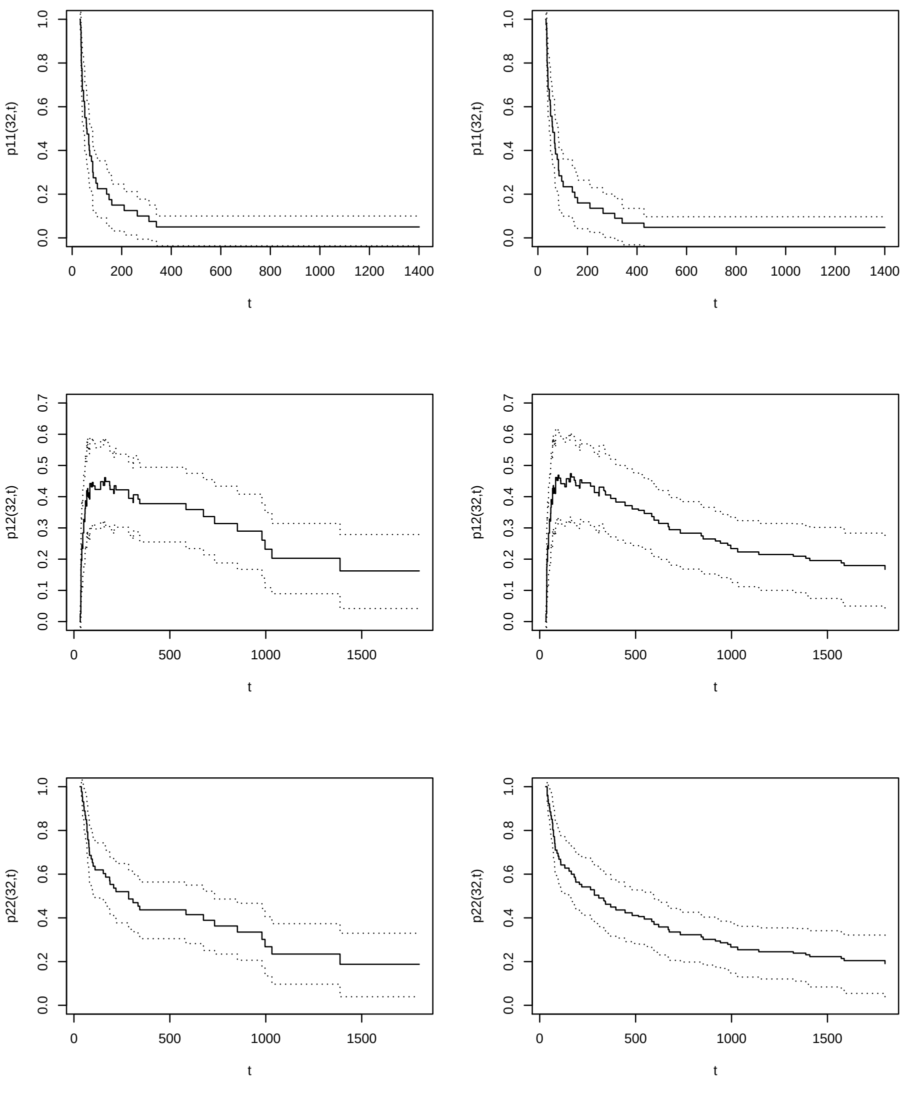

Figure 3.13: Estimated transition probabilities for $p_{ij}(s,t)$ with $s = 32$ based on the Aalen-Johansen estimator (on the left) and based on the presmoothed Aalen-Johansen estimator (on the right) with the corresponding 95% pointwise confidence bands. Stanford Heart Transplant data.

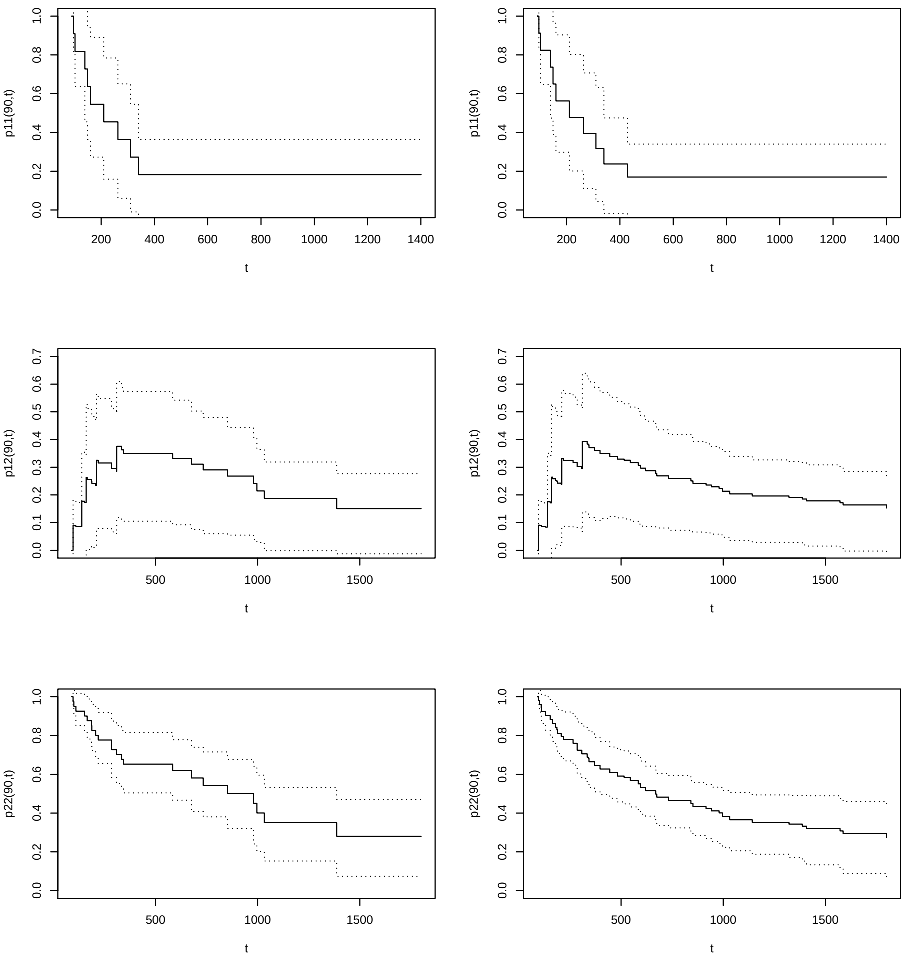

Figure 3.14: Estimated transition probabilities for $p_{ij}(s,t)$ with $s = 90$ based on the Aalen-Johansen estimator (on the left) and based on the presmoothed Aalen-Johansen estimator (on the right) with the corresponding 95% pointwise confidence bands. Stanford Heart Transplant data.

3.5 Technical Proofs

In this section we give the proof to Theorem 1. Throughout this proof $\widehat{p}_{ij}(s,t)$ stands for the presmoothed Aalen-Johansen estimator $\widehat{p}_{ij}^{PAJ}(s,t)$. Theorem 1(a) is a consequence of Dikta (1998). Now we prove Theorem 1(b), that is, the uniform strong consistency of

$$\widehat{p}_{22}(s,t)=\prod_{s<\widetilde{T}_i\le t}\left[1-\frac{m_{1n}(\widetilde{Z}_i,\widetilde{T}_i)I(\widetilde{Z}_i<\widetilde{T}_i)}{n\widetilde{M}_{1n}(\widetilde{T}_i)}\right]$$

where (recall) $m_{1n}(z,t)$ is an estimator of $m_1(z,t)=P(\Delta=1|\widetilde{Z}=z,\widetilde{T}=t,\widetilde{Z}<\widetilde{T})$ and where (recall) $\widetilde{M}_{1n}(y)=n^{-1}\sum_{j=1}^{n}I(\widetilde{Z}_j<y\le\widetilde{T}_j)$ is the empirical counterpart of $\widetilde{M}_1(y)=P(\widetilde{Z}<y\le\widetilde{T})$. Since continuity is assumed throughout, note that $\Delta_1\rho=I(\widetilde{Z}<\widetilde{T})$. The following notation will be used:

$$I(s,t)=\left\{i:s<\widetilde{T}_i\le t,\widetilde{Z}_i<\widetilde{T}_i\right\}$$

and

$$I^*(s,t)=\left\{i:s<\widetilde{T}_i\le t,\widetilde{Z}_i<\widetilde{T}_i,m_{1n}(\widetilde{Z}_i,\widetilde{T}_i)>0\right\}.$$

With this notation, we have

$$\widehat{p}_{22}(s,t)=\prod_{i\in I(s,t)}\left[1-\frac{m_{1n}(\widetilde{Z}_i,\widetilde{T}_i)}{n\widetilde{M}_{1n}(\widetilde{T}_i)}\right]=\prod_{i\in I^*(s,t)}\left[1-\frac{m_{1n}(\widetilde{Z}_i,\widetilde{T}_i)}{n\widetilde{M}_{1n}(\widetilde{T}_i)}\right].$$

Note that $\widehat{p}_{22}(s,t)=0$ may happen; indeed, this is the case whenever $n\widetilde{M}_{1n}(\widetilde{T}_i)=1$ and $m_{1n}(\widetilde{Z}_i,\widetilde{T}_i)=1$ for some $i\in I(s,t)$. In order to avoid problems when taking logarithms, introduce the following approximation to $\widehat{p}_{22}(s,t)$:

$$\overline{p}_{22}(s,t)=\prod_{i\in I(s,t)}\frac{n\widetilde{M}_{1n}(\widetilde{T}_i)}{n\widetilde{M}_{1n}(\widetilde{T}_i)+m_{1n}(\widetilde{Z}_i,\widetilde{T}_i)}.$$

Since $\left|\prod_j a_j-\prod b_j\right|\le\sum_j|a_j-b_j|$ for $|a_j|,|b_j|\le 1$, we have

$$|\widehat{p}_{22}(s,t)-\overline{p}_{22}(s,t)|\le\sum_{i\in I(s,t)}\frac{m_{1n}(\widetilde{Z}_i,\widetilde{T}_i)^2}{n^2\widetilde{M}_{1n}(\widetilde{T}_i)^2}.$$

We will refer to the following Lemma, which follows from e.g. Corollary 5.2.3 in de la Peña and Giné (1999).

Lemma 1. We have w.p. 1 $\sup_y \left|\widetilde{M}_{1n}(y) - \widetilde{M}_1(y)\right| \to 0.$■

Under condition M, from Lemma 1 we have eventually for $y \in [\tau_0, \tau_1]$ and some constant $c > 0$

$$\widetilde{M}_{1n}(y) \geq \inf_{\tau_0 \leq y \leq \tau_1} \widetilde{M}_1(y) - \sup_{\tau_0 \leq y \leq \tau_1} \left|\widetilde{M}_{1n}(y) - \widetilde{M}_1(y)\right| \geq c.$$

Hence we have w.p. 1

$$\sup_{\tau_0 \leq s < t \leq \tau_1} |\widehat{p}_{22}(s,t) - \overline{p}_{22}(s,t)| = O(n^{-1}). \tag{3.6}$$

Now write

$$\begin{aligned} \overline{p}_{22}(s,t) - p_{22}(s,t) &= \exp(\log \overline{p}_{22}(s,t)) - \exp(-\Psi_n(s,t)) \\ &\quad + \exp(-\Psi_n(s,t)) - \exp(-\Psi(s,t)) \end{aligned}$$

where

$$\Psi(s,t) = \int_s^t \frac{H^1(dy)}{\widetilde{M}_1(y)}, \qquad \text{with } H^1(y) = P(\widetilde{T} \leq y, \Delta = 1, \widetilde{Z} < \widetilde{T}),$$

and

$$\Psi_n(s,t) = \sum_{i \in I(s,t)} \frac{m_{1n}(\widetilde{Z}_i, \widetilde{T}_i)}{n\widetilde{M}_{1n}(\widetilde{T}_i)}.$$

Note that $p_{22}(s,t) = \exp(-\Psi(s,t))$ because of the Markov condition, and that

$$\Psi(s,t) = E\left[\frac{I(s < \widetilde{T} \leq t)\Delta I(\widetilde{Z} < \widetilde{T})}{\widetilde{M}_1(\widetilde{T})}\right] = E\left[\frac{I(s < \widetilde{T} \leq t) m_1(\widetilde{Z}, \widetilde{T}) I(\widetilde{Z} < \widetilde{T})}{\widetilde{M}_1(\widetilde{T})}\right].$$

It will be shown that $p_{22}(s,t) = \exp(-\Psi(s,t))$ is indeed the limit of $\exp(-\Psi_n(s,t))$. This will follow from the mean-value theorem after proving the uniform strong consistency of $\Psi_n(s,t)$, which is the goal of the following Lemma.

Lemma 2. Under U_2 and M we have w.p. 1 $\sup_{\tau_0 \le s < t \le \tau_1} |\Psi_n(s,t) - \Psi(s,t)| \to 0$.

Proof: Write

$$\Psi_n(s,t) = \sum_{i \in I(s,t)} \frac{m_1(\widetilde{Z}_i, \widetilde{T}_i)}{n\widetilde{M}_1(\widetilde{T}_i)} + \frac{1}{n} \sum_{i \in I(s,t)} \left[\frac{m_{1n}(\widetilde{Z}_i, \widetilde{T}_i)}{\widetilde{M}_{1n}(\widetilde{T}_i)} - \frac{m_1(\widetilde{Z}_i, \widetilde{T}_i)}{\widetilde{M}_1(\widetilde{T}_i)} \right]$$

$$\equiv \Psi_n^0(s,t) + R_n(s,t).$$

By the SLLN we have $\Psi_n^0(s,t) \to \Psi(s,t)$ w.p. 1. Furthermore, under M we have w.p. 1

$$\sup_{\tau_0 \le s < t \le \tau_1} \left| \Psi_n^0(s,t) - \Psi(s,t) \right| \to 0. \tag{3.7}$$

To see this, note that for $s, t \in [\tau_0, \tau_1]$ we have under M

$$\Psi(s,t) \le \frac{1}{\inf_{\tau_0 \le y \le \tau_1} \widetilde{M}_1(y)} E\left[I(\tau_0 < \widetilde{T} \le \tau_1) \Delta I(\widetilde{Z} < \widetilde{T}) \right] < \infty.$$

Introduce

$$\varphi_{s,t}(u,v) = \frac{I(s < v \le t) m_1(u,v) I(u < v)}{\widetilde{M}_1(v)}.$$

Now, $\{\varphi_{s,t} : \tau_0 \le s < t \le \tau_1\}$ is a VC-subgraph class (see Proposition 5.1.12 and comments following Definition 5.1.14 in de la Peña and Giné (1999)), and φ_{τ_0,τ_1} is an integrable envelope for that class. Hence, (3.7) follows from Corollary 5.2.3 in de la Peña and Giné (1999).

Now,

$$\frac{m_{1n}(\widetilde{Z}_i, \widetilde{T}_i)}{\widetilde{M}_{1n}(\widetilde{T}_i)} - \frac{m_1(\widetilde{Z}_i, \widetilde{T}_i)}{\widetilde{M}_1(\widetilde{T}_i)} = \frac{1}{\widetilde{M}_{1n}(\widetilde{T}_i)} \left[m_{1n}(\widetilde{Z}_i, \widetilde{T}_i) - m_1(\widetilde{Z}_i, \widetilde{T}_i) \right]$$

$$+ \frac{m_1(\widetilde{Z}_i, \widetilde{T}_i)}{\widetilde{M}_{1n}(\widetilde{T}_i) \widetilde{M}_1(\widetilde{T}_i)} \left[\widetilde{M}_1(\widetilde{T}_i) - \widetilde{M}_{1n}(\widetilde{T}_i) \right].$$

Under U_2 and M we have

$$\sup_{\tau_0 \leq s < t \leq \tau_1} |R_n(s,t)| \leq$$

$$\left[\frac{\sup_{z<t, \tau_0 \leq t \leq \tau_1} |m_{1n}(z,t) - m_1(z,t)|}{c} + \frac{\sup_{\tau_0 \leq y \leq \tau_1} \left| \widetilde{M}_{1n}(y) - \widetilde{M}_1(y) \right|}{c'} \right] \times$$

$$\times \frac{1}{n} \sum_{i=1}^{n} I(\tau_0 < \widetilde{T}_i \leq \tau_1) I(\widetilde{Z}_i < \widetilde{T}_i) = o(1) \text{ w.p. 1.}$$

Then the assertion of Lemma 2 follows.■

By the mean-value theorem,

$$\exp(\log \overline{p}_{22}(s,t)) - \exp(-\Psi_n(s,t))$$

$$= (\Psi_n(s,t) + \log \overline{p}_{22}(s,t)) \exp(-\xi_n^*(s,t))$$

for some ξ_n^* between Ψ_n and $-\log \overline{p}_{22}$. Now:

$$\begin{aligned} \log \overline{p}_{22}(s,t) &= \sum_{i \in I^*(s,t)} \log \left[\frac{n\widetilde{M}_{1n}(\widetilde{T}_i)}{n\widetilde{M}_{1n}(\widetilde{T}_i) + m_{1n}(\widetilde{Z}_i, \widetilde{T}_i)} \right] \\ &= \sum_{i \in I^*(s,t)} \log \left[1 - \frac{1}{x_i} \right] \end{aligned}$$

where

$$x_i = \frac{n\widetilde{M}_{1n}(\widetilde{T}_i)}{m_{1n}(\widetilde{Z}_i, \widetilde{T}_i)} + 1.$$

Note that x_i is well defined for $i \in I^*(s,t)$ and that $x_i > 1$ (because $n\widetilde{M}_{1n}(\widetilde{T}_i) \geq 1$ for $i \in I^*(s,t)$). Use

$$\log(1 - \frac{1}{x}) = -\sum_{k=1}^{\infty} \frac{1}{kx^k}, \quad x > 1,$$

to write

$$\log \overline{p}_{22}(s,t) = -\sum_{i\in I^*(s,t)} \sum_{k=1}^{\infty} \frac{m_{1n}(\widetilde{Z}_i, \widetilde{T}_i)^k}{k(n\widetilde{M}_{1n}(\widetilde{T}_i) + m_{1n}(\widetilde{Z}_i, \widetilde{T}_i))^k}.$$

Hence

$$\Psi_n(s,t) + \log \overline{p}_{22}(s,t) = \sum_{i\in I^*(s,t)} \frac{m_{1n}(\widetilde{Z}_i, \widetilde{T}_i)}{n\widetilde{M}_{1n}(\widetilde{T}_i)}$$

$$-\sum_{i\in I^*(s,t)} \sum_{k=1}^{\infty} \frac{m_{1n}(\widetilde{Z}_i, \widetilde{T}_i)^k}{k(n\widetilde{M}_{1n}(\widetilde{T}_i) + m_{1n}(\widetilde{Z}_i, \widetilde{T}_i))^k}$$

$$= \sum_{i\in I^*(s,t)} \frac{m_{1n}(\widetilde{Z}_i, \widetilde{T}_i)}{n\widetilde{M}_{1n}(\widetilde{T}_i)(n\widetilde{M}_{1n}(\widetilde{T}_i) + m_{1n}(\widetilde{Z}_i, \widetilde{T}_i))}$$

$$-\sum_{i\in I^*(s,t)} \sum_{k=2}^{\infty} \frac{m_{1n}(\widetilde{Z}_i, \widetilde{T}_i)^k}{k(n\widetilde{M}_{1n}(\widetilde{T}_i) + m_{1n}(\widetilde{Z}_i, \widetilde{T}_i))^k} \equiv I + II.$$

Under M we have, uniformly in $\tau_0 \le s < t \le \tau_1$, $I = O(n^{-1})$ w.p. 1. Besides, by noting

$$\sum_{k=2}^{\infty} x^k = \frac{1}{1-x} - 1 - x = \frac{x^2}{1-x}, \quad x < 1,$$

we have that the absolute value of II is bounded by (take $x = m_{1n}(\widetilde{Z}_i, \widetilde{T}_i)/ (n\widetilde{M}_{1n}(\widetilde{T}_i) + m_{1n}(\widetilde{Z}_i, \widetilde{T}_i))$)

$$\sum_{i\in I^*(s,t)} \sum_{k=2}^{\infty} \frac{m_{1n}(\widetilde{Z}_i, \widetilde{T}_i)^k}{(n\widetilde{M}_{1n}(\widetilde{T}_i) + m_{1n}(\widetilde{Z}_i, \widetilde{T}_i))^k}$$

$$= \sum_{i\in I^*(s,t)}^{n} \frac{m_{1n}(\widetilde{Z}_i, \widetilde{T}_i)^2}{n\widetilde{M}_{1n}(\widetilde{T}_i)(n\widetilde{M}_{1n}(\widetilde{T}_i) + m_{1n}(\widetilde{Z}_i, \widetilde{T}_i))} = O(n^{-1})$$

w.p. 1. uniformly in $\tau_0 \le s < t \le \tau_1$. This shows that

$$\sup_{\tau_0 \le s < t \le \tau_1} |\Psi_n(s,t) + \log \overline{p}_{22}(s,t)| = O(n^{-1}) \quad \text{w.p. 1}$$

and consequently

$$\sup_{\tau_0 \le s < t \le \tau_1} \left|\exp(\log \overline{p}_{22}(s,t)) - \exp(-\Psi_n(s,t))\right| = O(n^{-1}) \quad \text{w.p. 1.} \tag{3.8}$$

Now, use the mean-value theorem to write

$$\exp\left(-\Psi(s,t)\right) - \exp\left(-\Psi_n(s,t)\right) = \left[\Psi_n(s,t) - \Psi(s,t)\right] \exp(-\xi_n(s,t))$$

from which

$$\sup_{\tau_0 \le s < t \le \tau_1} \left|\exp\left(-\Psi(s,t)\right) - \exp\left(-\Psi_n(s,t)\right)\right| \le \sup_{\tau_0 \le s < t \le \tau_1} \left|\Psi_n(s,t) - \Psi(s,t)\right|.$$

Then Theorem 1(b) follows from Lemma 2, (3.8), (3.6), and the decomposition

$$\begin{aligned} \widehat{p}_{22}(s,t) - p_{22}(s,t) &= \widehat{p}_{22}(s,t) - \overline{p}_{22}(s,t) \\ &\quad + \exp(\log \overline{p}_{22}(s,t)) - \exp\left(-\Psi_n(s,t)\right) \\ &\quad + \exp\left(-\Psi_n(s,t)\right) - \exp\left(-\Psi(s,t)\right). \end{aligned}$$

In order to prove Theorem 1(c) write, with $J(s,t) = \left\{ i : s < \widetilde{Z}_i \le t, \widetilde{Z}_i < \widetilde{T}_i \right\}$,

$$\widehat{p}_{12}(s,t) = \frac{1}{n} \sum_{i \in J(s,t)} \frac{\widehat{p}_{11}(s, \widetilde{Z}_i^-) \widehat{p}_{22}(\widetilde{Z}_i, t)}{\widetilde{M}_{0n}(\widetilde{Z}_i)}$$

$$\begin{aligned} &= \frac{1}{n} \sum_{i \in J(s,t)} \left[\widehat{p}_{11}(s, \widetilde{Z}_i^-) - p_{11}(s, \widetilde{Z}_i)\right] \frac{\widehat{p}_{22}(\widetilde{Z}_i, t)}{\widetilde{M}_{0n}(\widetilde{Z}_i)} \\ &\quad + \frac{1}{n} \sum_{i \in J(s,t)} \left[\widehat{p}_{22}(\widetilde{Z}_i, t) - p_{22}(\widetilde{Z}_i, t)\right] \frac{p_{11}(s, \widetilde{Z}_i)}{\widetilde{M}_{0n}(\widetilde{Z}_i)} \\ &\quad + \frac{1}{n} \sum_{i \in J(s,t)} p_{11}(s, \widetilde{Z}_i) p_{22}(\widetilde{Z}_i, t) \left[\frac{1}{\widetilde{M}_{0n}(\widetilde{Z}_i)} - \frac{1}{\widetilde{M}_0(\widetilde{Z}_i)}\right] \\ &\quad + \frac{1}{n} \sum_{i \in J(s,t)} \frac{p_{11}(s, \widetilde{Z}_i) p_{22}(\widetilde{Z}_i, t)}{\widetilde{M}_0(\widetilde{Z}_i)} \\ &\equiv I(s,t) + II(s,t) + III(s,t) + IV(s,t) \end{aligned}$$

where $\widetilde{M}_0(y) = P(\widetilde{Z} \geq y)$. Since, because of the Markov condition,

$$E\left[\frac{p_{11}(s,\widetilde{Z}_i)p_{22}(\widetilde{Z}_i,t)}{\widetilde{M}_0(\widetilde{Z}_i)} I(s < \widetilde{Z}_i \leq t, \widetilde{Z}_i < \widetilde{T}_i)\right] = p_{12}(s,t),$$

the SLLN gives $IV(s,t) \to p_{12}(s,t)$ w.p. 1. Furthermore, by using Proposition 5.1.12 in de la Peña and Giné (1999) as in Lemma 2 above we get w.p. 1

$$\sup_{0 \leq s < t \leq \tau} |IV(s,t) - p_{12}(s,t)| \to 0.$$

It remains to show that $I(s,t)$, $II(s,t)$, and $III(s,t)$ go to zero w.p.1 uniformly on $[0,\tau]$. But this is easily seen by using Theorem 1(a),(b), Glivenko-Cantelli, and the fact that $\widetilde{M}_0$ is bounded away from zero on $[0,\tau]$.■

Chapter 4

Software

4.1 Introduction

One important goal in multi-state modelling is to study the relationship between the different covariates and disease evolution. Other issues of interest include the estimation of the bivariate distribution function, the estimation of the transition probabilities and survival rates. Despite its potential, multi-state modelling is not used by practitioners as frequently as other survival techniques we believe that lack of knowledge of available software of the new methodologies in user friendly software may be responsible for this lack of popularity. The aim of this chapter is therefore two fold. Firstly, to report an the existing software for implementing multi-state models using free (R) statistical software. Secondly, to describe the capabilities of the **survivalBIV** (Moreira and Meira Machado, 2012) package to implement of the methods described in Chapter 2.

The first contributions with software for implementing MSMs were written in difficult languages such as SAS (Paes and Lima, 2004; Hui-Min et al., 2004; Rosthøj et al., 2004) or Fortran (Marshall et al., 1995; Alioum and Commenges, 2001).

Recently several contributions have been made to the statistical software R (www.r-project.org). The first part of this chapter aims to provide users a guide about these contributions.

The **survival** package in the software R was one important contribution to this matter. Due to this package, survival analysis is no longer limited to Kaplan-Meier curves and simple Cox models (Therneau and Grambsch, 2000; Lumley, 2004). The library **survival** is available as part of S-plus and R statistical packages and can be used for modelling multi-state survival data. The key here is the creation of an appropriate data set representing each individual by several observations. This approach can be used to perform Markov and semi-Markov multi-state regression and can deal with any kind of process though it becomes complicated with the increase of the number of states. More details about this procedure will be given later in our applications.

In R (http://cran.r-project.org/web/packages/msm/), multi-state regression can also be performed using the **msm** package by Christopher Jackson (Jackson, 2007). Jackson implemented several functions for fitting continuous-time Markov and hidden Markov multi-state models (a model in which the stages are observed with misclassification) to longitudinal data. Covariates can be fitted to both the transition rates and misclassification probabilities. Allignol, Beyersmann and Schumacher have created two relevant packages for nonparametric estimation in multi-state models: the **mvna** package (Allignol et al., 2008) which allow to estimate transition hazards in multi-state models, potentially subject to left-truncation and right censoring; and the **etm** package provides a way to easily estimate and display the matrix of transition probabilities from MSM. The **etm** package handles both left-truncated and right-censored data. Recently, Wrangler et al. (2006) developed aRlibrary called **changeLOS** that compute and plot change in length of hospital stay. Now all the functionalities offered by **changeLOS** can be found in the **etm** package. The **mstate** package, developed by Putter (Putter et al., 2007), allows to estimate hazards and probabilities, possibly depending on covariates, and to obtain prediction probabilities in the context of competing risks and multi-state models. Recently, Meira-Machado and Roca-Pardiñas (2011) developed the R **p3state.msm** package that contains nonparametric statistical methods for estimating quantities of interest such as transition probabilities and the bivariate distribution function for censored gap times. This software can also be used to fit the time-dependent Cox regression model (TDCM) as well as semiparametric Cox proportional hazard regression models to all permitted transitions, by decoupling the whole process into various survival models. Meira-Machado et al. (2007) developed a R based library, called **tdc.msm**, for the analysis of multi-state survival data. Specifically, this software may be used to fit the TDCM but also several MSM regression models. The **TPmsm** package (`http://CRAN.R-project.org/package=TPmsm`) was recently developed that permits to estimate transition probabilities of an illness-death model or three-state progressive model. Another package that is important for the multi-state models is **genSurv** package (`http://CRAN.R-project.org/package=genSurv`) that permits to generate data with one binary time-dependent covariate and data stemming from a progressive illness-death model.

This chapter we will focus our attention to the available R packages for the analysis of multi-state survival data and describes the R-based **survivalBIV** (available from the Comprehensive R Archive Network at `http://CRAN.R-project.org/package=survivalBIV`) package's capabilities for implementing non-parametric and semiparametric estimators for the bivariate distribution function for censored gap times. In this chapter we explain and illustrate how numerical and graphical output for the four methods discussed in Section 2.2.2 (CKM, Lin, KMW and KMPW) can be obtained using the **survivalBIV** package.

The following section provides a brief introduction to the use of the available R packages for the multi-state models. An overview of the use of **survivalBIV** is given in Section 4.3. In Section 4.4 we explain how the package can be used to simulate bivariate censored data and how to use the several functions in the package. An example of its application is given using data from a Bladder cancer study in Section 4.5.

4.2 Available R based Packages for multi-state modelling

In this section we have the `3.0.1` version of `R` software to illustrate the capabilities of the `R` packages mentioned in the previous section. Most of these packages presents, however, some difficulties and limitations in practice. Some assumes the process to be Markovian and/or time-homogeneous; others do not provide graphical output. Furthermore, possible comparisons between different multi-state models are rather difficult because each of the programs requests its own data structure. We, therefore, developed a user-friendly `R` function, called `msmdata`, that provides the user the input data for all of the packages described in next section (see Appendix A). The `msmdata` function can be used to coerce objects from one class to another. The input data for this function is a data.frame that include the following variables: `times1` (time since entry into study to recurrence), `delta` (recurrence indicator), `times2` (time to death since the recurrence time),`time` (times1+times2) and `status`(censoring indicator: dead=1, alive=0). For illustration purposes we considered the Colon cancer data described in Chapter 1. For illustration purposes we only considered three covariates: *rx*, *sex* and *age*. The database has 922 patients. A sample of the original dataset is shown in Table 4.1.

Note that in the illness-death, possible courses for the individual include: $1 \to 1$ (the individual remains in state 1 until the end of the study; if `delta=0` and `status=0`); $1 \to 3$ (a direct transition from state 1 into state 3 is observed; if `delta=0` and `status=1`); $1 \to 2 \to 2$ (if `delta=1` and `status=0`); and $1 \to 2 \to 3$ (if `delta=1` and `status=1`).

This database (colon2) will be the basis for our analysis. In Table 4.2 we show what kind of outputs, state structure of the data and assumption of the `R` based packages.

Table 4.1: Sample of the original (Colon) data.

times1	delta	times2	time	status	rx	sex	age
968	1	553	1521	1	3	1	43
3087	0	0	3087	0	3	1	63
542	1	421	963	1	1	0	71
245	1	48	293	1	3	0	66

Table 4.2: Summary of output and state structure for the R based packages .

	Numerical Output	Graphical Output	State Structure	Assumption
survival	Regression Coefficients (TDCM, CMM, CSMM)	Survival	Any	
p3state.msm	Regression Coefficients , (TDCM, CMM, CSMM) Bivariate Distribution Function Transition Probabilities	Transition Probabilities, Bivariate Distribution Function Marginal Distribution	Progressive 3State illness-death	
msm	Regression Coefficients , (THMM, HMM) Transition Probabilities matrix Hazard Ratios	Survival, Transition Probabilities Expected Probability of Survival	Any	Time homogeneity Markov
mstate	Regression Coefficients	Estimated Cumulative Transition Intensities Transition Probabilities as estimated	Any	Markov
etm	Estimate Transition Probabilities Estimate Variance of the Aalen-Johansen	Estimates of the Transition Probabilities	Any	Markov
changeLOS	Aalen-Johansen estimator for the matrix of Transition Probabilities	Transition Probabilities	Any	Markov
mvna	Multivariate Nelson-Aalen estimator of the Cumulative Transitions Hazards	Estimates of the Cumulative Transitions Hazards	Any	Markov

survival

The analysis of the Cox model with time-dependent covariates can be obtained using almost all the existing statistical packages. To accommodate time-dependent effects, the `R` statistical packages use a counting process data structure introduced by Andersen and Gill (1982). In this data-structure, an individual's survival data is expressed by three variables: `start`, `stop` and `event`. In the Colon cancer data, recurrence (if `delta`=1) is the only time-dependent covariate (this covariate will be renamed as `tdcov`). Individuals without change in the time-dependent covariate (i.e. without recurrence) are represented by only one line of data, whereas patients with a change in the time-dependent covariate must be represented by two lines. For these patients, the first line represents the time period until the recurrence; the second line represents the time period that passes from the recurrence to the end of the follow-up. The remaining (time-fixed) covariates are the same for the two lines. For each row, variables start and stop mark the time interval (`start`, `stop`) for the data, while event is an indicator variable, taking on value 1, if there was a death at time stop, and 0 otherwise. As an example consider the information available from four patients. The structure of the new database is shown in Table 4.3 (using the same individuals as in Table 4.1). The first patient had a recurrence 968 days after enrolment and died at time 1521. For the second patient, the time from enrolment to censoring is 3087. Patients 3 and 4 had a recurrence at days 542 and 245 respectively. The time from enrolment to death for the these patients are 963 and 293 days,respectively.

Table 4.3: Sample of the Colon data in a counting process format.
Input data for the survival library.

id	start	stop	event	tdcov	rx	sex	age
1	0	968	0	0	3	1	43
1	968	1521	1	1	3	1	43
2	0	3087	0	0	3	1	63
3	0	542	0	0	1	0	71
3	542	963	1	1	1	0	71
4	0	245	0	0	3	0	66
4	245	293	1	1	3	0	66

This approach for representing standard survival data can be easily extended to more complex situations. Cox regression can be performed using **survival** library as follow:

```
> require("survival")
> colon.surv <- msmdata(colon2, pkg = "tdcm")
> cox.tdcm <- coxph(Surv(start, stop, event) ~ tdcov +
factor(rx) + sex + age, data = colon.surv)
> summary(cox.tdcm)
```

The effect of the recurrence (tdcov) leads to a increase in risk (Hazard Ratio, HR:64.6005; 95% confidence interval, 95% CI: 45.5597 - 91.5990). Age (HR:1.0116; 95% IC: 1.0038 - 1.0190) and rx are both important factors, while sex and rx has no significant effect (p-value=0.0585 $>$ 0.05 and p-value=0.7153, respectively).

A partial MSM can be obtained adding interactions with the time-dependent covariate:

```
> cox.tdcm2 <- coxph(Surv(start, stop, event) ~ tdcov:factor(rx)
+ tdcov:sex + tdcov:age, data = colon.surv)
> summary(cox.tdcm2)
```

Cox Markov models (Therneau and Grambsch, 2000; Meira-Machado et al., 2009) can be fitted through most of the statistical packages as long as we use a counting process notation, representing each patient by several observations. For the Colon cancer data, individuals without recurrence contribute with two lines of data (one for each of the transition leaving state 1) whereas individuals with a recurrence contribute with three lines of data (one for each transition). The counting process data structure has now one more variable representing the transition. The data structure now have the following variables: id, start, stop, event, tdcov and transition (Table 4.4).

In this data structure, transition = 1 denotes the mortality transition without recurrence, transition = 2 denotes the recurrence transition and transition = 3 the mortality transition after the recurrence. The events of interest are recurrence and death. The variable event denotes whether the main event time (death) is observed or censored.

The results for Cox Markov Model (CMM) can be obtained using the following input commands:

Table 4.4: Sample of the Colon data in a counting process format. Input data for the Cox Markov model.

id	start	stop	event	tdcov	transition	rx	sex	age
1	0	968	0	0	1	3	1	43
1	0	968	1	0	2	3	1	43
1	968	1521	1	1	3	3	1	43
2	0	3087	0	0	1	3	1	63
2	0	3087	0	0	2	3	1	63
3	0	542	0	0	1	1	0	71
3	0	542	1	0	2	1	0	71
3	542	963	1	1	3	1	0	71
4	0	245	0	0	1	3	0	66
4	0	245	1	0	2	3	0	66
4	245	293	1	1	3	3	0	66

```
> colon.cmm <- msmdata(colon2, pkg = "cmm")
> coxph(Surv(start, stop, event) ~ factor(rx) + sex + age,
data = colon.cmm, subset = c(transition == 1))
> coxph(Surv(start, stop, event) ~ factor(rx) + sex + age,
data = colon.cmm, subset = c(transition == 2))
> coxph(Surv(start,stop, event) ~ factor(rx) + sex + age,
data = colon.cmm, subset = c(transition == 3))
```

Before using MSMs, we have to evaluate whether the Markov assumption is tenable. The Markov assumption states that future evolution only depends on the current state at time t. The Markov assumption may be checked, among others, by including covariates depending on the history (Kay, 1986). For the illness-death model, the Markov assumption is only relevant for the mortality after recurrence. We can check this assumption by examining whether the time spent in the healthy (alive and disease-free) state (i.e. the past) is important on the transition from the disease (recurrence) state to death (i.e. the future). This can be done using the following Cox model:

```
> coxph(Surv(stop, event) ~ start, data=colon.cmm,
subset = c(transition == 3))
```

which revealed that the Markov assumption is not valid (p-value < 0.05). In situations like this, one alternative approach is to use a semi-Markov model in which the future of the process does not depend on the current time but rather on the duration in the current state. The Cox semi-Markov Model (CSMM) can be fitted using the following input command:

```
> coxph(Surv(stop-start, event) ~ factor(rx) + sex + age,
data = colon.cmm, subset = c(transition == 3))
```

The results obtained from fitting this model showed us that age, which revealed a strong effect on survival in the Cox model, under the CSMM, only obtains a significant effect on $1 \rightarrow 3$ (`transition=1`) (HR: 1.089; 95% CI: 1.049 - 1.13). The treatment variable, `rx` (treatment: Lev+5-FU), revealed to be the best predictor for the mortality transition $2 \rightarrow 3$ (`transition=3`) for patients that experienced a recurrence (HR: 1.39; 95% CI: 1.082 - 1.79), and also for transition $1 \rightarrow 2$ (`transition=2`), corresponding to recurrence (HR: 0.595; 95% CI: 0.470 - 0.752). The effect of `rx` (treatment: Lev+5-FU), on the mortality intensity in patients without recurrence was not significant (HR: 0.846; 95% CI: 0.401 - 1.79). No significant effect of `sex` and treatment with Levamisole alone was found.

p3state.msm

The **p3state.msm** package contains nonparametric statistical methods for estimating quantities of interest such as transition probabilities, bivariate distribution function for censored gap times, etc. This package can only be used for the progressive three-state model and the illness-death model. Records in the data file must contain the following variables: `times1`, `delta`, `times2`, `time` and `status`. The remaining variables are the covariates to be studied in the regression models. Each individual is represented by one line of data, just as shown in Table 4.1.

The **p3state.msm** software enables several semi-parametric Cox models to be fitted. The time-dependent Cox model (TDCM) or multi-state Cox-like models (CMM and CSMM) can be constructed with the following input commands:

```
> require("p3state.msm")
> res.p3state <- p3state(colon2, formula = ~ factor(rx) + sex
+ age)
> summary(res.p3state, model = "TDCM")
> summary(res.p3state, model = "CMM")
> summary(res.p3state, model = "CSMM")
```

For illustration purposes we show the results for all transition (Table 4.5).

The results are the same as for the **survival** package (CSMM). Note that in the CMM the results are only different on transition $2 \rightarrow 3$ (results not shown).

The patients course over time may also be studied through transition probabilities. For the **p3state.msm** package, the estimators for the transition probabilities can be considered as an alternative to Aalen-Johansen estimators since they do not rely on the Markov assumption (Meira-Machado et al., 2006). To obtain these estimates (for a model with no covariates), the following input command must be typed:

```
> summary(res.p3state, time1 = 100, time2 = 800)
```

```
Number of individuals experiencing the intermediate event: 461
Number of events for the direct transition from state 1 to state 3: 38
Number of individuals remaining in state 1: 423
Number of events on transition from state 2: 409
Number of censored observations on transition from state 2: 52


The estimate of the transition probability
```

```
P11( 100 , 800 ) is  0.6182574
P12( 100 , 800 ) is  0.1720199
P13( 100 , 800 ) is  0.2097227
P22( 100 , 800 ) is  0.05531639
P23( 100 , 800 ) is  0.9446836
```

The package also provides plots for several functions for the model illness-death. The transition probabilities (see Figure 4.1) can be obtained with:

```
> plot(res.p3state, plot.trans = "all", time1 = 100)
```

The **p3state.msm** package can also be used to obtain estimates and plots for the bivariate distribution function and for the marginal distribution of the second gap (time since recurrence). However this can only be obtained in the scope of the progressive three-state model.

Table 4.5: Cox Semi-Markov model for all transitions.

Cox Semi-Markov Model from state $1 \to 3$				
	coef	exp(coef)	95% CI	p-value
n=922				
factor(rx)2	-0.3353	0.7151	0.3132 - 1.6329	0.4261
factor(rx)3	-0.1670	0.8462	0.4011 - 1.7853	0.6611
sex	0.4238	1.5278	0.7922 - 2.9464	0.2059
age	0.0854	1.0892	1.0486 - 1.1313	1.0231e-05
Cox Semi-Markov Model from state $1 \to 2$				
	coef	exp(coef)	95% CI	p-value
n=922				
factor(rx)2	-0.0016	0.9984	0.8083 - 1.2333	0.9885
factor(rx)3	-0.5200	0.5945	0.4699 - 0.7522	1.4693e-05
sex	-0.1068	0.8987	0.7485 - 1.0791	0.2525
age	-0.0072	0.9928	0.9852 - 1.0005	0.0670
Cox Semi-Markov Model from state $2 \to 3$				
	coef	exp(coef)	95% CI	p-value
n=461				
factor(rx)2	0.1091	1.1153	0.8900 - 1.3976	0.3432
factor(rx)3	0.3317	1.3934	1.0818 - 1.7947	0.0102
sex	0.1603	1.1739	0.9638 - 1.4299	0.1111
age	0.0072	1.0073	0.9993 - 1.0153	0.0756

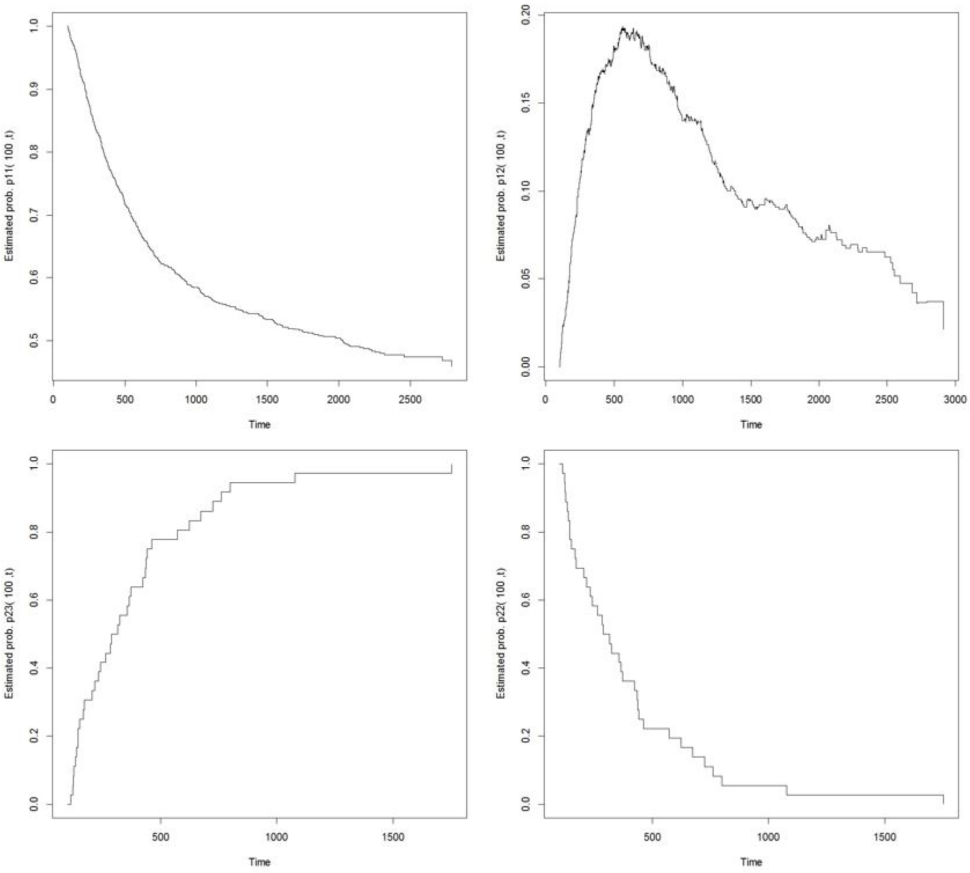

Figure 4.1: Transition probability estimates with first time equal to 100 days using the p3state.msm package for Colon cancer study.

msm

Like the preceding studied models, the homogeneous Markov model (HMM) offers a detailed description of the survival process, making use of all the available information to estimate the effect of prognostic factors and intensity rates. The **msm** package contains functions for fitting general continuous time Markov and hidden Markov multi-state models to longitudinal data. Transition rates and output processes can be modelled in terms of covariates. The first four patient histories are shown below. `ptnum` is the subject identifier; the state occupied is in the variable `state`, with possible values 1, 2, 3 representing alive, recurrence and death respectively; `dtime` is time to enter the state (1, 2 and 3), in days (Table 4.6).

Table 4.6: Sample of the Colon data. Input data for the msm package.

ptnum	dtime	state	rx	sex	age
1	0	1	3	1	43
1	968	2	3	1	43
1	1521	3	3	1	43
2	0	1	3	1	63
2	3087	1	3	1	63
3	0	1	1	0	71
3	542	2	1	0	71
3	963	3	1	0	71
4	0	1	3	0	66
4	245	2	3	0	66
4	293	3	3	0	66

To obtain the structure of data for this package we used the msmdata function. A useful way to summarising multi-state data is as a frequency table of pairs of consecutive states. This is implemented in the function statetable.msm. Can be used the following input commands:

```
> require("msm")
> colon.msm <- msmdata(colon2, pkg = "msm")
> statetable.msm(state, ptnum, data = colon.msm)
```

```
    to
from   1   2   3
   1 423 461  38
   2   0  52 409
```

Thus there were 38 healthy from deaths state and 409 deaths from state 2 (recurrence). 461 patients have been healthy for the recurrence.

We now specify the multi-state model to be fitted to the data. A model is governed by a transition intensity matrix Q. For the cancer Colon example, there are three possible states through which the patient can move. We assume that the patient can advance $1 \to 1$, $1 \to 2$, $1 \to 3$, $2 \to 2$ and $2 \to 3$.

$$\mathbf{Q} = \begin{pmatrix} -(q_{12}+q_{13}) & q_{12} & q_{13} \\ 0 & -q_{23} & q_{23} \\ 0 & 0 & 0 \end{pmatrix} \tag{4.1}$$

We have to indicate which transitions are allowed in our model. For this purpose we must define a matrix of the same size as Q, containing zeros in the positions where the entries of Q are zero. All other positions contain an initial value for the corresponding transition intensity. The diagonal entries supplied in this matrix do not matter, as the diagonal entries of Q are defined as minus the sum of all the other entries in the row. For example:

```
> qmat0 <- rbind(c(0, 0.25, 0.25), c(0, 0, 0.5), c(0, 0, 0))
```

The likelihood is maximized by numerical methods, which need a set of initial values to start the search for the maximum. Initial values for a model could be set by assuming that transitions between states take place only at the observation times. The **msm** package provides a function for calculating initial values. The input command is

```
> qmat1 <- crudeinits.msm(state ~ dtime, ptnum,
data = colon.msm, qmatrix = qmat0)
> qmat1

          [,1]          [,2]         [,3]
-0.0003847123  0.0003554156 2.929673e-05
 0.0000000000 -0.0016624800 1.662480e-03
 0.0000000000  0.0000000000 0.000000e+00
```

To fit the model, we have to call the msm function with the appropriate arguments. We need to have the data set cancer Colon in the appropriate format (as shown in Table 4.6), a matrix indicating the allowed transitions and the initial values. Then we may fit the homogeneous Markov model (HMM) using the following input commands:

```
> colon.msm2 <- msm(state ~ dtime, subject = ptnum,
data = colon.msm, qmatrix = qmat1, exacttimes = TRUE)
> colon.msm2
```

```
        State 1                          State 2
State 1 -0.0003847 (-0.00042,-0.0003524) 0.0003554 (0.0003244,0.0003894)
State 2 0                                -0.001662 (-0.001832,-0.001509)
State 3 0                                0
        State 3
State 1 2.93e-05 (2.132e-05,4.026e-05)
State 2 0.001662 (0.001509,0.001832)
State 3 0
```

To examine the effect of covariates (sex, age and rx), we have to supply a formula to the covariate argument.

```
> colon.msm.t <- msm(state ~ dtime, subject = ptnum,
data = colon.msm, qmatrix = qmat1, death = 3,
covariates = ~ factor(rx) + sex + age)
```

Then, the hazard.msm function gives hazard ratios, and confidence intervals, for the allowed transitions.

```
> hazard.msm(colon.msm.t)
```

```
$`factor(rx)2`
                         HR         L        U
State 1 - State 2 0.9700298 0.7856320 1.197708
State 1 - State 3 0.6608962 0.1114504 3.919088
State 2 - State 3 1.1483350 0.9119585 1.445979


$`factor(rx)3`
                         HR         L         U
State 1 - State 2 0.5947657 0.4749321 0.7448355
State 1 - State 3 0.9237622 0.2061749 4.1388962
State 2 - State 3 1.7146097 1.3423275 2.1901410


$sex
                         HR         L         U
State 1 - State 2 0.9237646 0.7725099  1.104634
State 1 - State 3 4.0005452 0.7410271 21.597540
State 2 - State 3 1.1976163 0.9841540  1.457378


$age
                         HR         L        U
State 1 - State 2 0.9976617 0.9899469 1.005437
State 1 - State 3 1.0825319 1.0040206 1.167183
State 2 - State 3 1.0086508 1.0006585 1.016707
```

The plot method for msm objects produces a plot of the expected probability of survival against time, from each transient state. Survival is defined as not entering the final absorbing state.

```
> plot(colon.msm2, legend.pos = c(1550, 1))
```

This shows that the 1500 day survival probability of health is approximately 0.7 and with as recurrence 0.2. With as relapse the survival probability diminishes very quickly to around 0.4 in the first 500 days after entry the study (Figure 4.2).

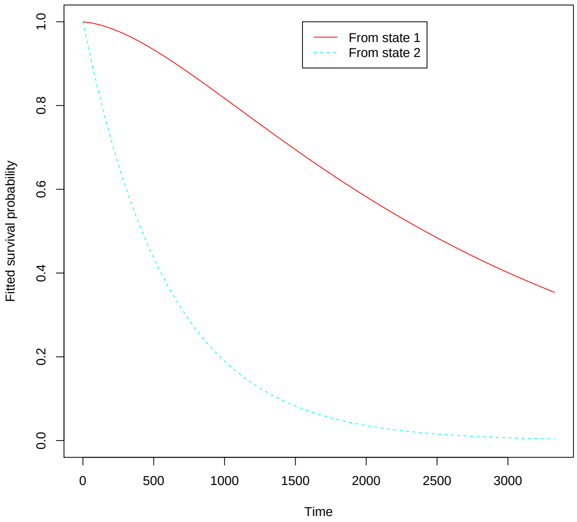

Figure 4.2: Plots of multi-state models.

We note that the **msm** package support a variety of observation schemes, including processes observed at arbitrary times, completely observed processes, and censored states. Any pattern of transitions between states can be specified.

mstate

The **mstate** package allow to estimate hazards and probabilities, possibly depending on covariates, and to obtain prediction probabilities in the context of competing risks and multi-state models. Again, we shall use the database of Colon cancer. The variables required by the package **mstate** are the following: `id`, `from`, `to`, `trans`, `Tstart`, `Tstop`, `time` and `status`. A sample of the data can be seen in Table 4.7.

After having prepared the data in long format using the `msmdata` as shown in the input commands below, estimation of covariate effects using Cox regression is straightforward using the `coxph` function of the survival package (previously shown). The delayed entry aspect of this model for transition 3 is achieved by specifying. We consider first the model without any proportionality assumption on the baseline hazards, for different values of trans (the transitions). The results for the CMM can

Table 4.7: Sample of the Colon data. Input data for the mstate package.

id	from	to	trans	Tstart	Tstop	time	status	rx	sex	age
1	1	2	1	0	968	968	1	3	1	43
1	1	3	2	0	968	968	0	3	1	43
1	2	3	3	968	1521	553	1	3	1	43
2	1	2	1	0	3087	3087	0	3	1	63
2	1	3	2	0	3087	3087	0	3	1	63
3	1	2	1	0	542	542	1	1	0	71
3	1	3	2	0	542	542	0	1	0	71
3	2	3	3	542	963	421	1	1	0	71
4	1	2	1	0	245	245	1	3	0	66
4	1	3	2	0	245	245	0	3	0	66
4	2	3	3	245	293	48	1	3	0	66

be obtained using the following commands:

```
> require("mstate")
> trans <- trans.illdeath(names = c("1", "2", "3"))
> colon.mstate <- msmdata(colon2, pkg = "mstate", tra = trans)
```

The number of events in the data can be summarized with the function `events`.

```
> events(colon.mstate)

$Frequencies
    to
from   1   2   3 no event total entering
   1   0 461  38      423            922
   2   0   0 409       52            461
   3   0   0   0        0              0
```

```
$Proportions
    to
from          1          2          3   no event
   1 0.00000000 0.50000000 0.04121475 0.45878525
   2 0.00000000 0.00000000 0.88720174 0.11279826
   3
```

Now add transition-specific covariates to the dataset, for a numerical covariate, the names of the expanded covariates are `cov.1`, `cov.2`, etc. The extension .i refers to transition number i.

```
> colon.mstate.cov <- expand.covs(colon.mstate,
covs = c("rx", "sex", "age"), append = TRUE, longnames = FALSE)
> c1 <- coxph(Surv(Tstart, Tstop, status) ~ rx1.1 + rx2.1 +
age.1 + sex.1 + rx1.2 + rx2.2 + age.2 + sex.2 + rx1.3 + rx2.3 +
age.3 + sex.3 + strata(trans), data = colon.mstate.cov)
> c1
```

Results, shown in Table 4.8, are in agree with those previously found (see, for example, Table 4.5). The first four lines represent the recurrence transition, from line 5 to line 8 the estimates for the mortality transition without recurrence. Finally, the remaining four lines are for the mortality transition after recurrence.

Table 4.8: Model Markov stratified hazards.

	coef	exp(coef)	p-value
rx1.1	-0.0016	0.9980	0.9900
rx2.1	-0.5200	0.5950	1.5e-05
age.1	-0.0072	0.9930	0.0670
sex.1	-0.1068	0.8990	0.2500
rx1.2	-0.3353	0.7150	0.4300
rx2.2	-0.1670	0.8460	0.6600
age.2	0.0854	1.0890	1.0e-05
sex.2	0.4238	1.5280	0.2100
rx1.3	0.0609	1.0630	0.6000
rx2.3	0.3072	1.3600	0.0170
age.3	0.0073	1.0070	0.0700
sex.3	0.1770	1.1940	0.0790

etm

The **etm** package provides estimates and plots for the transition probabilities for any multi-state model. It can also estimate the variance of the Aalen-Johansen estimator, and handles left-truncated data. The **etm** package permits to compute interesting quantities that depend on the matrix of transition probabilities. The variables required by the package **etm** are the following: `id`, `entry`, `exit`, `from`, `to`. A sample of the data can be seen in Table 4.9.

To use **etm** package, we first need to define the matrix that specifies the possible transitions. Rows represent the states from which a transition may occur whereas the columns designate states to which a transition may occur. For instance, the possible transitions are labeled TRUE. Next, we use `msmdata` function to obtain the structure of data for **etm** package. The `etm` function computes the empirical transition matrix, also called Aalen-Johansen estimator, of the transition probability matrix of any multi-state model. The s represents starting value for computing the transition probabilities

Table 4.9: Sample of the Colon data. Input data for the etm package.

id	entry	exit	from	to	rx	sex	age
1	0	968	1	2	3	1	43
1	968	1521	2	3	3	1	43
2	0	3087	1	cens	3	1	63
3	0	542	1	2	1	0	71
3	542	963	2	3	1	0	71
4	0	245	1	2	3	0	66
4	245	293	2	3	3	0	66

and t is ending value. This function also gives the number of absorbing and transient states and the possible transitions. Then we ran the following commands to obtain which we describe above:

```
> require("etm")
> trans <- matrix(FALSE, 3, 3)
> trans[1, 2:3] <- TRUE
> trans[2, 3] <- TRUE
> colon.etm <- msmdata(colon2, pkg = "etm", tra = trans,
state.names = c("1", "2", "3"), cens.name = "cens")
> etm(data = colon.etm, state.names = c("1", "2", "3"),
tra = trans, cens.name = "cens", s = 100, t = 800)
```

```
Multistate model with 2 transient state(s)
 and 1 absorbing state(s)

Possible transitions:
 from to
    1  2
    1  3
    2  3


Estimate of P(100, 800)
          1         2         3
1 0.6182574 0.1657187 0.2160239
2 0.0000000 0.2113512 0.7886488
3 0.0000000 0.0000000 1.0000000
```

This multi-state model have two transient states $(1, 2)$ and one absorbing state (3). An absorbing state is a process to will never leave an absorbing state once it enters. Is a state from which there is a zero probability of exiting. The probability of a patient to find the healthy 800 days since he was healthy at 100 days is 0.6183. A patient who is in state 2 at time 100, the probability of being in state 3 at time 800 is 0.7886 (The output is not complete because the complete one have the estimated of covariance).

We used in this case the `etm` function with the value zero initial (s) and final value (t) without any specification, will be the last data time. Then we ran the following commands to obtain a plot for the transition probabilities.

```
> my.etm <- etm(data = colon.etm, state.names = c("1", "2", "3"),
tra = trans, cens.name = "cens", s = 0)
> plot(my.etm, c("1 2", "1 3", "2 3"),
col = c("red", "blue", "black"))
```

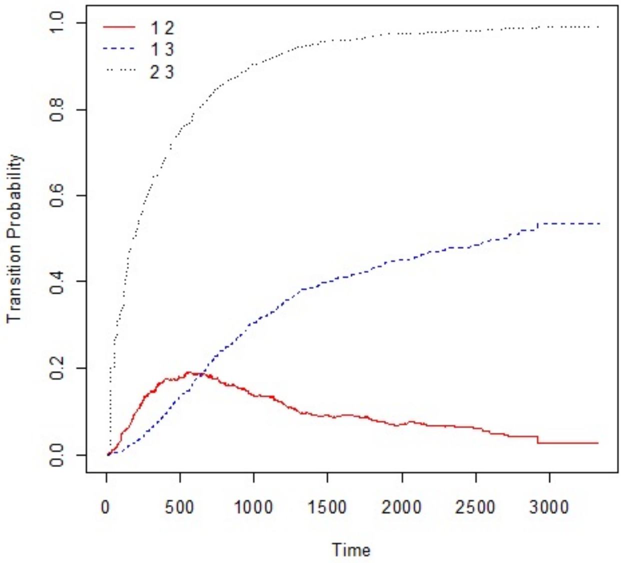

Figure 4.3: Plot with transition probabilities using the etm package. Colon cancer data.

The plot shown in Figure 4.3 has 3 transitions and their transition probability in certain times. Since state 3 is the state of death (absorbing) then the probability of transiting to state 3 (either the 1 or 2) increases over time. The probability of transition $1 \to 2$ increases to about the 500 days and then decreases.

We can calculate the transition probabilities and also the variance with the command trprob and trcov, respectively. We only put here the commands for the transition $1 \to 2$.

```
> p12 <- trprob(my.etm, "1 2", c(30, 365, 730, 1825, 2920))
> var12 <- trcov(my.etm, "1 2", c(30, 365, 730, 1825, 2920))
```

Then we can see again the transition probabilities in a chart, but now the transitions are discrete graphics. Should be remembered that the sum of $P_{11}+P_{12}+P_{13} = 1$ and that $P_{23} = 1 - P_{22}$. This chart also has confidence bands.

```
> xyplot(my.etm, data, c("1 3", "2 3", "1 1", "1 2"))
```

Table 4.10: Transition probabilities and variance for 30, 365, 730, 1825 and 2920 days.

Transition Probabilities	30	365	730	1825	2920
$1 \to 2$	0.0054	0.1649	0.1758	0.0805	0.0291
$1 \to 3$	0.0022	0.0824	0.2246	0.4331	0.5345
$2 \to 3$	0.0000	0.6626	0.8409	0.9677	0.9912
Variance	30	365	730	1825	2920
$1 \to 2$	5.8499e-06	1.4932e-04	1.5728e-04	8.0542e-05	1.7702e-04
$1 \to 3$	2.3476e-06	8.2033e-05	1.8892e-04	2.6661e-04	5.2895e-04
$2 \to 3$	0.0000	8.1231e-03	1.9208e-03	9.4962e-05	2.2719e-05

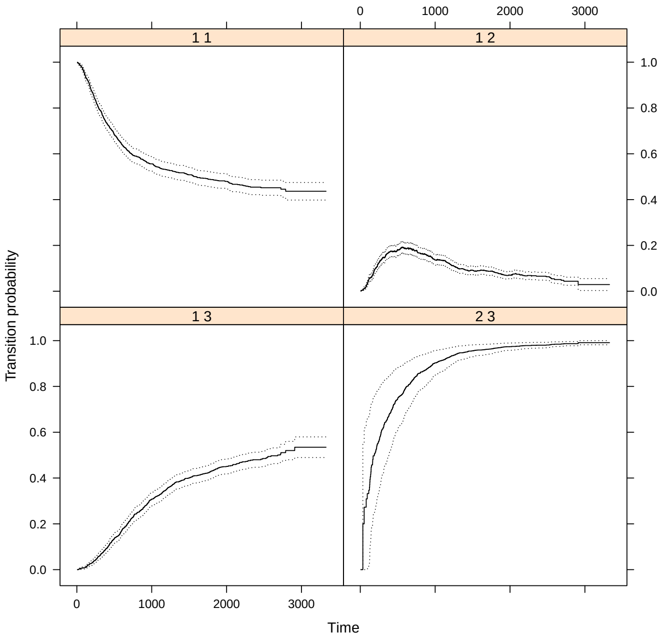

Figure 4.4: Transition probabilities for transitions $1 \to 1$, $1 \to 2$, $1 \to 3$ and $2 \to 3$.

changeLOS

This package was build for computing change in LOS (Change in length of hospital stay) based on methods described in (Schulgen and Schumacher, 1996). The main feature of this R package is: to compute and plot change in length of hospital stay (LOS is used to assess the utilization of hospital resources, the costs and the general impact of a disease)

The estimation techniques used are fully nonparametric, allowing for a time-inhomogeneous Markov process, i.e. the future development of the process depends only on the state currently occupied; the Markov assumption may be dropped for estimation of state occupation probabilities. The new format of the input dataset can be found in Table 4.11.

Table 4.11: Sample of the Colon data. Input data for the changeLOS package.

id	from	to	time	oid
1	0	1	968	1
1	1	2	1521	1
2	0	cens	3087	2
3	0	1	542	3
3	1	2	963	3
4	0	1	245	4
4	1	2	293	4

With the **changeLOS** R package it is possible describe the state names and possible transitions.

```
> require("changeLOS")
> trans <- matrix(FALSE, 3, 3)
> trans[1, 2:3] <- TRUE
> trans[2, 3] <- TRUE
> colon.los <- msmdata(colon2, pkg = "los", tra = trans,
cens.name = "cens")
> tr.prob <- etm(colon.los, c("0", "1", "2"), trans, "cens",s=0)
> cLOS <- etm::clos(tr.prob)
> plot(cLOS)
```

The function `clos` estimates the expected change in length of stay (LOS) associated with an intermediate event (IE), using the Aalen-Johansen estimator for the matrix of transition probabilities.

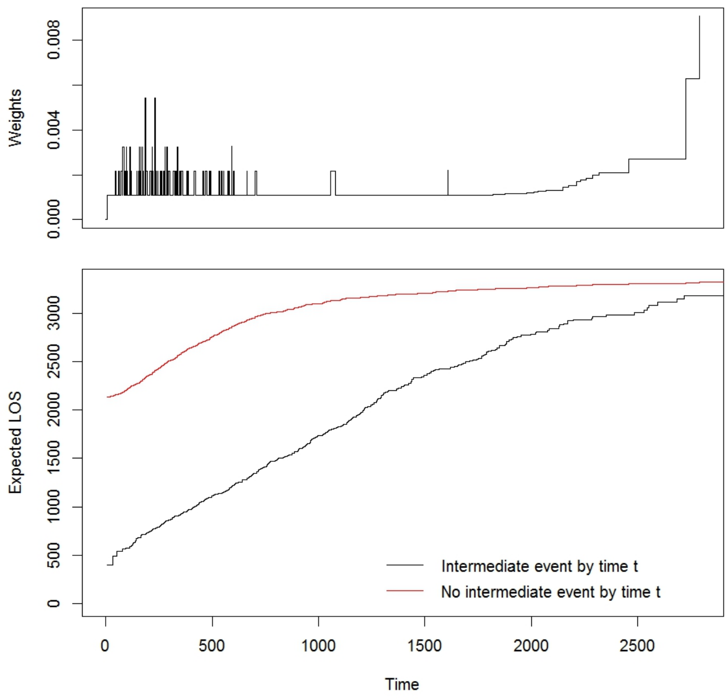

Figure 4.5: Expected change in length of hospital stay (LOS).

The upper graph displays the weights used to compute the weighted average. The lower graph displays the expected LOS for patients who have experienced the intermediate event and for those who have not (Figure 4.5). The black curve indicates the estimated expected time of hospital stay given recurrence has been acquired by time t. The red curve indicates the respective time, given still without recurrence by time t.

mvna

The multivariate Nelson-Aalen estimator of cumulative transition hazards is one important nonparametric estimator in event history analysis. The **mvna** package provides a way to easily estimate and display the cumulative transition hazards from a time-inhomogeneous markov multi-state model. The estimator may remain valid under even more general assumptions. The mvna contains the following functions: the function, `xyplot.mvna`, plots the cumulative hazard estimates in a lattice plot, along with pointwise confidence intervals; the `predict.mvna` function gives Nelson-Aalen estimates at time points given by the user. The main function, `mvna`, computes the Nelson-Aalen estimates at each of the observed event times, and two variance estimators. Finally, the summary function returns an object of class `mvna` which is a list of data frames named after possible transitions.

The variables required by the package are the following: `id`, `from`, `to`, `time`. A sample of the data can be seen in Table 4.12.

Table 4.12: Sample of the Colon data. Input data for the mvna package.

id	from	to	time	rx	sex	age
1	1	2	968	3	1	43
1	2	3	1521	3	1	43
2	1	cens	3087	3	1	63
3	1	2	542	1	0	71
3	2	3	963	1	0	71
4	1	2	245	3	0	66
4	2	3	293	3	0	66

Each data frame of the summary function contains the following columns:

- **na**: nelson-aalen estimates at each transition times
- **time**: the transition times
- **var.aalen**: variance estimator give
- **n.risk**: number at individual at risk in the transient states just before
- **n.event**: number of transitions at each event time

The input commands are the following:

```
> require("mvna")
> trans <- matrix(FALSE, 3, 3)
> trans[1, 2:3] <- TRUE
> trans[2, 3] <- TRUE
> colon.mvna <- msmdata(colon2, pkg = "mvna", tra = trans,
cens.name = "cens", state.names = c("1", "2", "3"))
> col.mvna <- mvna(data = colon.mvna,
state.names = c("1", "2", "3"), tra = trans, cens.name = "cens")
> summary(col.mvna)
```

For illustration purposes we only present the results for transition $1 \to 2$ and $2 \to 3$.

Table 4.13: Nelson-Aalen estimator in multi-state models.

Transition $1 \to 2$					Transition $2 \to 3$				
na	var.aalen	time	n.risk	n.event	na	var.aalen	time	n.risk	n.event
0.00	0	0	922	0	0.00	0.00	9	1	0
0.36	0	496	633	1	1.32	0.06	497	167	0
0.60	0	1178	490	0	2.55	0.07	1166	115	1
0.71	0	2209	274	0	3.79	0.09	2203	49	0
0.72	0	2598	108	0	4.05	0.10	2588	17	0
0.72	0	3325	2	0	4.59	0.22	3192	1	0

The plot method permits to draw several cumulative transition hazards on the same panel and the second one estimates of the cumulative hazards plotted as a function of time for all the transitions specified by the user, `xyplot` can also plot several types of pointwise confidence interval (Figure 4.6).

```
> plot(col.mvna, col = c("red", "blue", "black"))
> xyplot(x = col.mvna)
```

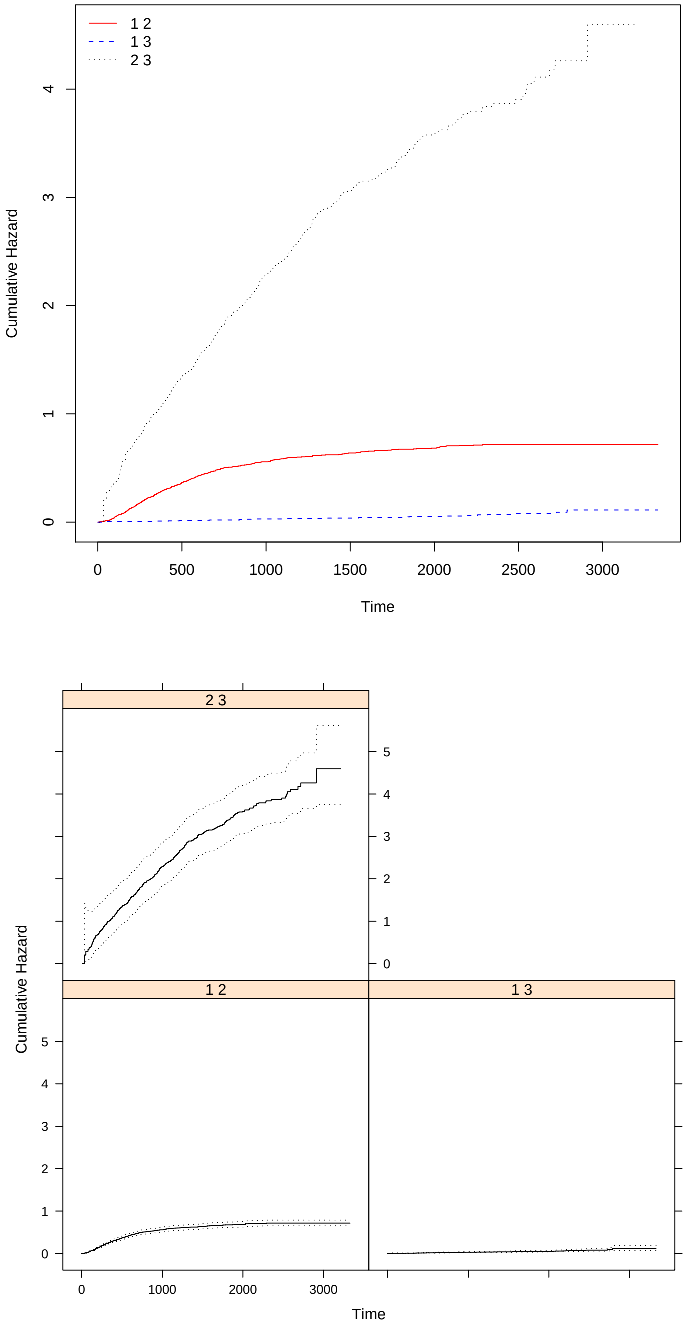

Figure 4.6: Plots for a mvna object for transition $1 \to 2$, $1 \to 3$ and $2 \to 3$.

TPmsm

The **TPmsm** package contains functions to compute estimates for the transition probabilities in the illness-death model and the progressive three-state model. This package can be used to implement seven methods (AJ, PAJ, KMW, KMPW, IPCW, LIN and LS). The Inverse Probability of Censoring (IPCW) and (LIN) estimators also permit to compute transition probabilities conditioned on a single covariate. The package also allow users to obtain plots of the transition probabilities with or without confidence bands. Records in the data must contain the following variables: `time1`, `event1`, `Stime` and `event`. A single covariate can also be included. Each individual is represented by one line of data, just as shown in Table 4.14. We construct the data with the following input commands:

```
> require("TPmsm")
> colnames(colon2) <- c("time1", "event1", "time2", "Stime", "event",
"rx", "sex", "age")
> p <- which(colon2$event == 1 & colon2$event1 == 0)
> colon.tpmsm <- colon2
> colon.tpmsm[p, ]$event1 <- 1
> colonTP_obj1 <- with(colon.tpmsm, survTP(time1, event1, Stime, event))
```

Table 4.14: Sample of the Colon data. Input data for the TPmsm package.

time1	event	Stime	event1
968	1	1521	1
3087	0	3087	0
542	1	963	1
245	1	293	1

The following four input commands provide the estimate for the KMW, KMPW, AJ and PAJ methods. With this commands we can obtain the estimates with or without 95% (`conf.level = 0.95`) pointwise confidence intervals (`conf = TRUE`) using 1000 bootstrap replicates (`n.boot = 1000`).

```
> transKMW(object = colonTP_obj1, s = 100, t = 800)
> transKMPW(object = colonTP_obj1, s = 100, t = 800)
> transAJ(object = colonTP_obj1, s = 100, t = 800)
> transPAJ(object = colonTP_obj1, s = 100, t = 800)
```

```
Kaplan-Meier Weighted transition probabilities

Estimates of P(100, 800)
          1          2         3
1 0.6182574 0.17201988 0.2097227
2 0.0000000 0.05531639 0.9446836
3 0.0000000 0.00000000 1.0000000

Presmoothed Kaplan-Meier Weighted transition probabilities

Estimates of P(100, 800)
          1          2         3
1 0.6176346 0.17211832 0.2102471
2 0.0000000 0.05618279 0.9438172
3 0.0000000 0.00000000 1.0000000

Aalen-Johansen transition probabilities

Estimates of P(100, 800)
          1         2         3
1 0.6182574 0.1657187 0.2160239
2 0.0000000 0.2113512 0.7886488
```

```
3 0.0000000 0.0000000 1.0000000


Presmoothed Aalen-Johansen transition probabilities


Estimates of P(100, 800)
          1         2         3
1 0.6176346 0.1656508 0.2167146
2 0.0000000 0.2114037 0.7885963
3 0.0000000 0.0000000 1.0000000
```

The results for the estimator KMW are the same as for the **p3state.msm** package. The AJ and PAJ methods are described in Chapter 3 and we can show that the results are the same as for the **etm** package in the case of the AJ estimator. In Moreira et al. (2013) we described both approaches (AJ and PAJ).

In addition to the numerical results graphical outputs can also be obtained. Figure 4.7 plots the transition probabilities for all allowed transitions using PAJ method. This plot can be obtained using the following input commands:

```
> AJ <- transPAJ(object=colonTP_obj1, s=0, conf=TRUE, conf.level=0.95)
> plot(AJ, tr.choice=c("1 1","1 2", "1 3","2 2", "2 3"),
ylab="Pij(0, Time)", xlab="Time",col=1:5, lty=1, conf.int=TRUE)
```

The graph in Figure 4.7 is a version presmoothed of the graph in Figure 4.3 that we see in **etm** package. **TPmsm** package provides all allowed transitions and 95% confidence bands.

Alternatively, we can view all transitions in the same chart but in differents plots using the following input commands:

```
> tr.choice <- colnames(AJ$est)
> par.orig <- par( c("mfrow", "cex") )
> par( mfrow = c(2, 3) )
> for ( i in seq_len( length(tr.choice) ) ) {
```

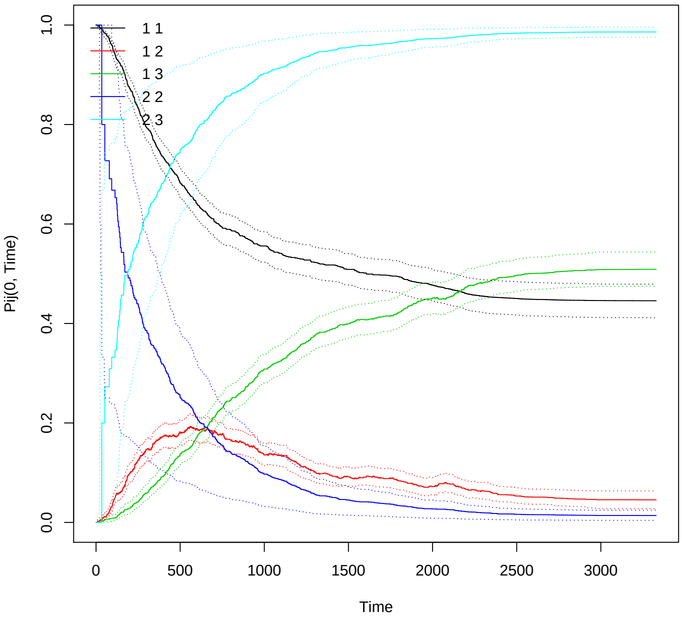

Figure 4.7: Plot with transition probabilities. The TPmsm package with Colon cancer data.

```
> plot(AJ, tr.choice = tr.choice[i], col = 1, lty = 1, legend = FALSE,
main = tr.choice[i], xlab = "", ylab = "", conf.int = TRUE)}
```

Figure 4.8 show the allowed transition for the PAJ method. This graph is the version presmoothed of the plot in Figure 4.4 (**etm** package).

genSurv

The **genSurv** package provides functions to generate data for different approaches from a progressive illness-death model. This package permits generate for the Cox Markov model (genCMM), Cox proportional hazard model (genCPHM), Cox model with time-dependent covariates (genTDCM) and time-homogeneous Markov model (genTHMM).

This package can be used to generate multi-state survival data, for example data arising from the widely used Cox Markov model represented by several lines. Results

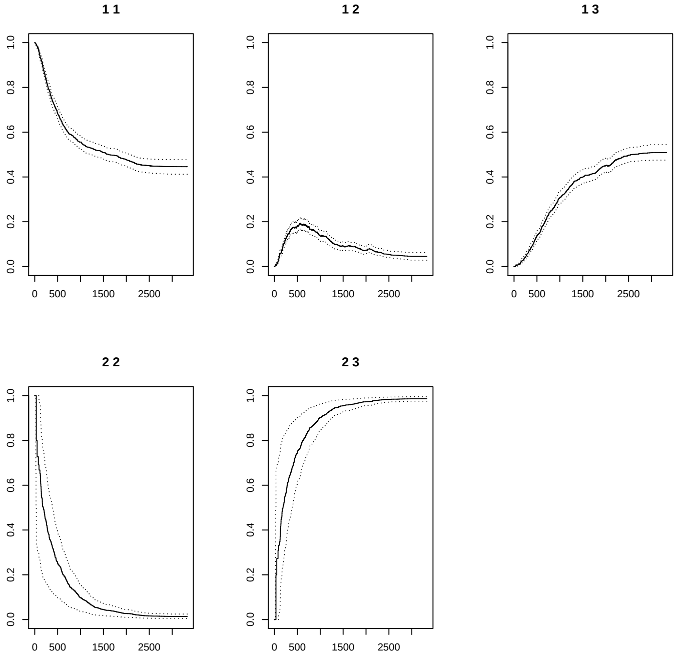

Figure 4.8: Transition probabilities estimates using the TPmsm package for Colon cancer data.

can easily obtained using the *R* **survival** package. Such data can be constructed with the following input commands:

```
> require("genSurv")
> cmmdata <- genCMM(n = 1000, model.cens = "uniform", cens.par = 2.5,
beta = c(2, 1, -1), covar = 10, rate =c (1, 5, 1, 5, 1, 5))
> head(cmmdata, n = 11)
> library("survival")
> fit_12 <- coxph(Surv(start, stop, event) ~ covariate, data = cmmdata,
subset = c(trans == 2))
```

```
   id      start       stop event covariate trans
1   1 0.00000000 0.51250161     0  1.890400     1
2   1 0.00000000 0.51250161     1  1.890400     2
3   1 0.51250161 0.54344208     0  1.890400     3
```

```
4    2 0.00000000 0.39193057      0  2.112890      1
5    2 0.00000000 0.39193057      0  2.112890      2
6    3 0.00000000 0.09490459      0  4.928487      1
7    3 0.00000000 0.09490459      1  4.928487      2
8    3 0.09490459 0.99449772      0  4.928487      3
9    4 0.00000000 0.02275807      0  7.640448      1
10   4 0.00000000 0.02275807      1  7.640448      2
11   4 0.02275807 1.17040300      0  7.640448      3
```

This kind of database have the structure that work with **survival** package (Table 4.4).

Using the following commands we simulated the time-homogeneous Markov model. In this output we can see the structure for use **msm** package (Table 4.6).

```
> thmmdata <- genTHMM(n = 100, model.cens = "uniform", cens.par = 80,
beta = c(0.09, 0.08, -0.09), covar = 80, rate = c(0.05, 0.04, 0.05))
> head(thmmdata, n = 11)
> library("msm")
> qmat0 <- rbind(c(0, 0.25, 0.25), c(0, 0, 0.5), c(0, 0, 0))
> qmat1 <- crudeinits.msm(state ~ time, PTNUM, data = thmmdata,
qmatrix = qmat0)
> msm.t <- msm(state ~ time, subject = PTNUM, data = thmmdata,
qmatrix = qmat1, exacttimes = TRUE, covariates = ~covariate)
> hazard.msm(msm.t)
```

```
    PTNUM       time state covariate
1       1  0.0000000     1  16.03930
2       1  1.0224553     2  16.03930
3       1 24.0678599     2  16.03930
4       2  0.0000000     1  37.27837
5       2  0.2045838     2  37.27837
```

```
6       2  2.4227099     2  37.27837
7       3  0.0000000     1  52.91009
8       3  0.1967006     3  52.91009
9       4  0.0000000     1  24.48898
10      4  1.2975183     2  24.48898
11      4 75.9107707     2  24.48898
```

4.3 The survivalBIV package

The **survivalBIV** software contains functions that calculate estimates for the bivariate distribution function. As mentioned in Chapter 2, this package can be used to implement four methods (CKM, KMW, KMPW and Lin). Here we will call IPCW to the Lin estimator (equation 2.5) in the Chapter 2, Section 2.2.2. This software is intended to be used with the `R` statistical program R Team (2010). Our package is composed of 9 functions that allow users to obtain estimates for the bivariate distribution function. Table 4.15 provides a summary of the functions in this package.

Users can obtain the estimates for the methods discussed in Chapter 2 by means of three functions, namely, `survBIV`, `summary` and `plot`. Details on the usage of these functions can be obtained with the corresponding help pages. It should be noted that to implement the methods described in Chapter 2 one needs the following variables: `time1`, `event1`, `time2` and `event2`. Covariates have not been included in any of the implemented methods, therefore they are not necessary. The variable `time1` represents the observed time of the first event (first gap time), and `event1` the status indicator of the first gap time (if the first gap time is a censored observation, the value is 0 and otherwise the value is 1). The variable `time2` represents the observed second time (second gap time). If `event1 = 0`, the second gap time is not observed and then `time2 = 0`. The variable `event2` is the final status of the individual (takes the value 1 if the second event of interest is observed and 0 otherwise).

Table 4.15: Summary of functions in the package.

Function	Description
`dgpBIV`	A function that generates bivariate censored gap times from some known copula functions. By default returns a dataset of class `survBIV`
`corrBIV`	Provides the correlation between the bivariate times for some copula distributions.
`survBIV`	Provides the adequate dataset for implementing all the four methods. The new dataset is of class `survBIV`.
`bivCKM`	Provides estimates for the bivariate distribution function for the Conditional Kaplan-Meier estimator, CKM.
`bivIPCW`	Provides estimates for the bivariate distribution function for the Inverse Probability of Censoring Weighted estimator, IPCW.
`bivKMW`	Provides estimates for the bivariate distribution function for the Kaplan-Meier Weighted estimator, KMW.
`bivKMPW`	Provides estimates for the bivariate distribution function for the Kaplan-Meier Presmoothed Weighted estimator, KMPW.
`plot`	A function that provides the plots for the bivariate distribution function and marginal distribution of the second time.
`summary`	Summary method for objects of class `survBIV`.

4.4 Data Generation

Users may use the function `dgpBIV` to generate bivariate survival data. This function can be used to generate bivariate survival times from two of the most known copula functions: Gumbel's bivariate exponential distribution (Lu and Bhattacharya, 1990, 1991), also known as the Farlie-Gumbel-Morgenstern distribution and the bivariate weibull distribution. In the book by Johnson and Kotz (1972) several bivariate distributions are discussed and procedures of construction are given.

It is well known that exponential and weibull distributions are very useful for modelling survival times. The Farlie-Gumbel-Morgenstern distribution is given by $F(x, y) = F_1(x)F_2(y)[1 + \delta(1 - F_1(x))(1 - F_2(y))]$ where the marginal distribution functions F_1 and F_2 are exponential with rate parameter θ_i, $i = 1, 2$ and where $|\delta| \leq 1$ is the association parameter. The case of independence is obtained for $\delta = 0$ while the maximum of correlation (between T_1 and T_2) for the bivariate exponential distribution is obtained for $\delta = 1$ with bound equal to 0.25. These and other theoretical correlations between the bivariate times for this copula distribution (with unit marginal distributions) can be obtained using the input commands shown below.

```
> library("survivalBIV")
> corrBIV(dist = "exponential", corr = 0, dist.par = c(1, 1))
> corrBIV(dist = "exponential", corr = 1, dist.par = c(1, 1))
```

In the following, using the `dgpBIV` function we will simulate bivariate exponential survival data (`dist = "exponential"`). We will use this data to explain and illustrate how numerical output for all methods can be obtained using the functions in the package. We will follow the simulation scenario described by Lin et al. (1999). We will simulate 1000 observations (`n = 1000`) assuming a maximum correlation of 0.25 (`corr = 1`) and use an independent uniform censoring time (`model.cens = "uniform"`), according to model $U(0, 3)$ (`cens.par = 3`).

```
> set.seed(1500)
> sim_data_exp <- dgpBIV(n = 1000, corr = 1, dist = "exponential",
model.cens = "uniform", cens.par = 3, dist.par = c(1, 1))
```

To obtain the estimates for the methods proposed in Chapter 2 we can use the functions shown in Table 4.15. As in the simulation by Lin et al. (1999) we are going to obtain estimates for bivariate distribution at values $t_1 = 0.5108$ and $t_2 = 0.9163$. The true value is 0.2976. The following input command provides the estimate for the `KMW` method. With this command we obtain the pointwise confidence intervals (`conf = TRUE`) using a 1000 bootstrap replicates (`n.boot = 1000`). The construction of the pointwise confidence intervals is obtained by randomly sampling the n items from the original data set with replacement. This can be achieved using percentile bootstrap (`method.boot = "percentile"`) or using basic bootstrap (`method.boot = "basic"`). By default all functions use the `percentile` bootstrap (Davison and Hinkley, 1997).

```
> bivKMW(object = sim_data_exp, t1 = 0.5108, t2 = 0.9163, conf = TRUE,
conf.level = 0.95, n.boot = 1000)


               2.5%     97.5%
0.3015313 0.2692518 0.3337527
```

One important issue is whether 1000 is a suitable number of resamples to generate. Since a second and a third set of 1000 resamples gave similar results for the bootstrap confidence intervals, this suggests that with these number of resamples the results are consistent. From this perspective 1000 would seem sufficient.

The CPU time needed for running the `bivKMW` function varies according to whether bootstrap confidence bands are requested or not, the sample size, and the type of processor in the PC computer. The command presented above took no more than 2 second on a PC with an Intel Core i7 processor with 8 GB memory. The same input command but with $n = 10000$ resamples took a little more than 17 seconds.

Results for the other methods are very similar and can be obtained using the functions `bivKMPW`, `bivCKM` and `bivIPCW` with the same arguments. The `bivIPCW` function has one extra argument which allows the user to choose how to estimate $\widehat{G}$ in (Chapter 2 equation 2.6) `method.cens = "KM"` for the Kaplan-Meier method and `method.cens = "prodlim"` for the method proposed in **prodlim** package. In general, the two methods (for estimating the survival of censoring times) provide similar results; without ties (e.g., using simulated data) they provide the same result. The method based on Kaplan-Meier is implemented in C language and is faster.

The `summary` function can be used to obtain estimates for the bivariate distribution function. This function allows the user to obtain the estimates for all four methods using `method = "all"`:

```
> summary(object = sim_data_exp, t1 = 0.5108, t2 = 0.9163, conf = TRUE,
conf.level = 0.95, n.boot = 1000, method = "all")
```

```
F( 0.5108 , 0.9163 )=
$CKM
               2.5%     97.5%
0.3001276 0.2695496 0.3321113
$IPCW
               2.5%     97.5%
0.2905816 0.2569556 0.3252051
$KMPW
               2.5%     97.5%
0.2982088 0.2669539 0.3274368
$KMW
               2.5%     97.5%
0.3015313 0.2688383 0.3338806
```

The CPU time needed for running the command presented below took a little more than 16 seconds. The same input command but with a sample size of $n = 10000$ took a little more than 68 seconds. Note that this input command is the one which requires more computational effort since all methods are implemented with bootstrap confidence bands (optional).

One limitation of the so-called Farlie-Gumbel-Morgenstern families of bivariate cdf's, is that the correlation of T_1 and T_2 can never exceed $1/3$ (0.25 in the bivariate exponential distribution). The bivariate weibull distribution allows for a larger correlation, which makes it superior to Gumbel's bivariate exponential. The `dgpBIV` function allows the user to generate a pair of times from the bivariate weibull distribution with two-parameter marginal distributions. Its survival function is given by

$$S(x,y) = P(T_1 > x, T_2 > y) = exp\left[-\left[\left(\frac{x}{\theta_1}\right)^{\frac{\beta_1}{\delta}} + \left(\frac{y}{\theta_2}\right)^{\frac{\beta_2}{\delta}}\right]^{\delta}\right]$$

where $0 < \delta \leq 1$, and each marginal distribution has shape parameter β_i and a scale parameter θ_i, $i = 1, 2$. The correlation between the two gap times may be obtained though it is a complicated function of the shape and scale parameters and of δ. Again, the function `corrBIV`, from the **survivalBIV** package can be used to calculate the theoretical correlation between times for this bivariate distribution. This function may be valuable for choosing the appropriate shape and scale parameters. For example, choosing $\delta = 0.6$, $\theta_1 = \theta_2 = 7$ and shape parameters $\beta_1 = \beta_2 = 2$, lead to about 54% of correlation. Below follow two input commands to illustrate the use these two functions. The first command provides the theoretical correlation while the second generates bivariate survival data from the bivariate weibull with exponential censoring with rate parameter 0.08.

```
> corrBIV(dist = "weibull", corr = 0.6, dist.par = c(2, 7, 2, 7))
> sim_data_wei <- dgpBIV(n = 200, corr = 0.6, dist = "weibull",
model.cens = "exponential", cens.par = 0.08,
dist.par = c(2 ,7 ,2 ,7), to.data.frame = TRUE)
```

It is important to note that the conditional Kaplan-Meier estimator can be obtained using the **survival** package alone. For example, for $t1 = 6.7006$ and $t2 = 8.8805$ this can be obtained through the following input commands:

```
> library("survival")
> KM1 <- survfit(Surv(time1, event1) ~ 1, data = sim_data_wei)
> KM2 <- survfit(Surv(time2, event2) ~ 1, data = sim_data_wei,
subset = c(time1 <= 6.7006 & event1 == 1))
> CKM <- (1 - summary(KM1, time = 6.7006) $ surv) *
(1 - summary(KM2, time = 8.8805) $ surv)
```

However, the `bivCKM` function in our package is simpler and allows the user to obtain the same estimate together with the bootstrap confidence bands:

```
> sim_data_wei2 <- with(sim_data_wei,
survBIV(time1, event1, time2, event2))
> bivCKM(object = sim_data_wei2, t1 = 6.7006, t2 = 8.8805)
```

The **survival** package can also be used to obtain the marginal distribution of the second gap time for the CKM method. According to equation in Chapter 2, this can be obtained using the following input commands:

```
> dft1 <- survfit(Surv(time1, event1) ~ 1, data = sim_data_wei)
> dft2 <- survfit(Surv(time2, event2) ~ 1, data = sim_data_wei,
subset = (event1 == 1))
> (1 - summary(dft2, time = 8.8805) $ surv) * (1 - summary(dft1,
time = max(summary(dft1) $ time)) $ surv)
```

Again, our package is simpler and it provides bootstrap confidence bands. Users can easily obtain these results for a specific method (using one of the four functions) or for all methods. The input commands are shown below.

```
> bivCKM(object = sim_data_wei2, t1 = Inf, t2 = 8.8805,
conf = TRUE, conf.level = 0.95, n.boot = 1000)
> summary(object = sim_data_wei2, t1 = Inf, t2 = 8.8805,
conf = TRUE, conf.level = 0.95, n.boot = 1000)
```

In addition to the numerical results graphical output can also be obtained. This will be shown in the next section using data from the well-known Bladder cancer study. Details about this dataset are given below.

4.5 Data Illustration

To illustrate our methods we will use data from a Bladder cancer study, previously presented in Chapter 1. From the total of 85 patients, 47 relapsed at least once and, among these, 29 experienced a new recurrence. We have a total amount of censoring of 66% from which 44.7% is obtained from censored observations on the first gap time. We have about 38% of censored total time among the uncensored first gap time. Here, only the first two recurrence times (in months) and the corresponding gap times, T_1 and T_2, are considered.

There is a high percentage of censored total time (T's) which in general lead to difficulties in the estimation of the bivariate distribution function. The presence of a reasonable amount of censored T's among the uncensored T_1's suggests that presmoothing could lead to an important reduction of variance in estimation (see de Uña-Álvarez and Amorim (2011)).

We will calculate estimates for the bivariate distribution function in several points and plot these estimates. This will be done using the **survivalBIV** package.

In the following, we will demonstrate the package capabilities using data from the Bladder cancer study. Below is an excerpt of the data with one row per individual.

```
> data("bladderBIV", package = "survivalBIV")
> head(bladderBIV)
```

```
 time1 event1 time2 event2
     1      0     0      0
     4      0     0      0
     7      0     0      0
    10      0     0      0
     6      1     4      0
    14      0     0      0
```

Each line represents the information from one individual in study. Among the first five observations, only individual represented by line 5 had a recurrence. This individual had a recurrence on month 6 and remained alive and without second recurrence until time 10 (months). Note that `event1 = 0` and `event2 = 0` (the remaining five observations) corresponds to a censored first gap time in the initial state ("remained alive without a recurrence"). All observations with `event1 = 1` and `event2 = 1` corresponds to individuals with a first recurrence and a second recurrence.

We computed the estimated values for the four estimators of $F_{12}(x, y)$, for x equals to 3, 13, 29 and 49 and y values 3, 10, 17.75 and 36.75, corresponding to marginal survival probabilities of 0.25, 0.5, 0.75 and 0.95. For illustration purposes we only report the estimated values of $F_{12}(x, y)$ for two pairs of gap times with 95% bootstrap confidence intervals.

```
> bladder_obj <- with(bladderBIV, survBIV(time1, event1,
time2, event2))
> summary(object = bladder_obj, t1 = 13, t2 = 10, method = "all",
conf = TRUE, n.boot = 10000)
> summary(object = bladder_obj, t1 = 29, t2 = 36.75,
method = "all", conf = TRUE, n.boot = 10000)
```

```
F( 13 , 10 )=
$CKM
                2.5%      97.5%
0.16836961 0.08841834 0.25478264
$IPCW
                2.5%      97.5%
0.15100626 0.06731521 0.24149771
$KMPW
                2.5%      97.5%
0.16815396 0.09382032 0.25254201
$KMW
                2.5%      97.5%
0.17192598 0.09163352 0.26381938

F( 29 , 36.75 )=
$CKM
               2.5%     97.5%
0.4498655 0.3276738 0.5755503
$IPCW
               2.5%     97.5%
0.4932222 0.3499283 0.6320595
$KMPW
               2.5%     97.5%
0.4303138 0.3090228 0.5603079
$KMW
               2.5%     97.5%
0.4349590 0.3119461 0.5603330
```

In this case it is clearly seen that the four methods can provide quite different results, specially for higher values of x or y (where the censoring effects are stronger). The CPU time needed for running the input commands presented above took no more than 2 minutes.

The outputs for the bivariate distribution function and for the marginal distribution of the second gap time are useful displays that greatly help to understand the patients course over time. Plots for these two quantities can easily be obtained. Figure 4.9 plots the marginal distribution function of the second gap time (time from first to second recurrence) for all methods. These plots are obtained using the following input commands:

```
> plot(bladder_obj, plot.marginal = TRUE, method = "KMW",
ylim = c(0, 0.65), xlim = c(0, 45))
> plot(bladder_obj, plot.marginal = TRUE, method = "KMPW",
ylim = c(0, 0.65), xlim = c(0, 45))
> plot(bladder_obj, plot.marginal = TRUE, method = "IPCW",
ylim = c(0, 0.65), xlim = c(0, 45))
```

```
> plot(bladder_obj, plot.marginal = TRUE, method = "CKM",
ylim = c(0, 0.65), xlim = c(0, 45))
```

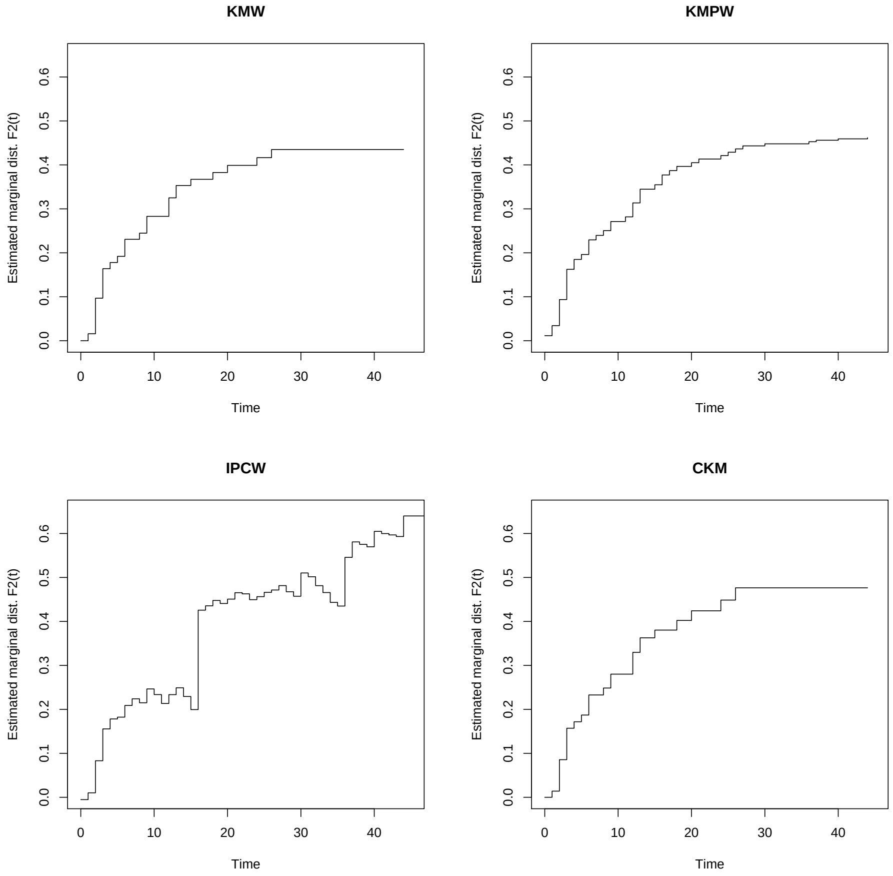

Figure 4.9: Marginal distribution function of the second gap time. Bladder cancer data.

In Figure 4.9 we can see new insights for each method, for example, about the number of jump points and monotonicity. In this graphical output we have on top the semiparametric estimator (right) and the method without presmoothing. The main difference between the first two methods is that the semiparametric estimator has more jump points, explicitly the censored values of the total time for which the first gap time is uncensored. Below, the method based on Bayes' theorem (CKM) and the method based on inverse censoring. Clearly, we can see that estimator based on inverse censoring (IPCW) provides a plot with more jump points than the remaining methods. Note also that this method provides non-monotone curves. In regard to the number of jump points and monotonicity, similar behaviors can be found in the plots for the bivariate distribution function (Figures 4.10 and 4.11). For illustration purposes we only present the plot for the semiparametric method. These plots are obtained through the following input command,

```
> plot(bladder_obj, plot.bivariate = TRUE, method = "KMPW")
```

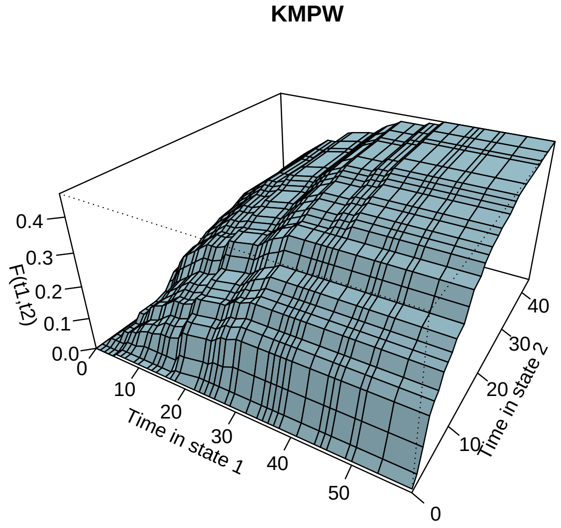

Figure 4.10: Bivariate distribution function. Bladder cancer data.

Plots for the different methods can be obtained by simply changing the `method` argument.

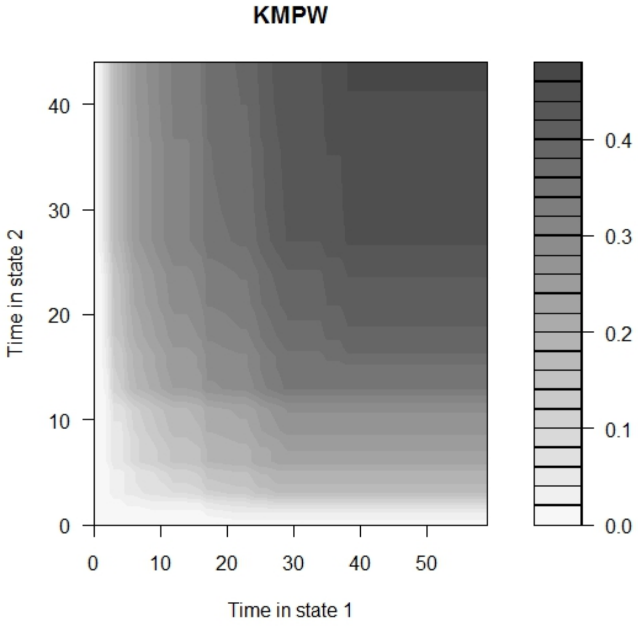

Figure 4.11: Contour plots for the bivariate distribution. Bladder cancer data.

Chapter 5

Conclusions and Future Research

In this dissertation we have presented several methodological contributions to the analysis of multi-state survival data and discussed their application to real biomedical datasets. Below, we go through the main results presented, jointly with some resulting open questions and related fields that motivate future research.

In the Chapter 2 we present a new estimator for the bivariate distribution function based on the Kaplan-Meier estimator. In addition, two estimation methods are also given for the bivariate distribution function conditionally on current or past covariate measures. Both estimators deal with the problem of dependent censoring. The performance of all methods is investigated through simulations and illustrated using real data. It would be interesting to provide some theoretical results for these quantities. We conjecture that this could be done by following lines similar to those in the paper by Akritas and Keilegom (2001), but the complete adaptation of to our context is still undeveloped. This is a topic of our current investigation and hopefully will be published soon.

There has been several recent contributions for the estimation of the transition probabilities in the context of multi-state models. However, the Aalen-Johansen estimator is still the standard method for estimating these quantities in Markov models. In Chapter 3 we propose a modification of Aalen-Johansen estimator in the illness-death model, based on a preliminary estimation (presmoothing) of the censoring probability for the total time (respectively, of the sojourn time in state 1), given the available information. An interesting open question is if this idea can be generalized (and how) to more complex multi-state models. We have derived the consistency of the proposed estimators. The consistency result is not restricted to parametric presmoothing, but it also includes the possibility of using some nonparametric estimators to this end. We verified through simulations that the method based on the presmoothing may be much more efficient than the original Aalen-Johansen estimators, even when there is some misspecification in the chosen parametric family. To this regard, it is worth mentioning that possible misspecifications in the presmoothing model will introduce some bias, while still allowing for a variance reduction. The size of the bias will depend on the misspecification level of the chosen presmoothing model, and on the amount of censored information. Dikta et al. (2005) studied this problem under a misspecified parametric model, showing that the bias component increases with the model's misspecification degree and the proportion of censored observations.

In a different context, the relative importance of introducing parametric information with censored data was investigated by Miller (1983). Similarly, in our scenario, relative advantages of presmoothing are more clearly seen with an increasing censoring degree and at the distribution's right tail. In such a case, standard corrections for censoring typically exhibit a large variance; however, presmoothing functions, when accurately estimated, offer a joint control of both the bias and the variance in estimation. Importantly, the validity of a given model for presmoothing can be checked graphically or formally, by applying a goodness-of-fit tests (e.g. Dikta et al. (2006) and Hosmer and Lemeshow (2008) for the logistic model). This implies that the risk of introducing a large bias through a misspecified model can be controlled in practice. We illustrated the proposed methodology and all this preliminary investigation of the presmoothing model using data from the Stanford Heart transplant study.

We have not investigated the semiparametric efficiency of the proposed presmoothed Aalen-Johansen estimator. Indeed, there is some lack of research in this line even for the basic estimators introduced in the seminal papers on semiparametric censorship models (Dikta, 1998; Dikta et al., 2005). As an exception, we point out that efficiency results are available for some particular family of semiparametric censorship models (see e.g. Zhang (2004)). We wonder if these type of results can be derived also for the semiparametric Aalen-Johansen estimator. This is an interesting topic for our future research.

In Chapter 3 we have not dealt with the possible effect of covariates on the transition probabilities. However, it is possible to include covariates in the presmoothed estimator following the usual approach for Markov models. For this, one just considers each transition probability as a certain transformation of the transition intensity functions. Then, transition intensities may be allowed to depend on covariates following Cox-type regression models. See e.g. Andersen et al. (2000). In order to estimate the regression parameters and the baseline transition intensities, one needs however to adapt the likelihood function to the new setting of presmoothing in which some parametric information on the conditional probability of uncensoring is available. Details are not obvious and will be considered in our future research.

The original and the presmoothed AJ estimators are consistent in Markov models. If the Markov property is violated, then the consistency of the time-honored Aalen-Johansen estimator and of its presmoothed version can not be ensured in general. Exceptions to this are the estimator for $p_{11}(s,t)$ (for which the Markov assumption is empty) or for $p_{ij}(0,t)$ (the so-called stage occupation probabilities, see Datta and Satten (2001)). Alternative estimators of the transition probabilities not relying on the Markov condition were recently proposed (Meira-Machado et al., 2006; Amorim et al., 2011). As a drawback, these alternative methods will suffer from a larger variance in estimation, particularly when the sample size is small and there is a large censoring degree. Consequently, AJ-type estimators will be preferred when there is no strong evidence against the Markov condition.

The main goal of Chapter 4 is to provide an up-to-date review of the existing software for implementing multi-state models. To illustrate the use of these packages we have used data from a Colon cancer study. We hope that this illustration will encourage the applied researches to use multi-state modelling more frequently or with greater confidence, as part of their routine data analysis techniques.

The Colon cancer data enables us to illustrate the use of several *R* packages for the analysis of multi-state survival data arising from the illness-death model. This model is probably the most used model in literature. However, it is important to mention that several of these packages go far beyond this model.

One severe difficulty in the analysis of multi-state survival data is that each of these packages require its own input dataset. To avoid this limitation we developed an *R* based function which can be used to obtain each of the required input format for each software. In this way, users may easily analyse the results offered by the various packages in order to compare them and make decisions accordingly. For the moment, our functions are only valid for the illness-death model. One important issue is the extension of these functions to a general multi-state model.

Chapter 4 discusses implementation in R of some newly developed methods for the bivariate distribution function for censored gap times. The **survivalBIV** package uses four nonparametric and semiparametric estimators. One of these estimators is the conditional Kaplan-Meier, based on Bayes' theorem and Kaplan-Meier estimator; also, two recent estimators based on the Kaplan-Meier weights pertaining to the distribution of the total time (time to the second or final event of interest). It also implements the inverse probability of censoring weighted estimator proposed by Lin et al. (1999). The package allows for numerical results as well as graphics to be easily obtained. Covariates have not been included in our methods. This is a topic of current research and hopefully will be implemented in future. We plan to constantly update **survivalBIV** package to cope with other estimators.

The methods developed in this thesis will be applied to a real dataset on breast cancer from the hospital of Guimarães. A protocol has been established and a dataset with more than 200 patients has been collected. Results from this study will be publisher elsewhere.

Bibliography

Aalen, O. and Johansen, S. (1978). An empirical transition matrix for non-homogeneous markov chains based on censored observations, *Scandinavian Journal of Statistics* **5**: 141–150.

Akritas, M. (1994). Nearest neighbor estimation of a bivariate distribution under random censoring, *The Annals of Statistics* **22**(3): 1299–1327.

Akritas, M. G. and Keilegom, I. V. (2001). Non-parametric estimation of the residual distribution, *Scandinavian Journal of Statistics* **28**: 549–567.

Akritas, M. G. and Keilegom, I. V. (2003). Estimation of bivariate and marginal distributions with censored data, *Journal of Royal Statistical Society, B* **65**: 457–471.

Alioum, A. and Commenges, D. (2001). Mkvpci: A computer program for markov models with piecewise constant intensities and covariates, *Computer Methods and Programs in Biomedicine* **64**: 109–119.

Allignol, A., Beyersmann, J. and Schumacher, M. (2008). mvna: An r package for the nelson-aalen estimator in multistate models, *R News* **8**: 48–50.

Amorim, A. P., de Uña-Álvarez, J. and Meira-Machado, L. (2011). Presmoothing the transition probabilities in the illness-death model, *Statistics & Probability Letters* **81**(7): 797–806.

Andersen, P., Esbjerj, S. and Sorensen, T. (2000). Multi-state models for bleeding episodes and mortality in liver cirrhosis, *Statistics in Medicine* **19**(4): 587–599.

Andersen, P. and Gill, R. D. (1982). Cox's regression model for counting processes: A large sample study, *The Annals of Statistics* **10**(4): 1100–1120.

Andersen, P. K., Borgan, Ø., Gill, R. D. and Keiding, N. (1993). *Statistical Models Based on Counting Processes*, Springer-Verlag, New York.

Beran, R. (1981). Nonparametric regression with randomly censored survival data, *Technical report, University of California, Berkeley.* .

Borgan, . (1998). Aalen-johansen estimator, *Encyclopedia of Biostatistics* **1**: 5–10.

Burke, M. (1988). Estimation of a bivariate distribution function under random censorship, *Biometrika* **75**(2): 379–382.

Byar, D. (1980). Veterans administration study of chemoprophylaxis for recurrent stage i bladder tumors: Comparisons of placebo, pyridoxine and topical thiotepa, *Bladder Tumors and Other Topics in Urological Oncology* **18**: 363–370.

Campbell, G. (1981). Nonparametric bivariate estimation with randomly censored data, *Biometrika* **68**: 417–422.

Cao, R. and Jácome, M. (2004). Presmoothed kernel density estimator for censored data, *Journal of Nonparametric Statistics* **16**(1-2): 289–309.

Cook, R. J. and Lawless, J. F. (2007). *The Statistical Analysis of Recurrent Events*, Springer.

Cox, D. R. (1972). Regression models and life-tables, *Journal of the Royal Statistical Society. Series B* **34**(2): 187–220.

Cox, D. R. and Oakes, D. (1984). *Analysis of Survival Data*, Chapman and Hall.

Crowley, J. and Hu, M. (1977). Covariance analysis of heart transplant survival data, *Journal of the American Statistical Association* **72**(357): 27–36.

Dabrowska, D. M. (1987). Non-parametric regression with censored survival time data, *Scandinavian Journal of Statistics* **14**: 181–197.

Dabrowska, D. M. (1988). Kaplan-meier estimate on the plane, *The Annals of Statistics* **16**(4): 1475–1489.

Dabrowska, D. M. (1989a). Uniform consistency of the kernel conditional kaplan-meier estimate, *Annals of Statistics* **17**(3): 1157–1167.

Dabrowska, D. M. (1989b). Kaplan-meier estimate on the plane: Weak convergence, lil, and the bootstrap, *Journal of Multivariate Analysis* **29**(2): 308–325.

Datta, S. and Satten, G. A. (2001). Validity of the aalen-johansen estimators of stage occupation probabilities and nelson-aalen estimators of integrated transition hazards for non-markov models, *Statistics & Probability Letters* **55**: 403–411.

Davison, A. and Hinkley, D. (1997). *Bootstrap Methods and Their Application*, Cambridge University Press.

de la Peña, V. H. and Giné, E. (1999). *Decoupling: From Dependence to Independence*, Springer.

de Uña-Álvarez, J. and Amorim, A. P. (2011). A semiparametric estimator of the bivariate distribution function for censored gap times, *Biometrical Journal* **53**: 113–127.

de Uña Álvarez, J. and Campos-Rodríguez, C. (2004). Strong consistency of presmoothed kaplan-meier integrals when covariables are present, *Statistics* **38**(6): 483–496.

de Uña-Álvarez, J. and Meira-Machado, L. (2008). A simple estimator of the bivariate distribution function for censored gap times, *Statistics and Probability Letters* **78**: 2440–2445.

Devroye, L. (1978a). The uniform convergence of nearest neighbor regression function estimators and their application in optimization, *IEEE Transactions on Information Theory* **24**(2): 142–151.

Devroye, L. (1978b). The uniform convergence of the nadaraya-watson regression function estimate, *Canadian Journal of Statistics* **6**(2): 179–191.

Dikta, G. (1998). On semiparametric random censorship models, *Journal of Statistical Planning and Inference* **66**: 253–279.

Dikta, G. (2000). The strong law under semiparametric random censorship models, *Journal of Statistical Planning and Inference* **83**: 1–10.

Dikta, G., Ghorai, J. and Schmidt, C. (2005). The central limit theorem under semiparametric random censorship models, *Journal of Statistical Planning and Inference* **127**(1-2): 23–51.

Dikta, G., Kvesic, M. and C., S. (2006). Bootsrap approximations in model checks for binary data, *Journal of the American Statistical Association* **101**(474): 521–530.

Gerds, T. and Schumacher, M. (2006). Consistent estimation of the expected brier score in general survival models with right-censored event times, *Biometrical Journal* **48**(6): 1029–1040.

Hardle, W. and Luckhaus, S. (1984). Uniform consistency of a class of regression functions estimators, *The Annals of Statistics* **12**(2): 612–623.

Hosmer, D. W. and Lemeshow, S. (2008). *Applied Logistic Regression*, John Wiley & Sons.

Hougaard, P. (1999). Multi-state models: a review, *Lifetime Data Analysis* **5**: 239–264.

Hougaard, P. (2000). *Analysis of Multivariate Survival Data*, Statistics for Biology and Health, Springer-Verlag, New York.

Hui-Min, W., Ming-Fang, Y. and Chen, T. H.-H. (2004). Sas macro program for non-homogeneous markov process in modelling multi-state disease progression, *Computer Methods and Programs in Biomedicine* **75**: 95–105.

Iglesias Pérez, M. C. and González Manteiga, W. (2003). Bootstrap for the conditional distribution function with truncated and censored data, *The Annals of the Institute of Statistical Mathematics* **55**(2): 331–357.

Jackson, C. (2007). Multi-state modelling with r: The msm package, *Medical research Council Biostatistics Unit* .

Johnson, N. and Kotz, S. (1972). *Distributions in statistics: continuous multivariate distributions*, John Wiley and Sons.

Kalbfleisch, J. D. and Prentice, R. L. (2002). *The Statistical Analysis of Failure Time Data*, John Wiley & Sons.

Kaplan, E. and Meier, P. (1958). Nonparametric estimation from incomplete observations, *Journal of the American Statistical Association* **53**(282): 457–481.

Kay, R. (1986). A markov model for analysing cancer markers and disease states in survival studies, *Biometrics* **42**(4): 855–865.

Klein, J. P. and Moeschberger, M. L. (1997). *Survival Analysis Techniques for Censored and Truncated Data*, Springer.

Lin, D., Sun, W. and Ying, Z. (1999). Nonparametric estimation of the time distributions for serial events with censored data, *Biometrika* **86**(1): 59–70.

Lin, D. Y. and Ying, Z. (1993). A simple nonparametric estimator of the bivariate survival function under univariate censoring, *Biometrika* **80**(3): 573–581.

Lu, J. and Bhattacharya, G. (1990). Some new constructions of bivariate weibull models, *Annals of Institute of Statistical Mathematics* **42**(3): 543–559.

Lu, J. and Bhattacharya, G. (1991). Inference procedures for a bivariate exponential model of gumbel based on life test of component and system, *Journal of Statistical Planning and Inference* **27**: 383–396.

Lumley, T. (2004). The survival package, *R News* **4**: 26–28.

Mack, Y. P. and Silverman, B. W. (1982). Weak and strong uniform consistency of kernel regression estimates, *Probability Theory and Related Fields* **61**: 405–415.

Marshall, G., Guo, W. and Jones, R. H. (1995). Markov: A computer program for multi-state markov models with covariables, *Computer Methods and Programs in Biomedicine* **47**: 147–156.

Meira-Machado, L., Cadarso-Suárez, C., de Uña Álvarez, J. and Andersen, P. (2009). Multi-state models for the analysis of time to event data, *Statistical Methods in Medical Research* **18**(2): 195–222.

Meira-Machado, L., Cadarso-Suárez, C. and de Uña-Álvarez, J. (2007). tdc.msm: An r library for the analysis of multi-state survival data, *Computer Methods and Programs in Biomedicine* **86**: 131–140.

Meira-Machado, L., de Uña-Álvarez, J. and Cadarso-Suárez, C. (2006). Nonparametric estimation of transition probabilities in a non-markov illness-death model, *Lifetime Data Analysis* **12**(3): 325–344.

Meira-Machado, L. and Roca-Pardiñas, J. (2011). p3state.msm: Analysing survival data from an illness-death model, *Journal of Statistical Software* **38**(3): 1–18.

Miller, R. G. (1983). What price kaplan-meier?, *Biometrics* **39**: 1077–1081.

Moreira, A., de Uña-Álvarez, J. and Machado, L. (2013). Presmoothing the aalen-johansen estimator in the illness-death model, *Electronic Journal of Statistics* **7**: 1491–1516.

Moreira, A. and Meira Machado, L. (2012). survivalbiv: Estimation of the bivariate distribution function for sequentially ordered events under univariate censoring, *Journal of Statistical Software* **46**: 1–18.

Nadaraya, E. A. (1965). On non-parametric estimates of density functions and regression curves, *Theory of Applied Probability* **10**(1): 186–190.

Paes, A. T. and Lima, A. C. P. (2004). A sas macro for estimating transition probabilities in semiparametric models for recurrent events, *Computer Methods and Programs in Biomedicine* **75**: 59–65.

Prentice, R. L. and Cai, J. (1992). Covariance and survivor function estimation using censored multivariate failure time data, *Biometrika* **79**: 495–512.

Prentice, R. L., Moodie, F. Z. and Wu, J. (2004). Hazard-based nonparametric survivor function estimation, *Journal of Royal Statistical Society, B* **66**: 305–319.

Putter, H., Fiocco, M. and Geskus, R. (2007). Tutorial in biostatistics: Competing risks and multi-state models, *Statistics in Medicine* **26**: 2389–2430.

R Team, D. C. (2010). *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing. ISBN 3-900051-07-0.

Rosthøj, S., Andersen, P. K. and Abildstrom, S. Z. (2004). Sas macros for estimation of the cumulative incidence functions based on a cox regression model for competing risk survival data, *Computer Methods and Programs in Biomedicine* **74**: 69–75.

Satten, G. A. and Datta, S. (2001). The kaplan-meier estimator as an inverse-probability-of-censoring weighted average, *The American Statistician* **55**(3): 207–210.

Satten, G. A., Datta, S. and Robins, J. (2001). Estimating the marginal survival function in the presence of time dependent covariates, *Statistics & Probability Letters* **54**: 397–403.

Schulgen, G. and Schumacher, M. (1996). Estimation of prolongation of hospital stay attributable to nosocomial infections: New approaches based on multistate models, *Lifetime Data Analysis* **2**: 219–240.

Therneau, T. M. and Grambsch, P. M. (2000). *Modeling survival data: Extending the Cox Model*, Springer.

Tsai, W. Y., Leurgans, S. and Crowley, J. (1986). Nonparametric estimation of a bivariate survival function in the presence of censoring, *The Annals of Statistics* **14**(4): 1351–1365.

Van Der Laan, M. J. (1996). Efficient estimation in the bivariate censoring model and repairing npmle, *Annals of Statistics* **24**: 596–627.

Van Keilegom, I. (2004). A note on the nonparametric estimation of the bivariate distribution under dependent censoring, *Journal of Nonparametric Statistics* **16**(3-4): 659–670.

Van Keilegom, I., Akritas, M. G. and Veraverbeke, N. (2001). Estimation of the conditional distribution in regression with censored data: a comparative study, *Computational Statistics and Data Analysis* **35**(4): 487–500.

Van Keilegom, I., de Uña Álvarez, J. and Meira-Machado, L. (2011). Nonparametric location-scale models for censored successive survival times, *Journal of Statistical Planning and Inference* **141**: 1118–1131.

Wand, M. P. and Jones, M. C. (1995). *Kernel Smoothing*, Chapman & Hall.

Wang, W. and Wells, M. T. (1997). Nonparametric estimators of the bivariate survival function under simplified censoring conditions, *Biometrika* **84**(4): 863–880.

Watson, G. S. (1964). Smooth regression analysis, *Sankhya* **26**(4): 359–372.

Wrangler, M., Beyersmann, J. and Schumacher, M. (2006). Changelos: An r-package for change in length of hospital stay based on the aalen-johansen estimator, *R News* **6**: 31–35.

Yuan, M. (2005). Semiparametric censorship model with covariates, *Test* **14**(2): 489–514.

Zhang, H. (2004). Asymptotic effciency of estimation in the partial koziol-green model, *Journal of Statistical Planning and Inference* **124**(2): 399–408.

Appendix A

msmdata function

```
eprep <- function (time, status, data, tra, state.names, cens.name = NULL,
    start = NULL, id = NULL, keep, pkg)
{
        if (pkg == "etm") {
        if (nrow(tra) != ncol(tra))
            stop("'tra' must be quadratic")
        if (missing(state.names)) {
            state.names <- as.character(0:(dim(tra)[2] - 1))
        }
        ls <- length(state.names)
        n <- nrow(data)
        if (ls != dim(tra)[2])
            stop("Discrepancy between 'tra' and the number of states specified
            in 'state.names'")
        if (length(time) != ls) {
            stop("The length of 'time' must be equal to the number of states")
        }
        colnames(tra) <- rownames(tra) <- state.names
        t.from <- lapply(1:dim(tra)[2], function(i) {
```

```
        rep(rownames(tra)[i], sum(tra[i, ]))
    })
    t.from <- unlist(t.from)
    t.to <- lapply(1:dim(tra)[2], function(i) {
        colnames(tra)[tra[i, ] == TRUE]
    })
    t.to <- unlist(t.to)
    trans <- data.frame(from = t.from, to = t.to)
    absorb <- setdiff(levels(trans$to), levels(trans$from))
    transient <- unique(state.names[!(state.names %in% absorb)])
    ind <- match(time[!is.na(time)], names(data))
    if (any(is.na(ind)))
        stop("At least one element in 'time' is not in 'data'")
    indd <- which(time %in% names(data))
    time <- matrix(NA, n, ls)
    time[, indd] <- as.matrix(data[, ind])
    if (length(status) != ls) {
        stop("The length of 'status' must be equal to the number of states")
    }
    ind <- match(status[!is.na(status)], names(data))
    if (any(is.na(ind)))
        stop("At least one element in 'status' is not in 'data'")
    indd <- which(status %in% names(data))
    status <- matrix(NA, n, ls)
    status[, indd] <- as.matrix(data[, ind])
    if (is.null(start)) {
        start.state <- as.integer(rep(state.names[1], n))
        start.time <- rep(0, n)
    }
```

```
else {
    if ((start$state != nrow(data)) | (start$time !=
        nrow(data)))
        stop("'start$state' or 'start$time' are not as long as the data")
    start.state <- start$state
    start.time <- start$time
}
if (is.null(id)) {
    id <- seq_len(n)
}
else id <- data[, id]
if (!missing(keep)) {
    cova <- data[, keep, drop = FALSE]
}
else keep <- NULL
newdata <- lapply(seq_len(n), function(i) {
    ind <- which(status[i, ] != 0)
    li <- length(ind)
    if (li == 0) {
        from <- start.state[i]
        to <- cens.name
        entry <- start.time[i]
        exit <- time[i, ncol(time)]
        idd <- id[i]
    }
    else {
        from <- c(start.state[i], state.names[ind[-li]])
        to <- state.names[ind]
        entry <- c(start.time[i], time[i, ind[-li]])
        exit <- time[i, ind]
        idd <- rep(id[i], length(exit))
        if (to[length(to)] %in% transient) {
          from <- c(from, to[length(to)])
          to <- c(to, cens.name)
          entry <- c(entry, exit[length(exit)])
          exit <- c(exit, time[i, ncol(time)])
          idd <- c(idd, id[i])
        }
    }
    if (is.null(keep)) {
        tmp <- data.frame(idd, entry, exit, from, to)
    }
    else {
        aa <- matrix(apply(cova[i, , drop = FALSE], 2,
          rep, length(exit)), length(exit), ncol(cova))
        tmp <- data.frame(idd, entry, exit, from, to,
          aa)
    }
    tmp
})
newdata <- do.call(rbind, newdata)
names(newdata) <- c("id", "entry", "exit", "from", "to",
    keep)
if (is.factor(newdata$from) || is.factor(newdata$to)) {
    aa <- unique(c(levels(newdata$from), levels(newdata$to)))
    newdata$from <- factor(as.character(newdata$from),
        levels = aa)
    newdata$to <- factor(as.character(newdata$to), levels = aa)
    }
    return(newdata)
}
if (pkg == "los") {
    if (nrow(tra) != ncol(tra))
        stop("'tra' must be quadratic")
    if (missing(state.names)) {
        state.names <- as.character(0:(dim(tra)[2] - 1))
    }
    ls <- length(state.names)
    n <- nrow(data)
    if (ls != dim(tra)[2])
        stop("Discrepancy between 'tra' and the number of states specified
        in 'state.names'")
    if (length(time) != ls) {
        stop("The length of 'time' must be equal to the number of states")
    }
    colnames(tra) <- rownames(tra) <- state.names
    t.from <- lapply(1:dim(tra)[2], function(i) {
        rep(rownames(tra)[i], sum(tra[i, ]))
    })
    t.from <- unlist(t.from)
    t.to <- lapply(1:dim(tra)[2], function(i) {
        colnames(tra)[tra[i, ] == TRUE]
    })
    t.to <- unlist(t.to)
    trans <- data.frame(from = t.from, to = t.to)
    absorb <- setdiff(levels(trans$to), levels(trans$from))
    transient <- unique(state.names[!(state.names %in% absorb)])
```

```
ind <- match(time[!is.na(time)], names(data))
if (any(is.na(ind)))
    stop("At least one element in 'time' is not in 'data'")
indd <- which(time %in% names(data))
time <- matrix(NA, n, ls)
time[, indd] <- as.matrix(data[, ind])
if (length(status) != ls) {
    stop("The length of 'status' must be equal to the number of states")
}
ind <- match(status[!is.na(status)], names(data))
if (any(is.na(ind)))
    stop("At least one element in 'status' is not in 'data'")
indd <- which(status %in% names(data))
status <- matrix(NA, n, ls)
status[, indd] <- as.matrix(data[, ind])
if (is.null(start)) {
    start.state <- as.integer(rep(state.names[1], n))
    start.time <- rep(0, n)
}
else {
    if ((start$state != nrow(data)) | (start$time !=
        nrow(data)))
        stop("'start$state' or 'start$time' are not as long as the data")
    start.state <- start$state
    start.time <- start$time
}
if (is.null(id)) {
    id <- seq_len(n)
}
```

```
else id <- data[, id]
if (!missing(keep)) {
    cova <- data[, keep, drop = FALSE]
}
else keep <- NULL
newdata <- lapply(seq_len(n), function(i) {
    ind <- which(status[i, ] != 0)
    li <- length(ind)
    if (li == 0) {
        from <- start.state[i]
        to <- cens.name
        entry <- start.time[i]
        exit <- time[i, ncol(time)]
        idd <- id[i]
    }
    else {
        from <- c(start.state[i], state.names[ind[-li]])
        to <- state.names[ind]
        entry <- c(start.time[i], time[i, ind[-li]])
        exit <- time[i, ind]
        idd <- rep(id[i], length(exit))
        if (to[length(to)] %in% transient) {
          from <- c(from, to[length(to)])
          to <- c(to, cens.name)
          entry <- c(entry, exit[length(exit)])
          exit <- c(exit, time[i, ncol(time)])
          idd <- c(idd, id[i])
        }
    }
```

```
            if (is.null(keep)) {
                tmp <- data.frame(idd, from, to, exit, idd)
            }
            else {
                aa <- matrix(apply(cova[i, , drop = FALSE], 2,
                  rep, length(exit)), length(exit), ncol(cova))
                tmp <- data.frame(idd, from, to, exit, idd)
            }
            tmp
        })
        newdata <- do.call(rbind, newdata)
        names(newdata) <- c("id", "from", "to", "time", "oid")
        if (is.factor(newdata$from) || is.factor(newdata$to)) {
            aa <- unique(c(levels(newdata$from), levels(newdata$to)))
            newdata$from <- factor(as.character(newdata$from),
                levels = aa)
            newdata$to <- factor(as.character(newdata$to), levels = aa)
        }
        return(newdata)
    }
    if (pkg == "mvna") {
        if (nrow(tra) != ncol(tra))
            stop("'tra' must be quadratic")
        if (missing(state.names)) {
            state.names <- as.character(0:(dim(tra)[2] - 1))
        }
        ls <- length(state.names)
        n <- nrow(data)
        if (ls != dim(tra)[2])
    stop("Discrepancy between 'tra' and the number of states specified
    in 'state.names'")
if (length(time) != ls) {
    stop("The length of 'time' must be equal to the number of states")
}
colnames(tra) <- rownames(tra) <- state.names
t.from <- lapply(1:dim(tra)[2], function(i) {
    rep(rownames(tra)[i], sum(tra[i, ]))
})
t.from <- unlist(t.from)
t.to <- lapply(1:dim(tra)[2], function(i) {
    colnames(tra)[tra[i, ] == TRUE]
})
t.to <- unlist(t.to)
trans <- data.frame(from = t.from, to = t.to)
absorb <- setdiff(levels(trans$to), levels(trans$from))
transient <- unique(state.names[!(state.names %in% absorb)])
ind <- match(time[!is.na(time)], names(data))
if (any(is.na(ind)))
    stop("At least one element in 'time' is not in 'data'")
indd <- which(time %in% names(data))
time <- matrix(NA, n, ls)
time[, indd] <- as.matrix(data[, ind])
if (length(status) != ls) {
    stop("The length of 'status' must be equal to the number of states")
}
ind <- match(status[!is.na(status)], names(data))
if (any(is.na(ind)))
    stop("At least one element in 'status' is not in 'data'")
    indd <- which(status %in% names(data))
    status <- matrix(NA, n, ls)
    status[, indd] <- as.matrix(data[, ind])
    if (is.null(start)) {
        start.state <- as.integer(rep(state.names[1], n))
        start.time <- rep(0, n)
    }
    else {
        if ((start$state != nrow(data)) | (start$time !=
            nrow(data)))
            stop("'start$state' or 'start$time' are not as long as the data")
        start.state <- start$state
        start.time <- start$time
    }
    if (is.null(id)) {
        id <- seq_len(n)
    }
    else id <- data[, id]
    if (!missing(keep)) {
        cova <- data[, keep, drop = FALSE]
    }
    else keep <- NULL
    newdata <- lapply(seq_len(n), function(i) {
        ind <- which(status[i, ] != 0)
        li <- length(ind)
        if (li == 0) {
            from <- start.state[i]
            to <- cens.name
            entry <- start.time[i]
        exit <- time[i, ncol(time)]
        idd <- id[i]
    }
    else {
        from <- c(start.state[i], state.names[ind[-li]])
        to <- state.names[ind]
        entry <- c(start.time[i], time[i, ind[-li]])
        exit <- time[i, ind]
        idd <- rep(id[i], length(exit))
        if (to[length(to)] %in% transient) {
          from <- c(from, to[length(to)])
          to <- c(to, cens.name)
          entry <- c(entry, exit[length(exit)])
          exit <- c(exit, time[i, ncol(time)])
          idd <- c(idd, id[i])
        }
    }
    if (is.null(keep)) {
        tmp <- data.frame(idd, from, to, exit)
    }
    else {
        aa <- matrix(apply(cova[i, , drop = FALSE], 2,
          rep, length(exit)), length(exit), ncol(cova))
        tmp <- data.frame(idd, from, to, exit, aa)
    }
    tmp
})
newdata <- do.call(rbind, newdata)
names(newdata) <- c("id", "from", "to", "time", keep)
```

```
        if (is.factor(newdata$from) || is.factor(newdata$to)) {
            aa <- unique(c(levels(newdata$from), levels(newdata$to)))
            newdata$from <- factor(as.character(newdata$from),
                levels = aa)
            newdata$to <- factor(as.character(newdata$to), levels = aa)
        }
        return(newdata)
    }
}


msmdata <- function (data, pkg, tra = NULL, state.names = NULL,
cens.name = NULL)
{
    if (missing(data))
        stop("Argument 'data' is missing with no default")
    if (missing(pkg))
        stop("Argument 'package' is missing with no default")
    if (!is.data.frame(data))
        stop("Argument 'data' must be a data.frame")
    if (any(names(data)[1:4] != c("time1", "event1", "Stime", "event")))
        stop("'data' must contain the right variables")
    if (any(data[, 2] != 0 & data[, 2] != 1))
        stop("The variable 'delta' in the argument 'data' must be 0 or 1")
    if (any(data[, 4] != 0 & data[, 4] != 1))
        stop("The variable 'status' in the argument 'data' must be 0 or 1")
    if (any(data[, 2] == 0 & data[, 3]-data[, 1] > 0))
        stop("The variable 'Stime' in the argument 'data' must be equal to 0
        when 'event1 = 0'")
```

```
if (any(data[, c(1, 3)] < 0))
    stop("The time variables in 'data' must be non negative")

nevent <- rep(0, dim(data)[1])
p<-which(data$time1<data$Stime)
nevent[p]<-1
data2<-cbind(data$time1, nevent, data$Stime-data$time1,
data$Stime, data$event)
if (dim(data)[2]>4) data2<-cbind(data2,data[5:dim(data)[2]])
data2<-data.frame(data2)
names(data2)[1:5]<-c("times1", "delta", "times2", "time", "status")


if (dim(data)[2]>4) names(data2)[6:(dim(data)[2]+1)]<-names(data
[5:dim(data)[2]]
data<-data2



if (pkg == "mstate") {
    if (dim(data)[2]<=5) stop("There aren't covariates in 'data'")
    lines <- nrow(data) + sum(data[, 2] == 1) + sum(data[,2] == 1) +
    sum(data[, 2] == 0)
    require(mstate)
    covs <- c(names(data)[6:(ncol(data))])
    colon.mstate <- msprep(time = c(NA, names(data)[1], names(data)[4]),
        status = c(NA, names(data)[2], names(data)[5]), data = data,
        trans = trans, keep = covs)
    return(colon.mstate)
}
```

```
if (pkg == "etm") {
    if (dim(data)[2]<=5) coxdata <- eprep(time = c(NA, names(data)[1],
    names(data)[4]),
        status = c(NA, names(data)[2], names(data)[5]), data = data,
        tra = tra, cens.name = cens.name, state.names = state.names,
        keep = NULL, pkg = "etm")



    else coxdata <- eprep(time = c(NA, names(data)[1], names(data)[4]),
        status = c(NA, names(data)[2], names(data)[5]), data = data,
        tra = tra, cens.name = cens.name, state.names = state.names,
        keep = c(names(data)[6:(ncol(data))]), pkg = "etm")
    return(coxdata)
}
if (pkg == "los") {
    coxdata <- eprep(time = c(NA, names(data)[1], names(data)[4]),
        status = c(NA, names(data)[2], names(data)[5]), data = data,
        tra = tra, cens.name = cens.name, pkg = "los")
    return(coxdata)
}
if (pkg == "mvna") {
    if (dim(data)[2]<=5)   coxdata <- eprep(time = c(NA, names(data)[1],
    names(data)[4]),
        status = c(NA, names(data)[2], names(data)[5]), data = data,
        tra = tra, cens.name = cens.name, state.names = state.names,
        keep = NULL, pkg = "mvna")

    else coxdata <- eprep(time = c(NA, names(data)[1], names(data)[4]),
        status = c(NA, names(data)[2], names(data)[5]), data = data,
            tra = tra, cens.name = cens.name, state.names = state.names,
            keep = c(names(data)[6:(ncol(data))]), pkg = "mvna")
        return(coxdata)
    }
    if (pkg == "tdcm") {
        lines <- nrow(data) + sum(data[, 2] == 1)
        coxdata <- matrix(data = NA, ncol = (ncol(data)), nrow = lines)
        q1 <- 5
        q2 <- ncol(coxdata)
        q3 <- q2 - q1
        p <- 0
        for (k in 1:nrow(data)) {
            if (data[k, 2] == 0 & data[k, 5] == 1) {
                coxdata[k + p, 1] <- k
                coxdata[k + p, 2] <- 0
                coxdata[k + p, 3] <- data[k, 1]
                coxdata[k + p, 4] <- 1
                coxdata[k + p, 5] <- 0
                for (j in 1:q3) coxdata[k + p, 5 + j] <- data[k,
                  5 + j]
            }
            if (data[k, 2] == 0 & data[k, 5] == 0) {
                coxdata[k + p, 1] <- k
                coxdata[k + p, 2] <- 0
                coxdata[k + p, 3] <- data[k, 1]
                coxdata[k + p, 4] <- 0
                coxdata[k + p, 5] <- 0
                for (j in 1:q3) coxdata[k + p, 5 + j] <- data[k,
                  5 + j]
```

```
}
if (data[k, 2] == 1 & data[k, 5] == 0) {
    coxdata[k + p, 1] <- k
    coxdata[k + p, 2] <- 0
    coxdata[k + p, 3] <- data[k, 1]
    coxdata[k + p, 4] <- 0
    coxdata[k + p, 5] <- 0
    for (j in 1:q3) coxdata[k + p, 5 + j] <- data[k,
      5 + j]
    p <- p + 1
    coxdata[k + p, 1] <- k
    coxdata[k + p, 2] <- data[k, 1]
    coxdata[k + p, 3] <- data[k, 4]
    coxdata[k + p, 4] <- 0
    coxdata[k + p, 5] <- 1
    for (j in 1:q3) coxdata[k + p, 5 + j] <- data[k,
      5 + j]
}
if (data[k, 2] == 1 & data[k, 5] == 1) {
    coxdata[k + p, 1] <- k
    coxdata[k + p, 2] <- 0
    coxdata[k + p, 3] <- data[k, 1]
    coxdata[k + p, 4] <- 0
    coxdata[k + p, 5] <- 0
    for (j in 1:q3) coxdata[k + p, 5 + j] <- data[k,
      5 + j]
    p <- p + 1
    coxdata[k + p, 1] <- k
    coxdata[k + p, 2] <- data[k, 1]
```

```
                coxdata[k + p, 3] <- data[k, 4]
                coxdata[k + p, 4] <- 1
                coxdata[k + p, 5] <- 1
                for (j in 1:q3) coxdata[k + p, 5 + j] <- data[k,
                  5 + j]
            }
        }
        nomes2 <- c("id", "start", "stop", "event", "tdcov")
        coxdata <- data.frame(coxdata)
        names(coxdata) <- c(nomes2, names(data)[6:(ncol(data))])
        return(coxdata)
    }
    if (pkg == "msm") {
        lines <- nrow(data) + sum(data[, 2] == 1) + sum(data[,
            2] == 1) + sum(data[, 2] == 0)
        coxdata <- matrix(data = NA, ncol = ncol(data), nrow = lines)
        q1 <- 5
        q2 <- ncol(coxdata)
        q3 <- q2 - q1
        p <- 0
        for (k in 1:nrow(data)) {
            if (data[k, 2] == 0 & data[k, 5] == 1) {
                coxdata[k + p, 1] <- k
                coxdata[k + p, 2] <- 0
                coxdata[k + p, 3] <- 1
                for (j in 1:q3) coxdata[k + p, 3 + j] <- data[k,
                  5 + j]
                p <- p + 1
                coxdata[k + p, 1] <- k
            coxdata[k + p, 2] <- data[k, 1]
            coxdata[k + p, 3] <- 3
            for (j in 1:q3) coxdata[k + p, 3 + j] <- data[k,
              5 + j]
        }
        if (data[k, 2] == 0 & data[k, 5] == 0) {
            coxdata[k + p, 1] <- k
            coxdata[k + p, 2] <- 0
            coxdata[k + p, 3] <- 1
            for (j in 1:q3) coxdata[k + p, 3 + j] <- data[k,
              5 + j]
            p <- p + 1
            coxdata[k + p, 1] <- k
            coxdata[k + p, 2] <- data[k, 1]
            coxdata[k + p, 3] <- 1
            for (j in 1:q3) coxdata[k + p, 3 + j] <- data[k,
              5 + j]
        }
        if (data[k, 2] == 1 & data[k, 5] == 0) {
            coxdata[k + p, 1] <- k
            coxdata[k + p, 2] <- 0
            coxdata[k + p, 3] <- 1
            for (j in 1:q3) coxdata[k + p, 3 + j] <- data[k,
              5 + j]
            p <- p + 1
            coxdata[k + p, 1] <- k
            coxdata[k + p, 2] <- data[k, 1]
            coxdata[k + p, 3] <- 2
            for (j in 1:q3) coxdata[k + p, 3 + j] <- data[k,
          5 + j]
        p <- p + 1
        coxdata[k + p, 1] <- k
        coxdata[k + p, 2] <- data[k, 4]
        coxdata[k + p, 3] <- 2
        for (j in 1:q3) coxdata[k + p, 3 + j] <- data[k,
          5 + j]
    }
    if (data[k, 2] == 1 & data[k, 5] == 1) {
        coxdata[k + p, 1] <- k
        coxdata[k + p, 2] <- 0
        coxdata[k + p, 3] <- 1
        for (j in 1:q3) coxdata[k + p, 3 + j] <- data[k,
          5 + j]
        p <- p + 1
        coxdata[k + p, 1] <- k
        coxdata[k + p, 2] <- data[k, 1]
        coxdata[k + p, 3] <- 2
        for (j in 1:q3) coxdata[k + p, 3 + j] <- data[k,
          5 + j]
        p <- p + 1
        coxdata[k + p, 1] <- k
        coxdata[k + p, 2] <- data[k, 4]
        coxdata[k + p, 3] <- 3
        for (j in 1:q3) coxdata[k + p, 3 + j] <- data[k,
          5 + j]
    }
}
nomes2 <- c("ptnum", "dtime", "state")
```

```
        coxdata <- data.frame(coxdata)
        names(coxdata) <- c(nomes2, names(data)[6:(ncol(data))])
        msm <- coxdata[, 1:(ncol(coxdata) - 2)]
        return(msm)
    }
    if (pkg == "cmm") {
        lines <- nrow(data) + sum(data[, 2] == 1) + sum(data[,
            2] == 1) + sum(data[, 2] == 0)
        coxdata <- matrix(data = NA, ncol = ncol(data) + 1, nrow = lines)
        q1 <- 6
        q2 <- ncol(coxdata)
        q3 <- q2 - q1
        p <- 0
        for (k in 1:nrow(data)) {
            if (data[k, 2] == 0 & data[k, 5] == 1) {
                coxdata[k + p, 1] <- k
                coxdata[k + p, 2] <- 0
                coxdata[k + p, 3] <- data[k, 1]
                coxdata[k + p, 4] <- 1
                coxdata[k + p, 5] <- 0
                coxdata[k + p, 6] <- 1
                for (j in 1:q3) coxdata[k + p, 6 + j] <- data[k,
                  5 + j]
                p <- p + 1
                coxdata[k + p, 1] <- k
                coxdata[k + p, 2] <- 0
                coxdata[k + p, 3] <- data[k, 1]
                coxdata[k + p, 4] <- 0
                coxdata[k + p, 5] <- 0
        coxdata[k + p, 6] <- 2
        for (j in 1:q3) coxdata[k + p, 6 + j] <- data[k,
          5 + j]
    }
    if (data[k, 2] == 0 & data[k, 5] == 0) {
        coxdata[k + p, 1] <- k
        coxdata[k + p, 2] <- 0
        coxdata[k + p, 3] <- data[k, 1]
        coxdata[k + p, 4] <- 0
        coxdata[k + p, 5] <- 0
        coxdata[k + p, 6] <- 1
        for (j in 1:q3) coxdata[k + p, 6 + j] <- data[k,
          5 + j]
        p <- p + 1
        coxdata[k + p, 1] <- k
        coxdata[k + p, 2] <- 0
        coxdata[k + p, 3] <- data[k, 1]
        coxdata[k + p, 4] <- 0
        coxdata[k + p, 5] <- 0
        coxdata[k + p, 6] <- 2
        for (j in 1:q3) coxdata[k + p, 6 + j] <- data[k,
          5 + j]
    }
    if (data[k, 2] == 1 & data[k, 5] == 0) {
        coxdata[k + p, 1] <- k
        coxdata[k + p, 2] <- 0
        coxdata[k + p, 3] <- data[k, 1]
        coxdata[k + p, 4] <- 0
        coxdata[k + p, 5] <- 0
    coxdata[k + p, 6] <- 1
    for (j in 1:q3) coxdata[k + p, 6 + j] <- data[k,
      5 + j]
    p <- p + 1
    coxdata[k + p, 1] <- k
    coxdata[k + p, 2] <- 0
    coxdata[k + p, 3] <- data[k, 1]
    coxdata[k + p, 4] <- 1
    coxdata[k + p, 5] <- 0
    coxdata[k + p, 6] <- 2
    for (j in 1:q3) coxdata[k + p, 6 + j] <- data[k,
      5 + j]
    p <- p + 1
    coxdata[k + p, 1] <- k
    coxdata[k + p, 2] <- data[k, 1]
    coxdata[k + p, 3] <- data[k, 4]
    coxdata[k + p, 4] <- 0
    coxdata[k + p, 5] <- 1
    coxdata[k + p, 6] <- 3
    for (j in 1:q3) coxdata[k + p, 6 + j] <- data[k,
      5 + j]
}
if (data[k, 2] == 1 & data[k, 5] == 1) {
    coxdata[k + p, 1] <- k
    coxdata[k + p, 2] <- 0
    coxdata[k + p, 3] <- data[k, 1]
    coxdata[k + p, 4] <- 0
    coxdata[k + p, 5] <- 0
    coxdata[k + p, 6] <- 1
```

```
                for (j in 1:q3) coxdata[k + p, 6 + j] <- data[k,
                  5 + j]
                p <- p + 1
                coxdata[k + p, 1] <- k
                coxdata[k + p, 2] <- 0
                coxdata[k + p, 3] <- data[k, 1]
                coxdata[k + p, 4] <- 1
                coxdata[k + p, 5] <- 0
                coxdata[k + p, 6] <- 2
                for (j in 1:q3) coxdata[k + p, 6 + j] <- data[k,
                  5 + j]
                p <- p + 1
                coxdata[k + p, 1] <- k
                coxdata[k + p, 2] <- data[k, 1]
                coxdata[k + p, 3] <- data[k, 4]
                coxdata[k + p, 4] <- 1
                coxdata[k + p, 5] <- 1
                coxdata[k + p, 6] <- 3
                for (j in 1:q3) coxdata[k + p, 6 + j] <- data[k,
                  5 + j]
            }
        }
        nomes2 <- c("id", "start", "stop", "event", "tdcov",
            "transition")
        coxdata <- data.frame(coxdata)
        names(coxdata) <- c(nomes2, names(data)[6:(ncol(data))])
        return(coxdata)
    }
}
```