Use R!

Dirk F. Moore

Applied Survival Analysis Using R

Springer

Use R!

More information about this series at http://www.springer.com/series/6991

Use R!

Wickham: ggplot2 (2nd ed. 2016)
Luke: A User's Guide to Network Analysis in R
Monogan: Political Analysis Using R
Cano/M. Moguerza/Prieto Corcoba: Quality Control with R
Schwarzer/Carpenter/Rücker: Meta-Analysis with R
Gondro: Primer to Analysis of Genomic Data Using R
Chapman/Feit: R for Marketing Research and Analytics
Willekens: Multistate Analysis of Life Histories with R
Cortez: Modern Optimization with R
Kolaczyk/Csàrdi: Statistical Analysis of Network Data with R
Swenson/Nathan: Functional and Phylogenetic Ecology in R
Nolan/Temple Lang: XML and Web Technologies for Data Sciences with R
Nagarajan/Scutari/Lèbre: Bayesian Networks in R
van den Boogaart/Tolosana-Delgado: Analyzing Compositional Data with R
Bivand/Pebesma/Gòmez-Rubio: Applied Spatial Data Analysis with R (2nd ed. 2013)
Eddelbuettel: Seamless R and C++ Integration with Rcpp
Knoblauch/Maloney: Modeling Psychophysical Data in R
Lin/Shkedy/Yekutieli/Amaratunga/Bijnens: Modeling Dose-Response Microarray Data in Early Drug Development Experiments Using R
Cano/M. Moguerza/Redchuk: Six Sigma with R
Soetaert/Cash/Mazzia: Solving Differential Equations in R

Dirk F. Moore

Applied Survival Analysis Using R

Springer

Dirk F. Moore
Department of Biostatistics
Rutgers School of Public Health
Piscataway, NJ, USA

ISSN 2197-5736 ISSN 2197-5744 (electronic)
Use R!
ISBN 978-3-319-31243-9 ISBN 978-3-319-31245-3 (eBook)
DOI 10.1007/978-3-319-31245-3

Library of Congress Control Number: 2016940055

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG Switzerland

To Lynne, Molly, and Emily

Preface

This book serves as an introductory guide for students and analysts who need to work with survival time data. The minimum prerequisites are basic applied courses in linear regression and categorical data analysis. Students who also have taken a master's level course in statistical theory will be well prepared to work through this book, since frequent reference is made to maximum likelihood theory. Students lacking this training may still be able to understand most of the material, provided they have an understanding of the basic concepts of differential and integral calculus. Specifically, students should understand the concept of the limit, and they should know what derivatives and integrals are and be able to evaluate them in some basic cases.

The material for this book has come from two sources. The first source is an introductory class in survival analysis for graduate students in epidemiology and biostatistics at the Rutgers School of Public Health. Biostatistics students, as one would expect, have a much firmer grasp of more mathematical aspects of statistics than do epidemiology students. Still, I have found that those epidemiology students with strong quantitative backgrounds have been able to understand some mathematical statistical procedures such as score and likelihood ratio tests, provided that they are not expected to symbolically differentiate or integrate complex formulas. In this book I have, when possible, used the numerical capabilities of the R system to substitute for symbolic manipulation. The second source of material is derived from collaborations with physicians and epidemiologists at the Rutgers Cancer Institute of New Jersey and at the Rutgers Robert Wood Johnson Medical School. A number of the data sets in this text are derived from these collaborations. Also, the experience of training statistical analysts to work on these data sets provided additional inspiration for the book.

The first chapter introduces the concepts of survival times and how right censoring occurs and describes several of the datasets that will be used throughout the book. Chapter 2 presents fundamentals of survival theory. This includes hazard, probability density, survival functions, and how they are related. The hazard function is illustrated using both life table data and using some common parametric distributions. The chapter ends with a brief introduction to properties of maximum likelihood estimates using the exponential distribution as an illustration. Chapter 3 discusses the Kaplan-Meier estimate of the survival function and several related concepts such as the median survival and its confidence interval. Also discussed in this chapter are smoothing of the hazard function and how to accommodate left truncation into the Kaplan-Meier estimate.

Chapter 4 discusses the log-rank test for comparing survival distributions and also some modified linear rank tests. Stratified tests are also discussed, along with an example where stratification can reverse the apparent direction of a treatment effect in a survival example of Simpson's paradox. In Chapter 5, we present the Cox proportional hazards model and partial likelihood function in the context of comparing two groups of survival data. There we illustrate the Wald, score, and likelihood ratio tests in this basic context. Left-truncated survival data and the partial likelihood are also discussed.

Chapter 6 presents methods for model selection and extends and illustrates the proportional hazards model in situations where there are multiple possible predictor covariates. Chapter 7 presents diagnostic residual plots that are useful for assessing model assumptions. Chapter 8 discusses how to adapt the survival models discussed earlier to allow for time-dependent covariates.

The next few chapters discuss some important special situations. Chapter 9 discusses multiple outcomes, which can occur as clustered survival times or in a competing risks framework, where only the first of multiple outcomes can be observed. Chapter 10 discusses parametric survival models, and Chapter 11 covers the critically important design question of how to determine the power and sample size of a proposed study that has a survival outcome. Finally, Chapter 12 presents some additional topics, including the piecewise exponential distribution, methods for handling interval censoring, and the lasso method for handling survival data with large numbers of predictors. Many of the data sets discussed in the text are available in the accompanying R package "asaur" (for "Applied Survival Analysis Using R"), while others are in other packages. All are freely available for download from the Central R Archive Network at cran.r-project.org. The R-code discussed in the book is available for download at http://www.springer.com/us/book/9783319312439

A key feature of this book is the integration of the R statistical system with the survival analysis material. Not only do we show the reader how to use R functions to fit survival models and how to interpret the results, but we also use R to illustrate how survival quantities are computed. Typically we use small examples to illustrate in detail how one constructs survival tests, partial likelihood models, and diagnostics and then proceed to more complicated examples. Most of the survival functions will require that the "survival" library be attached using the "library(survival)" statement. The "survival" package is included by default; other packages referred to in the text must be explicitly downloaded and installed. The appendix includes both some basics of the R language and special features relevant to the survival calculations used elsewhere in the book. Users not already familiar with the R system should refer to one of the many online resources for more detailed information.

I would like to thank Rebecca Moss for permission to use the “pancreatic” data and Michael Steinberg for permission to use the “pharmacoSmoking” data. Both of these data sets are used repeatedly throughout the text. I would also like to thank Grace Lu-Yao, Weichung Joe Shih, and Yong Lin for years-long collaborations on using the SEER-Medicare data for studying the survival trajectories of prostate cancer patients. These collaborations led to the development of the “prostateSurvival” data set discussed in this text in Chapter 9. I thank the Division of Cancer Epidemiology and Genetics of the US National Cancer Institute for providing the “asheknazi” data. I also thank Wan Yee Lau for making the “hepatoCellular” data publically available in the online Dryad data repository and for allowing me to include it in the “asaur” R package.

Piscataway, NJ, USA Dirk F. Moore
October 2015

Contents

Chapter 1
Introduction

1.1 What Is Survival Analysis?

Survival analysis is the study of survival times and of the factors that influence them. Types of studies with survival outcomes include clinical trials, prospective and retrospective observational studies, and animal experiments. Examples of survival times include time from birth until death, time from entry into a clinical trial until death or disease progression, or time from birth to development of breast cancer (that is, age of onset). The survival endpoint can also refer a positive event. For example, one might be interested in the time from entry into a clinical trial until tumor response. Survival studies can involve estimation of the survival distribution, comparisons of the survival distributions of various treatments or interventions, or elucidation of the factors that influence survival times. As we shall see, many of the techniques we study have analogues in generalized linear models such as linear or logistic regression.

Survival analysis is a difficult subject, and a full exposition of its principles would require readers to have a background not only in basic statistical theory but also in advanced topics in the theory of probability. Fortunately, many of the most important concepts in survival analysis can be presented at a more elementary level. The aim of this book is to provide the reader with an understanding of these principles and also to serve as a guide to implementing these ideas in a practical setting. We shall use the R statistical system extensively throughout the book because (1) it is a high-quality system for doing statistics, (2) it includes a wealth of enhancements and packages for doing survival analysis, (3) its interactive design will allow us to illustrate survival concepts, and (4) it is an open source package available for download to anyone at no cost from the main R website, www.R-project.org. This book is meant to be *used* as well as read, and the reader is encouraged to use R to try out the examples discussed in the text and to do the exercises at the end of each chapter. It is expected that readers are already familiar with the R language; for

D. Moore, *Applied Survival Analysis Using R*, Use R!,
DOI 10.1007/978-3-319-31245-3_1

those who are not, an overview of R may be found in the appendix, and links to more extensive R guides and manuals may be found on the main R website. Readers who master the techniques in this book will be equipped to use R to carry out survival analyses in a practical setting, and those who are familiar with one of the many excellent commercial statistical packages should be able to adapt what they have learned to the particular command syntax and output style of that package.

1.2 What You Need to Know to Use This Book

Survival analysis resembles linear and logistic regression analysis in several ways: there is (typically) a single outcome variable and one or more predictors; testing statistical hypotheses about the relationship of the predictors to the outcome variable is of particular interest; adjusting for confounding covariates is crucial; and model selection and checking of assumptions through analysis of residuals and other methods are key requirements. Thus, readers should be familiar with basic concepts of classical hypothesis testing and with principles of regression analysis. Familiarity with categorical data analysis methods, including contingency tables, stratified contingency tables, and Poisson and logistic regression, are also important. However, survival analysis differs from these classical statistical methods in that censoring plays a central role in nearly all cases, and the theoretical underpinnings of the subject are far more complex. While I have strived to keep the mathematical level of this book as assessable as possible, many concepts in survival analysis depend on some understanding of mathematical statistics. Readers at a minimum must understand key ideas from calculus such as limits and the meaning of derivatives and integrals; the definition of the hazard function, for example, underlies everything we will do, and its definition depends on limits. And its connection to the survival function depends on an integral. Those who are already familiar with basic concepts of likelihood theory at the level of a Masters program in statistics or biostatistics will have the easiest time working through this book. For those who are less familiar with these topics I have endeavored to use the numerical capabilities of R to illustrate likelihood principles as they arise. Also, as already mentioned, the reader is expected to be familiar with the basics of using the R system, including such concepts as vectors, matrices, data structures and components, and data frames. He or she should also be sufficiently familiar with R to carry out basic data analyses and make data plots, as well as understand how to install in R packages from the main CRAN (Comprehensive R Archive Network) repository.

1.3 Survival Data and Censoring

A key characteristic of survival data is that the response variable is a non-negative discrete or continuous random variable, and represents the time from a well-defined origin to a well-defined event. A second characteristic of survival analysis, censoring, arises when the starting or ending events are not precisely observed. The most common example of this is right censoring, which results when the final endpoint is only known to exceed a particular value. Formally, if T^* is a random variable representing the time to failure and U is a random variable representing the time to a censoring event, what we observe is $T = min(T^*, U)$ and a censoring indicator $\delta = I[T^* < U]$. That is, δ is 0 or 1 according to whether T is a censored time or an observed failure time. Less commonly one may have left censoring, where events are known to have occurred *before* a certain time, or interval censoring, where the failure time is only known to have occurred within a specified interval of time. For now we will address the more prevalent right-censoring situation.

Censoring may be classified into three types: Type I, Type II, or random. In Type I censoring, the censoring times are pre-specified. For example, in an animal experiment, a cohort of animals may start at a specific time, and all followed until a pre-specified ending time. Animals which have not experienced the event of interest before the end of the study are then censored at that time. Another example, discussed in detail in Example 1.5, is a smoking cessation study, where by design each subject is followed until relapse (return to smoking) or 180 days, whichever comes first. Those subjects who did not relapse within the 180 day period were censored at that time.

Type II censoring occurs when the experimental objects are followed until a pre-specified fraction have failed. Such a design is rare in biomedical studies, but may be used in industrial settings, where time to failure of a device is of primary interest. An example would be one where the study stops after, for instance, 25 out of 100 devices are observed to fail. The remaining 75 devices would then be censored. In this example, the smallest 25 % of the ordered failure times are observed, and the remainder are censored.

The last general category of censoring is *random* censoring. Careful attention to the cause of the censoring is essential in order to avoid biased survival estimates. In biomedical settings, one cause of random censoring is patient dropout. If the dropout occurs truly at random, and is unrelated to the disease process, such censoring may not cause any problems with bias in the analysis. But if patients who are near death are more likely to drop out than other patients, serious biases may arise. Another cause of random censoring is competing events. For instance, in Example 1.4, the primary outcome is time to death from prostate cancer. But when a patient dies of another cause first, then that patient will be censored, since the time he would have died of prostate cancer (had he not died first of the other cause) is unknown. The question of independence of the competing causes is, of course, an important issue, and will be discussed in Sect. 9.2.

In clinical trials, the most common source of random censoring is *administrative* censoring, which results because some patients in a clinical trial have not yet died at the time the analysis is carried out. This concept is illustrated in the following example.

Example 1.1. Consider a hypothetical cancer clinical trial where subjects enter the trial over a certain period of time, known as the accrual period, and are followed for an additional period of time, known as the follow-up period, to determine their survival times. That is, for each patient, we would like to observe the time between when a patient entered the trial and when that patient died. But unless the type of cancer being studied is quickly fatal, some patients will still be alive at the end of the follow-up time, and indeed many patients may survive long after this time. For these patients, the survival times are only partially observed; we know that these patients survived until the end of follow-up, but we don't know how much longer they will survive. Such times are said to be right-censored, and this type of censoring is both the most common and the most easily accommodated. Other types of censoring, as we have seen, include left and interval censoring. We will discuss these briefly in the last chapter.

Figure 1.1 presents data from a hypothetical clinical trial. Here, five patients were entered over a 2.5-year accrual period which ran from January 1, 2000 until June 30, 2002. This was followed by 4.5 years of additional follow-up time, which lasted until December 31, 2007. In this example, the data were meant to be analyzed at this time, but three patients (Patients 1, 3 and 4) were still alive. Also shown in this example is the ultimate fate of these three patients, but this would not have been known at the time of analysis. Thus, for these three patients, we have incomplete information about their survival time. For example, we know that Patient 1 survived at least 7 years, but as of the end of 2007 it would not have been known how long the patient would ultimately live.

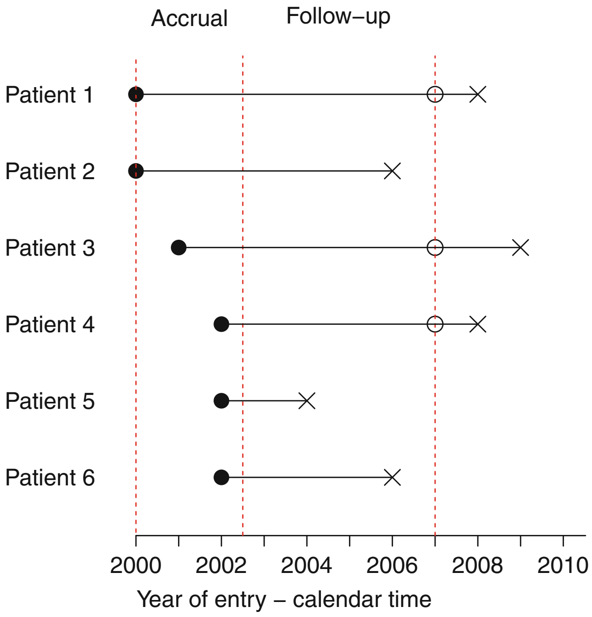


Fig. 1.1 Clinical trial accrual and follow-up periods. The *vertical dashed lines* indicate the trial start, end of accrual, and end of follow-up. The X's denote deaths and the *open circles* denote censoring events

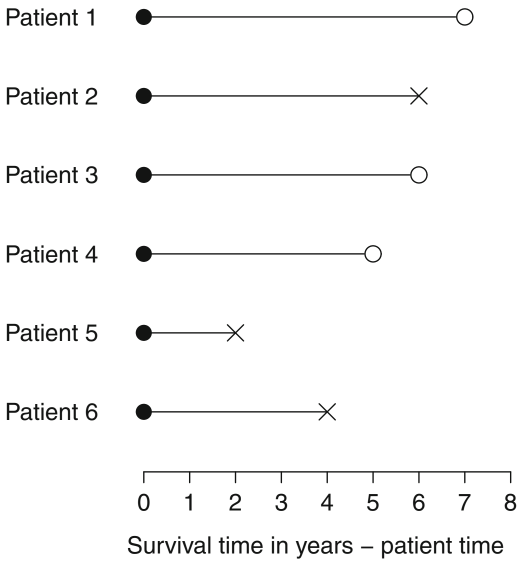


Fig. 1.2 Clinical trial survival data, patient time

Table 1.1 Survival data

Patient	Survtime	Status
1	7	0
2	6	1
3	6	0
4	5	0
5	2	1
6	4	1

Figure 1.2 presents this data set in terms of patient time, where each patient is shown as starting at time zero. Here we again see that three of the patients have complete information; that is, we know when they started the trial and when they died. The other three patients were right-censored; for these patients, the last follow-up times (the last times at which the patient is known to be alive) are indicated by open circles.

The data may be represented in tabular form as shown in Table 1.1. Here, the variable "Survtime" refers to the time from entry into the trial until death or loss to follow-up, whichever comes first, and "Status" indicates whether the survival time represents an event (Status = 1) or is censored (Status = 0).

Administrative censoring has the property that the censoring mechanism is (ordinarily) independent of the survival mechanism, and such censoring can be accommodated using the techniques described in the remainder of the book. Right censoring due to dropout is more problematic. If these patients drop out for reasons unrelated to the outcome, this form of censoring, like that due to patients remaining alive at the end of the follow-up period, is said to be *non-informative*, and can be directly accommodated using the methods to be discussed in the next few chapters.

Informative censoring, by contrast, may (for example) result if individuals in a clinical trial tend to drop out of the study (and become lost to follow-up) for reasons related to the failure process. This type of censoring can introduce biases into the analysis that are difficult to adjust for. The methods we discuss will require the assumption that censoring is non-informative.

The goals of survival analysis are to estimate the survival distribution, to compare two or more survival distributions, or (more generally) to assess the effects of a number of factors on survival. The techniques bear some resemblance to regression analysis, with the important distinctions that the outcome variable (time) is always positive and often censored.

1.4 Some Examples of Survival Data Sets

Following are a few examples of studies using survival analysis which we will refer to throughout the text. The data sets may be obtained by installing the text's package "asaur" from the main CRAN repository. Data for these examples is presented in a number of different formats, reflecting the formats that a data analyst may see in practice. For example, most data sets present survival time in terms of time from the origin (typically entry into a trial). One contains specific dates (date of entry into a trial and date of death) from which we compute the survival time. All contain additional variables, such as censoring variables, which indicate that partial time information on some subjects is available. Most also contain treatment indicators and other covariate information.

Example 1.2. Xelox in patients with advanced gastric cancer

This is a Phase II (single sample) clinical trial of Xeloda and oxaliplatin (XELOX) chemotherapy given before surgery to 48 advanced gastric cancer patients with para-aortic lymph node metastasis (Wang et al. [74]). An important survival outcome of interest is progression-free survival, which is the time from entry into the clinical trial until progression or death, whichever comes first. The data, which have been extracted from the paper, are in the data set "gastricXelox" in the "asaur" package; a sample of the observations (for patients 23 through 27) are as follows:

```
> library (asaur)
> gastricXelox[23:27,]
   timeWeeks delta
23        42     1
24        43     1
25        43     0
26        46     1
27        48     0
```

The first column is the patient (row) number. The second is a list of survival times, rounded to the nearest week, and the third is "delta", which is the censoring indicator. For example, for patient number 23, the time is 42 and delta is 1, indicating that the observed endpoint (progression or death) had been observed 42 weeks after entry into the trial. For patient number 25, the time is 43 and delta is 0, indicating that the patient was alive at 43 weeks after entry and no progression had been observed. We will discuss this data set further in Chap. 3.

Example 1.3. Pancreatic cancer in patients with locally advanced or metastatic disease

This is also a single sample Phase II study of a chemotherapeutic compound, and the main purpose was to assess overall survival and also "progression-free survival", which is defined as the time from entry into the trial until disease progression or death, whichever comes first. A secondary interest in the study is to compare the prognosis of patients with locally advanced disease as compared to metastatic disease. The results were published in Moss et al. [51] The data are available in the data set "pancreatic" in the "asaur" package. Here are the first few observations:

```
> head(pancreatic)
  stage    onstudy progression      death
1     M 12/16/2005    2/2/2006 10/19/2006
2     M   1/6/2006   2/26/2006  4/19/2006
3    LA   2/3/2006    8/2/2006  1/19/2007
4     M  3/30/2006           .  5/11/2006
5    LA  4/27/2006   3/11/2007  5/29/2007
6     M   5/7/2006   6/25/2006 10/11/2006
```

For example, Patient #3, a patient with locally advanced disease (stage = "LA"), entered the study on February 3, 2006. That person was found to have progressive disease on August 2 of that year, and died on January 19 of the following year. The progression-free survival for that patient is the difference of the progression date and the on-study date. Patient #4, a patient with metastatic disease (stage = "M"), entered on March 30 2006 and died on May 11 of that year, with no recorded date of progression. The progression-free survival time for that patients is thus the difference of the death date and the on-study date. For both patients, the overall survival is the difference between the date of death and the on-study date. In this study there was no censoring, since none of these seriously ill patients survived for very long. In Chap. 3 we will see how to compare the survival of the two groups of patients.

Example 1.4. Survival prospects of prostate cancer patients with high-risk disease

In this data set there are two outcomes of interest, death from prostate cancer and death from other causes, so we have what is called a competing risks survival analysis problem. In this example, we have simulated data from 14,294 prostate cancer patients based on detailed competing risks analyses published by Lu-Yao et al. [46]. For each patient we have grade (poorly or moderately differentiated), age of diagnosis (66-70, 71-75, 76-80, and 80+), cancer stage (T1c if screen-diagnosed using a prostate-specific antigen blood test, T1ab if clinically diagnosed without screening, or T2 if palpable at diagnosis), survival time (days from diagnosis to death or date last seen), and an indicator ("status") for whether the patient died of prostate cancer (status = 1), died of some other cause (status = 2), or was still alive at the date last seen (status = 0). The simulated data set matches the original in the number of patients in each of the two grades, three stages, and four age groups (24 categories). For each of the 24 categories, Lu-Yao et al. [46] also presented competing risks survival plots for death from prostate cancer and from other causes, and these 24 plots were used to simulate the data presented here. Thus, the simulated data preserve many of the key characteristics of the original. This data set, "prostateSurvival", is available in the "asaur" package. Here is a list of the data for a few patients (88–95):

```
> prostateSurvival[88:95,]
   grade stage ageGroup survTime status
88  poor    T2    75-79       33      0
89  mode    T2    75-79        6      0
90  mode   T1c    75-79       15      2
91  mode    T2    70-74        6      2
92  mode  T1ab      80+       93      1
93  poor    T2      80+       60      2
94  mode    T2      80+        1      0
95  mode  T1ab    75-79       34      0
```

When analyzing such a large data set, we will typically apply statistical models to selected subsets rather than to the entire data set, for two distinct reasons. First, patients of different ages or disease types may have vastly different disease trajectories, and depend on measured covariates in quite different ways. Thus, attempting to construct a single model for the entire data set is likely to involve complicated interaction terms, and interpreting these can be rather difficult. Second, the types of questions one is interested in asking can be very different for different classes of patients. For example, for the younger men in this data set (say, 65 through 74), we may be particularly interested in teasing out the projected time to death from prostate cancer as compared to death from other causes, a topic we address Sect. 9.2 (competing risks). For older patients (say 85 and older), one can also look at competing risks if desired, and we do this in Sect. 9.2. But practically speaking, the main interest may be in overall mortality among these men, an issue we address in Sect. 12.1.

Example 1.5. Comparison of medical therapies to aid smokers to quit

The purpose of this study (Steinberg et al. [63]) was to evaluate extended duration of a triple-medication combination versus therapy with the nicotine patch alone in smokers with medical illnesses. Patients with a history of smoking were randomly assigned to the triple-combination or patch therapy and followed for up to six months. The primary outcome variable was time from randomization until relapse (return to smoking); individuals who remained non-smokers for six months were censored. The data set, "pharmacoSmoking", is available in the "asaur" package. Here is a listing of a few cases and variables:

```
> pharmacoSmoking[1:6, 2:8]
  ttr relapse         grp age gender     race employment
1 182       0   patchOnly  36   Male    white         ft
2  14       1   patchOnly  41   Male    white      other
3   5       1 combination  25 Female    white      other
4  16       1 combination  54   Male    white         ft
5   0       1 combination  45   Male    white      other
6 182       0 combination  43   Male hispanic         ft
```

The variable “ttr” is the number of days without smoking (“time to relapse”), and “relapse=1” indicates that the subject started smoking again at the given time. The variable “grp” is the treatment indicator, and “employment” can take the values “ft” (full time), “pt” (part time), or “other”. The primary objectives were to compare the two treatment therapies with regard to time to relapse, and to identify other factors related to this outcome.

Example 1.6. Prediction of survival of hepatocellular carcinoma patients using biomarkers

This study (Li et al. [42, 43]) focused on using expression of a chemokind known as CXCL17, and other clinical and biomarker factors, to predict overall and recurrence-free survival. This example contains data on 227 patients, each with a wide range of clinical and biomarker values. The “hepatoCellular” data are publicly available in the Dryad online data repository [43] as well as in the “asaur” R package that accompanies this text. Here, for illustration, is a small selection of cases and covariates.

```
> hepatoCellular[c(1, 2, 3, 65, 71),c(2, 3, 16:20, 24, 47)]
   Age Gender OS Death RFS Recurrence   CXCL17T CD4N     Ki67
1   57      0 83     0  13          1 113.94724    0  6.04350
2   58      1 81     0  81          0  54.07154   NA       NA
3   65      1 79     0  79          0  22.18883   NA       NA
65  38      1  5     1   5          1 106.78169    0 44.24411
71  57      1 11     1  11          1  98.49680    0 99.59232
```

The survival outcomes are “OS” (overall survival) and “RFS” (recurrence-free survival), and the corresponding censoring indicators are “Death” and “Recurrence”. The full data set has 48 columns. In columns 23 to 48 there are many patients with missing values, with only 117 patients having complete data.

1.5 Additional Notes

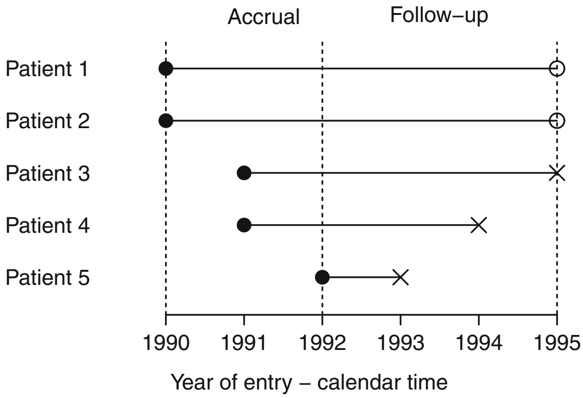
1. Another type of incomplete observation with survival data is truncation, a result of length-biased sampling. We discuss left truncation in Sect. 3.5. Right truncation is less common and more difficult to model. See Klein and Moeschberger [36] for further discussion.
2. The Healthcare Delivery Research Program of the Division of Cancer Control and Population Sciences, National Cancer Institute, USA maintains the SEER-Medicare Linked Database, which provided the data used in Lu-Yao et al. [46].

This NCI-based research program makes this data available for research only, and will not permit it to be distributed for educational purposes. Thus it cannot be used in this book. Fortunately, however, the Lu-Yao publication contains detailed cause-specific survival curves for patients cross-classified by four age groups, three stage categories, and two Gleason stages, as well as precise counts of the numbers of patients in each category. This information was used to simulate a survival data set that maintains many of the characteristics of the original SEER-Medicare data used in the paper. This simulated data set, “prostateSurvival”, is what is used in this book for instructional purposes.

3. Numerous excellent illustrative survival analysis data sets are freely available to all. The standard “survival” library that is distributed with the R system has a number of survival analysis data sets. Also, the “KMsurv” R package contains a rich additional set of data sets that were discussed in Klein and Moeschberger [36]. The “asaur” R package contains data sets used in the current text.

Exercises

1.1. Consider a simple example of five cancer patients who enter a clinical trial as illustrated in the following diagram:



Re-write these survival times in terms of patient time, and create a simple data set listing the survival time and censoring indicator for each patient. How many patients died? How many person-years are there in this trial? What is the death rate per person-year?

1.2. For the “gastricXelox” data set, use R to determine how many patients had the event (death or progression), the number of person-weeks of follow-up time, and the event rate per person-week.

Chapter 2
Basic Principles of Survival Analysis

2.1 The Hazard and Survival Functions

Survival analysis methods depend on the survival distribution, and two key ways of specifying it are the survival function and the hazard function. The *survival function* defines the probability of surviving up to a point t. Formally,

$$S(t) = pr(T > t), \quad 0 < t < \infty$$

This function takes the value 1 at time 0, decreases (or remains constant) over time, and of course never drops below 0. As defined here it is *right continuous*.[1]

The survival function is often defined in terms of the *hazard function*, which is the instantaneous failure rate. It is the probability that, given that a subject has survived up to time t, he or she fails in the next small interval of time, divided by the length of that interval. Formally, this may be expressed as

$$h(t) = \lim_{\delta \to 0} \frac{pr(t < T < t + \delta | T > t)}{\delta}$$

This function is also known as the *intensity function* or the *force of mortality*.

The hazard and survival functions are two ways of specifying a survival distribution. To illustrate the connection between the two, consider first a case where the hazard is initially very high. Such a hazard might be appropriate to describe the lifetimes of animals with high mortality early in life. Figure 2.1 illustrates such a hazard function (a) and the corresponding survival function (b). Next consider the

[1] In some texts the survival function is defined as $S(t) = Pr(T \geq t)$, resulting in a left-continuous survival function. This issue arises with step function survival curves, e.g. the Kaplan-Meier estimate discussed in the next chapter.

© Springer International Publishing Switzerland 2016
D. Moore, *Applied Survival Analysis Using R*, Use R!,
DOI 10.1007/978-3-319-31245-3_2

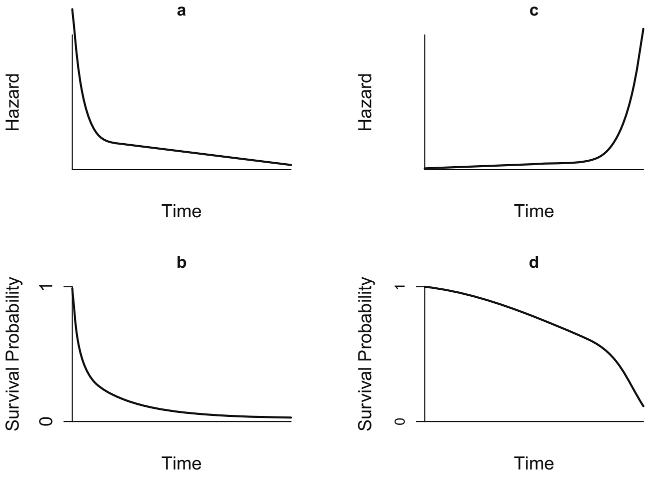


Fig. 2.1 Hazard and survival functions with high initial hazard (**a** and **b**) and low initial hazard (**c** and **d**)

opposite, where the hazard is initially low, and increases later in life. Such a hazard would describe organisms with low initial hazard of death. This is illustrated in Fig. 2.1 (c) and the corresponding survival in (d).

Demographic data provide another illustration of hazard and survival functions, as shown in the next example, where we see elements of both high early and high late hazard functions.

Example 2.1. The daily hazard rates of men and women by age in each calendar year from 1940 to 2004 are contained in the three-dimensional array “survexp.us”, which is part of the R package “survival”. These hazard rates were derived from US life tables using methodology described in Therneau and Offord [70]. Figure 2.2 shows the distribution of lifetimes in the United States in 2004 for males and females. The hazard plot, here plotted on a log scale, shows several features of a human lifespan. First, the initial days and weeks of life are particularly dangerous, and the risk recedes rapidly after the first month of life. The hazard increases during the teen years, then levels off, until it starts a steady increase in midlife. Males exhibit a higher mortality than females, as is well-known. The corresponding survival function is also shown. This example also demonstrates that the hazard function may show details of changes in risk that may not be apparent in the survival curves.

To get the hazards in R, we use the following R code, which is run after a “library(survival)” command. The “#” character is called a comment character; text following this character is explanatory, and not executed.

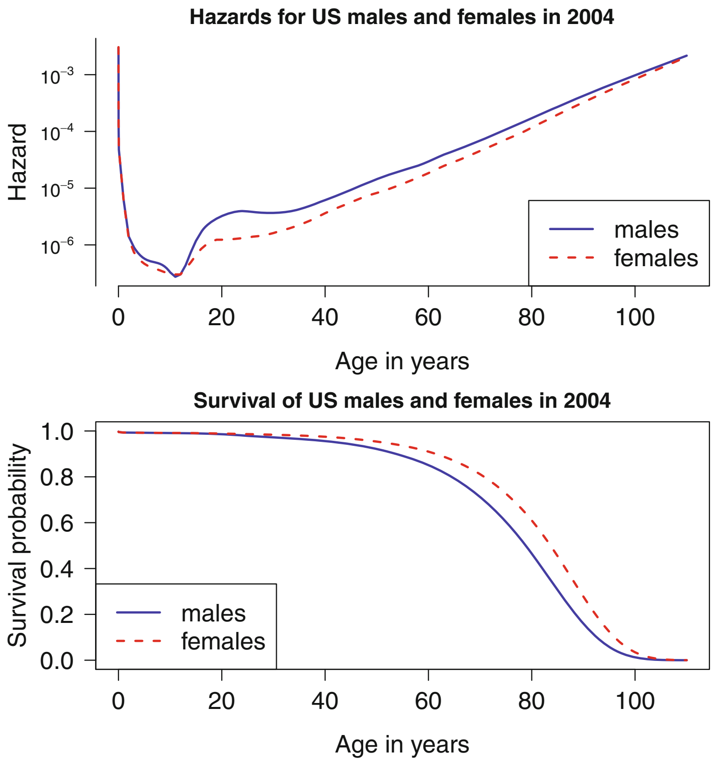


Fig. 2.2 Hazard and survival functions for US males and females in 2004. The hazard function is plotted on a log scale

```
> tm <- c(0,                  # birth
        1/365,                # first day of life
        7/365,                # seventh day of life
        28/365,               # fourth week of life
        1:110)                # subsequent years
> hazMale <- survexp.us[,"male","2004"]          # 2004 males
> hazFemale <- survexp.us[,"female","2004"]      # 2004 females
```

The hazard plot in Fig. 2.2 is obtained by plotting “hazMale” and “hazFemale” versus “tm”. In Sect. 2.5 we will show how to derive the lifetime survival distributions for males and females from the corresponding hazard functions.

2.2 Other Representations of a Survival Distribution

In addition to the survival and hazard functions, there are several other ways to define a survival distribution. The cumulative distribution function (CDF), which is commonly used outside of survival analysis, is given by

$$F(t) = pr(T \leq t), \ \ 0 < t < \infty$$

This is the complement of the survival function and, like the survival function, it is right continuous. In survival analysis, this function is known as the *cumulative risk function* (not to be confused with the cumulative hazard defined below). The probability density function (PDF),

$$f(t) = -\frac{d}{dt}S(t) = \frac{d}{dt}F(t)$$

is the rate of change of the CDF, or minus the rate of change of the survival function. The hazard function is related to the PDF and survival functions by

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

That is, the hazard at time t is the probability that an event occurs in the neighborhood of time t divided by the probability that the subject is alive at time t. The *cumulative hazard function* is defined as the area under the hazard function up to time t, that is,

$$H(t) = \int_0^t h(u)du$$

The survival function may be defined in terms of the hazard function by

$$S(t) = \exp\left(-\int_0^t h(u)du\right) = \exp\left(-H(t)\right) \tag{2.2.1}$$

It is this relationship that allows us to compute the survival function corresponding to a hazard function, as in Figs. 2.1 and 2.2.

2.3 Mean and Median Survival Time

The mean survival is the expected value of the survival time,

$$\mu = E(T) = \int_0^\infty tf(t)dt$$

which, using standard integration by parts, and using the fact that $f(t) = -d/dtS(t)$, may be written as[2]

$$\mu = \int_0^\infty S(t)dt. \tag{2.3.1}$$

The mean survival time is only defined if $S(\infty) = 0$, that is, if all subjects eventually fail. This might not be the case if, for example, the survival outcome is time to cancer recurrence, and some fraction c of the subjects are cured and thus have no recurrence. In that case, $S(\infty) = c$, and the area under the survival curve is infinite. In theory the mean survival also cannot be computed with the Kaplan-Meier survival curve when the curve does not reach zero, an issue we will discuss in the next chapter. If a mean survival time is required in this situation, a work-around is to specify a maximum possible survival time, so that the integral becomes finite.

The median survival time is defined as the time t such that $S(t) = 1/2$. If the survival curve is not continuous at $1/2$ (if the survival function is a step function, for example), then the median is taken to be the smallest t such that $S(t) \leq 1/2$. If the survival curve does not drop below $1/2$ during the observation period, then of course the median survival is undefined.

2.4 Parametric Survival Distributions

Several survival distributions are available for modeling survival data. The exponential distribution, the simplest survival distribution, has a constant hazard, $h(t) = \lambda$. The cumulative hazard function may be easily derived using the relationships in the previous section:

$$H(t) = \int_0^t h(u)du = \int_0^t \lambda du = \lambda t|_0^t = \lambda t$$

Thus, the cumulative hazard at time t is just the area λt of the shaded rectangle in Fig. 2.3.

The survival function is

$$S(t) = e^{-H(t)} = e^{-\lambda t}$$

and the probability density function is given by

$$f(t) = h(t)S(t) = \lambda e^{-\lambda t}.$$

[2] To establish this formula, we also need the result that $\lim_{t\to\infty}(t \cdot S(t)) = 0$. This is easy to show for the exponential distribution, but it is non-trivial to prove in general.

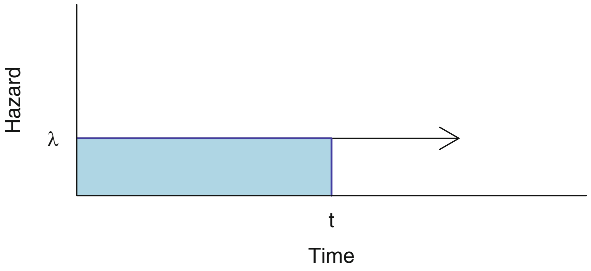


Fig. 2.3 Exponential hazard, with shaded area indicating the cumulative hazard at time *t*

The mean of an exponential random variable is given by (using Eq. 2.3.1)

$$E(T) = \int_0^\infty S(t)\,dt = \int_0^\infty e^{-\lambda t}\,dt = 1/\lambda .$$

The median is the value of t that satisfies $0.5 = e^{-\lambda t}$, so that $t_{\text{med}} = \log(2)/\lambda$. (In this text, as in the R language, "log" refers to the natural logarithm.)

The exponential distribution is easy to work with, but the constant hazard assumption is not often appropriate for describing the lifetimes of humans or animals. The Weibull distribution, which offers more flexibility in modeling survival data, has hazard function

$$h(t) = \alpha\lambda(\lambda t)^{\alpha-1} = \alpha\lambda^\alpha t^{\alpha-1}.$$

The cumulative hazard and survival functions are given by, respectively,

$$H(t) = (\lambda t)^\alpha$$

and

$$S(t) = e^{-(\lambda t)^\alpha}.$$

Figure 2.4 shows the shape of the hazard for several parameter choices. The exponential distribution is a special case with $\alpha = 1$. It is monotone increasing for $\alpha > 1$ and monotone decreasing for $\alpha < 1$.

The mean and median of the Weibull distribution are, respectively,

$$E(T) = \frac{\Gamma(1 + 1/\alpha)}{\lambda} \tag{2.4.1}$$

and

$$t_{\text{med}} = \frac{[\log(2)]^{1/\alpha}}{\lambda}. \tag{2.4.2}$$

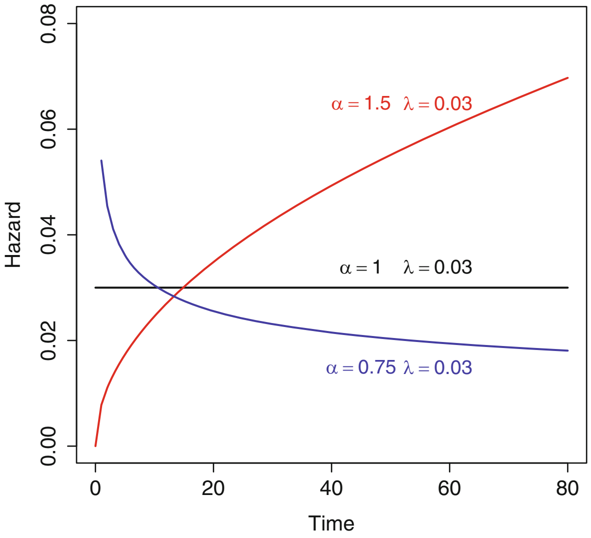


Fig. 2.4 Weibull hazard functions

For integers, the gamma function is given by $\Gamma(n) = (n-1)!$. For the special case $\alpha = 1$, of course, the mean and median are identical to those of the exponential distribution. For non-integers, it must be evaluated numerically; in R, this may be done using the "gamma" function.

In R the functions "dweibull" and "pweibull" compute the p.d.f. and c.d.f., respectively, of the Weibull distribution. These functions use the arguments "shape" and "scale" to represent the parameters α and $1/\lambda$, respectively. To obtain the survival function, we can specify "lower.tail = F" as an option in the "pweibull" function. For example, we can plot the Weibull survival function with $\alpha = 1$ and $\lambda = 0.03$ by first defining a function "weibSurv" with these parameters and then using the "curve" function to plot the curve as follows (figure not shown):

```
weibSurv <- function(t, shape, scale) pweibull(t, shape=shape,
            scale=scale, lower.tail=F)
curve(weibSurv(x, shape=1.5, scale=1/0.03), from=0, to=80,
       ylim=c(0,1), ylab='Survival probability', xlab='Time')
```

To plot the hazard function with this shape and scale, as shown by the red curve in Fig. 2.4, we can use the following code to first define the hazard function as the p.d.f. divided by the survival function,

```
weibHaz <- function(x, shape, scale) dweibull(x, shape=shape,
     scale=scale)/pweibull(x, shape=shape, scale=scale,
        lower.tail=F)
curve(weibHaz(x, shape=1.5, scale=1/0.03), from=0, to=80,
      ylab='Hazard', xlab='Time', col="red")
```

The other two curves may be obtained using "shape = 1" and "shape = 0.75" when calling the "curve" function. To place the additional curves on the plot, add "add = T" as an option to the "curve" function.

We may generate random variables from the exponential or Weibull distribution using the functions "rexp" and "rweib". For example, we may generate 1000 Weibull random variables with shape 1.5 and scale 1/0.03, and compute their mean and median, as follows:

```
> tt.weib <- rweibull(1000, shape=1.5, scale=1/0.03)
> mean(tt.weib)
[1] 31.35497
> median(tt.weib)
[1] 26.84281
```

The theoretical mean and median, using Eqs. 2.4.1 and 2.4.2, are as follows:

```
> gamma(1 + 1/1.5)/0.03    # mean
[1] 30.09151

> (log(2)^(1/1.5))/0.03      # median
[1] 26.10733
```

The empirical mean and median are close to their theoretical values, as they must be.

The gamma distribution (not to be confused with the gamma function) provides yet another choice for survival modeling. The probability density function is given by

$$f(t) = \frac{\lambda^{\beta} t^{\beta-1} \exp(-\lambda t)}{\Gamma(\beta)}$$

While the hazard and survival functions cannot be written in closed form, they can be computed using the formulas in the previous section. Figure 2.5 shows several examples of hazard functions. It is monotone increasing for $\beta > 1$ and monotone decreasing for $\beta < 1$. When $\beta = 1$, the gamma distribution reduces to an exponential distribution.

To plot the gamma hazard function for $\beta = 1.5$ and $\lambda = 0.03$, we can use the following code:

```
gammaHaz <- {function(x, shape, scale) dgamma(x, shape=shape,
   scale=scale)/pgamma(x, shape=shape, scale=scale, lower.tail=F)}
curve(gammaHaz(x, shape=1.5, scale=1/0.03), from=0, to=80,
   ylab='Hazard', xlab='Time', col="red")
```

This produces the red curve in Fig. 2.5. The other two curves may be obtained using "shape = 1" and "shape = 0.75", along with the "add = T" option. Other parametric families of survival distributions include the log-normal (see Exercise 2.6 for this one), log-logistic, Pareto, and many more. See for example Klein and Moeschberger [36] and Cox and Oakes [11] for details.

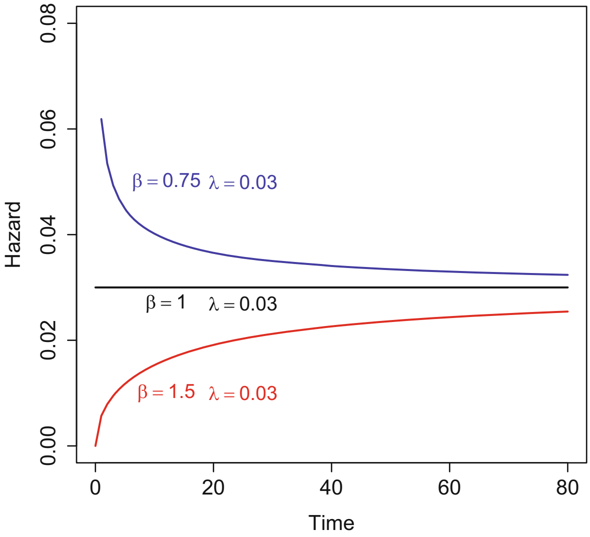


Fig. 2.5 Gamma hazard functions

2.5 Computing the Survival Function from the Hazard Function

If we know the hazard function of a survival random variable, we may derive the survival function using Eq. 2.2.1. For some parametric families, this is simple to do. But if the hazard function is more complicated, we need to use numerical methods to evaluate the integral. For example, consider again the human hazard functions in Fig. 2.2. To get the corresponding survival plots, we first compute a vector of differences, “tm.diff”, then we find the cumulative hazard functions using the “cumsum” function, and finally we use the relationship of the survival function to the cumulative hazard to get “survMale” and “survFemale”. The survival functions in Fig. 2.2 result from plotting these survival functions versus “tm”. In the following code, “tm.diff” is the width of each rectangle, and “survMale” and “survFemale” represent the survival curves for males and females, respectively.

```
> tm.diff <- diff(tm)
> survMale <- exp(-cumsum(hazMale*tm.diff)*365.24)
> survFemale <- exp(-cumsum(hazFemale*tm.diff)*365.24)
```

The diagram in Fig. 2.6 illustrates the computation of the cumulative hazard at time $t = 1.5$, which is the shaded area.

This representation of the hazard function is an example of a piecewise exponential distribution, since the hazard is constant on specified time intervals.

Now that we have the survival distributions for men and women, we can compute the mean age of death for men and women in 2004 in the US, which is the area under the respective survival curve in Fig. 2.2. In the following code, “tm.diff” is the width

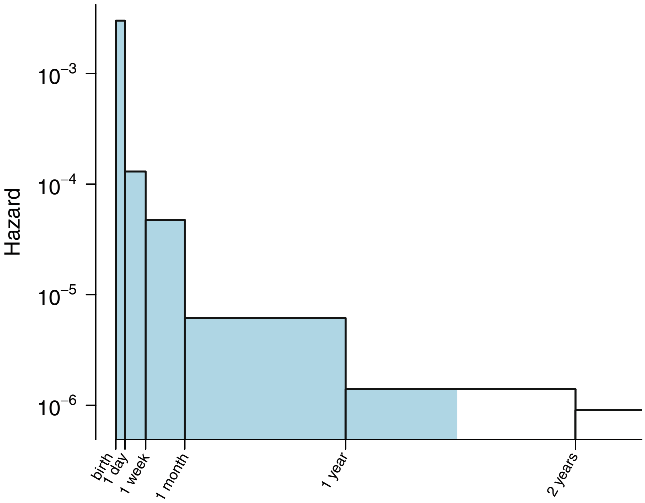


Fig. 2.6 Illustration of the calculation of the cumulative hazard for males in 2004. The hazard function stored in "survexp.us" is actually a step function evaluated at times 1 day, 1 week, 1, month, 1 year, and every year thereafter. Shown here are only the first two years, and the shaded area represents the cumulative hazard at 1.5 years

of each rectangle and "survMale" and "survFemale" are the heights of the rectangles for men and women, respectively. The sum of the areas of the rectangles gives the value of the integral:

```
> sum(survMale*tm.diff)      # mean age of male death in 2004
[1] 73.8084
> sum(survFemale*tm.diff)    # mean age of female death in 2004
[1] 78.90526
```

2.6 A Brief Introduction to Maximum Likelihood Estimation

The previous sections show us how to compute probabilities for a specific probability distribution (e.g., exponential, Weibull, gamma) for specified values of the parameters. For example, if we know that a random variable T has an exponential distribution with parameter $\lambda = 0.03$, we can directly compute the probability that T exceeds a particular value. But suppose that we have a series of observations $t_1, t_2, \ldots, t_n$from an exponential distribution with unknown parameter λ. How can we estimate λ? The theory of maximum likelihood estimation provides a mathematical framework for doing this. While a comprehensive discussion of likelihood theory is beyond the scope of this book, we may get an overview of the technique by considering a simple example using the exponential distribution. We construct a likelihood by taking a product of terms from the exponential distribution, one for each observation. If there is no censoring, the likelihood function takes the general form

$$L(\lambda; t_1, t_2, \ldots, t_n) = f(t_1, \lambda) \cdot f(t_2, \lambda) \cdot \cdots \cdot f(t_n, \lambda) = \prod_{i=1}^{n} f(t_i, \lambda).$$

If some observations are censored, we have to make an adjustment to this expression. For an observation of an observed death, we put in the p.d.f. as above. But for a right-censored observation, we put in the survival function, indicating that observation is known only to exceed a particular value. The likelihood in general then takes the form

$$L(\lambda; t_1, t_2, \ldots, t_n) = \prod_{i=1}^{n} f(t_i, \lambda)^{\delta_i} S(t_i, \lambda)^{1-\delta_i} = \prod_{i=1}^{n} h(t_i, \lambda)^{\delta_i} \cdot S(t_i, \lambda). \quad (2.6.1)$$

This expression means that when t_i is an observed death, the censoring indicator is $\delta_i = 1$, and we enter a p.d.f. factor. When t_i is a censored observation, we have $\delta_i = 0$ we enter a survival factor. Alternatively, we may enter a hazard factor for each censored observation and a survival factor for every observation, censored or not.

For the exponential distribution the likelihood, we substitute the expressions for the p.d.f. and survival distributions, and simplify as follows:

$$L(\lambda) = \prod_{i=1}^{n} \left[\lambda e^{-t_i/\mu}\right]^{\delta_i} \left[e^{-\lambda t_i}\right]^{1-\delta_i} = \lambda^d e^{-\lambda V}$$

Alternatively, we may substitute a hazard factor λ for the censored observations and a survival factor $e^{-\lambda t_i}$ for all observations. This of course leads to the same form for the likelihood function. We have the total number of deaths, $d = \sum_{i=1}^{n} \delta_i$ and the total amount of time of patients on the study, $V = \sum_{i=1}^{n} t_i$. This latter term is known in epidemiology as person-years (or person-months or person-days, according to the time unit). We need to find the value of λ that maximizes this function, and that value is known as the *maximum likelihood estimate*. Now, this product formula is difficult to work with, so we use a logarithmic transformation to convert it into a sum, known as the *log-likelihood*,

$$l(\lambda) = d \log \lambda - \lambda V.$$

Since the log transformation is monotonic, the value of λ that maximizes the log-likelihood also maximizes the original likelihood function. We use standard calculus to find the first derivative, also called the *score function*,

$$l'(\lambda) = \frac{d}{\lambda} - V$$

which we set equal to zero to obtain the *maximum likelihood estimate*, $\hat{\lambda} = d/V$. That is, our estimate is the number of deaths divided by the number of person-years.

Next, we compute the second derivative of the log-likelihood,

$$l''(\lambda) = -\frac{d}{\lambda^2} = -I(\lambda)$$

which, when we switch the sign, is known as the *information*. This is important for two reasons. First, since the information is positive (the second derivative is negative), the likelihood function is concave down, which shows that we have indeed found a maximum. Second, using standard mathematical statistics theory, the inverse of the information is approximately the variance of the m.l.e.,

$$\text{var}\left(\hat{\lambda}\right) \approx I^{-1}(\lambda) = \lambda^2/d$$

Now we substitute $\hat{\lambda}$ for λ to obtain the observed information $I(\hat{\lambda})$, and from there we get an estimate of the variance of the parameter:

$$\widehat{\text{var}}\left(\hat{\lambda}\right) \approx I^{-1}\left(\hat{\lambda}\right) = \hat{\lambda}^2/d = d/V^2$$

We may use this formula to carry out hypothesis tests or find a confidence interval for λ.

Consider for example the six observations in Table 1.1, and suppose that they are derived from an exponential distribution with unknown parameter λ. There are three deaths, which gives us $d = 3$. Also, the total patient time on study is $V = 7+6+6+5+2+4 = 30$. The log-likelihood function is $l(\lambda) = 3 \log \lambda - 30\lambda$, and the maximum likelihood estimate is given by $\hat{\lambda} = 3/30 = 0.1$ (Fig. 2.7).

Maximum likelihood methods may be applied to a wide range of statistical problems, using other distributions and more than one parameter, and (under technical conditions that are often satisfied), the m.l.e. is asymptotically normal with a mean that approaches the true mean of the parameter and a variance (or, when there are multiple parameters, a covariance matrix) that is the inverse of the information, or minus the second derivative of the log-likelihood theory. The generality of the method makes it a central part of statistical theory and practice.

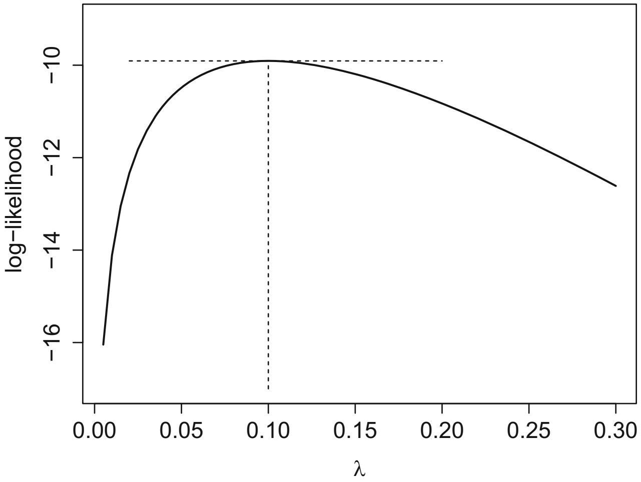


Fig. 2.7 Log-likelihood function for data from Table 1.1, showing the maximum likelihood estimate, and the horizontal tangent at the maximum

2.7 Additional Notes

1. Methods for the analysis if human life tables pre-date modern survival analysis as described here. See Preston, Heuveline, and Guillot [55] for a thorough modern exposition of the methods used in demography.
2. Many sources, including R, express the Weibull distribution using $\beta = 1/\lambda$. Then α is known as the "shape" parameter and β as the "scale" parameter. Still others, e.g. Klein and Moeschberger [36], express this distribution in terms of $\lambda^* = \lambda^\alpha$, so that $h(t) = \lambda^* \alpha t^{\alpha-1}$. The terms "shape" and "scale" refer to the shape and scale of the probability density function; these terms are not particularly relevant to survival analysis, where the emphasis on the hazard and survival functions. In fact, in later chapters, the term "scale" will take on a completely different meaning when we use the Weibull distribution for modeling survival data with covariates. Despite the potential confusion over two meanings for "scale", we must continue use the "shape" and "scale" terminology as defined here since these are the names of the parameters used by the R Weibull functions.
3. Thorough discussions of maximum likelihood methods in survival analysis may be found in the classical references Kalbfleisch and Prentice [34] and Cox and Oakes [11].

Exercises

2.1. Using the “survexp.us” data described in Example 2.2, plot the hazard functions for men and women in 1940 and 2000. Comment on the change in mortality rates in children.

2.2. Find the mean age of death separately for men and women for 1940 and 2000.

2.3. The data set “survexp.usr” in the “survival” package is a four dimensional array of hazards in format similar to the “survexp.us” data set, with race (black or white) in the added dimension. Plot the hazard functions for black males and white males for 1940 and 2000.

2.4. Consider the survival data in Exercise 1.1. Assuming that these observations are from an exponential distribution, find $\hat{\lambda}$ and an estimate of $\text{var}(\hat{\lambda})$.

2.5. Consider a survival distribution with constant hazard $\lambda = 0.07$ from $t = 0$ until $t = 5$ and then hazard $\lambda = 0.14$ for $t > 5$. (This is known as a piecewise constant hazard.) Plot this hazard function and the corresponding survival function for $0 < t < 10$. What is the median survival time?

2.6. Another parametric survival distribution is the log-normal distribution. Use the density and cumulative distribution R functions “dlnorm” and “plnorm” to compute and plot the lognormal hazard functions with the parameter “meanlog” taking the values 0, 1, and 2, and with “sdlog” fixed at 0.25. Describe the risk profile a disease would have if it followed one of these hazard functions.

Chapter 3
Nonparametric Survival Curve Estimation

3.1 Nonparametric Estimation of the Survival Function

We have seen that there are a wide variety of hazard function shapes to choose from if one models survival data using a parametric model. But which parametric model should one use for a particular application? When modeling human or animal survival, it is hard to know what parametric family to choose, and often none of the available families has sufficient flexibility to model the actual shape of the distribution. Thus, in medical and health applications, nonparametric methods, which have the flexibility to account for the vagaries of the survival of living things, have considerable advantages. In this chapter we will discuss non-parametric estimators of the survival function. The most widely used of these is the product-limit estimator, also known as the Kaplan-Meier estimator. This estimator, first proposed by Kaplan and Meier [35], is the product over the failure times of the conditional probabilities of surviving to the next failure time. Formally, it is given by

$$\hat{S}(t) = \prod_{t_i \leq t} (1 - \hat{q}_i) = \prod_{t_i \leq t} \left(1 - \frac{d_i}{n_i}\right)$$

where n_i is the number of subjects at risk at time t_i, and d_i is the number of individuals who fail at that time. The example data in Table 1.1 may be used to illustrate the construction of the Kaplan-Meier estimate, as shown in Table 3.1.

D. Moore, *Applied Survival Analysis Using R*, Use R!,
DOI 10.1007/978-3-319-31245-3_3

Table 3.1 Kaplan-Meier estimate

t_i	n_i	d_i	q_i	$1 - q_i$	$S_i = \prod(1 - q_i)$
2	6	1	0.167	0.833	0.846
4	5	1	0.200	0.800	0.693
6	3	1	0.333	0.667	0.497

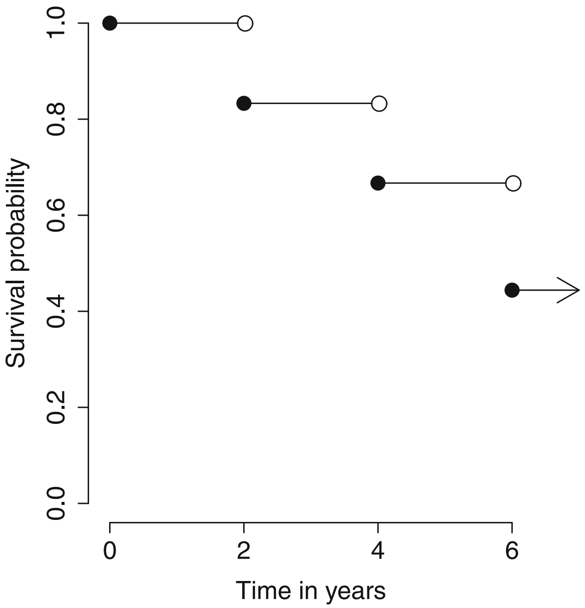


Fig. 3.1 Right-continuous Kaplan-Meier survival function estimate

The columns represent, respectively, the failure time t_i, the number n_i at risk at that time, the number d_i who fail at that time, the failure probability $q_i = d_i/n_i$, the conditional survival probability $1 - q_i$, and the cumulative product, which is the estimate of the survival probability. For example, the probability 0.667 of being alive at time $t_i = 4$ is the probability 0.833 of being alive at time $t_i = 2$ times the probability 0.800 of being alive at time $t_i = 4$ *given* that patients is alive at the previous time, and so on.

Figure 3.1 shows the Kaplan-Meier estimate of the survivor function using these data. This function is a non-increasing step function, and the open and closed circles explicitly show the right-continuity. For example, $S(4) = 0.667$, while $S(3.99) = 0.833$. In practice, the Kaplan-Meier function is plotted as a step function, with the indicators of right-continuity not shown. The median survival time is at $t = 6$, which is the smallest time t such that $S(t) \leq 0.5$, as discussed in Sect. 2.4.

To obtain confidence limits for the product-limit estimator, we first use what is known as the "delta method"[1] to obtain the variance of log($\hat{\mathrm{S}}$(t)),

$$\operatorname{var}\left(\log \hat{S}(t_k)\right) = \sum_{t_i \leq t} \operatorname{var}\left(\log(1-\hat{q}_i)\right) \approx \sum_{t_i \leq t} \frac{d_j}{n_j(n_j - d_j)} \tag{3.1.1}$$

To get the variance of $\hat{S}(t)$ itself, we use the delta method again to obtain

$$\operatorname{var}\left(\hat{S}(t)\right) \approx \left[\hat{S}(t)\right]^2 \sum_{t_i \leq t} \frac{d_i}{n_i(n_i - d_i)} \tag{3.1.2}$$

Unfortunately, confidence intervals computed based on this variance may extend above one or below zero. While one could truncate them at one and zero, a more satisfying approach is to find confidence intervals for the complementary log-log transformation of $\hat{S}(t)$ as follows,

$$\operatorname{var}\left(\log\left[-\log \hat{S}(t)\right]\right) \approx \frac{1}{\left[\log \hat{S}(t)\right]^2} \sum_{t_i \leq t} \frac{d_i}{n_i(n_i - d_i)} \tag{3.1.3}$$

To obtain estimates of the Kaplan-Meier estimator in R for the data in Table 1.1, we first load the "survival" library, and then enter the data. Note that the "Surv" function produces a special structure for censored survival data.

```
> library(survival)
> tt <- c(7,6,6,5,2,4)
> cens <- c(0,1,0,0,1,1)
> Surv(tt, cens)
[1] 7+ 6  6+ 5+ 2  4
```

For the estimation itself we use the "survfit" function,

```
> result.km <- survfit(Surv(tt, cens) ~ 1, conf.type="log-log")
```

To compute confidence intervals based on our preferred method, the complementary log-log transformation, we have to explicitly specify that. The results of the "survfit" procedure are placed into a data structure which we have named "result.km". To see a few basic results, including the median survival and 95 % confidence intervals, just type the structure name,

[1] The delta method allows one to approximate the variance of a continuous transformation $g(\cdot)$ of a random variable. Specifically, if a random variable X has mean μ and variance σ^2, then $g(X)$ will have approximate mean $g(\mu)$ and approximate variance $\sigma^2 \cdot \left[g'(\mu)\right]^2$ for a sufficiently large sample size. Refer to any textbook of mathematical statistics for a more precise formulation of this principle. In the context of the Kaplan Meier survival curve estimate, see Klein and Moeschberger [36] for further details.

```
> result.km

records   n.max n.start   events   median 0.95LCL 0.95UCL
      6       6       6        3        6       2      NA
```

This prints out the number of "records" (here six patients), the number of patients (n.max and n.start), the number of events (three deaths), the median survival time (6 years), and a 95 % confidence interval for the median. Note that the upper 95 % confidence limit is undefined, indicated by a missing value "NA". To see the full Kaplan-Meier estimate, and plot it, we use the "summary" and "plot" functions:

```
> summary(result.km)

 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    2      6       1    0.833   0.152       0.2731        0.975
    4      5       1    0.667   0.192       0.1946        0.904
    6      3       1    0.444   0.222       0.0662        0.785
> plot(result.km)
```

This lists the distinct failure times (2, 4, and 6 years), the number at risk at each time interval, and the number of events at each failure time. Also given are the 95 % confidence intervals for the survival probabilities. The survival function estimate is plotted in Fig. 3.2. This is the same figure as in Fig. 3.1 but without the continuity notation.

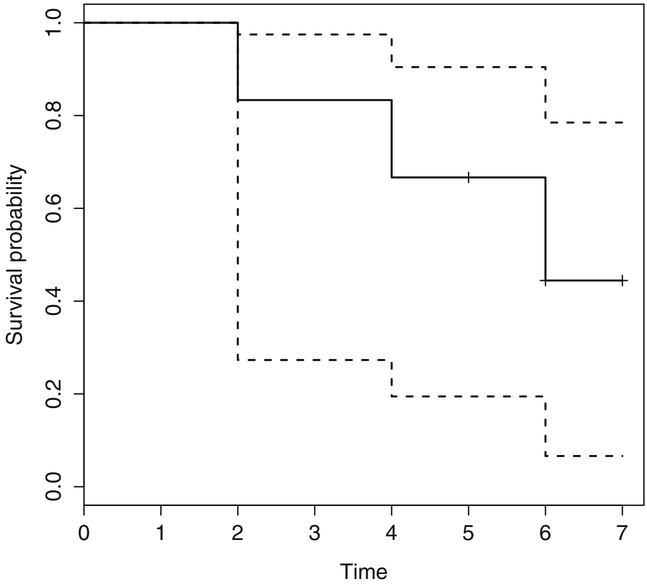


Fig. 3.2 Kaplan-Maier survival curve estimate with 95 % confidence intervals

Table 3.2 Nelson-Altschuler estimate of the survival function

t_i	n_i	d_i	q_i	$H_i = \sum q_i$	$\hat{S}_i = exp(-H_i)$
2	6	1	0.167	0.167	0.846
4	5	1	0.200	0.367	0.693
6	3	1	0.333	0.700	0.497

An alternative estimator of the survival function is known as the *Nelson-Altschuler* estimator.[2] It is based on the relationship of the survival function to the hazard function. An estimate of the cumulative hazard function is the sum of the estimated hazards up to a time t_i:

$$H(t) = \sum_{t_i \leq t} \frac{d_i}{n_i} \tag{3.1.4}$$

and the survival function estimate is simply

$$S(t) = e^{-H(t)}.$$

We may illustrate this by again referring to the data in Table 1.1 of Example 1.1; the calculations are in Table 3.2.

In R, the Nelson-Altschuler estimate may be obtained using the “survfit” function with the option “type = ‘fh’ “, the letters “fh” being taken from the initials of Fleming and Harrington:

```
> result.fh <− survfit(Surv(tt, cens) ~ 1, conf.type="log−log",
+   type="fh")
> summary(result.fh)

 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    2      6       1    0.846   0.155       0.2401        0.981
    4      5       1    0.693   0.200       0.1799        0.925
    6      3       1    0.497   0.248       0.0585        0.841
```

We now consider data from an actual clinical trial. The data set “gastricXelox” is a Phase II (single sample) clinical trial of the chemotherapeutic agent Xelox administered to patients with advanced gastric cancer prior to surgery (Wang et al. [74]). The primary outcome of interest is “progression-free survival.” This quantity is defined as the time from entry into a clinical trial until progression or death, whichever comes first. The survival data set was extracted from the paper, and the survival times rounded to the nearest week. The product-limit estimator may be estimated and plotted as follows, after converting the time scale from weeks to months:

[2] Other names associated with this estimator are Aalen, Fleming, and Harrington.

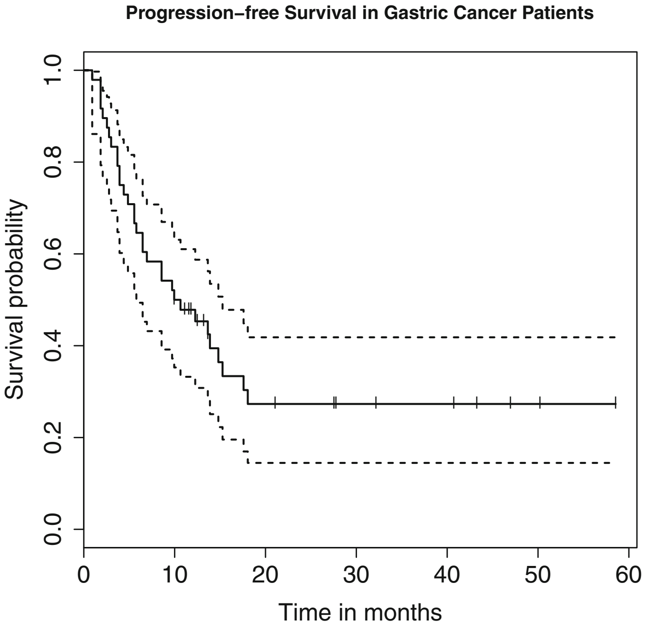


Fig. 3.3 Progression-free survival of gastric cancer patients treated with Xelox

```
> timeMonths <- gastricXelox$timeWeeks*7/30.25
> delta <- gastricXelox$delta
> result.km <- survfit(Surv(timeMonths, delta) ~ 1,
+       conf.type="log-log")
> plot(result.km, conf.int=T, mark="|", xlab="Time in months",
+   ylab="Survival probability")
> title("Progression-free Survival in Gastric Cancer Patients")
```

Figure 3.3 shows the survival plot.

3.2 Finding the Median Survival and a Confidence Interval for the Median

Formally, the median survival time may be defined as $\hat{t}_{\text{med}} = \inf\left\{t : \hat{S}(t) \leq 0.5\right\}$; that is, it is the smallest t such that the survival function is less than or equal to 0.5. To find a $1-\alpha$ confidence interval for the median, we consider the following inequality:

$$-z_{\alpha/2} \leq \frac{g\left\{\hat{S}(t)\right\} - g(0.5)}{\sqrt{\operatorname{var}\left[g\left\{\hat{S}(t)\right\}\right]}} \leq z_{\alpha/2}$$

where $g(u) = \log[-\log(u)]$ and $\text{var}\left[g\left\{\hat{S}(t)\right\}\right]$ is given by Eq. 3.1.3. For details, see Barker [5]. To obtain a 95 % confidence interval, we search for the smallest value of t such that the middle of the expression is at least -1.96 (for the lower limit) and the largest value of t such that the middle expression does not exceed 1.96 (for the upper limit). By default, the "survfit" function prints out 95 % confidence limits for the median. To obtain the median survival time for the gastric cancer data, and a 95 % confidence interval, just enter the result of the "survfit" function:

```
> result.km

records   n.max n.start   events   median 0.95LCL 0.95UCL
  48.00   48.00   48.00    32.00    10.30    5.79   15.27
```

Here we see that the median PFS time is 10.30 months, and a 95 % confidence interval ranges from 5.79 to 15.27 months. The median and associated 95 % confidence interval are illustrated in Fig. 3.4. If the upper limit of the pointwise 95 % confidence interval were above the red line, the upper limit would be undefined; if the survival curve itself were entirely above this red line, the median survival would also be undefined.

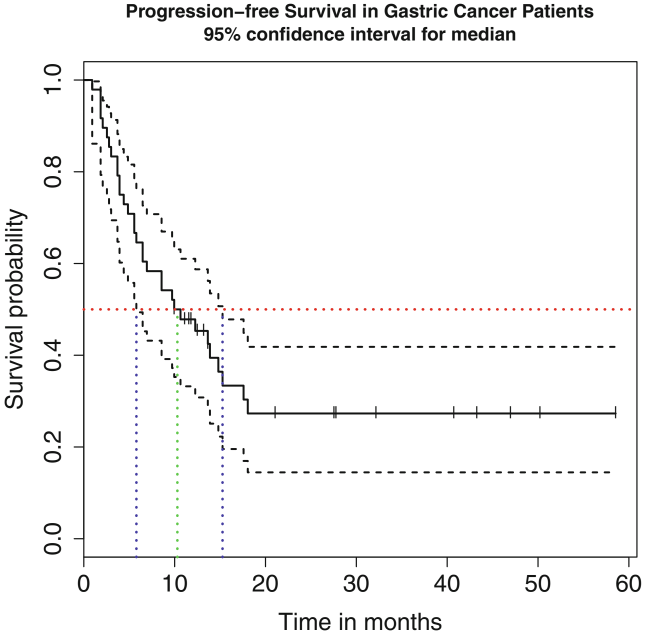


Fig. 3.4 The median is indicated by the *vertical green line* at 10.3 months, which intersects the survival curve estimate at 0.5. The 95 % confidence interval is indicated by the *vertical blue lines* at 5.79 and 10.27; they intersect the lower and upper survival curve confidence limits at 0.5

3.3 Median Follow-Up Time

One measure of the quality of a clinical trial is the duration of follow-up, as measured by the median follow-up time. This is a measure that captures how long, on average, patients have been followed. But defining this median is not straightforward. A simple definition is to consider all of the survival times, whether censored or not, and find the median. A disadvantage of this is that a trial with many early deaths, but a long observation period, would appear not to have a long median follow-up time. A perhaps better way of looking at median survival is the "potential" median survival. To obtain this estimate, one first switches the censoring and death indicators, so that a "censored" observation is the "event", while a death is viewed as a censored observation, in the sense that the observation time would have been much longer had the patient not died. One then computes the Kaplan-Meier "survival" estimate using these reversed censoring indicators, and finds the median survival, as discussed in the previous section. This method is also known as the "reverse" Kaplan-Meier [59]. We may find these two estimates of the median follow-up time for the "gastricXelox" data as follows:

```
> delta.followup <- 1 - delta
> survfit(Surv(timeMonths, delta.followup) ~ 1)

records   n.max n.start  events  median 0.95LCL 0.95UCL
   48.0    48.0    48.0    16.0    27.8    21.1    50.2

> median(timeMonths)
[1] 9.950413
```

The simple median follow-up time is only 9.95 months, whereas the potential follow-up time is 27.8 months.

3.4 Obtaining a Smoothed Hazard and Survival Function Estimate

In some applications we may wish to examine the hazard function in addition to the survival curve. The hazard function at the i'th failure time is d_i/n_i, the number of deaths at that time divided by the number at risk at that time. In fact, the Nelson-Altschuler estimate of the cumulative hazard function at time t_i, given in the previous section, is the sum of these hazard estimates up to that time. Unfortunately, this estimate of the hazard function is quite unstable from one time to the next, and thus is of limited value in illustrating the true shape of the hazard function. A better way to visualize the hazard function estimate is by using a "kernel" smoother [22, 30, 52]. A kernel is a function $K(u)$, which we center at each failure time. Typically we choose a smooth-shaped kernel, with the amount of smoothing controlled by a parameter b. The estimate of the hazard function is given by

$$\hat{h}(t) = \frac{1}{b}\sum_{i=1}^{D} K\left(\frac{t - t_{(i)}}{b}\right)\frac{d_i}{n_i} \tag{3.4.1}$$

where $t_{(1)} < t_{(2)} < \cdots < t_{(D)}$ are distinct ordered failure times, the subscript "(i)" in $t_{(i)}$ indicates that this is the i'th ordered failure time, d_i is the number of deaths at time $t_{(i)}$, and n_i is the number at risk at that time. Note that in the special case where the kernel function $K(u) = 1$ when u is a failure time and zero elsewhere, this estimator is just the Nelson-Altschuler hazard estimator. While there are many ways to define the kernel function, a common one is the Epanechnikov kernel, $K(u) = \frac{3}{4}(1 - u^2)$, defined for $-1 \le u \le 1$, and zero elsewhere. In the above formula for the hazard, there is one kernel function placed at each failure time, scaled by the smoothing parameter b. Larger values of b result in wider kernel functions, and hence more smoothing. This is illustrated in Fig. 3.5. Here the three failure times $t = 2, 4, 6$ are indicated by gray triangles, and the kernels, adjusted for height as in equation, are dashed gray. The sum, the smoothed estimate of the hazard, is given by the blue curve.

One problem with this simple approach to hazard estimation is that a kernel may put mass at negative times. In the above example, the first kernel function is centered at time $t = 2$, and it ranges from $t - b = 2 - 2.5 = -0.5$ to $t + b = 2 + 2.5 = 4.5$. Since the minimum time is 0, the actual area under the first kernel is too small. To correct for this, one may use a modified Epanechnikov kernel; for details, see Muller and Wang [52].

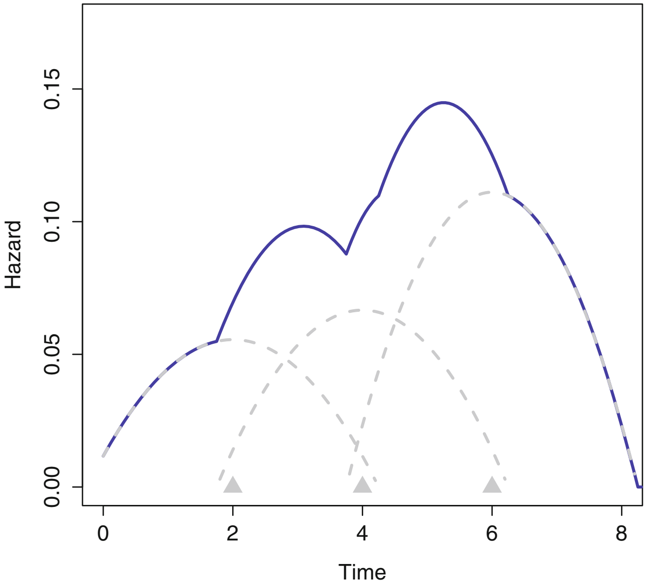


Fig. 3.5 Illustration of the hazard kernel smoother using the example data from Table 1.1 and the Kaplan-Meier estimate in Table 3.1

In the R package, there is a library "muhaz" for estimating and plotting nonparametric hazard functions. This package must be downloaded and installed into R. To reproduce the nonparametric curve in Fig. 3.5, use the function "muhaz" as in the following R code:

```
> library(muhaz)
> t.vec <- c(7,6,6,5,2,4)
> cens.vec <- c(0,1,0,0,1,1)
>
> result.simple <- muhaz(t.vec, cens.vec, max.time=8,
          bw.grid=2.25, bw.method="global", b.cor="none")
> plot(result.simple)
```

The first two arguments are the failure times and censoring indicators, respectively; the maximum time is set at 8; the smoothing parameter b is specified by "bw.grid=2.25"; the "global" option means that a constant smoothing parameter is use for all times; and the "b.cor" option is set to "none" indicating that no boundary correction is to be done.

We now illustrate estimation of the hazard function for the "gastricXelox" data. First, let us divide time into equal intervals of width 5 months, and observe the number of events (progression or death) d_i and the number of patients at risk each interval, n_i; the hazard estimate for that interval is $h_i = d_i/n_i$. The hazard estimate using this method may be obtained using the "pehaz" function:

```
result.pe5 <- pehaz(timeMonths, delta, width=5, max.time=20)
plot(result.pe5, ylim=c(0,0.15), col="black")
```

The resulting estimate is the solid step function in Fig. 3.6. In the same figure, we also present the step function for 1-month intervals:

```
result.pe1 <- pehaz(timeMonths, delta, width=1, max.time=20)
lines(result.pe1)
```

The "lines" function adds the step function to the same plot. The one-month hazard function jumps around quite a bit from one interval to the next, which limits its utility in visualizing the hazard function. For best results for visualizing the hazard function, we may compute a smooth hazard estimate using the following code:

```
result.smooth <- muhaz(timeMonths, delta, bw.smooth=20,
            b.cor="left", max.time=20)
lines(result.smooth)
```

Here we choose a smoothing parameter $b = 20$. The parameter "b.cor" is set to "left" to indicate that we want a boundary correction at the left, for small times t.

Selection of the appropriate amount of smoothing is one of the most difficult problems in non-parametric hazard estimation. If the bandwidth parameter is too small, the estimate may gyrate widely. Chose a parameter too wide and the hazard function may be too smooth to observe real variations in the hazard function over time. The "muhaz" function includes an automatic method for selecting a variable width bandwidth, so that for time regions with few events, a wider smoothing parameter is used than for time regions densely populated with events. To use this

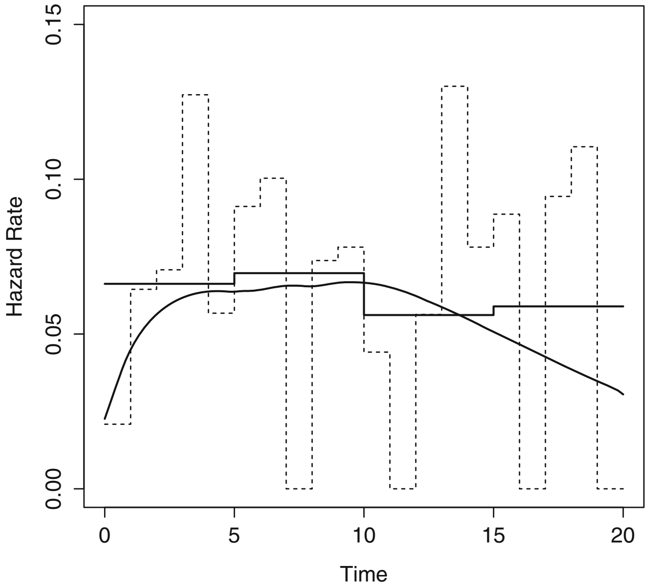


Fig. 3.6 Smoothed and step function estimates of the hazard function for the gastricXelox data

automatic variable bandwidth procedure, set the parameter “bw.option” equal to “local” instead of “global”. More information about the use of “pehaz” and “muhaz” may be obtained from the R help system.

One use of smoothing the hazard function is to obtain a smooth estimate of the survival function, using the relationship $\tilde{S}(t) = e^{-\int_{u=0}^{t} \hat{h}(u)\,du}$. To get this estimate, we need to extract the hazard estimate and list of times at which the hazard is estimated as follows:

```
haz <- result.smooth$haz.est
times <- result.smooth$est.grid
surv <- exp(-cumsum(haz[1:(length(haz)-1)]*diff(times)))
```

The survival curve estimation uses the “cumsum” function, which is a vector of the cumulative sum of the hazard estimates, and the “diff” function, which computes the widths (“differences”) of the vector “times”. Since the length of “diff(times)” is one less than the length of “times” and “haz”, we need to drop the last element of “haz”. This expression is a numerical evaluation of the integral, which works by adding up the area of the rectangles under the hazard curve. We may compare our smoothed survival estimate to the Kaplan-Meier estimate as follows:

```
result.km <- survfit(Surv(timeMonths, delta) ~ 1,
    conf.type="none")
plot(result.km, conf.int=T, mark="|", xlab="Time in months",
    xlim=c(0,30), ylab="Survival probability")
lines(surv ~ times[1:(length(times) - 1)])
```

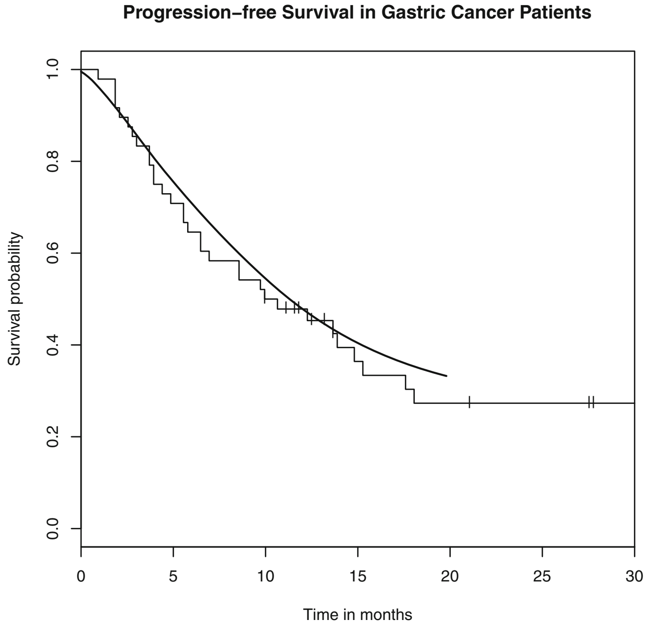


Fig. 3.7 Kaplan-Meier and smoothed survival curve estimate for the "gastricXelox" dataset

The smoothed hazard function follows the survival curve fairly well (Fig. 3.7). Only the first 30 months are shown here, because the smoothing procedure doesn't produce estimates beyond the last failure time. While certain specialized applications may require a smooth survival curve estimate, most published studies of survival data prefer to report the Kaplan-Meier step function estimate. This estimate has the theoretical property of being the maximum likelihood estimate of the survival function. In addition, the step function plot is an effective visual display of the data, in that it shows when the failures and censoring times occurred.

3.5 Left Truncation

While we have focused on right censoring as a type of incomplete data, there is another type of incompleteness, called "left truncation," which we are sometimes faced with. To understand left truncation, consider again the data from Table 1.1. Now, instead of examining the time from entry into the clinical trial until censoring or death, let us use as the time origin the time of diagnosis. The time from diagnosis to death (or censoring) may be of more practical interest than the time from entry into the trial to death. To get this additional information, we interview each patient

Table 3.3 Data from Table 1.1, with the addition of the time of diagnosis

Patient	Diagnosis	Survtime	Censor	SurvtimeDiag
1	−2	7	0	9
2	−5	6	1	11
3	−3	6	0	9
4	−3	5	0	8
5	−2	2	1	4
6	−5	4	1	9
X	−4	−2	1	

The time units are still the same, with time 0 indicating the time of entry into the trial and the time "Diagnosis" indicating the prior time of diagnosis. The new variable "SurvtimeDiag" denotes the time from diagnosis until censoring or death. The variables "Survtime" and "Censor" are as they were in Table 1.1. The new "Patient X" is a hypothetical patient with a short time from diagnosis until death. Practically speaking, such a patient is never observed; even if we somehow had a record of his diagnosis and early death, we could not possibly know for certain if that person would have entered the trial had he lived long enough. Such patients with short survival times are less likely to be enrolled in the trial than other patients, resulting in length-biased sampling

when he or she enters the trial to determine the time that the disease was diagnosed. The times between diagnosis and entry into the trial are known as the "backward recurrence times," and are given in Table 3.3. For example, Patient 1 was diagnosed 2 time units before entry into the trial, and was censored at time 7, which refers to the time from entry into the trial until censoring. Then the total time from diagnosis to censoring is $7 + 2 = 9$ time units. The data are plotted in Fig. 3.8.

Entry into the trial is still at time 0, but we have added diagnosis times, indicated by triangles. "Patient X," as discussed in Table 3.3.

We may realign so that the time of diagnosis is time 0, as shown in Fig. 3.9. Here, "Patient X" is no longer shown; such a patient would have died before he or she were able to register for the clinical trial, and thus would not have been observed. What are shown are times from diagnosis to death (or censoring), and "left truncation" times. Had a patient died during one of these intervals (denoted by dashed lines) that patient would not have been observed. To obtain an unbiased estimate of the survival distribution, we need to condition on the survival time being greater than the left truncation time. To do this, we construct the Kaplan-Meier estimator as we did earlier, but now a patient only enters the risk set at the left truncation time. Thus, unlike before, the size of the risk set can increase as well as decrease. For example, the first death is Patient 5, at time 4. at that time, patients 1, 3, 4, and 5 are in the risk set. After that patient dies, Patients 2 and 6 enter the risk set, and Patient 4 is censored at time 6. Thus, at time 9, then Patient 6 dies, patients 1, 3, and 6 are at risk (Table 3.4).

Fig. 3.8 Data from Table 1.1, now with diagnosis times

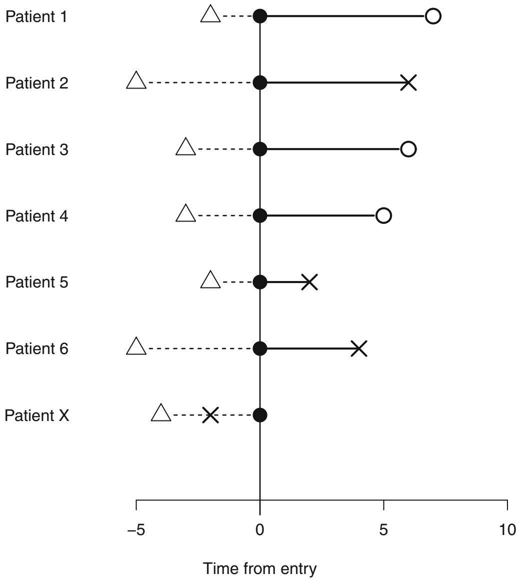
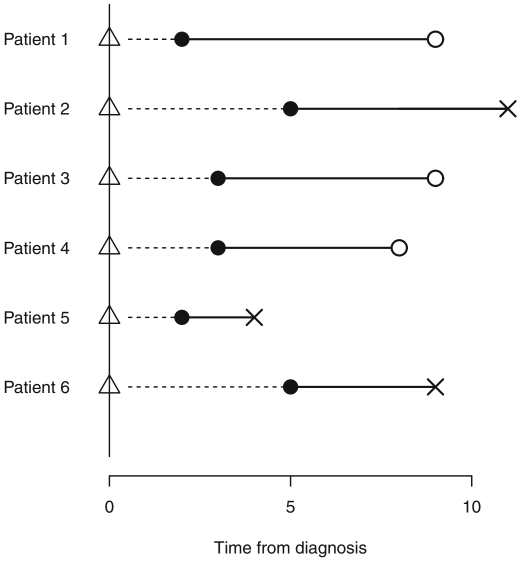


Fig. 3.9 Time from diagnosis to death. Entry into the clinical trial is denoted by *solid circles*. The *dashed lines* are "left truncation" times. Had the event occurred during these intervals, the patient would not have been observed



In R, we may obtain both estimates as follows:

```
> tt <- c(7, 6, 6, 5, 2, 4)
> status <- c(0, 1, 0, 0, 1, 1)
> backTime <- c(-2, -5, -3, -3, -2, -5)
> tm.enter <- -backTime
> tm.exit <- tt - backTime
> result.left.trunc.km <- survfit(Surv(tm.enter, tm.exit, status,
```

Table 3.4 Nelson-Altschuler estimate of the survival function for the data from Table 3.3

t_i	n_i	d_i	q_i	$1-q_i$	$S_i = \prod(1-q_i)$	$H(t) = \sum q_i$	$\hat{S}_{NAA}(t) = exp(-H(t))$
4	4	1	0.250	0.750	0.750	0.250	0.779
9	4	1	0.250	0.750	0.562	0.500	0.607
11	1	1	1.000	0.000	0.000	1.500	0.223

```
+    type="counting") ~ 1, conf.type="none")
> summary(result.left.trunc.km)

 time n.risk n.event entered censored survival std.err
    4      4       1       0        0    0.750   0.217
    9      4       1       0        2    0.562   0.230
   11      1       1       0        0    0.000     NaN

> result.left.trunc.naa <- survfit(Surv(tm.enter, tm.exit, status,
+    type="counting") ~ 1, type="fleming-harrington", conf.
     type="none")
> summary(result.left.trunc.naa)

 time n.risk n.event entered censored survival std.err
    4      4       1       0        0    0.779   0.225
    9      4       1       0        2    0.607   0.248
   11      1       1       0        0    0.223     Inf
```

We have used the terms “tm.enter” and “tm.exit” for the left truncation and survival times, respectively. The reason is derived from the counting process theory, where a subject “enters” the observation period at a particular time and then “exits” it at the time of death or censoring; events that may occur outside of this observation period are not visible to us.

A serious problem arises with left-truncated data if the risk set becomes empty at an early survival time. Consider for example the Channing House data, “ChanningHouse”.[3] This data set contains information on 96 men and 361 women who entered the Channing House retirement community, located in Palo Alto, Californ. For each subject, the variable “entry” is the age (in months) that the person entered the Channing House and “exit” is the age at which the person either died, left the community, or was still alive at the time the data were analysed. The variable “cens” is 1 if the patient had died and 0 otherwise. This data is subject to left truncation because subjects who die at older ages are more likely to have enrolled in the center than patients who died at younger ages. Thus, to obtain an unbiased estimate of the age distribution, it is necessary to treat “entry” as a left truncation time. The following code shows the first few records in the data set, converts “entry” and “exit” from months to years, and selects the men only:

[3] The data set “ChanningHouse” is included in the “asaur” package. It contains the cases in “channing” in the “boot” package, but with five cases removed for which the recorded entry time was later than the exit time.

```
> head(ChanningHouse)
   sex entry exit time cens
1 Male   782  909  127    1
2 Male  1020 1128  108    1
3 Male   856  969  113    1
4 Male   915  957   42    1
5 Male   863  983  120    1
6 Male   906 1012  106    1

> ChanningHouse <- within(ChanningHouse, {
+   entryYears <- entry/12
+   exitYears <- exit/12})
> ChanningMales <- ChanningHouse[ChanningHouse$sex == "Male",]
```

Next we estimate the survival distribution for men using first the Kaplan-Meier estimate and then the Nelson-Altschuler-Aalen estimator, and plot them. In the following code, the function "Surv" combines the left truncation time, the death (or censoring) time, and the censoring variable into a single survival variable.

```
result.km <- survfit(Surv(entryYears, exitYears, cens,
   type="counting") ~ 1, data=ChanningMales)
plot(result.km, xlim=c(64, 101), xlab="Age",
   ylab="Survival probability", conf.int=F)
result.naa <- survfit(Surv(entryYears, exitYears, cens,
   type="counting") ~ 1, type="fleming-harrington",
   data=ChanningMales)
lines(result.naa, col="blue", conf.int=F)
```

The plot is shown in Fig. 3.10. The black curve is the Kaplan-Meier estimate; it plunges to zero at age 65 because, at this early age, the size of the risk set is small, and in fact reduces to 0. This forces the survival curve to zero. And, since the Kaplan-Meier curve is a cumulative product, once it reaches zero it can never vary from that. The NAA estimate, shown in blue, is based on exponentiating a cumulative sum, so it doesn't share this problem of going to zero early on. Still, it does take an early plunge, also due to the small size of the risk set at the younger ages. The problem here is that there is too little data to accurately estimate the overall survival distribution of men.

Instead, we can condition on men reaching the age of 68, using the "start.time" option, and estimate the survival among that cohort:

```
> result.km.68 <- survfit(Surv(entryYears, exitYears, cens,
+    type="counting") ~ 1, start.time=68, data=ChanningMales)
> lines(result.km.68, col="green", conf.int=F)
> legend("topright", legend=c("KM", "NAA", "KM 68 and older"),
 +   lty=1, col=c("black", "blue", "green"))
```

This survival curve, shown in green, is much better behaved. So the only solution to the problem of a small risk set with left-truncated data is to select a realistic target (here, survival of men conditional on living to age 68) for which there is sufficient data to obtain a valid estimate.

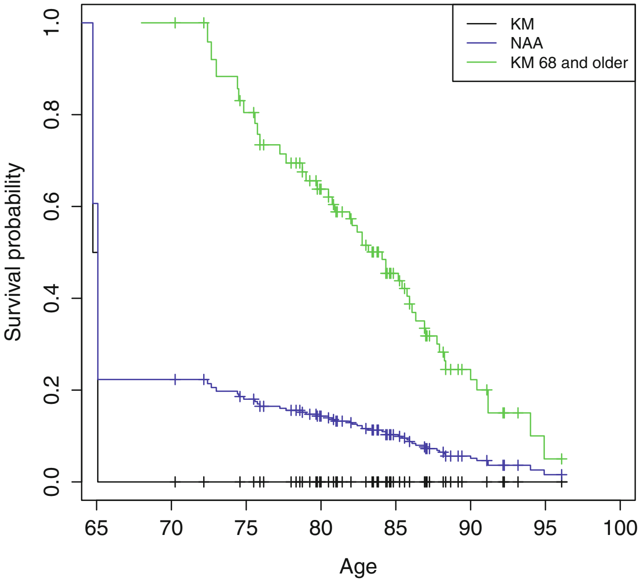


Fig. 3.10 Estimates of the survival (i.e. age at death) function for men entering the Channing House. The *black curve* is the Kaplan-Meier estimate, accounting for age at entry as a left truncation time, and the *blue curve* is the corresponding Nelson-Altschuler-Aalen estimator. The *green curve* is the Kaplan-Meier estimate, also accounting for left truncation, of the survival distribution conditional on living to age 68

3.6 Additional Notes

1. The "bshazard" function in the package of the same name provides an alternative method, based on B-splines, for finding smooth estimates of the hazard function. This function can accommodate left-truncated as well as right censored survival data.
2. We may estimate other percentiles and confidence intervals in a manner analogous to what we did for the median. Specifically, to estimate the p'th quantile, we find
   $\hat{t}_p = \inf\left\{t : \hat{S}(t) \leq 1 - p\right\}$. To get the 95 % confidence interval, we solve, for p,

$$-z_{\alpha/2} \leq \frac{g\left\{\hat{S}(t)\right\} - g(1-p)}{\sqrt{\operatorname{var}\left[g\left\{\hat{S}(t)\right\}\right]}} \leq z_{\alpha/2}$$

   where a good choice for the function $g(u)$ is the complementary log-log transformation.

3. As discussed in the text, the confidence bands for the survival curve are only valid at a pre-specified time point. Simultaneous confidence bands for an entire survival curve were developed by Hall and Wellner [27]; see also Kleinbaum and Klein [37]. This method was generalized by Matthews [47] and implemented by the same author in the R package "kmconfband".
4. Right truncation is another form of length-biased sampling, but it is much more difficult to accommodate than left truncation. See Lagakos et al. [39] for one methodological approach, and an application to estimating the latency time of HIV. Turnbull [73] discusses another approach based on the EM algorithm. The R package "DTDA" can estimate non-parametric survival curves for data with censoring and left and right truncation. This package also includes a copy of the HIV latency blood transfusion dataset used by Lagakos et al. [39].

Exercises

3.1. Refer to Fig. 3.2. Find the median survival, and a 95 % confidence interval for the median. Explain why the upper limit of the confidence interval is undefined.

3.2. In Fig. 3.3, find the first and third quartiles, and 95 % confidence intervals for these quartiles. If any of these quantities are undefined, explain.

3.3. Find a smooth hazard function estimate for the gastric cancer data using kernel width "bw.grid = 20". Explain reason for the multiple peaks in the estimate.

3.4. Estimate the survival distribution for men, conditional on reaching the age of 68, ignoring the left truncation times. Discuss the bias of this estimate by comparing to the estimate presented in Sect. 3.4.

Chapter 4
Nonparametric Comparison of Survival Distributions

4.1 Comparing Two Groups of Survival Times

Testing the equivalence of two groups is a familiar problem in statistics. Typically we are interested in testing a null hypothesis that two population means are equal versus an alternative that the means are not equal (for a two-sided test) or that the mean for an experimental treatment is greater than that for a standard treatment (one-sided test). We compute a test statistic from the observed data, and reject the null hypothesis if the test statistic exceeds a particular constant. The significance level of the test is the probability that we reject the null hypothesis when the null hypothesis is in fact true. A widely known test is the two-sample Students t-test for continuous observations, which requires the assumption that the observations are normally distributed. If the normal distribution assumption is in doubt, a rank-based test called the Mann-Whitney test may be used, which gives valid test results without making parametric assumptions. With survival data, if we are willing to assume that the data follow a particular parametric distribution, we can use likelihood theory to construct a test for equivalence of the two distributions, as we shall see in Chap. 10. However, as we have discussed in the previous chapters, survival data from biomedical experiments or clinical trials generally doesn't lend itself to analysis by parametric methods. Thus, we shall construct nonparametric tests of equivalence of two survival functions, $H_0 : S_1(t) = S_0(t)$. Typically, S_1 and S_0 will represent the survival distributions for, respectively, an experimental and a control therapy. Now, a statistical hypothesis test (in the classical hypothesis testing framework) also requires us to specify an alternative hypothesis, and one might at first try to specify a one-sided alternative $H_A : S_1(t) > S_0(t)$ or two-sided alternative $H_A : S_1(t) \neq S_0(t)$. Unfortunately, things aren't so simple in survival analysis, since the alternative can take a wide range of forms. What if the survival distributions are similar for some values of t and differ for others? What if the survival distributions cross? How do we want our test statistic to behave under these different scenarios? One solution is to consider what is called a Lehman alternative,

D. Moore, *Applied Survival Analysis Using R*, Use R!,
DOI 10.1007/978-3-319-31245-3_4

$H_A : S_1(t) = [S_0(t)]^{\psi}$. Equivalently, we can view Lehman alternatives in terms of proportional hazards as $h_1(t) = \psi h_0(t)$. Either way we would construct a one sided test as $H_0 : \psi = 1$ versus $H_A : \psi < 1$, so that under the alternative hypothesis $S_1(t)$ will be uniformly higher than $S_0(t)$ and $h_1(t)$ uniformly lower than $h_0(t)$ (i.e. subjects in Group 1will have longer survival times than subjects in Group 0). As we shall see, we can construct a test statistic using the ranks of the survival times. While these rank-based tests are similar to the Mann-Whitney test, the presence of censoring complicates the assignment of ranks. Thus, we initially take an alternative approach to developing this test, where we view the numbers of failure and numbers at risk at each distinct time as a two-by-two table. That is, for each failure time t_i we may construct a two-by-two table showing the numbers at risk (n_{0i} and $n_{1i}$for the control and treatment arms, respectively) and the number of failures (d_{0i} and d_{1i}, respectively). Also shown in the table are the "marginals", that is, the row and column sums. For example, we have $d_i = d_{0i} + d_{1i}$ and $n_i = n_{0i} + n_{1i}$. We first order the distinct failure times. Then for the i'th failure time, we have the following table:

	Control	Treatment	
Failure	d_{0i}	d_{1i}	d_i
Non-failures	$n_{0i} - d_{0i}$	$n_{1i} - d_{1i}$	$n_i - d_i$
	n_{0i}	n_{1i}	n_i

Suppose that the numbers of failures in the control and treatment groups are independent. If one then *conditions* on the margins; that is, if one holds d_i, n_i, n_{0i}, and n_{1i} fixed, then the distribution of d_{0i} follows what is known as a *hypergeometric distribution.*

$$p(d_{0i}|n_{0i}, n_{1i}, d_i) = \frac{\binom{n_{0i}}{d_{0i}}\binom{n_{1i}}{d_{1i}}}{\binom{n_i}{d_i}} \tag{4.1.1}$$

where

$$\binom{n}{d} = \frac{n!}{d!(n-d)!}$$

represents the number of combinations of n items taken d at a time, and n-factorial is given by $n! = n(n-1)\cdots 2$. This probability mass function allows one to compute the probability of each possible table with the margins fixed. One way to better understand this distribution is to imagine an urn with n_{0i} blue balls and n_{1i} red balls. From the urn we draw, without replacement, d_i balls. The number of blue balls in our sample, d_{0i}, follows a hypergeometric distribution, assuming that there is no difference between treatments. The mean and variance are given by

$$e_{0i} = E(d_{0i}) = \frac{n_{0i}d_i}{n_i}$$

where $E(d_{0i})$ is the expected value of d_{0i}, and

$$v_{0i} = \text{var}(d_{0i}) = \frac{n_{0i} n_{1i} d_i (n_i - d_i)}{n_i^2 (n_i - 1)}$$

We may sum up over all the tables the differences between the observed and expected values to get a linear test statistic U_0, and also the sum of the variances V_0 as follows, where N is the number of subjects:

$$U_0 = \sum_{i=1}^{N} (d_{0i} - e_{0i}) = \sum d_{0i} - \sum e_{0i}$$

$$\text{var}(U_0) = \sum v_{0i} = V_0$$

Then we may construct a test statistic that is standard normal,

$$\frac{U_0}{\sqrt{V_0}} \sim N(0,1)$$

or equivalently we may use the square of that to get a chi-square random variable with one degree of freedom,

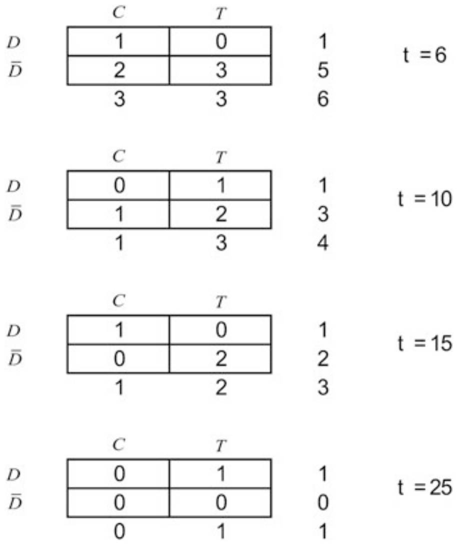
$$\frac{U_0^2}{V_0} \sim \chi_1^2$$

This test is known as the log-rank test. We illustrate it's calculation in the following example.

Example 4.1. Consider a hypothetical comparative clinical trial with six subjects assigned to either a control or treatment group. The survival data for the control group are 6, 7+, and 15, and for the treatment group they are 10, 19+, and 25 (Table 4.1). In tabular form, with the survival times in increasing order, we have where "C" denotes a control patient and "T" denotes a treatment patient. Since there

Table 4.1 Survival data

Patient	Survtime	Censor	Group
1	6	1	C
2	7	0	C
3	10	1	T
4	15	1	C
5	19	0	T
6	25	1	T



	C	T		
D	1	0	1	t =6
$\bar{D}$	2	3	5	
	3	3	6	

	C	T		
D	0	1	1	t =10
$\bar{D}$	1	2	3	
	1	3	4	

	C	T		
D	1	0	1	t =15
$\bar{D}$	0	2	2	
	1	2	3	

	C	T		
D	0	1	1	t =25
$\bar{D}$	0	0	0	
	0	1	1	

Fig. 4.1 Example data expressed as a series of two-by-two tables

t_i	n_i	d_i	n_{0i}	d_{0i}	n_{1i}	d_{1i}	e_{0i}	v_{0i}
6	6	1	3	1	3	0	0.500	0.2500
10	4	1	1	0	3	1	0.250	0.1875
15	3	1	1	1	2	0	0.333	0.2222
25	1	1	0	0	1	1	0.000	0.0000
				2			1.083	0.6597

are four distinct failure times, we may express this data set as a series of four two-by-two tables, where D indicates failure and $\bar{D}$, or "not D", indicates a non-failure, as shown in Fig. 4.1.

In tabular form, data and calculations of the log-rank test statistic are as follows: We have $U_0 = \sum d_{0i} - \sum e_{0i} = 2 - 1.083 = 0.917$, $V_0 = \sum v_{0i} = 0.6597$, and finally the log-rank statistic $X^2 = U_0^2/V_0 = 1.27$, which we compare to a chi-square distribution with one degree of freedom. Using the function "survdiff" in the R "survival" package, we obtain the same value of the chi-square statistic (which is rounded to 1.3 in the last row of the output):

```
> tt <- c(6, 7, 10, 15, 19, 25)
> delta <- c(1, 0, 1, 1, 0, 1)
> trt <- c(0, 0, 1, 0, 1, 1)
> survdiff(Surv(tt, delta) ~ trt)

      N Observed Expected (O-E)^2/E (O-E)^2/V
trt=0 3        2     1.08     0.776      1.27
trt=1 3        2     2.92     0.288      1.27

 Chisq= 1.3  on 1 degrees of freedom, p= 0.259
```

The p-value is 0.259, indicating that the group difference is not statistically significant (which is not surprising due to the extremely small sample size in this illustration). Also given in the output, for the first row (trt = 0), $\sum d_{0i} = 2$, $\sum e_{0i} = 1.083$, and in the last column, the chi-square statistic 1.27. The second row gives the corresponding results for the group trt=1. Due to symmetry, we get the same value (1.27) for the chi-square statistic calculated using the observed and expected quantities in the treatment group as we did for the control group.

Interestingly, the log-rank statistic is identical to a classical test statistic from epidemiology, the Cochran-Mantel-Haenzel test [2]. This is a test for independence of two factors (here, treatment and outcome) adjusted for a potential confounder, and is expressed as series of two-by-two tables with a time-stratified confounding factor. The log-rank test may also be derived from the proportional hazards model, as we will see in the next chapter.

An important generalization of this test makes use of a series of N weights w_i, with which we may define a weighted log-rank test by

$$U_0(w) = \sum w_i (d_{0i} - e_{0i})$$

and

$$\text{var}(U_0) = \sum w_i^2 v_{0i} = V_0(w).$$

The most common way of setting weights is given by the following expression, which uses the product-limit estimator from the combined sample, ignoring group:

$$w_i = N \left\{ \hat{S}(t_i) \right\}^{\rho}$$

A log-rank test using these weights is called the Fleming-Harrington $G(\rho)$ test [11]. If $\rho = 0$ this test is equivalent to the log-rank test, since then $w_i = n$ for all survival times t_i, and of course the constant n cancels out of the test statistics. If $\rho = 1$, we get what is often known as the Prentice modification (also known as the Peto-Peto modification) of the Gehan-Wilcoxon test. The effect of this test is then to place higher weight on earlier survival differences. The following example illustrates this.

Example 4.2. The data set “pancreatic” in the “asaur” package consists of pancreatic cancer data from a Phase II clinical trial where the primary outcome of interest is progression-free survival. As we saw in the previous chapter, this quantity is defined as the time from entry into a clinical trial until progression or death, whichever comes first. The data consist of, for each patient, the stage, classified as “LAPC” (locally advanced pancreatic cancer) or “MPC” (metastatic pancreatic cancer), the date of entry into the clinical trial, the date of death (all of the patients in this study died), and the date of progression, if that was observed before death. The first six observations are shown in this output,

```
> head(pancreatic)
    stage    onstudy progression      death
1     MPC 12/16/2005    2/2/2006 10/19/2006
2     MPC   1/6/2006   2/26/2006  4/19/2006
3    LAPC   2/3/2006    8/2/2006  1/19/2007
4     MPC  3/30/2006        <NA>  5/11/2006
5    LAPC  4/27/2006   3/11/2007  5/29/2007
6     MPC   5/7/2006   6/25/2006 10/11/2006
```

Patient #4, for example, died with no recorded progression (shown using the missing value indicator "NA"), so that person's PFS is time to death. For the five other patients in this list the PFS is time to the date of progression. Following is code to compute PFS for all 41 patients:

```
> attach(pancreatic)     # make the variable names accessible
>
> # convert the text dates into R dates
> Progression.d <- as.date(as.character(progression))
> OnStudy.d <- as.date(as.character(onstudy))
> Death.d <- as.date(as.character(death))
>
> # compute progression-free survival
>
> progressionOnly <- Progression.d - OnStudy.d
> overallSurvival <- Death.d - OnStudy.d
> pfs <- pmin(progressionOnly, overallSurvival)
> pfs[is.na(pfs)] <- overallSurvival[is.na(pfs)]
>
> # convert pfs to months
> pfs.month <- pfs/30.5
> # note that no observations are censored. This is advanced
    stage pancreatic cancer.
>
> plot(survfit(Surv(pfs.month) ~ stage), xlab="Time in months",
    ylab="Survival probability",
          col=c("blue", "red"), lwd=2)
> legend("topright", legend=c("Locally advanced", "Metastatic"),
    col=c("blue","red"), lwd=2)
```

(An alternative version of the data set, "pancreatic2", with PFS and over-all survival already computed, is also available in the "asaur" package.) The log-rank test may be fitted to this data as follows:

```
> survdiff(Surv(pfs) ~ stage, rho=0)

          N Observed Expected (O-E)^2/E (O-E)^2/V
stage=LA  8        8     12.3      1.49      2.25
stage=M  33       33     28.7      0.64      2.25
 Chisq= 2.2  on 1 degrees of freedom, p= 0.134
```

Here, the number of patients in each group equals the corresponding observed number of events, since there is no censoring. The value of chi-square statistics is 2.2 with 1 degree of freedom, and the p-value is 0.134, which is not statistically

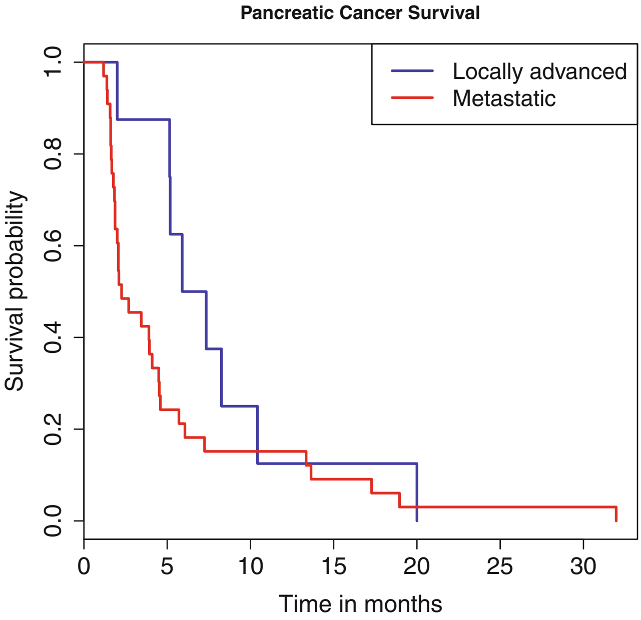


Fig. 4.2 Survival for pancreatic cancer patients with locally advanced or metastatic disease

significant. Here, we specified that $\rho = 0$. Since this is the default value, it is not necessary. If we use the Prentice modification, we must specify that $\rho = 1$:

```
> survdiff(Surv(pfs) ~ stage, rho=1)

           N Observed Expected (O-E)^2/E (O-E)^2/V
stage=LA   8     2.34     5.88     2.128      4.71
stage=M   33    18.76    15.22     0.822      4.71
 Chisq= 4.7  on 1 degrees of freedom, p= 0.0299
```

We obtain a p-value of 0.0299, which is statistically significant at the 5 % level. What changed is that this version of the test places higher weight on earlier survival times. From Fig. 4.2 we see that indeed the metastatic group shows an early survival advantage over the locally advanced group, but the survival curves converge after about 10 months. The reason for the difference is that these two tests, with $\rho = 0$ or 1, are optimized for different alternatives. We will return to this issue when we discuss time dependent covariates and non-proportional hazards.

4.2 Stratified Tests

If there is a need to compare two groups while adjusting for another covariate, there are two approaches one can use. One is to include the other covariate (or multiple covariates) as regression terms for the hazard function, an approach we will discuss in the next chapter. Alternatively, if the covariate we are adjusting for is categorical with a small number of levels G, we may construct a stratified log-rank test. This is a test of the null hypothesis $H_0 : h_{0j}(t) = h_{1j}(t)$ for $j = 1, 2, \ldots, G$. Essentially, for each level of the second variable, we compute a score statistic U_{0g} and variance V_{0g}, where $g = 1, \ldots, G$ is the group indicator. The test statistic is given by

$$X^2 = \frac{\left(\sum_{g=1}^{G} U_{0g}\right)^2}{\sum_{g=1}^{G} V_{0g}^2},$$

which (as for the unstratified log-rank statistic) may be compared to a chi-square distribution with one degree of freedom. Treatment center, age group, or gender are examples of variables on which we might need to stratify. As an example, let us consider the data set "pharmacoSmoking" in the "asaur" package, where the primary goal is to compare the time to relapse (defined in this study as return to smoking) between two treatment groups. We may compare the two groups using a log-rank test as follows:

```
> attach (pharmacoSmoking)
> survdiff(Surv(ttr, relapse) ~ grp)

                  N Observed Expected (O-E)^2/E (O-E)^2/V
grp=combination 61       37     49.9      3.36      8.03
grp=patchOnly   64       52     39.1      4.29      8.03

Chisq= 8  on 1 degrees of freedom, p= 0.00461
```

If we are concerned that the group comparison may differ by age, we may define a categorical variable, "ageGroup2", that divides the subjects into those 49 and under and those 50 and above. We may summarize this variable as follows:

```
> table(ageGroup2)
ageGroup2
21-49   50+
   66    59
```

The variable "ageGroup2" has two levels, with 66 patients in the 21-49 age group and 59 patients 50 years old and older. The log-rank test stratified on "ageGroup2" may be computed as follows:

```
> survdiff(Surv(ttr, relapse) ~ grp + strata(ageGroup2))

                  N Observed Expected (O-E)^2/E (O-E)^2/V
grp=combination 61       37     49.1      2.99      7.03
grp=patchOnly   64       52     39.9      3.68      7.03

Chisq= 7  on 1 degrees of freedom, p= 0.008
```

The chi-square test in this case differs only slightly from the unadjusted value, indicating that it was not necessary to stratify on this variable.

In the next example we illustrate the impact of a confounder.

Example 4.3. We shall set up a simulated data set from a clinical trial comparing a standard therapy (control) to an experimental therapy (treated). For simplicity, we suppose that the survival times are exponentially distributed, and that the disease is rapidly fatal, so that there is no censoring. We also suppose that there is a confounding variable, "genotype", which can either be wild type (i.e. normal) or mutant, and that patients carrying the mutant genotype have a considerably poorer prognosis. Specifically, we set the hazard rate for a mutant patient in the control group at 0.03 per day, and we assume that the effect of treatment is to reduce the hazard by a factor of 0.55. We also assume that the hazard rate for wild type patients is reduced by a factor of 0.2 as compared to mutant patients, and that the multiplicative effect of treatment on the wild type patients is the same as for the mutant patients. In R, we set up the four hazard rates as follows:

```
lambda.mutant.0 <- 0.03
lambda.mutant.1 <- 0.03*0.55
lambda.wt.0 <- 0.03*0.2
lambda.wt.1 <- 0.03*0.2*0.55
```

Next, we (1) set a "seed" for the random variable generator, so that this example may be reproduced exactly, (2) generate exponential random variables and string them together into the variable "ttAll", (3) create the censoring variable "status", and (4) create the treatment variable "trt" and genotype variable, as follows:

```
set.seed(4321)

tt.control.mutant <- rexp(25, rate=lambda.mutant.0)
tt.treat.mutant <- rexp(125, rate=lambda.mutant.1)
tt.control.wt <- rexp(125, rate=lambda.wt.0)
tt.treat.wt <- rexp(25, rate=lambda.wt.1)
ttAll <- c(tt.control.mutant, tt.treat.mutant, tt.control.wt,
    tt.treat.wt)

status <- rep(1, length(ttAll))

genotype <- c(rep("mutant", 150), rep("wt", 150))
trt <- c(rep(0, 25), rep(1, 125), rep(0, 125), rep(1, 25))
```

The survival plots comparing the two treatments is shown in left plot in Fig. 4.3, and appears to show that the treatment reduces survival. The log-rank test appears to confirm this with a very strong p-value:

```
> survdiff(Surv(ttAll, status) ~ trt)

         N Observed Expected (O-E)^2/E (O-E)^2/V
trt=0 150      150      183      6.00      15.9
trt=1 150      150      117      9.41      15.9

 Chisq= 15.9  on 1 degrees of freedom, p= 6.66e-05
```

However, when we plot the survival curves comparing treatment to control separately for the mutant and wild type patients (Fig. 4.3, right), we see that within

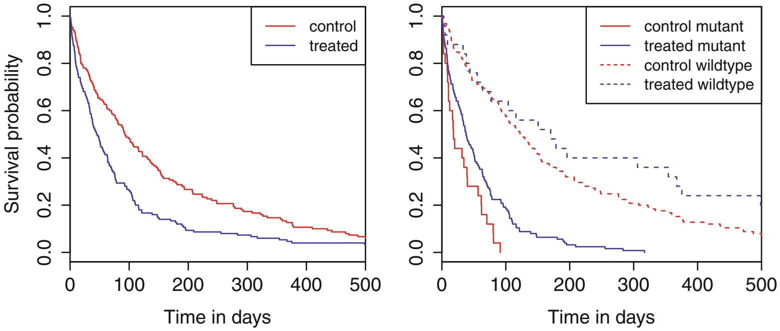


Fig. 4.3 Comparison of the Kaplan-Meier survival curves for two treatments ignoring the gene confounder (*left*) and accounting for it (*right*)

each genotype the treatment is actually superior to the control. We can confirm this using a stratified log-rank test, which shows the difference is highly significant:

```
> survdiff(Surv(ttAll, status) ~ trt + strata(genotype))

        N Observed Expected (O-E)^2/E (O-E)^2/V
trt=0 150      150      133      2.17      7.57
trt=1 150      150      167      1.73      7.57

 Chisq= 7.6  on 1 degrees of freedom, p= 0.00595
```

The output from the "survdiff" function does not make it clear which treatment is the superior one, so it is important to also consult the plot to ascertain the directional effect of treatment.

The explanation for the confounding is that (1) the treatment improves survival compared to the control, (2) patients carrying the wild type form of the gene have better survival than do patients carrying the mutation, and (3) there are more mutation-carrying patients in the treatment group than in the control group, whereas the reverse is true for wild type patients. Confounding of this type can easily arise in an observational study. For example, the frequency of mutants in one ethnic group may differ significantly from the frequency in the other, and at the same time one of the groups may have had more access to the experimental therapy than did the other. If the confounding factor can be observed, then it can be adjusted for, as we have seen.

4.3 Additional Note

1. The Gehan test, an adaptation of the Wilcoxon rank-sum test to censored data, is equivalent to a weighted rank test, with weights $w_i = n_i$, that is, each term is weighted by the number of subjects at risk at that time. The Prentice modification

of the Gehan test uses weights given by $w_i = n\hat{S}(t)$. These weights are similar to those of the Gehan test, but are more stable in small samples [11].

Exercises

4.1. Using the pharmacoSmoking data, compare the two treatments using the Prentice modification of the Gehan test, and compare your results to those from the log-rank test.

4.2. Again using the pharmacoSmoking data, carry out a log-rank test comparing the two treatments stratifying on employment status.

4.3. Using the "pancreatic" data set, which has no censored observations, compare the two groups using a Wilcoxon rank-sum test, using the "wilcox.test" function in base R. Compare your results to those from the log-rank and Prentice-modified Gehan tests.

4.4. Again using the "pancreatic" data set, compare the two groups using overall survival as the outcome, using both the log-rank test and the Prentice modification of the Gehan test. Do these two tests yield different results?

Chapter 5
Regression Analysis Using the Proportional Hazards Model

5.1 Covariates and Nonparametric Survival Models

In the previous chapter we saw how to compare two survival distributions without assuming a particular parametric form for the survival distributions, and we also introduced a parameter ψ that indexes the difference between the two survival distributions via the Lehmann alternative, $S_1(t) = [S_0(t)]^{\psi}$. Using Eq. 2.2.1 we can see that we can re-express this relationship in terms of the hazard functions, yielding the *proportional hazards assumption*,

$$h_1(t) = \psi h_0(t). \tag{5.1.1}$$

This equation is the key to quantifying the difference between two hazard functions, and the proportional hazards model is widely used. (Later we will see how to assess the validity of this assumption, and ways to relax it when necessary.) Furthermore, we can extend the model to include covariate information in a vector z as follows:

$$\psi = e^{z\beta}. \tag{5.1.2}$$

While other functional relationships between the proportional hazards constant ψ and covariates z are possible, this is by far the most common in practice. This proportional hazards model will allow us to fit regression models to censored survival data, much as one can do in linear and logistic regression. However, not assuming a particular parametric form for $h_0(t)$, along with the presence of censoring, makes survival modeling particularly complicated. In this chapter we shall see how to do this using what we shall call a *partial likelihood*. This modification of the standard likelihood was developed initially by D.R. Cox [12], and hence is often referred to as the *Cox proportional hazards model*.

D. Moore, *Applied Survival Analysis Using R*, Use R!,
DOI 10.1007/978-3-319-31245-3_5

5.2 Comparing Two Survival Distributions Using a Partial Likelihood Function

We begin our discussion of the partial likelihood by considering the simple case of comparing two groups of survival data. In Sect. 2.6 we constructed a likelihood based on the exponential distribution by taking a product of terms, one for each failure and each censoring time. We also saw that we could use the same procedure using other parametric distributions. But parametric distributions require strong assumptions about the form of the underlying survival distribution. The partial likelihood will allow us to use an unspecified baseline survival distribution to define the survival distributions of subjects based on their covariates. The partial likelihood differs from a likelihood in two ways. First, it is a product of expressions, one for each failure time, while censoring times do not contribute any factors. Second, the factors of a partial likelihood are conditional probabilities.

Let's fix some notation. We will use j to denote the j'th failure time (where the failure times are sorted from lowest to highest). The hazard function for Subject i at failure time t_j is $h_i(t_j)$. Under the proportional hazards model, we may write this hazard function as $h_i(t_j) = h_0(t_j)\psi_i$, and $\psi_i = e^{z_i\beta}$. Since we are now considering the very simple case of comparing a control and experimental group, the covariate z_i is either 1 (if the patient is in the experimental group) or 0 (of the patient is in the control group). Since patients in the experimental group are, we hope, less likely than control patients to experience the event, we expect that $\beta < 0$, and hence $\psi < 1$. In other words, $\psi_i = 1$ if a patient is in the control group or $\psi_i = \psi$ if that patient is in the experimental group.

Consider now the first failure time t_1. The set of all subjects in the trial "at risk" for failure at this time is denoted by R_1. (Just before the first failure, this set is comprised of all of the patients.) Among the patients in the risk set R_1, all are at risk of failure (i.e. of experiencing the event), and one of them, say Patient i, does fail. (We assume for now that there are no ties.) The probability that Patient i is the one who fails is the hazard, $h_i(t_1) = h_0(t_1)\psi_i$, for that patient divided by the sum of the hazards of all of the patients:

$$p_1 = \frac{h_i(t_1)}{\sum\limits_{k \in R_1} h_k(t_1)} = \frac{h_0(t_1)\psi_i}{\sum\limits_{k \in R_1} h_0(t_1)\psi_k} \tag{5.2.1}$$

where $h_0(t_1)$ is the hazard for a subject from the control group. The expression "$k \in R_1$" under the summation sign indicates that the sum is taken over all patients in the risk set R_1. A key fact here is that the baseline hazard $h_0(t_1)$ cancels out of the numerator and denominator, so that we have

$$p_1 = \frac{\psi_i}{\sum\limits_{k \in R_1} \psi_k}.$$

After the event at t_1, that patient drops out of the risk set R_1, as do any censored observations that occur after t_1 up to and including the second failure time t_2, resulting in a new (and smaller) risk set R_2. We then repeat this calculation to obtain p_2, and so on up to the last failure time. The partial likelihood is the product $L(\psi) = p_1 p_2 \cdots p_D$, assuming that there are D failure times. In each factor the baseline hazard cancels out of the numerator and denominator, so that it plays no role in the final partial likelihood.

Example 5.1. Let us consider again the synthetic data in Table 4.1. At time 0, there are six patients in the data set, all of which are at risk of experiencing an event. We call this group of patients the initial risk set R_1. As we can see in the first table in Fig. 4.1, just before the first failure time, $t = 6$, there are still six patients at risk, any one of which could experience the event. For our simple example, we have $\psi_1 = \psi_2 = \psi_4 = 1$ and $\psi_3 = \psi_5 = \psi_6 = \psi$. Substituting into Eq. 5.2.1 we have, for the event at time 6.

$$p_1 = \frac{1 \cdot h_0(t_1)}{3 \cdot h_0(t_1)\psi + 3 \cdot h_0(t_1)} = \frac{1}{3\psi + 3}$$

That is, there are six patients at risk, with six corresponding terms in the denominator. One of them fails, a control patients, so a "1" appears in the numerator. Note that $h_0(t_1)$ cancels out of the numerator and denominator. This cancellation is crucial since it removes parameters for the baseline survival distribution from the partial likelihood. The factor for the second failure time may be found in the same way. Of the six patients at risk at the first time, one dropped out because of a failure, and also a control patient dropped out at time 7 due to censoring. The factor for the second failure time, $t = 10$, is thus

$$p_2 = \frac{\psi}{3\psi + 1}.$$

Here, as in the first factor, the hazard h_{02} at the second failure time cancels out of the numerator and denominator. For the third failure time, $t = 15$, there are three patients at risk, one control and two treated. A control patient fails, so we have

$$p_3 = \frac{1}{2\psi + 1}.$$

Finally, at the last event time $t = 25$, there is only one subject at risk, who has the event, so the last factor is just 1. The partial likelihood is the product of these expressions,

$$L(\psi) = \frac{\psi}{(3\psi + 3)(3\psi + 1)(2\psi + 1)}.$$

Working with this function will be easier if we make the variable transformation $\psi = e^{\beta}$. Then we have

$$l(\beta) = \beta - \log\left(3e^{\beta} + 3\right) - \log\left(3e^{\beta} + 1\right) - \log\left(2e^{\beta} + 1\right).$$

The maximum partial likelihood estimate is the value of ψ that maximizes this function which, as we have said, is independent of the baseline hazard function $h_0(t)$. Notice that the particular values of the failure times do not contribute to this function; only the order matters. Also notice that, unlike a likelihood function, this partial likelihood is not a probability, since factors for censored times are not included. Still, one can treat this as if it were a likelihood, and find the maximum partial likelihood estimate of β. In R, we may do this by first defining the function $l(\beta)$:

```
plsimple <- function(beta) {
  psi <- exp(beta)
  result <- log(psi) - log(3*psi + 3) -
       log(3*psi + 1) - log(2*psi + 1)
  result   }
```

We may find the m.p.l.e. (maximum partial likelihood estimate) using the “optim” function. The control parameter “fnscale” is set to -1 so that optim will find the maximum of the function “plsimple”. (The default would be to find the minimum.)

```
> result <- optim(par=0, fn = plsimple, method = "L-BFGS-B",
+                 control=list(fnscale = -1),
+                 lower = -3, upper = 1)
> result$par
[1] -1.326129
```

Thus, the m.p.l.e. is $\hat{\beta} = -1.326129$. We may see this by plotting $l(\beta)$ versus β, as in Fig. 5.1:

The solid curved black line is a plot of the log partial likelihood over a range of values of β. The maximum is indicted by the vertical dashed blue line, and the value of the l.p.l. at that point is -3.672. Also shown is the value -4.277 of the l.p.l. at the null hypothesis value, $\beta = 0$. The tangent to the $l(\beta)$ curve at $\beta = 0$ is shown by the straight red line. Its slope is the derivative of the log-likelihood (i.e. the score function) evaluated at $\beta = 0$. Interestingly, this is exactly the value of the log-rank statistic U which we obtained in the previous chapter. This simple example illustrates a general principle: *The score function, obtained by taking the derivative of the log partial likelihood, evaluated at $\beta = 0$, is equivalent to the log-rank statistic*. In Sect. 5.4 we will discuss the score test in more detail and also other tests derived from the partial likelihood.

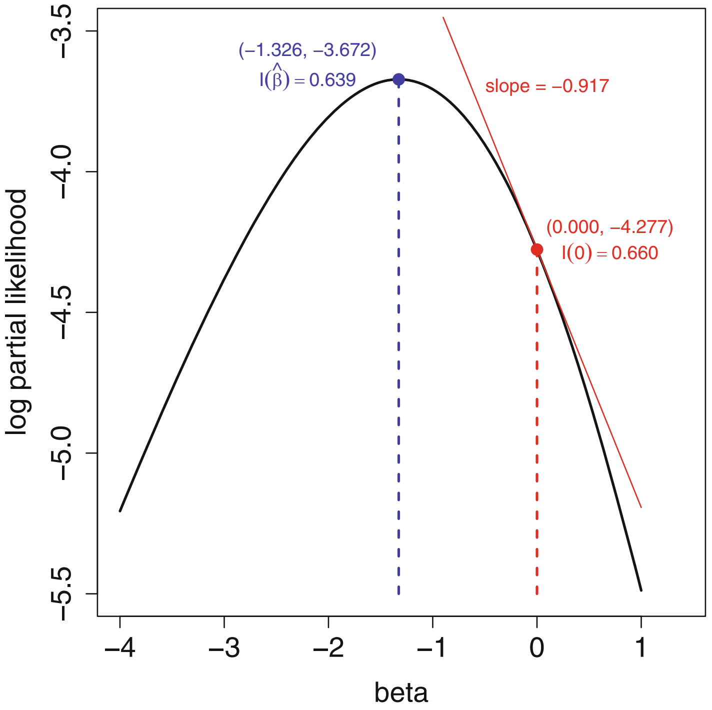


Fig. 5.1 Plot of the log partial likelihood function versus β. The maximum partial likelihood estimate is indicated by the *vertical dashed blue line*. The null hypothesis at $\beta = 0$ is indicated by the *vertical dashed red line*. The slope of the *red tangent line* is the value of the score statistic. The observed information values $I(\hat{\beta})$ and $I(0)$ are also given

5.3 Partial Likelihood Hypothesis Tests

In standard likelihood theory, one can derive three forms of the test of $H_0 : \beta = 0$: the Wald test, the score test, and the likelihood ratio test. In survival analysis, we may use the partial likelihood to derive these three tests, although the underlying statistical theory for the partial likelihood is far more complex than that for standard likelihood theory [19]. Often - but not always - the three tests yield similar results. To develop the tests, we need two functions derived from the partial log likelihood. The first, the score function, is the first derivative of the log likelihood, $S(\beta) = l'(\beta)$. The second function, the *information*, is minus the derivative of the score function, or equivalently minus the second derivative of the log-likelihood, $I(\beta) = -S'(\beta) = -l''(\beta)$. The second derivative $l''(\beta)$ is also known as the *Hessian*. With the substitution of the parameter estimate $\hat{\beta}$ into the information, we obtain the *observed information*.

5.3.1 The Wald Test

The Wald test is perhaps the most commonly used test, and carrying it out is straightforward from computer output. The test statistic is of the form $Z = \hat{\beta}/s.e.(\hat{\beta})$, where "s.e." stands for "standard error." In the previous section, we saw that $\hat{\beta}$ was the value of β that maximizes $l(\beta)$. We know from basic differential calculus that we may find the maximum by solving $S(\beta) = l'(\beta) = 0$ for β. Ordinarily this is a non-linear function that must be solved numerically using an iterative procedure. To find the variance of $\hat{\beta}$, we evaluate the information, $I(\hat{\beta}) = -l''(\hat{\beta})$. That is, the information (technically, the "observed information") is minus the second derivative of the partial likelihood, evaluated at $\hat{\beta}$. $I(\hat{\beta})$ is a measure of the curvature of the likelihood at $\hat{\beta}$. Intuitively, higher values of the curvature reflect a sharper curve, more "information", and lower variance. Lower curvatures, by contrast, corresponds to flatter curves and higher variance. The variance of $\hat{\beta}$ is approximately $1/I(\hat{\beta})$, and the standard error is $s.e.(\hat{\beta}) = 1/\sqrt{I(\hat{\beta})}$. We may use this to construct a normalized test statistic $Z_w = \hat{\beta}/s.e(\hat{\beta})$, and reject $H_0 : \beta = 0$ if $|Z_w| > z_{\alpha/2}$. We can also construct a $1 - \alpha$ confidence interval, $\hat{\beta} \pm z_{\alpha/2} \cdot s.e.(\hat{\beta})$. Equivalently, we can use the fact that the square of a standard normal random variable has a chi-square distribution with one degree of freedom, and reject the null hypothesis if $Z_w^2 > \chi^2_{\alpha,1}$.

5.3.2 The Score Test

The score function is the first derivative of the partial log-likelihood, $S(\beta) = l'(\beta)$. The variance of the score statistic is $I(\beta)$. We evaluate the score and information at the null hypothesis value of β, normally $\beta = 0$. The test statistic is $Z_s = S(\beta = 0)/\sqrt{I(\beta = 0)}$, and we reject $H_0 : \beta = 0$ if $|Z_s| > z_{\alpha/2}$, or equivalently if $Z_s^2 > \chi^2_{\alpha,1}$. The score test is equivalent to the log-rank test, as we saw in the previous section. This test can be carried out without finding the maximum likelihood estimate $\hat{\beta}$.

5.3.3 The Likelihood Ratio Test

The likelihood ratio test uses the result from statistical theory that $2\left[l(\beta = \hat{\beta}) - l(\beta = 0)\right]$ follows approximately a chi-square distribution with one degree of freedom. The key advantage of this test over the other two is that it is invariant to monotonic transformations of β. For example, whether the test is computed in terms of β or in terms of $\psi = e^{\beta}$ has no effect at all on the p-value for testing $H_0 : \beta = 0$.

We may illustrate these three tests using the simple data of Example 5.1.

Example 5.1 (continued). Let us continue with this simple data set we discussed in Sect. 5.2. We begin by presenting the output from the "coxph" function. The result is put into a data structure called "result.cox", and a complete summary of the results we obtain using the "summary" function,

```
 1  > result.cox <- coxph(Surv(tt, status) ~ grp)
 2  > summary(result.cox)
 3  Call: coxph(formula = Surv(tt, status) ~ grp)
 4
 5    n= 6, number of events= 4
 6
 7          coef exp(coef) se(coef)      z Pr(>|z|)
 8  grp -1.3261    0.2655   1.2509 -1.06    0.289
 9
10      exp(coef) exp(-coef) lower .95 upper .95
11  grp    0.2655      3.766   0.02287     3.082
12
13  Concordance= 0.7  (se = 0.187 )
14  Rsquare= 0.183   (max possible= 0.76 )
15  Likelihood ratio test= 1.21  on 1 df,   p=0.2715
16  Wald test            = 1.12  on 1 df,   p=0.2891
17  Score (logrank) test = 1.27  on 1 df,   p=0.2591
```

We will explain the computations of the estimates and test statistics as follows. We use the associated log partial likelihood function "plsimple" that we created in Sect. 5.2. We may compute the derivative of the log-likelihood (i.e. the score) evaluated at $\beta = 0$ numerically using the "gradient" function in the package "numDeriv" (which must be separately downloaded and installed),

```
> library(numDeriv)
> grad(func=plsimple, x=0)
[1] -0.917
```

The result -0.917 is thus the score evaluated at the null hypothesis, and is the slope of the red tangent line in Fig. 5.1. To carry out the score test, we also need the information, which we obtain using the "hessian" function as follows:

```
> hessian(func=plsimple, x=0)
[,1] [1,] -0.660
```

This is the curvature of the log-likelihood at the point where the tangent touches the log-likelihood in Fig. 5.1. The score test statistic, expressed as Z_s^2, is the square of the score at $\beta = 0$ divided by minus the hessian (information), also at $\beta = 0$, as follows:

$$(-0.917)^2/0.660 = 1.274.$$

This is the result given on line 17 of the summary output. We compare this to a chi-square distribution with one degree of freedom. The score test p-value is given by the upper tail,

```
> pchisq(1.274, df=1, lower.tail=F)
[1] 0.259
```

This score test p-value is also given on line 17.

To compute the Wald test, we need the maximum partial likelihood estimate, which we saw in Sect. 5.2 is $\hat{\beta} = -1.326129$. We also need the information at this point, which is the curvature at the peak of the curve in Fig. 5.1. We compute this as for the score test, but evaluated at $\hat{\beta}$. This is "result.cox$par", and here is the hessian for the Wald test:

```
> hessian(func=plsimple, x=result.cox$par)
          [,1]
[1,] -0.639
```

The square root of minus the reciprocal of the hessian is the standard error,

```
> sqrt(1/0.639)
[1] 1.251
```

The parameter estimate and standard error are given on line 8. Finally, the Wald test statistic Z_w and two-sided p-value for the test are given by

```
> -1.326/1.251
[1] -1.060
> 2*pnorm(1.060, lower.tail=F)
[1] 0.289
```

These results may also be found on line 8. The square of the test statistic is 1.124, and this result may be found on line 16, along with the same Wald p-value of 0.289.

The likelihood ratio statistic is twice the difference in the log partial likelihoods evaluated at $\hat{\beta}$ and at 0:

```
> betahat <- result.cox$par
> 2*(plsimple(betahat) - plsimple(0))
[1] 1.209
```

In the figure, this is twice the vertical difference between the log-likelihood at $\hat{\beta}$ and at 0. This result may be found on line 15, along with the p-value derived from the chi-square distribution,

```
> pchisq(1.209, 1, lower.tail=F)
[1] 0.271
```

Two additional portions of the output are often useful. The statistic "r-squared" is an adaptation to survival analysis of the R^2 statistic from linear regression. Here it is defined as follows:

$$R^2 = 1 - \left(\frac{l(0)}{l(\hat{\beta})} \right)^{2/n}$$

and reflects the improvement in the fit of the model with the covariate compared to the null model. The "Concordance" is the C-statistic, a measure of the predictive discrimination of a covariate. See Harrell [28] for more details.

5.4 The Partial Likelihood with Multiple Covariates

We now develop in greater generality the partial likelihood we introduced in Sect. 5.2. We define the hazard ratio (relative to the baseline hazard) for subject i by $\psi_i = e^{z_i'\beta}$. As in the previous section, z_i is a vector of covariate values for subject i, and β is a vector of coefficients, with one coefficient for each covariate. The hazard ratio parameter could be written more completely as $\psi(z_i, \beta)$, but for we will generally use ψ_i for brevity. Just before the first failure time, all of the subjects are said to be "at risk" for failure, and among these, one will fail. The "risk set" is the set of all individuals at risk for failure, and is denoted by R_j. The partial likelihood is a product of terms, one for each failure time. For each factor, (i.e., for each i), the denominator is the sum of all risks in the risk set R_i (denoted by "$k \in R_j$") of $h_k = h_0\psi_k$ and the numerator is the hazard $h_i = h_0\psi_i$ for the individual in the risk set R_i who experienced the failure. As we saw previously, the baseline hazards $h_0(t_j)$ cancel out of all of the terms, as follows:

$$L(\beta) = \prod_{j=1}^{D} \frac{h_0(t_j)\psi_j}{\sum_{k\in R_j} h_0(t_j)\psi_k} = \prod_{j=1}^{D} \frac{\psi_j}{\sum_{k\in R_j} \psi_k} \tag{5.4.1}$$

This function is called a partial likelihood because it lacks factors for the censored observations. Nevertheless, it may be used as if it were a likelihood, an idea first proposed by Cox [12]. The log partial likelihood is as follows, using D to represent the number of deaths in the set $\mathscr{D}$:

$$l(\beta) = \sum_{j=1}^{D}\left[\log(\psi_j) - \log\left(\sum_{k\in R_j}\psi_k\right)\right] = \sum_{j=1}^{D} z_j'\beta - \sum_{i=1}^{D}\log\left(\sum_{k\in R_j} e^{z_k'\beta}\right)$$

The score function, which is the first derivative of $l(\beta)$, has p components, one for each of the p covariates. The l'th component is given by (recalling that $\log(\psi_j) = z_j'\beta$, and using the fact that $z_{jl} = \partial\log(\psi_j)/\partial\beta_l$)

$$S_l(\beta) = \frac{\partial l(\beta)}{\partial \beta_l} = \sum_{j=1}^{D}\left(z_{jl} - \frac{\sum_{k\in R_j} z_{jk}e^{z_j'\beta}}{\sum_{k\in R_j} e^{z_j'\beta}}\right).$$

As we will see in Chap. 7, we may view the score function as the sum of "residuals", each of which consists of the observed value z_{ij} of the covariate minus an "expected" value. In the special case where z_i is a single binary covariate, $S(\beta = 0)$ is the log-rank statistic.

To construct test statistics as we did in Sect. 5.3, we will need the second derivative of the log-likelihood with respect to all pairwise combinations of the k

covariates. Writing the score function as a vector with k components, we may define the *observed information* matrix as follows:

$$I(\beta; z) = -\frac{\partial^2 l(\beta)}{\partial\beta\,\partial\beta'} = -\frac{\partial S(\beta)}{\partial\beta}.$$

Also known as the *Hessian* matrix, this may be derived using standard differential calculus. The Wald, score, and likelihood ratio test statistics of $H_0 : \beta = 0$ are, respectively,

$$X_w^2 = \hat{\beta}' I(\hat{\beta}; z)\hat{\beta},$$

$$X_s^2 = S'(\beta = 0; z) \cdot I^{-1}(\beta = 0; z) \cdot S(\beta; z),$$

and

$$X_l^2 = 2\left\{l(\beta = \hat{\beta}) - l(\beta = 0)\right\}.$$

All three are, under the null hypothesis, asymptotically chi-square random variables with $k - 1$ degrees of freedom. We shall see specific examples of these tests in the next chapter.

5.5 Estimating the Baseline Survival Function

An estimate of the baseline hazard function is given by

$$h_0(t_i) = \frac{d_i}{\sum\limits_{j \in R_j} \exp(z_j\hat{\beta})}.$$

In the case of a single sample, with $\beta = 0$, this reduces to the Nelson-Altschuler-Aalen estimator in Eq. 3.1.4. The baseline survival function is

$$S_0(t) = \exp\left[-H_0(t)\right],$$

and an estimate may be obtained by estimating $H_0(t)$ as a cumulative sum of the estimated hazards $h_0(t_j)$ for $t_j \leq t$. This is the estimator of the survival function of an individual with all covariate values set to zero. For many cases this is not a desirable or even sensible estimate. For example, if one of the covariates is "age of onset", setting that to zero will not result in a baseline survival curve with any practical meaning. To find a survival curve for a particular covariate value z use

$$S(t|z) = \left[S_0(t)\right]^{\exp(z\hat{\beta})}.$$

In R the “basehaz” function will compute a cumulative baseline hazard function. Be sure to use the option “centered = F” to cause it to estimate the cumulative hazard at $\beta = 0$. The default is to estimate it at the mean of the covariates. This will often not make sense, particularly for categorical covariates such as treatment indicator, sex, or race.

5.6 Handling of Tied Survival Times

Tied survival time can arise in two ways. If the underlying data are continuous, ties may arise due to rounding. For example, survival data may be recorded in terms of whole days, weeks, or months. In this case, the ties are a form of incomplete data, in that the true times are not tied. The other way tied survival data may arise is when the survival times are genuinely discrete. For example, in an industrial setting, the survival time may represent the number of compressions of a device until it fails. Or, in a study of effectiveness of birth control methods, survival time may be represented by the number of menstrual cycles until pregnancy. We will consider these two cases separately. (If censoring times are tied with failure times, the general convention is to consider the failures to precede the censoring times, so that the censored individuals are still in the risk set at the time of failure.) We shall illustrate these two cases using a simple data set.

Example 5.2. Suppose that we are comparing a control to a treatment group, with control survival times 7+, 6, 6+, 5+, 2, and 4, and treated times 4, 1, 3+ and 1. In this data set there are four distinct failure times, with ties at the first failure time t $=$ 1 and at the third failure time t $=$ 4. If the underlying times are actually continuous, we use the proportional hazards model

$$h(t;z) = e^{z\beta} h_0(t)$$

where $z = 1$ or 0 for a treated or control patient, respectively.

The partial likelihood is then the product of four factors, one for each distinct failure time. At the first time, t $=$ 1, all 10 patients are at risk, and two of them, both from the treatment group, and either of those two patients may have failed first. The first factor of the partial likelihood may be represented as the sum of these two possibilities:

$$L_1(\beta) = \frac{e^\beta}{4e^\beta + 6} \cdot \frac{e^\beta}{3e^\beta + 6} + \frac{e^\beta}{4e^\beta + 6} \cdot \frac{e^\beta}{3e^\beta + 6}.$$

Since both events are in the treatment group, the form of the two terms is the same. At the second failure time, t $=$ 2, there are eight subjects at risk, two in the treatment group and six in the control group, and only one failure, a subject in the control group. The second factor is thus

$$L_2(\beta) = \frac{1}{2e^\beta + 6}.$$

At the third failure time, t = 4, there are six subjects at risk, and two failures, one in the treatment and one in the control group. The third factor is thus a sum of the two possible orderings of these two failures,

$$L_3(\beta) = \frac{1}{e^\beta + 5} \cdot \frac{e^\beta}{e^\beta + 4} + \frac{e^\beta}{e^\beta + 5} \cdot \frac{1}{5}.$$

The final factor is a constant. Thus, we may write the full partial likelihood as the product of these three terms. Since this method essentially averages over an enumeration all possible orderings of the tied failure times, we refer to this method as the marginal method for ties.

If the times are in fact discrete, and the tied survival times are true ties, then we may model these using the discrete logistic model,

$$\frac{h(t;z)}{1 - h(t;z)} = e^{z\beta} \frac{h_0(t)}{1 - h_0(t)}.$$

At the first failure time, t = 1, there are $\binom{10}{2} = 45$ possible pairs that could represent the two failures. We may enumerate these possibilities by listing the proportionality terms as rows and columns, and the products as the lower diagonal as in Fig. 5.2.

$$\begin{array}{c|cccccccccc}
 & e^\beta & e^\beta & e^\beta & e^\beta & 1 & 1 & 1 & 1 & 1 & 1 \\
\hline
e^\beta & \bullet & & & & & & & & & \\
e^\beta & e^{2\beta} & \bullet & & & & & & & & \\
e^\beta & e^{2\beta} & e^{2\beta} & \bullet & & & & & & & \\
e^\beta & e^{2\beta} & e^{2\beta} & e^{2\beta} & \bullet & & & & & & \\
1 & e^\beta & e^\beta & e^\beta & e^\beta & \bullet & & & & & \\
1 & e^\beta & e^\beta & e^\beta & e^\beta & 1 & \bullet & & & & \\
1 & e^\beta & e^\beta & e^\beta & e^\beta & 1 & 1 & \bullet & & & \\
1 & e^\beta & e^\beta & e^\beta & e^\beta & 1 & 1 & 1 & \bullet & & \\
1 & e^\beta & e^\beta & e^\beta & e^\beta & 1 & 1 & 1 & 1 & \bullet & \\
1 & e^\beta & e^\beta & e^\beta & e^\beta & 1 & 1 & 1 & 1 & 1 & \bullet
\end{array}$$

Fig. 5.2 Computation of partial likelihood term for tied discrete failure times

The numerator of the first partial likelihood factor is $e^{2\beta}$ since both of the subjects who failed at this time were in the treatment group. The denominator is the sum over all possible pairs:

$$L_1(\beta) = \frac{e^{2\beta}}{6e^{2\beta} + 24e^{\beta} + 15}.$$

The second factor is the same as it was previously,

$$L_2(\beta) = \frac{1}{2e^{\beta} + 5}.$$

For the third failure time, there are $\binom{6}{2} = 15$ possible pairs, of which one is from the treatment group and one from the control group. So the numerator is $e^{\beta} \cdot 1$ and the denominator has 15 terms,

$$L_3(\beta) = \frac{e^{\beta} \cdot 1}{5e^{\beta} + 10}$$

and the partial likelihood using this method is, of course, the product of these three factors. We shall call this method the exact discrete method.

We may enter this data set into R as follows:

```
> tt <- c(7, 6, 6, 5, 2, 4, 4, 1, 3, 1)
> status <- c(0, 1, 0, 0, 1, 1, 1, 1, 0, 1)
> grp <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1)
```

The partial log-likelihoods for the continuous exact and discrete exact may be defined as

```
loglikContinuous <- function(b) {
  result <- 3*b + log(exp(b) + 9) - log(4*exp(b) + 6) -
        log(3*exp(b) + 6) - log(2*exp(b) + 6) -
        log(exp(b) + 5) - log(exp(b) + 4)
 result
 }

loglikDiscrete <- function(b) {
  resultA <- exp(2*b)/(6*exp(2*b) + 24*exp(b) + 15)
  resultB <- 1/(6 + 2*exp(b))
  resultC <- exp(b)/(10+5*exp(b))
  result <- log(resultA) + log(resultB) + log(resultC)
  result
  }
```

We may find the maximum partial likelihood estimates using the "optim" function,

```
> result.optim.continuous <- optim(par=1.4, fn=loglikContinuous,
+   method="BFGS", control=list(fnscale = -1) )

> result.optim.discrete <- optim(par=1.4, fn=loglikDiscrete,
+   method="BFGS", control=list(fnscale = -1) )
```

The results are as follows:

```
> result.optim.continuous$par
[1] 1.838591

> result.optim.discrete$par
[1] 1.856719
```

We may compare these results to those from "coxph" with the "exact" method,

```
> result.coxph <- coxph(Surv(tt, status) ~ grp, ties="exact")
> result.coxph$coef
     grp
1.856768
```

The "exact" method in "coxph" corresponds to the discrete exact method, which typically will be similar in value to the marginal method.

Both of these methods require exhaustive enumeration for tied survival times, and they become computationally burdensome for data sets with more than a small number of tied observations. Fortunately, approximate methods are available. The first, and simplest, is the Breslow approximation, adjusts both terms of the marginal method so that they have the same denominator, corresponding to all subjects at risk. The first and third factors are just

$$L_1(\beta) = \frac{2e^{2\beta}}{\left(6e^{\beta} + 4\right)^2}$$

and

$$L_3(\beta) = \frac{2(1 \cdot e^{\beta})}{\left(e^{\beta} + 5\right)^2}.$$

(The second term is the same as before, as there are no ties.)

A more refined method is the Efron method, in which the Breslow method denominator is replaced by a better approximation,

$$L_1(\beta) = \frac{e^{\beta}}{\left(6e^{\beta} + 4\right)} \cdot \frac{e^{\beta}}{\left(0.5e^{\beta} + 0.5e^{\beta} + 4e^{\beta} + 4\right)}$$

and

$$L_3(\beta) = \frac{1}{\left(e^{\beta} + 5\right)^2} \cdot \frac{e^{\beta}}{\left(0.5 + 0.5e^{\beta} + 3\right)}.$$

At the first failure time, the denominator of the first factor contains terms for all 10 subjects at risk, while in the second factor it has the 8 subjects still at risk after the failures, plus one-half of each of the subjects that fail at that time. Intuitively, each of these subjects has a chance of one-half of being in the second denominator, since one of them would have been the first failure. Similarly, for the third failure time, the denominator of the second factor has the three subjects that do not fail, and one-half of each of the subjects that will fail; one of these is a control and one a treatment subject.

5.7 Left Truncation

In Sect. 3.5 we discussed how left-truncation can arise in a clinical trial setting when the question of interest is time from diagnosis (rather than time from enrollment) to death or censoring. The same considerations arise in a comparative clinical trial. To illustrate this, consider data from a hypothetical trial of six patients, three receiving an experimental treatment and three receiving a standard therapy. The time "tt" represents the time from entry into the trial until death or censoring, "status" indicates whether or not a death was observed, and "grp" indicates which group the patient is in. The time "backTime" refers to the backwards recurrence time, that is, the time before entry when the patient was diagnosed. We may enter the data into R as follows:

```
tt <- c(6, 7, 10, 15, 19, 25)
status <- c(1, 0, 1, 1, 0, 1)
grp <- c(0, 0, 1, 0, 1, 1)
backTime <- c(-3, -11, -3, -7, -10, -5)
```

The data are plotted in Fig. 5.3. The standard way to compare the two groups is to ignore the backwards recurrence times:

```
> coxph(Surv(tt, status) ~ grp)

     coef exp(coef) se(coef)      z    p
grp -1.33     0.266     1.25 -1.06 0.29

Likelihood ratio test=1.21  on 1 df, p=0.271  n= 6, number
     of events= 4
```

This result shows that the experimental group has a lower hazard than the control group, but this difference is not statistically significant (p-value = 0.271 based on the likelihood ratio test). There is nothing wrong with this standard and widely-used method; since there is no reason to believe that the backwards recurrence times would differ between the two groups, there should be no concern about bias. However, in some circumstances one may wish to compare survival times starting from time from diagnosis, and then it is essential to account for the left truncation. The data can be re-configured so that the diagnosis occurs at time 0 as follows:

```
tm.enter <- -backTime
tm.exit <- tt - backTime
```

These data are plotted in Fig. 5.4.

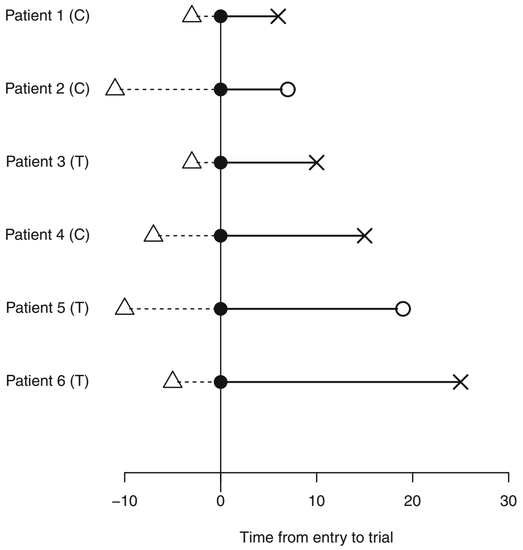


Fig. 5.3 Survival times and backwards recurrence times for data from a comparative clinical trial. Patients marked "T" received the experimental treatment, and those marked "C" received the standard therapy

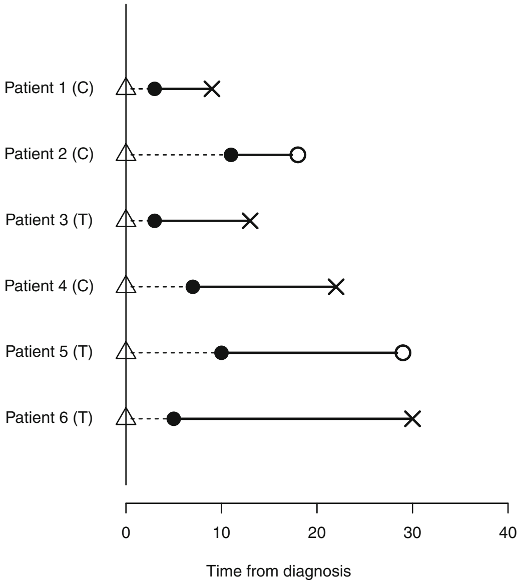


Fig. 5.4 Re-aligned data with left truncation

The left-truncated data may be compared as follows:

```
> coxph(Surv(tm.enter, tm.exit, status, type="counting") ~ grp)

     coef exp(coef) se(coef)      z    p
grp -1.07     0.342     1.24 -0.869 0.39

Likelihood ratio test=0.81  on 1 df, p=0.368
```

In this example, using the full survival times (from diagnosis) with left truncation leads to a similar non-significant treatment difference conclusion. (The option "type = 'counting' " is not required, since the "Surv" function will use it by default in this case.)

Another example is the Channing House data, which we discussed in Sect. 3.5. We may compare the survival of men and women, accounting for the different ages of entry. As before, we condition on subjects reaching the age of 68. We have to do this explicitly, since the "start.time" option we used previously is not available in the "coxph" function,

```
channing68 <- ChanningHouse[ChanningHouse$exitYears >= 68,]
```

Here are the results, which show that men have a higher hazard (and hence lower survival) than do women, but this difference is not statistically significant:

```
> coxph(Surv(entryYears, exitYears, cens, type="counting") ~ sex,
+    data=channing68)

          coef exp(coef) se(coef)    z    p
sexMale 0.273      1.31    0.176 1.55 0.12

Likelihood ratio test=2.3  on 1 df, p=0.129
```

Note that the variables "entryYears" and "exitYears" were defined in Sect. 3.5 and added to the "ChanningHouse" data set.

5.8 Additional Notes

1. An alternative R program for fitting the Cox proportional hazards model is "cph", in the "rms" package developed by Frank Harrell [28]. This function actually calls the main fitting program for the standard "coxph" function in the survival library, so that both functions produce identical results. The "cph" function adds additional options and is compatible with other routines in the "rms" package.
2. The theoretical properties of the partial likelihood using counting process theory were first elucidated by Aalen [1], For a full treatment, see for example Andersen et al. [3] and Fleming and Harrington [19].

Exercises

5.1. Consider the data set "aml", which is included in the "survival" package. This is a study of whether or not maintenance therapy increases survival among patients with acute myelogenous leukemia, with survival time measured in weeks. The basic Cox model may be fitted as follows:

```
result <- coxph(Surv(time, status) ~ x, data=aml)
```

Create a coarser time variable by expressing it in months instead of weeks as follows:

```
time.months <- cut(aml$time, breaks=seq(0,161,4), labels=F)
```

Now re-fit the model, modeling ties using the Breslow, Efron, and exact methods. Which approximate method gives a result closest to that from the exact method?

5.2. Consider again the synthetic data in Table 4.1, discussed in Example 5.1 in Sect. 5.2. Use the "basehaz" function to obtain an estimate of the baseline cumulative hazard function. Use this to compute the predicted survival curves for the control and experimental groups based on the proportional hazards model we fitted in Sect. 5.2.

Chapter 6
Model Selection and Interpretation

6.1 Covariate Adjustment

Survival analysis studies typically include a wealth of clinical, demographic, and biomarker information on the patients as well as indicators for a therapy or other intervention. If the study is a randomized clinical trial, the focus will be on comparing the effectiveness of different treatments. A successful randomization procedure should ensure that confounding covariates are balanced between the treatments. Still, we may wish to include such covariates in the model to adjust for any differences that may have arisen, and also to understand how these other factors affect survival. If the study is based on observational data, and if there is a primary intervention of interest, then adjustment for potential confounders is essential to obtaining a valid estimate of the intervention effect. The effect of other covariates on survival will also be of interest in such a study, and in some applications discovery and quantification of explanatory variables may be the primary goal. Regardless of the type of study, we will need methods to sift through a potentially large number of potential explanatory variables to find the important ones.

To illustrate the importance of covariate adjustment, let us again look at the simulated data in Example 4.3 of Chap. 4, which presented a study of the effect of treatment on survival in the presence of a genetic confounder. Here is a Cox proportional hazards model of the effect of treatment on survival unadjusted for the genetic mutation status of the patients:

```
> coxph(Surv(ttAll, status) ~ trt)

     coef exp(coef) se(coef)    z       p
trt 0.464      1.59    0.117 3.96 7.6e-05

Likelihood ratio test=15.5  on 1 df, p=8.2e-05
```

We see that the estimate of the log hazard ratio treatment effect, $\hat{\beta}$, is 0.464. Since this is positive, higher hazards are associated with the treatment than with

D. Moore, *Applied Survival Analysis Using R*, Use R!,
DOI 10.1007/978-3-319-31245-3_6

the control. That is, the treatment appears to reduce survival, which would be an unfortunate result. The value of $e^{\hat{\beta}} = 1.59$ is also given, suggesting (incorrectly, as we know) that the treatment is associated with a 59 % additional risk of death over the control. We can stratify on genotype, just as we did previously with the log-rank test, as follows:

```
> coxph(Surv(ttAll, status) ~ trt + strata(genotype))

      coef exp(coef) se(coef)     z      p
trt -0.453     0.636    0.164 -2.76 0.0058

Likelihood ratio test=7.66  on 1 df, p=0.00566
```

Now the coefficient is negative, indicating that, within each genotype, the treatment is effective. With the Cox model, we also have the option of explicitly estimating the genetic effect,

```
> coxph(Surv(ttAll, status) ~ trt + genotype)

              coef exp(coef) se(coef)     z      p
trt         -0.452     0.636    0.163 -2.77 0.0056
genotypewt  -1.568     0.209    0.183 -8.59 0.0000

Likelihood ratio test=93.4  on 2 df, p=0
```

Here we also see the correct treatment effect. We also see that the wild type genotype has lower hazard than the reference (mutant) genotype, and thus that the mutant genotype incurs additional risk of death.

6.2 Categorical and Continuous Covariates

The previous sections considered a partial likelihood for comparing two groups, indexed by a covariate z. Since z can take the values 0 or 1 depending on which of two groups a subject belongs to, this covariate is called an *indicator* or *dummy* variable. Typically in survival analysis, as in linear or logistic regression, we will want to include in our model a variety of types of patient data. In addition to group assignment for a randomized trial, we may have demographic information; examples might include the patient's age, gender, race, and income level. Furthermore, there may be clinical variables, such as blood measurements and disease stage indicators. All of this information will be encoded as covariates, some of which are continuous (e.g. age or blood pressure), and others which are categorical (e.g. gender or race). Categorical variables with only two levels can be handled with dummy variables as we did for treatment group. For gender, for example, we arbitrarily choose one gender, say males, as the reference group and code that with a zero. Then females would be coded with a one. With categorical variables with three or more variables, we will need multiple dummy variables. Suppose, for example, that the variable "race" has four levels, "white", "asian", "black", and "other". We first need to select one level as a reference level, to which all the others will be compared. This choice could be arbitrary, or driven by the goals of the research project. For example, if an important research question is how survival in non-white groups compares to survival in whites, one would select "white" as the reference variable. Since there are four levels, we need to create three dummy variables, say, z_2, z_3, and z_4 to represent "race". Then for a white patient, all three would take the value zero. For an Asian person, we would have $z_2 = 1$, and $z_1 = z_3 = 0$. For persons of race black or other, we make the corresponding assignments. In this model, at most one of the three dummy variables can be 1, and the others must be 0. (Dealing with persons of mixed race would be handled in a more complex way, certainly not by making more than one dummy variable take the value 1.)

Once we have settled on a set of k covariates, some of which are dummy variables and some continuous variables, we may write the model as follows:

$$\log(\psi_i) = z_{1i}\beta_1 + z_{2i}\beta_2 + \cdots + z_{ki}\beta_k.$$

For each covariate, the parameter β_j is the log hazard ratio for the effect of that parameter on survival, adjusting for the other covariates. For continuous covariates, it represents the effect of a unit change in the covariate; for dummy variables, it represents the effect of the corresponding level as compared to the reference covariate. We will write this in more compact form as $\log(\psi_i) = z_i'\beta$ (for Patient i), where z_i' (the transpose of z_i) is a $1 \times k$ matrix (i.e. a row matrix) of covariates, and β is a $k \times 1$ matrix (i.e. a column matrix) of parameters.

We may enhance this model in two ways. First, it is possible that a continuous variable is not linearly related to the log hazard. In that case, we may consider transforming it using, say, a logarithmic or square root function before entering it into the model. Or we can enter a variable twice, once as a linear variable and once as the square of that variable. Another choice is to "discretize" a variable. For example, an age variable could be split into three pieces, "under 50" and "50-64", and "65 and above" and entered into the model as a categorical variable.

The second enhancement to the model is to incorporate interaction terms. For example, suppose that gender and age do not contribute additively to the log hazard. Then one can directly enter into the model gender and age and also an interaction term constructed as the product of age and gender. Interactions with categorical variables with more than two levels are also possible. For example, the interaction of age with race (with four levels, say) would involve adding three terms composed of the product of age with the three race dummy variables.

While these models are similar to ones used in linear and logistic regression, there are also some key differences. For example, since survival data evolve over time, there is a possibility that some covariate values may also change as time passes. Initially, however, we require that all covariates are fixed at the beginning of the trial, and thus cannot change in response to evolving conditions. For example, if there are two treatment groups, the assignment to a group must be made at time 0, and not depend on anything that may happen later in the trial. Time-related variables such as age must also be defined by taking their value at the beginning of the trial. For example, in a clinical trial, “age” might be defined as the age at the time of randomization, so that it’s value is fixed even though (obviously) patients will age as the trial progresses. Later we will see how to modify the model to accommodate time-varying covariates.

Another way that proportional hazards models differ from those used in other types of regression is that there is no intercept term; if there were one, it would appear in both the numerator and denominator of the partial likelihood, and cancel out just as the baseline hazard canceled out. Another way to think of it is that any intercept term would be absorbed into the baseline hazard.

Example 6.1. Suppose that we have two black patients, two white patients, and two patients of other races, with ages 48, 52, 87, 82, 67, and 53, respectively. We may enter these data values as follows:

```
race <- factor(c("black", "black", "white", "white", "other",
           "other"))
age <- c(48, 52, 87, 82, 67, 53)
```

We use the function “factor” to convert a vector of character variables to one of type “factor”; this conversion will make it easier to incorporate this variable into statistical models. We may create a matrix of dummy variables for race and also a column for age using the “model.matrix” function as follows:

```
> model.matrix(~ race + age)[,-1]
  raceother racewhite age
1         0         0  48
2         0         0  52
3         0         1  87
4         0         1  82
5         1         0  67
6         1         0  53
```

Here we have removed the first column of the matrix (using the “−1” selection), since it is a column of 1s for the intercept. As explained above, in survival analysis, we do not include an intercept term. The first column contains indicators for “other race” and the second for “white race”; both are compared to “black race” here. If we need to use whites as the reference, we can change the race factor to have “white” as the reference level,

```
> race <- relevel(race, ref="white")
> model.matrix(~ race + age)[,-1]
  raceblack raceother age
1         1         0  48
2         1         0  52
3         0         0  87
4         0         0  82
5         0         1  67
6         0         1  53
```

In this example we have three covariates, say, z_1, z_2, and z_3, the first two of which are dummy variables for black race and other race, and the third a continuous variable, age. For the first subject, a black 48-year old person, the log hazard ratio is

$$\log(\psi_1) = z_{11}\beta_1 + z_{12}\beta_2 + z_{13}\beta_3 = 1 \times \beta_1 + 0 \times \beta_2 + 48 \times \beta_3.$$

Thus, β_1 represents the log hazard ratio for blacks as compared to whites, and β_3 represents the change in log hazard ratio that would correspond to a one-year change in age.

If we wish to include an interaction between race and age, we can express it as follows:

```
> model.matrix(~ race + age + race:age)[,-1]
   raceblack raceother age raceblack:age raceother:age
1          1         0  48            48             0
2          1         0  52            52             0
3          0         0  87             0             0
4          0         0  82             0             0
5          0         1  67             0            67
6          0         1  53             0            53
```

The interaction terms (last two columns) are just the product of the first two columns and the third (age) column.

To show how models are incorporated into a survival problem, we will generate a small survival data set in this example:

Example 6.2. We first generate 60 ages between 40 and 80 at random:

```
age <- runif(n=60, min=40, max=80)
```

Next, we set the race variable so that there are 20 of each category, and make "white" the reference category:

```
race <- factor(c(rep("white", 20), rep("black", 20),
          rep("other", 20)))
race <- relevel(race, ref="white")
```

The survival variables in our simulated data will be exponentially distributed with a particular rate parameter that depends on the covariates. Specifically, we set the log rate parameter to have baseline −4.5, and the race variable to take the values 1 and 2 for "black" and "other" respectively, when compared to "white". Finally, we let "age" increase the log rate by 0.05 per year:

```
log.rate.vec <- -4.5 + c(rep(0,20), rep(1,20), rep(2,20))
    + age*0.05
```

Finally, we define the exponential survival variables, with no censoring:

```
tt <- rexp(n=60, rate=exp(log.rate.vec))
status <- rep(1, 60)
```

Now we can fit a Cox proportional hazards model,

```
> library(survival)
> result.cox <- coxph(Surv(tt, status) ~ race + age)
> summary(result.cox)
```

```
  n= 60, number of events= 60
              coef exp(coef) se(coef)     z Pr(>|z|)
raceblack  1.15154   3.16305  0.36752 3.133 0.00173 **
raceother  2.49905  12.17087  0.42936 5.820 5.87e-09 ***
age        0.07798   1.08110  0.01448 5.385 7.24e-08 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

We see that the coefficient estimates, 1.15, 2.50, and 0.08, are close to the true values from the simulation, (1, 2, and 0.05). These estimates are log hazard ratios. To describe the estimated effect of, say, "black race" compared to "white race", we can look at the "exp(coef)" column, and conclude that blacks have 3.16 times the risk of death as do whites. The model matrix for "race + age" is as discussed above, and is created within the "coxph" function. The parameter estimates are maximum partial likelihood estimates. The Z test statistics and p-values for statistical tests are generalizations of the two-group comparison Wald tests described in the previous section. In the next section, we discuss how to handle proportional hazards models such as this one where there are multiple covariates.

6.3 Hypothesis Testing for Nested Models

Now that we have the tools to fit models with multiple covariates, let's use these tools to compare models for the "pharmacoSmoking" data, which was introduced in Chap. 1. When constructing statistical tests, it is necessary to compare what are called "nested" models. That is, when comparing two models, the covariates of one model must be a subset of the covariates in the other. For example, consider the following two models, which we define by listing the covariates to be included in the proportional hazards model:

Model A: ageGroup4
Model B: employment
Model C: ageGroup4 + employment

Here, Model A is nested in Model C, and Model B is also nested in Model C, so these models can be compared using statistical tests. But Models A and B can't be directly compared in this way. Now, "ageGroup4" and "employment" are covariates with four and three levels, respectively:

```
> levels(ageGroup4)
[1] "21-34" "35-49" "50-64" "65+"
> levels(employment)
[1] "ft"    "other" "pt"
```

where "ft" and "pt" refer to full-time and part-time employment, respectively. When we fit these models in R, it will by default choose the first level as the reference level:

```
> modelA.coxph <- coxph(Surv(ttr, relapse) ~ ageGroup4)
> modelA.coxph
                  coef exp(coef) se(coef)      z     p
ageGroup435-49  0.0293     1.030    0.309  0.0947 0.920
```

```
ageGroup450-64 -0.7914     0.453    0.336 -2.3551 0.019
ageGroup465+   -0.3173     0.728    0.444 -0.7153 0.470

Likelihood ratio test=12.2  on 3 df, p=0.00666

> modelB.coxph <- coxph(Surv(ttr, relapse) ~ employment)
> modelB.coxph
                 coef exp(coef) se(coef)     z    p
employmentother 0.198      1.22    0.237 0.836 0.40
employmentpt    0.450      1.57    0.323 1.394 0.16

Likelihood ratio test=2.06  on 2 df, p=0.357

> modelC.coxph <- coxph(Surv(ttr, relapse) ~ ageGroup4 +
+      employment)
> modelC.coxph
                  coef exp(coef) se(coef)      z      p
ageGroup435-49  -0.130     0.878    0.321 -0.404 0.6900
ageGroup450-64  -1.024     0.359    0.359 -2.856 0.0043
ageGroup465+    -0.782     0.457    0.505 -1.551 0.1200
employmentother  0.526     1.692    0.275  1.913 0.0560
employmentpt     0.500     1.649    0.332  1.508 0.1300

Likelihood ratio test=16.8  on 5 df, p=0.00492
```

From the results of Model C, we can see that some levels of the predictors are statistically significant based on the Wald tests in the last column. For example, we see that the "50-64" age group has a lower hazard when compared to the reference (which we noted above is the "21-34" age group), with log-hazard ratio of -1.024 and a p-value of 0.0043. We also see that those with part-time employment have a higher hazard when compared to the baseline (which we noted above is the "full-time" group), with a log-hazard ratio of 0.526 and a p-value of 0.056, which may be seen as not quite statistically significant at the 0.05 level. But we cannot easily see from these p-values whether or not the term "ageGroup4" or the term "employment" belong in the model. These we can assess using a (partial) likelihood ratio test. The log-likelihoods for the three models are as follows:

```
> logLik(modelA.coxph)
'log Lik.' -380.043 (df=3)

> logLik(modelB.coxph)
'log Lik.' -385.1232 (df=2)

> logLik(modelC.coxph)
'log Lik.' -377.7597 (df=5)
```

Let us begin be determining if "ageGroup4" belongs in the model by comparing Models A and C. The null hypothesis is that the three coefficients for "ageGroup4" are zero, and the alternative is that they are not all zero. The likelihood ratio statistic is

$$2\left(l(\hat{\beta}_{\text{full}} - l(\hat{\beta}_{\text{reduced}}\right) = 2(-377.7597 + 380.043) = 4.567.$$

This is twice the difference between the partial log-likelihood evaluated at the "full" model (Model C) and the value at the "reduced" model (Model A), We compare this to a chi-square distribution with $5 - 3 = 2$ degrees of freedom, which is the difference in degrees of freedom for the two models. The p-value is thus

```
> pchisq(4.567, df=2, lower.tail=F)
[1] 0.1019268
```

and we would conclude that the effect of "ageGroup4" is not statistically significant when "employment" is included in the model.

Similarly we can compare Models B and C to test for the importance of "employment" in the presence of "age":

$$2\left(l(\hat{\beta}_{\text{full}} - l(\hat{\beta}_{\text{reduced}}\right) = 2(-377.7597 + 385.1232) = 14.727.$$

We compare this to a chi-square distribution with $5 - 2 = 3$ degrees for freedom:

```
> pchisq(14.727, df=3, lower.tail=F)
[1] 0.002065452
```

We thus conclude that "employment" belongs in the model if "ageGroup4" is also included, since the p-value for the former is extremely small.

Should "ageGroup4" even be in the model? To carry out a likelihood ratio test for this factor we need to refer to what we shall call the "null" model, one with no covariates. We may evaluate this as follows:

```
> model.null.coxph <- coxph(Surv(ttr, relapse) ~ 1)
> logLik(model.null.coxph)
'log Lik.' -386.1533 (df=0)
```

(The "logLik" function also returns a warning connected to having zero degrees of freedom. We may ignore this, since the value of the log likelihood is correct.) This null model is nested within Model A (and is actually nested in all other models), so we may compute the likelihood ratio test as follows:

$$2\left(l(\hat{\beta}_{\text{full}} - l(\hat{\beta}_{\text{reduced}}\right) = 2(-380.043 + 386.1533) = 12.2206$$

which we compare to a chi-square distribution with $3 - 0 = 3$ degrees of freedom:

```
> pchisq(12.2206, df=3, lower.tail=F)
[1] 0.006664445
```

This result, which shows that "ageGroup4" by itself is strongly statistically significant, is identical to the results given above in the output from Model A. In fact, the function "coxph" always prints the value of the likelihood ratio test for the fitted model as compared to the null model. Since the reference model is always the null model, another way to carry out the likelihood ratio test for, for example, Model B to Model C, is to take the differences of the printed log-likelihood statistics, e.g., $16.8 - 12.2 = 4.6$, which (to one decimal place) is identical to the value we computed using the “logLik” function. A more direct way to compare models is using the “anova” function, which directly computes test statistic, degrees of freedom, and p-value:

```
> anova(modelA.coxph, modelC.coxph)
Analysis of Deviance Table
 Cox model: response is  Surv(ttr, relapse)
Model 1: ~ ageGroup4
Model 2: ~ ageGroup4 + employment
   loglik  Chisq Df P(>|Chi|)
1 -380.04
2 -377.76 4.5666  2    0.1019
```

6.4 The Akaike Information Criterion for Comparing Non-nested Models

When we have a specific hypothesis to test, the methods of the previous section are appropriate. But often we have a large number of potential factors and need to prune the model so that only necessary covariates are included. There are a number of tools available to aid this process. A well-known method is “stepwise” model selection. In the “forward” version of this method, we first fit univariate models, one for each covariate. The covariate with the smallest p-value is chosen and added to the base model. Then, with that covariate included, a separate model is fitted with each single additional covariate also included. We then select the best second variable (the one with the smallest p-value), so that we have a model with two covariates. We continue until no additional covariate has a p-value less than a certain critical value; common critical p-values are 5 % and 10 %. The result is the “final” model, presumably including all the covariates that are related to the outcome, and excluding the ones unrelated to it. In another version, known as the “backwards” stepwise procedure, we start with all the covariates in the model, and then remove them one by one, each time removing the one with the largest p-value. The procedure continues until the p-values are below the critical p-value.

There are a number of problems with the stepwise procedure. For one thing, due to multiple comparisons, the p-values that are produced from one stage to the next are not what they appear to be. Thus, the decision criterion for model selection (e.g. to continue until all p-values are less than a particular value, often 0.05) does not necessarily produce a list of covariates that are statistically significant at that level. Another problem is that p-values are only valid for nested models, as discussed in the previous section. Thus, this procedure does not allow one to compare non-nested models. A better way of looking at the model search procedure is to compute a quantity known as the Akaike Information Criterion, or AIC. This quantity is given by $AIC = -2 \cdot l(\hat{\beta}) + 2 \cdot k$, where $l(\hat{\beta})$ denotes the value of the partial log likelihood at the M.P.L.E. for a particular model, and k is the number of parameters in the model. The value of the AIC balances two quantities which are properties of a model. The first is goodness of fit,$-2 \cdot l(\hat{\beta})$. This quantity is smaller for models that fit the data well. The second quantity, the number of parameters, is a measure of complexity. This enters the AIC as a penalty term. Thus, a "good" model is one that fits the data well (small value of $-2 \cdot l(\hat{\beta})$) with few parameters ($2k$), so that smaller values of AIC should in theory indicate better models. For example, again considering the "pharmacoSmoking" model, we can compute the AIC for model A as follows: $AIC = 2 \times 380.043 + 2 \times 2 = 766.086$. But it is more convenient to use the "AIC" function:

```
> AIC(modelA.coxph)
[1] 766.086
> AIC(modelB.coxph)
[1] 774.2464
> AIC(modelC.coxph)
[1] 765.5194
```

The best fitting model from among these three, using the AIC criterion, is then Model C. This is the model that includes both "ageGroup4" and "employment". Model A, which includes only "ageGroup4", is a close second choice.

While we could in principle compute the AIC for all possible combinations of covariates, in practice this may be computationally impractical. An alternative is to return to the stepwise procedure, using AIC (instead of p-values) to drive the covariate selection. Here is an example for the pharmacoSmoking data, where we start with all of the covariates, and use the "step" function to find a more parsimonious model using the AIC criterion:

```
modelAll.coxph <- coxph(Surv(ttr, relapse) ~ grp + gender + race
            + employment + yearsSmoking + levelSmoking +
           ageGroup4 + priorAttempts + longestNoSmoke)

result.step <- step(modelAll.coxph, scope=list(upper=~ grp +
            gender + race + employment + yearsSmoking +
            levelSmoking + ageGroup4 + priorAttempts +
            longestNoSmoke, lower=~grp) )
```

The results of the first step are as follows:

```
Start:  AIC=770.2 Surv(ttr, relapse) ~ grp + gender + race +
     employment + yearsSmoking + levelSmoking + ageGroup4 +
     priorAttempts + longestNoSmoke
                  Df    AIC
- race             3 766.98
- yearsSmoking     1 768.20
- gender           1 768.20
- priorAttempts    1 768.24
- levelSmoking     1 768.47
- longestNoSmoke   1 769.04
<none>               770.20
- employment       2 772.45
- ageGroup4        3 774.11
```

The terms in the model are listed in order from the one which, when deleted, yields the greatest AIC reduction ("race" in this case) to the smallest reduction ("ageGroup4"). Thus, "race" is deleted. This procedure continues until the last step:

```
Step:  AIC=758.42 Surv(ttr, relapse) ~ grp + employment
       + ageGroup4
                 Df    AIC
<none>              758.42
+ longestNoSmoke  1 759.10
- employment      2 760.31
+ yearsSmoking    1 760.34
+ gender          1 760.39
+ priorAttempts   1 760.40
+ levelSmoking    1 760.41
+ race            3 761.53
- ageGroup4       3 767.24
```

The "+" sign shows the effect on AIC of adding certain terms. This table shows that no addition or subtraction of terms results in further reduction of the AIC. The coefficient estimates for the final model are

```
> result.step
                  coef exp(coef) se(coef)      z      p
grppatchOnly     0.656     1.928    0.220  2.986 0.0028
employmentother  0.623     1.865    0.276  2.254 0.0240
employmentpt     0.521     1.684    0.332  1.570 0.1200
ageGroup435-49  -0.112     0.894    0.322 -0.348 0.7300
ageGroup450-64  -1.023     0.359    0.360 -2.845 0.0044
ageGroup465+    -0.707     0.493    0.502 -1.410 0.1600

Likelihood ratio test=25.9  on 6 df, p=0.000233
```

We may display these results as a forest plot in Fig. 6.1 (see appendix for a discussion of forest plots),

This is a plot of the coefficient estimates and 95 % confidence intervals, each with respect to a reference level. For example, we can see that triple therapy

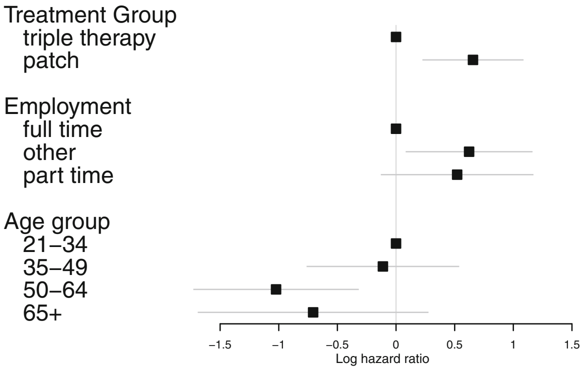


Fig. 6.1 Forest plot of estimates of log hazard ratios for the final model fit to the pharmacoSmoking data

(the reference) is better than the patch alone, that those with full-time work (the reference) have a better success rate than those working part time and those with the "other" employment status. We also see that the upper age groups (50 and above) had better results than younger patients.

An alternative to the AIC is the "Bayesian Information Criterion", sometimes called the "Schwartz criterion". It is given by $BIC = -2 \cdot \log(L) + k \cdot \log(n)$. The key difference is that the BIC penalizes the number of parameters by a factor of $\log(n)$ rather than by a factor of 2 as in the AIC. As a result, using the BIC in model selection will tend to result in models with fewer parameters as compared to AIC.

6.5 Including Smooth Estimates of Continuous Covariates in a Survival Model

When a covariate is continuous, we are interested in whether that covariate is related to survival and, if so, in what manner. That is, is the relationship to the log-hazard linear? Or is it a quadratic or other non-linear relationship? Let us consider again the pharmacoSmoking data. We found, in the previous section, that treatment group, employment status, and age are related to time to relapse. We entered "age" in the model by dividing it into four age groups, 21-34, 35-49, 50-64, and 65 and older, and found that the two older age groups were associated with increased time to relapse as compared to the two younger groups. From Fig. 6.1 we see that this relationship (on the log-hazard ratio scale) appears not to be linear. An alternative way to model a non-linear relationship is via "smoothing splines". Splines are mathematical constructs made of pieces of polynomial functions that are stitched together to form a smooth curve. The points where these pieces are joined are called "knots". The challenges in using smoothing splines are, first, to select the location of the knots, and second, to find an optimal set of polynomials to model the statistical relationship. A classical treatment of splines is de Boor [13], and their use in statistics has been discussed by many authors. In survival analysis, an effective method of finding a smoothing spline is via "penalized partial likelihood." The quantity to be optimized consists of two parts, the partial log likelihood discussed in earlier chapters, and a penalty term. Splines with many knots are complex and tend to increase the partial log-likelihood, since they improve the fit of the model. The penalty term is an integral of the second derivative, so that increasing complexity of the spline curve *decreases* this second term. The sum of these two parts, the penalized partial log-likelihood, is a quantity that, when maximized, balances goodness of fit against complexity. For details of this method, see Therneau and Grambsch [68].

This "pspline" function may be used with "coxph" to fit a smoothing spline to the pharmacoSmoking data as follows:

```
> modelS4.coxph <- coxph(Surv(ttr, relapse) ~ grp + employment +
+       pspline(age, df=4) )
> modelS4.coxph
```

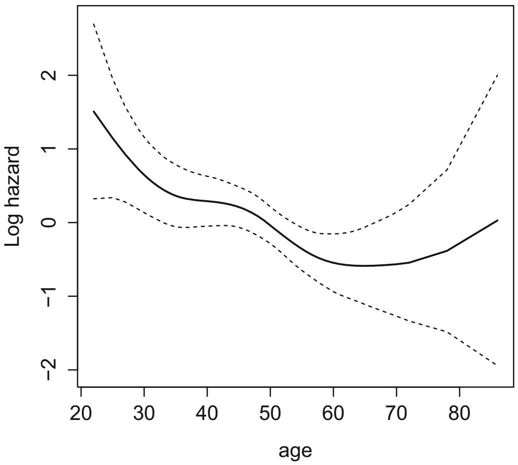


Fig. 6.2 Penalized spline fit of age

```
                               coef  se(coef) se2    Chisq DF   p
grppatchOnly                   0.651 0.221    0.219  8.67 1.00 0.0032
employmentother                0.633 0.277    0.275  5.21 1.00 0.0220
employmentpt                   0.570 0.340    0.333  2.81 1.00 0.0940
pspline(age, df = 4), lin -0.034 0.010    0.010 11.07 1.00 0.0009
pspline(age, df = 4), non                        4.20 3.08 0.2500

Iterations: 3 outer, 9 Newton-Raphson
     Theta= 0.709
Degrees of freedom for terms= 1.0 2.0 4.1
Likelihood ratio test=27.3  on 7.02 df, p=0.000297  n= 125
```

We see, as we saw previously, that “grp”, “employment”, and “age” are important predictors of the log-hazard. Now, however, the continuous variable “age” is fitted as a linear component and a nonlinear component. The linear component is −0.0339, indicating that older individuals have a lower log hazard of relapse. The non-linear component, with a p-value of 0.25, is not statistically significant, indicating that there is not enough data to state definitively that the relationship is non-linear. We may plot the relationship of age to log hazard using the “termplot” function:

```
termplot(modelS4.coxph, se=T, terms=3, ylabs="Log hazard")
```

The option “se=T” produces the confidence limits, and the “terms=3” option specifies that the third variable (age) is the one we want to plot. The plot is shown in Fig. 6.2. This shows a decreasing relationship with age, as we have seen previously, with a slight upward turn after age 65. However, the data is rather sparse at these older ages, as reflected by the wide confidence interval. Thus, this confirms our observation that the departure from linearity cannot be established.

6.6 Additional Note

1. See, for example, Harrell [28] for a general discussion of model-selection issues. Hosmer, Lemeshow, and May [32] have extensive advice on model selection in the context of survival analysis.

Exercises

6.1. The data set "hepatocelluar" is in the "asaur" package. It contains 17 clinical and biomarker measurements on 227 patients, as well as overall survival and time to recurrence, both recorded in months [42, 43]. There are three measures of CXCL17 activity, CXCL17T (intratumoral), CXCL17P (peritumoral), and CXCL17N (nontumoral). There is a particular interest in whether they are related to overall and also recurrence-free survival. Which of the three is most strongly related for each survival outcome? For the one most strongly related with survival, fit a spline model and plot it, as we did in Sect. 6.5. Does this suggest that categorizing CXCL17 would be appropriate?

6.2. For the covariates with complete data (in Columns 1–22), use stepwise regression with AIC to identify the best model for (a) overall survival, and (b) recurrence-free survival.

Chapter 7
Model Diagnostics

7.1 Assessing Goodness of Fit Using Residuals

The use of residuals for model checking has been well-developed in linear regression theory (see for example Weisberg, 2014 [77]). The residuals are plotted versus some quantity, such as a covariate value, and the observed pattern is used to diagnose possible problems with the fitted model. Some residuals have the additional property of not only indicating problems but also suggesting remedies. That is the pattern of the plotted residuals may suggest an alternative model that fits the data better. Many of these residuals have been generalized to survival analysis. In addition, the fact that survival data evolves over time, and requires special assumptions such as proportional hazards, makes it necessary to develop additional diagnostic residual methods.

7.1.1 Martingale and Deviance Residuals

An important tool for assessing the goodness of fit of a model is to compare the censoring indicator (0 for censored, 1 for death) for each subject to the expected value of that indicator under the proportional hazards Cox model. If there are no time dependent covariates and if the survival times are right-censored, this is given by

$$m_i = \delta_i - \hat{H}_0(t_i)\exp(z_i'\hat{\beta}).$$

These residuals, which originate from the counting process theory underlying the Cox proportional hazards model, sum to 1, range in value from $-\infty$ to a maximum of 1, and each has an expected value of zero. The residual is essentially the difference between the observed value (1 or 0) of the censoring indicator and its

D. Moore, *Applied Survival Analysis Using R*, Use R!,
DOI 10.1007/978-3-319-31245-3_7

expected value under a particular Cox model. The asymmetry of the residuals might appear to be a disadvantage, but Therneau and Grambsch [68], in Chap. 5, show that a plot of these residuals versus the covariate z can reveal not only discrepancies in the model but also the actual functional form of this covariate.

Martingale residuals may be used much as residuals that the reader may be familiar with from linear regression. However, unlike those, the sum of squares of Martingale residuals cannot be used as a measure of goodness of fit. The "deviance" residual is an alternative that does have this property. It may defined in terms of the martingale residual as follows [69]:

$$d_i = \operatorname{sign}(m_i) \cdot \{-2 \cdot [m_i + \delta_i \log(\delta_i - m_i)]\}^{1/2}$$

These residuals are symmetrically distributed with expected value 0 (if the fitted model is correct). The sum of squares of these residuals is the value of the likelihood ratio test, which is analogous to the deviance from generalized linear model theory [48]. While the properties of deviance residuals might seem preferable to those of martingale residuals, only the latter have the property of showing us the functional form of a covariate. For this reason, in practice, the martingale residuals are more useful.

Example 7.1. Consider the "pharmacoSmoking" data set, and a fit of the "null" Cox proportional hazards model. A null model is one with no fitted covariates. There is still a partial likelihood, and the model produces martingale residuals which take the form $m_i = \delta_i - \hat{H}_0(t_i)\exp(z'\beta)$. We may plot these against continuous predictors to get a preliminary assessment of which of these predictors should be in the model, and what form they should take. We first read in the data and truncate the variable "priorAttempts" at 20, since recorded values of this variable that exceed 20 are not likely to be correct,

```
> pharmacoSmoking <- read.csv("PharmacoSmoking.csv")
> priorAttemptsT <- priorAttempts
> priorAttemptsT[priorAttempts > 20] <- 20
```

We may fit the null model and obtain these residuals as follows:

```
> library(survival)
> result.0.coxph <- coxph(Surv(ttr, relapse) ~ 1)
> rr.0 <- residuals(result.0.coxph, type="martingale")
```

We next assess the potential relationship of survival to age, prior attempts at quitting, and longest prior period without smoking. We plot these null model residuals versus each of these variables and also versus log transformations of these variables. In the following, since the smallest value of "priorAttemptsT" and "longestNoSmoke" is zero, we add one to each before taking the log transformation. We fit a "loess" smooth curve through each set of residuals to better assess the shapes of the relationships, using the "loess" function. In order to also get 95 % confidence intervals for these smooth curves, we use the function "smoothSEcurve", a simple plotting function that is defined in the appendix. The results are in Fig. 7.1.

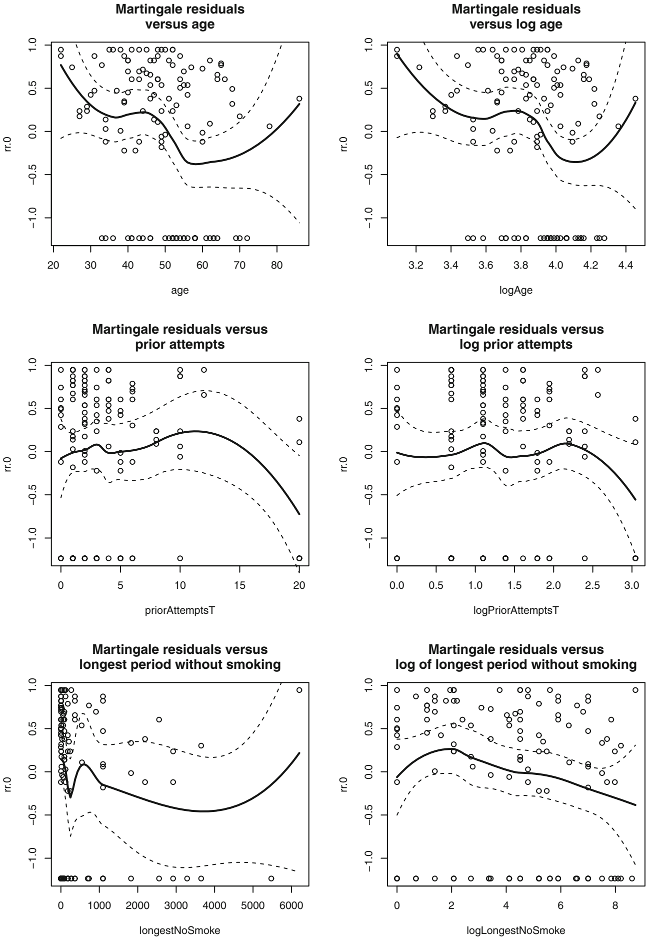


Fig. 7.1 Martingale residuals from a null model fit to the pharmacoSmoking data, plotted versus three continuous predictors and one log-transformed predictor

In this figure, the three plots on the left are for untransformed variables, and the three on the right are for log transformations of these variables.

```
> par(mfrow=c(3,2))
> plot(rr.0 ~ age)
> smoothSEcurve(rr.0, age)
> title("Martingale residuals\nversus age")

> logAge <- log(age)
> plot(rr.0 ~ logAge)
> smoothSEcurve(rr.0, logAge)
> title("Martingale residuals\nversus log age")
> plot(rr.0 ~ priorAttemptsT)
> smoothSEcurve(rr.0, priorAttemptsT)
> title("Martingale residuals versus\nprior attempts")

> logPriorAttemptsT <- log(priorAttemptsT + 1)
> plot(rr.0 ~ logPriorAttemptsT)
> smoothSEcurve(rr.0, logPriorAttemptsT)
> title("Martingale residuals versus\nlog prior attempts")

> plot(rr.0 ~ longestNoSmoke)
> smoothSEcurve(rr.0, longestNoSmoke)
> title("Martingale residuals versus\n
+ longest period without smoking")

> logLongestNoSmoke <- log(longestNoSmoke+1)
> plot(rr.0 ~ logLongestNoSmoke)
> smoothSEcurve(rr.0, logLongestNoSmoke)
> title("Martingale residuals versus\n
+ log of longest period without smoking")
```

In each case, the untransformed variables show considerable non-linearity. Note in particular the first plot, versus age, that shows the same non-linear relationship we modeled directly (with "pspline") in Sect. 6.5. This illustrates that the use of martingale residuals with a null model is an alternative way to identify the form of a non-linear relationship. The log-transformation of "LongestNoSmoke" produces a curve that will be easier to adjust for using linear models, whereas with "age" and "priorAttempts", the log-transformation doesn't appear to help.

As in the previous chapter, we use the "step" function to identify a model with low AIC:

```
result.grp.coxph <- coxph(Surv(ttr, relapse) ~ grp)
result.step <- step(result.grp.coxph, scope=list(upper=~ grp +
                  gender + race + employment + yearsSmoking +
                  levelSmoking + age + priorAttemptsT +
                  logLongestNoSmoke, lower=~grp) )
```

The resulting model is as follows:

```
> result.step
                   coef exp(coef) se(coef)     z      p
grppatchOnly     0.6079     1.837   0.2184  2.78 0.0054
age             -0.0353     0.965   0.0108 -3.28 0.0010
employmentother  0.7035     2.021   0.2693  2.61 0.0090
employmentpt     0.6537     1.923   0.3273  2.00 0.0460

Likelihood ratio test=22  on 4 df, p=0.000198  n= 125, number of
    events= 89
```

We may now assess the final model by creating residuals and then plotting them versus the final selected predictors,

```
> rr.final <- residuals(result.step, type="martingale")
> par(mfrow=c(2,2))
> plot(rr.final ~ age)
> smoothSEcurve(rr.final, age)
> title("Martingale residuals\nversus age")

> plot(rr.final ~ grp)
> title("Martingale residuals\nversus treatment group")

> plot(rr.final ~ employment)
> title("Martingale residuals\nversus employment")
```

The results, shown in Fig. 7.2, show that the residuals treatment group and employment are evenly distributed over the values of the covariates. The variable "age" still shows some possible deviation, but it is much improved over the plot for the null model.

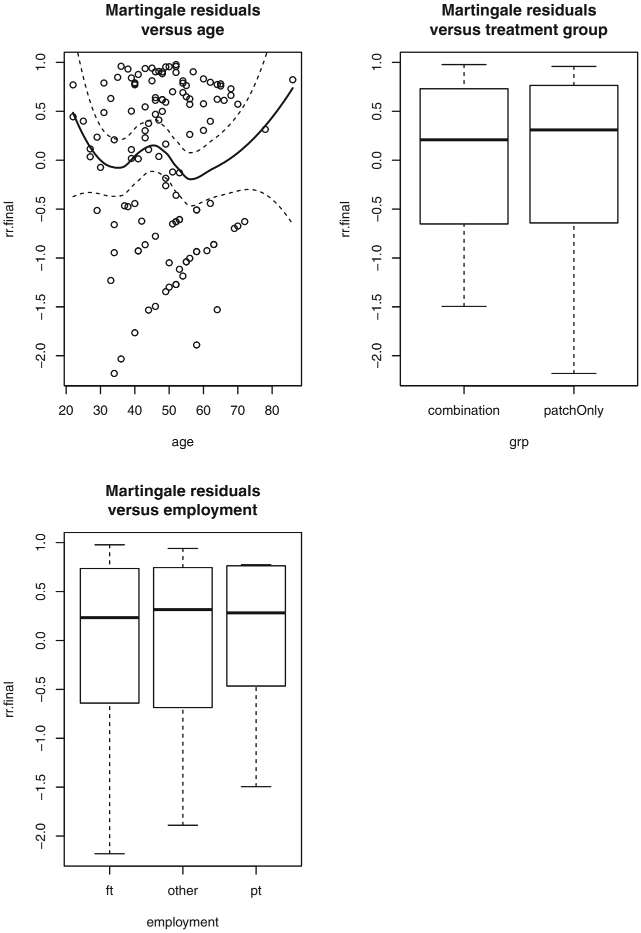


Fig. 7.2 Martingale residuals from final model

7.1.2 Case Deletion Residuals

Some subjects may have an especially large influence on the parameter estimates. Since some such influential subjects may indicate a problem with the data, it is helpful for the data analyst to have tools that can identify those subjects. Case deletion residuals (also called "jackknife residuals") serve this purpose. For each subject, a case deletion residual is the difference in the value of the coefficient using all of the data and its value when that subject is deleted from the data set. Using the pharmacoSmoking data set, we illustrate computation of these residuals using "age" as a predictor. First, we find the coefficient for age (the fourth coefficient) using all the data:

```
> result.coxph <- coxph(Surv(ttr, relapse) ~ grp + employment
   + age)
> coef.all <- result.coxph$coef[4]
> coef.all
        age
-0.03528934
```

Next, for each subject in turn ("n.obs" subjects in all), we delete the *i*th subject from the survival time "tt", censoring indicator "relapse", and covariates "grp", "employment", and "age", and fit a Cox model to this reduced data set. The results for the *i*th subject go into "result.coxph.i". We extract the age coefficient (the fourth element of the vector of coefficient estimates) into "coef.i" and compute the jackknife residual as the difference of "coef.i" and "coef.all":

```
n.obs <- length(ttr)
jkbeta.vec <- rep(NA, n.obs)
for (i in 1:n.obs) {
  tt.i <- ttr[-i]
  delta.i <- relapse[-i]
  grp.i <- grp[-i]
  employment.i <- employment[-i]
  age.i <- age[-i]
  result.coxph.i <- coxph(Surv(tt.i, delta.i) ~ grp.i +
        employment.i + age.i)
  coef.i <- result.coxph.i$coef[4]
  jkbeta.vec[i] <- (coef.all - coef.i)
  }
```

We may plot these residuals versus the patient id's, which we place in the vector "index.obs". In the plot function, "type='h' " causes the residuals to be plotted as spikes. Finally, "abline(h=0)" plots a horizontal line through 0.

```
index.obs <- 1:n.obs
plot(jkbeta.vec ~ index.obs, type="h",
    xlab="Observation", ylab="Change in coefficient for age",
   cex.axis=1.3, cex.lab=1.3)
abline(h=0)
```

The "identify" function allows us to identify the index numbers of select patients by manually selecting them with a mouse:

```
identify(jkbeta.vec ~ index.obs)
```

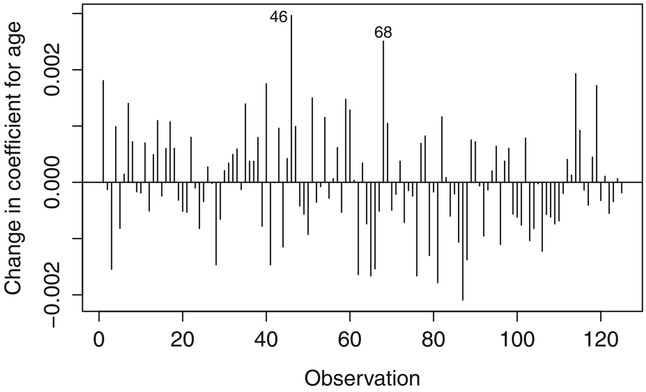


Fig. 7.3 Change in coefficient (dfbeta) residuals for age

The plot is shown in Fig. 7.3. There we see that no single patient changes the estimate of the "age" coefficient by more than 0.003, which is less than 10 % of the value of that coefficient. Still, we see that patients 46 and 68 have the most influence over the parameter estimate for age, and one may check these data points to ensure that there are no errors in recording the data.

A more convenient way to obtain case deletion residuals is using the "residuals" function with "type = 'dfbeta' ". These residuals are approximations to case-deletion residuals that require less numerical computation by eliminating all but one interaction in the partial likelihood maximization. The simplest way to compute such as residual is exactly as outlined above for case-deletion residuals, but one starts the search for a maximum partial likelihood at $\beta = 0$ and then only permit a single iteration of the "coxph" function for each subject. Since the first iteration of the coxph function yields an estimate of the coefficient that is near the final value, these residuals should be nearly as effective at identifying influential subjects as the case deletion residuals. These may be computed using the "residuals" function as follows:

```
resid.dfbeta <- residuals(result.coxph, type="dfbeta")
n.obs <- length(ttr)
index.obs <- 1:n.obs
plot(resid.dfbeta[,4] ~ index.obs, type="h",
   xlab="Observation", ylab="Change in coefficient")
abline(h=0)
identify(resid.dfbeta[,4] ~ index.obs)
```

The resulting dfbeta residuals plot (not shown) is nearly identical to that in Fig. 7.3. While the reduction in computation time is of minimal value in most applications, using the "residuals" function to obtain dfbeta residuals has the advantage that it is slightly easier to use it to produce multiple plots for all of the coefficients—one only need select the corresponding element of "resid.dfbeta" for each coefficient in the model.

A modification of the dfbeta residual is to standardize it by an estimate of its standard error. In the residuals function, we obtain these residuals exactly as above, but using the option "type = 'dfbetas' ". In this example this refinement makes no perceptible difference in which cases are found to be influential (see Exercise 7.1), but potentially could be of value with other data sets.

7.2 Checking the Proportion Hazards Assumption

The proportional hazards assumption is key to the construction of the partial likelihood, since it is this property that allows one to cancel out the baseline hazard function from the partial likelihood factors. If one has a binary predictor variable, such as experimental vs. standard treatment, what this assumption means is that the hazard functions are proportional, and hence that the log-hazards are separated by a constant at all time points. Similarly, a categorical variable with many levels will result in parallel log hazard functions. This assumption is at best an approximation in practice, and minor violations are unlikely to have major effects on inferences on model parameters. For this reason, formal hypothesis tests of the proportional hazards assumption are often of limited value. Still, it is useful to assess, in a particular data set, if this assumption is reasonable, and what one can do if it is not. Here we will examine some commonly used assessment methods.

7.2.1 Log Cumulative Hazard Plots

If we are comparing survival times between two groups, there is a simple plot that can help us assess the proportional hazards assumption. Under this assumption, we have

$$S_1(t) = [S_0(t)]^{\exp(\beta)}$$

where $\exp(\beta)$ is the proportional hazards constant. Taking logs of both sides, we have

$$\log[S_1(t)] = \exp(\beta) \cdot \log[S_0(t)] .$$

Since the survival functions are less than 1, their logarithms are negative. Thus, we must negate them before we take a second logarithm,

$$\log\{-\log[S_1(t)]\} = \beta + \log\{-\log[S_0(t)]\} .$$

The function $g(u) = \log\{-\log(u)\}$is called a complementary log-log transformation, and has the effect of changing the range from $(0, 1)$ for u to $(-\infty, \infty)$ for

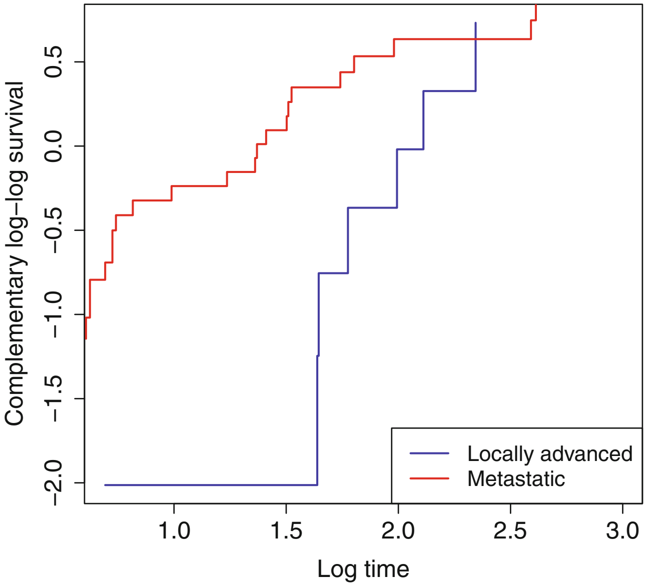


Fig. 7.4 Complementary log-log plot for the two pancreatic cancer groups

$g(u)$. A plot of $g\left[S_1(t)\right]$ and $g\left[S_0(t)\right]$ versus t or $\log(t)$ will yield two parallel curves separated by β if the proportional hazards assumption is correct.

We may illustrate this with the pancreatic cancer data from Chap. 3. There we found that the Prentice-modification test showed a stronger group difference than did the log-rank test, and the suggestion was that this difference was caused by non-proportional hazards. We can investigate the proportional hazards assumption as follows, with results shown in Fig. 7.4.

```
> result.surv.LA <- survfit(Surv(pfs.month) ~ stage,
+   subset={stage == "LA"})
> time.LA <- result.surv.LA$time
> surv.LA <- result.surv.LA$surv
> cloglog.LA <- log(-log(surv.LA))
> logtime.LA <- log(time.LA)
> result.surv.M <- survfit(Surv(pfs.month) ~ stage,
+   subset={stage == "M"})
> time.M <- result.surv.M$time
> surv.M <- result.surv.M$surv
> cloglog.M <- log(-log(surv.M))
> logtime.M <- log(time.M)
> plot(cloglog.LA ~ logtime.LA, type="s", col="blue", lwd=2)
> lines(cloglog.M ~ logtime.M, col="red", lwd=2, type="s")
> legend("bottomright", legend=c("Locally advanced",
+  "Metastatic"), col=c("blue","red"), lwd=2)
```

The curves are clearly not parallel. However, one problem with this approach is that we don't have a clear way to assess statistical significance. This is a critical issue here due to the small sample size, particularly in the locally advanced group.

7.2.2 Schoenfeld Residuals

Schoenfeld residual plots provide a useful way to assess this assumption. To see how they are derived, recall the partial log-likelihood function,

$$l(\beta) = \sum_{i \in D} \left\{ \log(\psi_i) - \log\left(\sum_{k \in R_i} \psi_k \right) \right\} = \sum_{i \in D} \left\{ z_i\beta - \log\left(\sum_{k \in R_i} e^{z_k\beta} \right) \right\}$$

and its derivative, which is the score function.

$$l'(\beta) = \sum_{i \in D} \left\{ z_i - \sum_{k \in R_i} z_k \cdot p(\beta, z_k) \right\},$$

where

$$p(\beta, z_k) = \frac{e^{z_k\beta}}{\sum\limits_{j \in R_k} e^{z_j\beta}}.$$

The Schoenfeld residuals are the individual terms of the score function, and each term is the observed value of the covariate for patient i minus the expected value $E(Z_i) = \bar{z}(t_i)$, which is a weighted sum, with weights given by $p_k(\beta)$, of the covariate values for subjects at risk at that time. Each weight may be viewed as the probability of selecting a particular person from the risk set at time t_i. For an estimate $\hat{\beta}$, the residual for the ith failure time is

$$\hat{r}_i = z_i - \sum_{k \in R_i} z_k \cdot p(\hat{\beta}, z_k) = z_i - \bar{z}(t_i). \tag{7.2.1}$$

A plot of theses residuals versus the covariate z_i will yield a pattern of points that are centered at zero, if the proportional hazards assumption is correct. Note that these residuals are defined only for the failure (and not the censoring) times. If there are multiple covariates, then one obtains a series of residuals for each covariate.

The "expected" value of the covariate at that time is the weighted sum (weighted by p_i). To clarify how we compute these, let us return to our simple example from Chap. 3 comparing two sets of three observations. We first need $\hat{\beta}$, the log hazard ratio estimate, which we may compute as follows:

```
> tt <- c(6, 7, 10, 15, 19, 25)
> delta <- c(1, 0, 1, 1, 0, 1)
> trt <- c(0, 0, 1, 0, 1, 1)
> result.coxph <- coxph(Surv(tt, delta) ~ trt)
> result.coxph$coef
trt  -1.326
```

We see that $\hat{\beta} = -1.326$. To compute he Schoenfeld residuals, we first compute the weights as follows:

t_i	n_{0i}	n_{1i}	$p(\beta, z_k = 0)$	$p(\beta, z_k = 1)$
6	3	3	$\frac{1}{3+3e^{-1.326}}$	$\frac{e^{-1.326}}{3+3e^{-1.326}}$
10	1	3	$\frac{1}{1+3e^{-1.326}}$	$\frac{e^{-1.326}}{1+3e^{-1.326}}$
15	1	2	$\frac{1}{1+2e^{-1.326}}$	$\frac{e^{-1.326}}{1+2e^{-1.326}}$
25	1	1	$\frac{1}{e^{-1.326}}$	$\frac{e^{-1.326}}{e^{-1.326}} = 1$

Next, we compute the expected values, and then the residuals:

t_i	$E(Z_i) = \sum\limits_{k \in R_i} z_k \cdot p_k(\hat{\beta})$	z_i	$\widehat{r_i} = z_i - E(Z_i)$
6	$3 \times 0 \times \frac{1}{3+3e^{-1.326}} + 3 \times 1 \times \frac{e^{-1.326}}{3+3e^{-1.326}} = 0.2098$	0	-0.2098
10	$1 \times 0 \times \frac{1}{1+3e^{-1.326}} + 3 \times 1 \times \frac{e^{-1.326}}{1+3e^{-1.326}} = 0.4434$	1	0.5566
15	$1 \times 0 \times \frac{1}{1+2e^{-1.326}} + 2 \times 1 \times \frac{e^{-1.326}}{1+2e^{-1.326}} = 0.3468$	0	-0.3468
25	1	1	0

At the first failure time ($t_i = 6$) there are three at risk in the control group and three in the treatment group. The expected value of the covariate is weighted sum of the three control covariate values, $z_i = 0$, the weights being $1/[3 + 3\exp(-1.326)]$, and the three treatment covariate values, $z_i = 1$, with weights $\exp(-1.326)/[3 + 3\exp(-1.326)]$. This works out to 0.2098, and the residual is $0 - 0.2098 = -0.2098$. The remaining residuals are computed analogously. These residuals may be computed in R as follows:

```
> residuals(result.coxph, type="schoenfeld")
        6          10          15          25
-0.2098004  0.5566351 -0.3468347  0.0000000
```

Gramsch and Therneau [25] proposed scaling each residual by an estimate of its variance. This scaled residual may be conveniently approximated as follows:

$$r_i^* = r_i \cdot d \cdot \operatorname{var}\left(\hat{\beta}\right)$$

where d is the total number of deaths, and var $\left(\hat{\beta}\right)$ is the variance of the parameter estimate. (If there is more than one covariate, then this is a covariance matrix.) We may compute these as follows:

```
> resid.unscaled <- residuals(result.coxph, type="schoenfeld")
> resid.scaled <- resid.unscaled*result.coxph$var*sum(delta)
> resid.unscaled
   6          10          15          25
-0.2098004  0.5566351 -0.3468347  0.0000000
> resid.scaled
[1] -1.313064  3.483776 -2.170712  0.000000
```

Therneau and Gramsch showed that, if the hazard ratio is a function of t, $\beta(t)$, then the expected value of the scaled residuals is

$$E(r_i^*) \approx \beta + \beta(t)$$

so that an approximate estimate of $\beta(t)$ may be obtained by adding the estimate $\hat{\beta}$ from the Cox proportional hazards model (which assumes proportional hazards) to the standardized residuals,

```
> resid.scaled + result.coxph$coef
[1] -2.639193  2.157647 -3.496841 -1.326129
```

and plotting these versus time, or the log of time. This may be done more conveniently using the "cox.zph" function, which yields the same residuals directly,

```
> resid.sch <- cox.zph(result.coxph)
> resid.sch$y
         trt
6  -2.639193
10  2.157647
15 -3.496841
25 -1.326129
```

Here we see that the "y" component of "resid.sch" yields the sum of the scaled residuals and the parameter estimate from the Cox model.

The benefits of using Schoenfeld residuals become especially apparent when we apply the method to the pancreatic data. We compute the scaled Schoenfeld residuals as follows:

```
> result.coxph <- coxph(Surv(pfs.month) ~ stage)
> result.sch.resid <- cox.zph(result.coxph, transform="km")
```

The "transform" option specifies that the time axis is scaled to conform with Kaplan-Meier-transformed time. We may plot the residuals, and a smooth curve through them using a technique called "loess", as follows (Fig. 7.5):

```
> plot(result.sch.resid)
```

The shape of the smoothed (loess) curve is an estimate of the difference parameter as a function of time, which appears to be constant or slightly increasing initially, followed by a steady decline after about 2 months. Also given is a 95 %

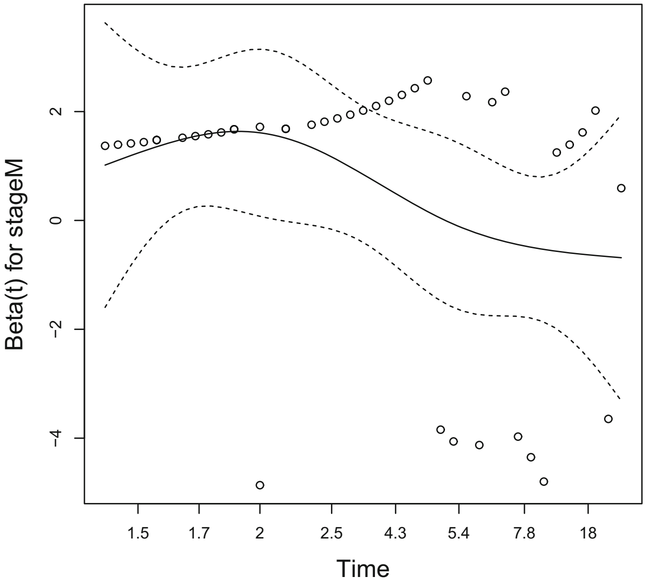


Fig. 7.5 Schoenfeld residual plot for the pancreatic cancer data

confidence band for the smooth curve. The hypothesis test for a constant β, which is a test of the proportional hazards function, may be obtained by fitting a straight line to the residuals plotted versus, yielding a p-value of 0.0496:

```
> result.sch.resid
          rho chisq      p
stageM -0.328  3.86 0.0496
```

Alternatively, variations of this test may be obtained by plotting $\beta(t)$ versus time or versus other transformations of time. The "cox.zph" function offers a "rank" option, where the time axis is ordered by the ranks of the times, and an "identity" option, where the time variable is untransformed. The results using these transformations are as follows:

```
> cox.zph(result.coxph, transform="rank")
          rho chisq      p
stageM -0.33  3.89 0.0486
> cox.zph(result.coxph, transform="identity")
          rho chisq     p
stageM -0.197  1.39 0.239
```

The rank transformation yields a similar p-value to what we found with the "km" transformation. The "identity" transform results in a much higher p-value. This is most likely due to the fact that deaths occur more sparsely at larger times, and as a result residuals at these larger times hold a great deal of influence on the fitted regression line. Thus, this transform is likely not a preferred one when failure times are not uniformly spaced over time. A wide range of tests for proportional hazards that have been developed may be viewed as linear regressions of Schoenfeld residuals versus various transformations of time. See Therneau and Grambsch [68] for further discussion of this point.

7.3 Additional Note

A thorough exposition of martingale residuals from a counting process point of view may be found in Chaps. 4 and 5 of Therneau and Grambsch [68]. Additional details are in chapter 11 of Klein and Moeschberger [36].

Exercises

7.1. Consider the case deletion and dfbeta residuals discussed in Sect. 7.1.2. For each of the covariates in the final pharmacoSmoking model (grp, employment levels 2 and 3 vs. 1, and age), plot the case deletion residuals versus the dfbeta residuals. Also plot the “dfbeta” residuals versus the “dfbetas” residuals. Do you see any differences?

7.2. Consider the CXCL17 model you fitted in Exercise 6.1. Check the functional form using martingale residuals, and use case-deletion residuals to identify any outlying points. Also use Schoenfeld residuals to check the proportional hazards assumption.

Chapter 8
Time Dependent Covariates

8.1 Introduction

The partial likelihood theory for survival data, introduced in Chap. 5, allows one to model survival times while accommodating covariate information. An important caveat to this theory is that the values of the covariates must be determined at time $t = 0$, when the patient enters the study, and remain constant thereafter. This issue arises with survival data because such data evolve over time, and it would be improper to use the value a covariate to model survival information that is observed before the covariate's value is known. To accommodate covariates that may change their value over time ("time dependent covariates"), special measures are necessary to obtain valid parameter estimates. An intervention that occurs after the start of the trial, or a covariate (such as air pollution exposure) that changes values over the course of the study are two examples of time dependent variables.

The rule is clear: we cannot predict survival using covariate values from the future. Unfortunately, this deceptively simple principle can ensnare even an experienced researcher. An oft cited and extensively studied example of this is the Stanford heart transplant study, published by Clark et al. in the Annals of Internal Medicine in 1971[9]. This study of the survival of patients who had been enrolled into the transplant program appeared to show that patients who received heart transplants lived significantly longer than those who did not. The data are in the "survival" package in a data set named "jasa" after a journal article that discussed analysis methods for the data. Here is a naive analysis:

```
> result.heart <- coxph(Surv(futime, fustat) ~ transplant + age +
+    surgery, data=jasa)
> summary(result.heart)

  n= 103, number of events= 75
                coef exp(coef) se(coef)      z Pr(>|z|)
transplant -1.71711   0.17958  0.27853 -6.165 7.05e-10 ***
age         0.05889   1.06065  0.01505  3.913 9.12e-05 ***
```

D. Moore, *Applied Survival Analysis Using R*, Use R!,
DOI 10.1007/978-3-319-31245-3_8

```
surgery    -0.41902   0.65769  0.37118 -1.129    0.259
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

The key covariate is "transplant", which takes the value 1 for those patients who received a heart transplant and 0 for those who did not. The estimate of the transplant coefficient is −1.717, and the p-value is very small. This result may appear to indicate (as it did to Clark et al. in 1971) that transplants are extremely effective in increasing the lifespan of the recipients. Soon after publication of this result, Gail [21], in an article in the same journal, questioned the validity of the result, and numerous re-analyses of the data followed. The problem here is that receipt of a transplant is a time dependent covariate; patients who received a transplant had to live long enough to receive that transplant. Essentially, the above analysis only shows that patients who live longer (i.e. long enough to receive a transplant) have longer lives than patients who don't live as long, which of course is a tautology.

A simple fix is to define a "landmark" time to divide patients into two groups. In this approach, patients who receive the intervention prior to the landmark go into the intervention group and those who did not are placed in the comparison group. Key requirements of this approach are that (a) only patients who survive up to the landmark are included in the study, and (b) all patients (in particular, those in the comparison group) remain in their originally assigned group regardless of what happens in the future, i.e., after the landmark. For example, for the heart transplant data, we may set a landmark at 30 days. We first select those patients who lived at least 30 days (79 of the 103 patients lived this long). Of these 79 patients, 33 had a transplant within 30 days, and 46 did not. Of these 46, 30 subsequently had a heart transplant, but we still count them in the "no transplant within 30 days" group. In this way we have created a variable (we shall call it "transplant30") which has a fixed value (that is, it does not change over time) for all patients in our set of 30-day survivors. Here is how we set things up:

```
> ind30 <- jasa$futime >= 30
> transplant30 <- {{jasa$transplant == 1} & {jasa$wait.
     time < 30}}
> summary(coxph(Surv(futime, fustat) ~ transplant30 + age +
+  surgery, data=jasa, subset=ind30 ))

  n= 79, number of events= 52

                     coef exp(coef) se(coef)      z Pr(>|z|)
transplant30TRUE -0.04214   0.95874  0.28377 -0.148   0.8820
age               0.03720   1.03790  0.01714  2.170   0.0300 *
surgery          -0.81966   0.44058  0.41297 -1.985   0.0472 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

The coefficient for transplant30 (a true/false indicator for transplant within the first 30 days) is −0.042, and the p-value is 0.88, which is not at all statistically significant. This "landmark method" analysis indicates that there is little or no difference in survival between those who got a transplant and those who did not. Although the landmark method is straightforward to implement, we have no

Table 8.1 Sample of six patients from the Stanford heart transplant data set

id	wait.time	futime	fustat	transplant
2	–	5	1	0
5	–	17	1	0
10	11	57	1	1
12	–	7	1	0
28	70	71	1	1
95	1	15	1	1

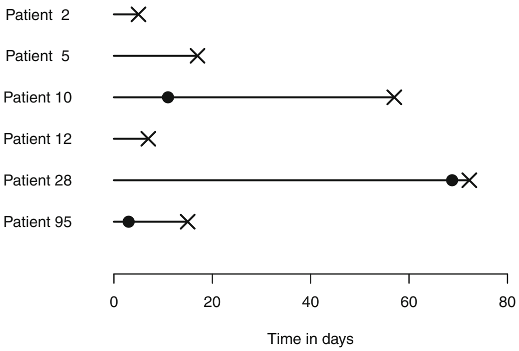


Fig. 8.1 Sample of six patients from the Stanford heart transplant data set. In this plot, death is denoted by an "X", and the time of transplant (for Patients 1, 3, and 6) by a *solid dot*. In the plot on the right, the timelines of patients who received a transplant are split into pre- and post-transplant components

guidance as to when to set the landmark. Why 30 days? Why not 15? Or why not 100? There is no clear way to answer this question. Furthermore, this 30-day landmark method requires that we discard almost a quarter of the patients from the analysis. Fortunately there is a better way, which is to directly model the variable "transplant" as a time dependent variable. This can be done in the framework of the classical Cox proportional hazards model, but important adjustments are required to obtain unbiased estimates. To see how to do this, it is helpful to look at a small data set, which we construct by selecting an illustrative subset of six patients, three of which had a transplant and three who did not (Table 8.1). We may plot them in Fig. 8.1.

We may set up the data in R as follows:

```
id <- 1:nrow(jasa)
jasaT <- data.frame(id, jasa)
id.simple <- c(2, 5, 10, 12, 28, 95)
heart.simple <- jasaT[id.simple,c(1, 10, 9, 6, 11)]
```

In this simple data set, all of the patients died within the follow-up time (stat = 1 for all patients). We may model the data incorrectly (ignoring the fact that "transplant" is time dependent) as follows:

```
> summary(coxph(Surv(futime, fustat) ~ transplant,
+   data=heart.simple))

  n= 6, number of events= 6

              coef exp(coef) se(coef)      z Pr(>|z|)
transplant -1.6878    0.1849   1.1718 -1.44     0.15
```

To do this correctly, we need to modify the partial likelihood function to accommodate these types of variables. Essentially, at each failure time, there are a certain number of patients at risk, and one fails, as we discussed in Chap. 5. However, the contributions of each subject can change from one failure time to the next. The hazard function is given by $h(t) = h_0(t)e^{z_k(t_i)\beta}$, where the covariate $z_k(t_i)$ is the value of the time-varying covariate for the kth subject at time t_i. The modified partial likelihood, in general, is as follows:

$$L(\beta) = \prod_{i=1}^{D} \frac{\psi_{ii}}{\sum\limits_{k \in R_i} \psi_{ki}}$$

where $\psi_{ki} = e^{z_k(t_i)\beta}$. In previous chapters the covariates were fixed at time 0, so that $z_k(t_i) = z_k$ for all failure times t_i, and the denominator at each time could be computed by, as time passes, successively deleting the value of ψ_i for the subject (or subjects) that failed at that time. With a time dependent covariate, by contrast, the entire denominator has to be recalculated at each failure time, since the values of the covariates for each subject may change from one failure time to the next. For example, from Table 8.1 and Fig. 8.1, we see that Patient #2 is the first to fail, at t $=$ 5. At this time, all six patients are at risk, but only one, Patient #2, has had a transplant at this time. So the denominator for the first factor is $5 + e^{\beta}$, and the numerator is 1, since it was a non-transplant patient who died. Patient 12 is the next to die, at time $t = 7$, and none of the patients in the risk set have changed their covariate value. But when the third patient dies, Patient #95, at $t = 15$, one of the other patients (#10) has switched from being a non-transplant patient to one who has had one. There are now four patients at risk, of which two (#10 and #95) are transplant patients. The denominator is thus $2 + 2e^{\beta}$ and the numerator is e^{β}, since it was a transplant patient that died. The full partial likelihood is

$$L(\beta) = \frac{1}{5 + e^{\beta}} \cdot \frac{1}{4 + e^{\beta}} \cdot \frac{e^{\beta}}{2 + 2e^{\beta}} \cdot \frac{1}{2 + e^{\beta}} \cdot \frac{e^{\beta}}{1 + e^{\beta}} \cdot \frac{e^{\beta}}{e^{\beta}}. \tag{8.1.1}$$

We may use the "coxph" function to accommodate time dependent variables by first pre-processing the data into what we shall call "start-stop" format. The validity of this approach may be derived from the counting process theory of partial likelihoods [68]. Essentially, this approach divides the time data for patients who had a heart transplant into two time periods, one before the transplant and one after. For example, Patient #10 was a non-transplant patient from entry until day 11. Since that patient received a transplant at that time, the future for that patient, had he or she not received a transplant, is unknown. Thus, we censor that portion of the patient's life experience at $t = 11$. Following the transplant, we start a new record for Patient #10. This second piece of the record is left-truncated at time $t = 11$, and a death is recorded at time $t = 57$. It is left-truncated because that patient's survival experience with the transplant starts at that point. For the first part of this patient's experience, the "start" time is 0, and the "stop" time is 11, which is recorded as a censored observation. For the second piece of that patient's experience, the start time is 11 and the stop time is 57. Thus, to put the data in start-stop format, the record of every patient with no transplant is carried forward as is, whereas the record of each patient who received a transplant is split into pre-transplant and post-transplant records. The R survival package includes a function "tmerge" to simplify this conversion. We may transform the "heartSimple" data set into start/stop format as follows:

```
> sdata <- tmerge(heart.simple, heart.simple, id=id,
+             death=event(futime, fustat),
+                 transpl=tdc(wait.time))
> heart.simple.counting <- sdata[,-(2:5)] # drop columns 2
    through 5
> heart.simple.counting
  id tstart tstop death transpl
1  2      0     5     1       0
2  5      0    17     1       0
3 10      0    11     0       0
4 10     11    57     1       1
5 12      0     7     1       0
6 28      0    70     0       0
7 28     70    71     1       1
8 95      0     1     0       0
9 95      1    15     1       1
```

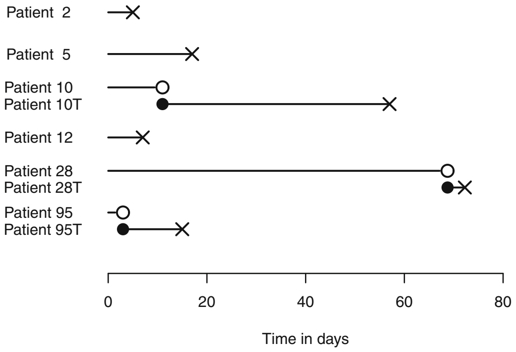
These data are diagrammed in Fig. 8.2. Once the data are in this format, we may use the coxph function as we did with left-truncated data:

```
> summary(coxph(Surv(tstart, tstop, death) ~ transpl,
+    data=heart.simple.counting))

  n= 9, number of events= 6

          coef exp(coef) se(coef)     z Pr(>|z|)
transpl 0.2846    1.3292   0.9609 0.296    0.767
```

Inspection of Fig. 8.2, when compared to Fig. 8.1, reveals that the partial likelihood is identical to that in Eq. 8.1.1.

We may apply this method to the full heart transplant data in the same way as described in Therneau and Crowson (2015) [67]. In the following, we define "tdata"

Fig. 8.2 Plot of sample of heart transplant patients in start-stop counting process format



as a temporary data set, leaving off the dates and transplant-specific covariates. Also, we add 0.5 to the death time on day 0, and break a tied transplant time.

```
> tdata <- jasa[, -c(1:4, 11:14)]
> tdata$futime <- pmax(.5, tdata$futime)
> indx <- {{tdata$wait.time == tdata$futime} &
+           !is.na(tdata$wait.time)}
> tdata$wait.time[indx] <- tdata$wait.time[indx] - .5
> sdata <- tmerge(tdata, tdata, id=1:nrow(tdata),
+                 death = event(futime, fustat),
+                 trans = tdc(wait.time))
> jasa.counting <- sdata[,c(7:11, 2:3)]
> head(jasa.counting)
  id tstart tstop death trans surgery      age
1  1      0    49     1     0       0 30.84463
2  2      0     5     1     0       0 51.83573
3  3      0    15     1     1       0 54.29706
4  4      0    35     0     0       0 40.26283
5  4     35    38     1     1       0 40.26283
6  5      0    17     1     0       0 20.78576
```

Patients 1, 2, and 3 did not have a transplant, so “tstart” takes the value 0 for all three, and “tstop” are the death times for those patients. For Patient 4, who had a heart transplant on day 35 and died on day 38, there are two records for each period of this patient’s experience, as described above. The results of fitting a time dependent Cox model are as follows:

```
> summary(coxph(Surv(tstart, tstop, death) ~ trans + surgery +
+    age, data=jasa.counting))

  n= 170, number of events= 75
            coef exp(coef) se(coef)      z Pr(>|z|)
trans    0.01405   1.01415  0.30822  0.046   0.9636
surgery -0.77326   0.46150  0.35966 -2.150   0.0316 *
age      0.03055   1.03103  0.01389  2.199   0.0279 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

We now see, as with the landmark analysis given earlier, that there is no evidence that receiving a heart transplant increases survival. This method is valid even though (unlike with the landmark method) no data are discarded.

8.2 Predictable Time Dependent Variables

An alternative way of modeling non-proportional hazards is to allow the coefficient for a particular covariate to vary with time. Specifically, if there is only one covariate, we have $h(t) = h_0(t)e^{z_k\beta(t)}$, where now it is β that varies with time (rather than the covariate z_k as in the previous section). Characterizing the functional form of the non-proportional hazards is a much harder problem than simply testing for a difference, as we did in Chap. 4. Although here it is the coefficient β that is changing rather than the covariate z, we may model this by defining a new time dependent variable with fixed coefficients that achieves the same effect. Because the time-varying relationship in the model is defined by the analyst, we refer to the variable as a predictable time dependent variable. In this section we will see how to use the pattern of Schoenfeld residuals to help us identify an appropriate time dependent function, and then model it using the time transfer function in the survival package.

8.2.1 *Using the Time Transfer Function*

Consider again the pancreatic data first discussed in Sect. 4.1. There we found that a log-rank test comparing the two groups did not yield a statistically significant result. Here we need to define a numerical (0/1) group variable, and fit the following model using the “pancreatic2” data in the “asaur” package:

```
> stage.n <- rep(0, nrow(pancreatic2))
> stage.n[pancreatic2$stage == "M"] <- 1
> result.panc <- coxph(Surv(pfs) ~ stage.n)
> result.panc

         coef exp(coef) se(coef)     z    p
stage.n 0.593      1.81    0.401 1.48 0.14

Likelihood ratio test=2.43  on 1 df, p=0.119
```

The p-value (0.14) for the likelihood ratio test, which is similar to that from the log-rank test in Sect. 4.1, shows little evidence of a group difference, as we saw there. Later in that section a plot of Schoenfeld residuals indicated that the hazard ratio appears not to be constant. One way of dealing with this was to use the Prentice modification of the Wilcoxon test (using “rho = 1” in the “survdiff” function). An alternative is to accommodate the changing hazard ratio by defining a time dependent covariate, $g(t) = z \cdot \log(t)$. In the survival package, the “time transfer” function “tt” allows us to do this. We define the “tt” function within the coxph function, and this function computes the necessary terms for the coxph fitting function, as follows:

```
> result.panc2.tt <- coxph(Surv(pfs) ~ stage.n + tt(stage.n),
+     tt=function(x,t, ...) x*log(t))
> result.panc2.tt
             coef exp(coef) se(coef)     z     p
stage.n      6.01   407.339    3.060  1.96 0.050
tt(stage.n) -1.09     0.338    0.589 -1.84 0.065

Likelihood ratio test=6.33  on 2 df, p=0.0423
```

The fitted function is $\beta(t) = 6.01 - 1.09 \cdot \log(t)$. Here we see that, while the p-value for the time dependent variable is 0.065, the likelihood ratio test for both stage and the time dependent variable together is 0.0423. This indicates that the group indicator combined with a time-varying hazard ratio yields evidence of a group difference. This is consistent with what we found in Sect. 4.1 using the weighted log-rank test with weights defined using the option "rho = 1". We may visually check this function by constructing a Schoenfeld residual plot (this time using a logarithmic transform scale), and then plotting the fitted function on the same plot,

```
result.sch.resid <- cox.zph(result.panc,
    transform=function(pfs) log(pfs))
plot(result.sch.resid)
abline(coef(result.panc2.tt), col="red")
```

Here the "transform" option in "cox.zph" is a log function defined within the function call. (As an alternative, one could define this simple function outside of "cox.zph" and then specify it by name within "cox.zph")

In this plot, the curved line is a loess (smooth) curve through the residuals. The tick marks on the horizontal axis follow a logarithmic scale, as specified by the "transform" argument in the "coxph.zph" function. The red line is from the fitted time transfer function, not from a fit to the residuals; it is a log function whose plot appears straight because the horizontal axis is a logarithmic scale. This time transfer function indicates that overall, the log hazard ratio decreases over time (Fig. 8.3).

Other time dependent functions may not yield this result. For example, if $g(t) = z \cdot t$, we get a non-significant result (p-value = 0.102) for the effect of "stage.n" on survival:

```
> coxph(Surv(pfs) ~ stage.n + tt(stage.n),
+    tt=function(x,t, ...) x*t)

                coef exp(coef) se(coef)     z     p
stage.n      1.27810     3.590  0.66103  1.93 0.053
tt(stage.n) -0.00366     0.996  0.00253 -1.44 0.150

Likelihood ratio test=4.56  on 2 df, p=0.102
```

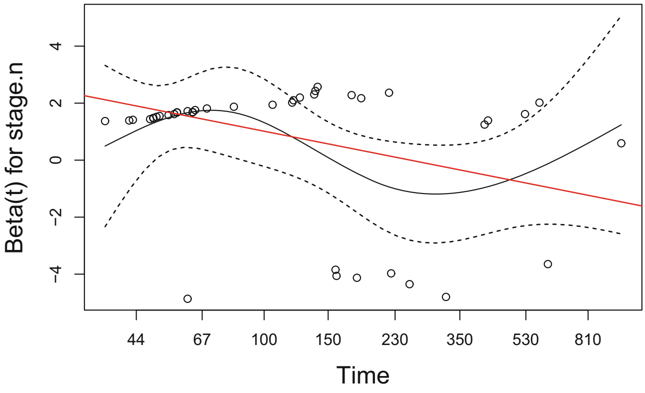


Fig. 8.3 Schoenfeld residual plot for the pancreatic data, using a log scale for time. The *curved line* is from a loess curve fitted to the residual plot, while the *straight red line* is based on the fitted time dependent estimate of $\beta(t)$ using the time transfer facility

Thus, it is important to identify a hazard-ratio function that well-approximates the actual changing hazard ratio.

8.2.2 Time Dependent Variables That Increase Linearly with Time

A common source of confusion is whether or not one could treat patient age as a time dependent variable. We have seen the use of "age at entry" as a covariate in survival analysis, and this is a fixed quantity at time 0; the age of a patient at that time is fixed by definition. But we know that the age of a patient increases in lock step with time itself, so can't we treat increasing age as a time dependent variable? The answer is yes, but doing so has no effect on the model. We could illustrate this with any survival data set that includes age as a covariate; for convenience, we shall choose an example from the "lung" data set in the survival package. This data set consists of survival times in days of 228 patients with advanced lung cancer. A number of covariates are included, but we shall focus on "age" to illustrate what happens when it is treated as time dependent. First, here is the result of fitting a model to this data with "age" (age at entry into the clinical trial) as the sole covariate:

```
> coxph(Surv(time, status==2) ~ age, data=lung)

      coef exp(coef) se(coef)    z     p
age 0.0187      1.02   0.0092 2.03 0.042

Likelihood ratio test=4.24  on 1 df, p=0.0395
```

We see that the log hazard increases with increasing age, with a p-value of 0.042. Now let us define "age" as a time dependent variable in the time transfer function, noting that "age" is in years, and the survival time, being measured in days, should be converted to years:

```
> coxph(Surv(time, status==2) ~ tt(age), data=lung,
+   tt=function(x, t, ...) {
+     age <- x + t/365.25
+     age})

          coef exp(coef) se(coef)    z     p
tt(age) 0.0187      1.02   0.0092 2.03 0.042

Likelihood ratio test=4.24  on 1 df, p=0.0395
```

There is no change at all in the fitted values. To see why this happens, let us denote age at entry into the trial by $z(0)$ and current age by $z(t) = z(0) + t$. Then the hazard function is given by

$$h(t) = h_0(t)e^{\beta z(t)} = \left\{h_0(t)e^{\beta t}\right\} \cdot e^{\beta z(0)}.$$

If one inserts this expression into the partial likelihood in Eq. 5.4.1, the time dependent part, $e^{\beta t}$, appears in both the numerator and the denominator of each factor, as does the baseline hazard. Both cancel, leaving only the age at entry variable $z(0)$. Thus, the coefficient β for the time dependent model is identical to that from the non-time dependent model. The same happens with any time dependent covariate that increases in lock step with time; continuous and unchanging exposure to a toxic substance would be a common example. However, if the variable doesn't change at a constant rate, this equivalence no longer holds. A simple example would be to use the log of current age, where $\{\text{current age}\} = \{\text{age at entry}\} + \{\text{survival time}\}$. See Exercise 8.5 for details.

8.3 Additional Note

Further details concerning time dependent covariates and the time-transfer function may be found in the vignette distributed with the R package on this topic (Therneau and Crowson [67]).

Exercises

8.1. Encode the log of the partial likelihood in Eq. 8.1.1 into an R function, and find the maximum using "optim" (as in Sect. 5.2). Verify that the result matches that from the "coxph" procedure in Sect. 8.1.

8.2. Consider the following synthetic time dependent data:

id	wait.time	futime	fustat	transplant
1	12	58	1	1
2	–	8	1	0
3	–	37	1	0
4	18	28	1	1
5	–	35	1	0
6	17	77	1	1

First model the data ignoring the wait time. Then transform the data into start-stop format, then use that form of the data to model "transplant" as a time dependent covariate. Write out the partial likelihood for these data, and use this partial likelihood to find the maximum partial likelihood estimate of the coefficient for transplant. Compare your answer to the results of "coxph".

8.3. For the pancreatic data, construct a Schoenfeld residual plot and loess smooth curve for an identity transform, using transform = "identity" in the coxph.zph function. Then fit a linear time transfer function, as in Sect. 8.2.1, and plot the fitted line on the residual plot.

8.4. Again using the pancreatic data, construct the residual plot and plot the transfer function for $g(t) = \log(t - 30)$. How does the evidence for a treatment effect differ from the result in Sect. 8.2.1 using $g(t) = \log(t - 30)$?

8.5. Using the lung data as in Sect. 8.2.2, compute log(age) and fit a Cox model using this as a fixed covariate. Then fit log(age) as a time dependent variable, using the time transfer function. Do the results differ? Why?

Chapter 9
Multiple Survival Outcomes and Competing Risks

Until now the type of survival data we have considered has, as an endpoint, a single cause of death, and the survival times of each case have been assumed to be independent. Methods for analyzing such survival data will not be sufficient if cases are not independent or if the event is something that can occur repeatedly. An example of the first type would be clustered data. For instance, one might be interested in survival times of individuals that are in the same family or in the same unit, such as a town or school. In this case, genetic or environmental factors mean that survival times within a cluster are more similar to each other than to those from other clusters, so that the independence assumption no longer holds. In the second case, if the event of interest is, for example, the occurrence of a seizure, the event may repeat indefinitely. Then we would have multiple times per person. Special methods are needed to handle these types of data structures, which we shall discuss in Sect. 9.1. A different situation arises when only the first of several outcomes is observable, a topic we will take up in Sect. 9.2.

9.1 Clustered Survival Times and Frailty Models

Example 9.1. Struewing et al. [64], in the Washington Ashkenazi study, examined the effect of mutations of the BRCA gene on risk of breast cancer in an Ashkenazi Jewish population. The original data set consisted of a set of probands who were volunteers of Ashkenazi ancestry. Each proband was genotyped for the BRCA breast cancer gene to determine if she was a mutation carrier. The proband was also interviewed by the investigators to determine if she had any female first-degree relatives, and the relatives age at the time she developed breast cancer or current age if that relative had never been diagnosed with breast cancer. A subset of this data set was constructed to use as an example. This subset consists of 1,960 families with two or more female relatives; for those with three or more female relatives, two were

D. Moore, *Applied Survival Analysis Using R*, Use R!,
DOI 10.1007/978-3-319-31245-3_9

selected at random. This data set, "askenazi", is in the "asaur" package. The subset was constructed so that it contains information on the age of onset of breast cancer (or current age for women without breast cancer) for the two female relatives, the BRCA mutation status of the proband, and the age of onset (or current age) of two female relatives. Following is a sample (sets 1, 9, and 94) of the family sets. We select them using the "%in" operator, which indicates set membership.

```
> ashkenazi[ashkenazi$famID %in% c(1, 9, 94), ]
   famID brcancer age mutant
1      1        0  73      0
2      1        0  40      0
7      9        0  89      0
8      9        1  60      0
87    94        1  44      1
88    94        0  45      1
```

Family #1 consists of two first degree female relatives (most likely a mother and daughter), ages 73 and 40. Neither of them has ever had breast cancer, nor does their proband have a BRCA mutation. In Family #9, one relative was 89 with no history of breast cancer, and the other relative had breast cancer at age 60. Their proband was not a BRCA mutation carrier. In Family #94 one relative had breast cancer at age 44 and the other, a sister, was 45 and had never had breast cancer. The proband for these sisters was a mutation carrier. Note that, since the variable "mutant" refers to a common proband for each family, both entries must either be 0 or 1, according to whether the proband was not a carrier (mutant = 0) or was a carrier (mutant = 1). The survival variable is age of onset (or age at interview for those with no history of breast cancer), the censoring variable is "brcancer" (1 if a breast cancer case, 0 if not), and the covariate of interest is "mutant", whether or not the relative's proband was a BRCA mutation carrier. A concern is that family members may share environmental and genetic characteristics (other than BRCA mutation status), so it may not be appropriate to treat them as independent.

Example 9.2. The Diabetic Retinopathy Study [33, 41, 44, 57] was a randomized clinical trial conducted to evaluate the effect of laser photo-coagulation versus control on time to onset of blindness. For each patient on eye was randomly assigned to receive the laser treatment and the other was untreated. The time of blindness was defined as the first occurrence of visual acuity less than 5/100 at two consecutive examinations; any eye that did not meet this criterion was treated as a censored observation. A secondary objective was to determine if diabetes type (early or late onset) affected time to blindness, and whether this influenced the effectiveness of the laser treatment. The data are in "diabetes" in the "timereg" package, which must be downloaded and installed. Here are the first few rows:

```
> library(timereg)
> head(diabetes)
  id     time status trteye treat adult agedx
1  5 46.24967      0      2     1     2    28
2  5 46.27553      0      2     0     2    28
3 14 42.50684      0      1     1     1    12
```

```
4 14 31.34145       1       1      0      1     12
5 16 42.30098       0       1      1      1      9
6 16 42.27406       0       1      0      1      9
```

For example, Patient #5 was observed for 46 months, and didn't lose sight in either eye. Patient #14 lost sight in the untreated eye (treat $= 0$ and status $= 1$) at 31 months, but still had sight in the treated eye (treat $= 1$ and status $= 0$) at 42 months. And Patient #16 had not lost sight in either eye at 42 months.

9.1.1 Marginal Survival Models

In the marginal approach, the proportional hazards assumption is presumed to hold for all subjects, despite the structure in the data due to clusters. With this approach, the parameter estimates are obtained from a Cox proportional hazards model as described in earlier chapters, ignoring the cluster structure. Where the clusters come into play is in computing standard errors for the parameter estimates. This model is described in detail in the pair of articles Wei, Lin, and Weissfeld [76] and Lin and Wei [45]. Suppose first that there is only one covariate, and its estimate is $\hat{\beta}$. We shall denote the estimate of its variance (from the Cox model, ignoring the clustering) by $\hat{V}$, and the standard error of the estimate is then $\hat{V}^{1/2} = \sqrt{\hat{V}}$. This is the estimate of the standard error assuming that all subjects are independent. To obtain a correction to this variance that accounts for the clustering structure, we need to define a score residual for subject j in cluster i:

$$s_{ij} = \delta_{ij}\left[z_{ij} - \bar{z}(t_{ij})\right] - \sum_{t_u \le t_{ij}} \left[z_i - \bar{z}(t_{ij})\right] e^{z_i \beta} \left[\hat{H}_0(t_u) - \hat{H}_0(t_{u-1})\right]$$

Note that the first part of this residual is the Schoenfeld residual given in Eq. 7.2.1. We formulate a quantity C defined by

$$C = \sum_{i=1}^{G} \sum_{j=1}^{n_i} \sum_{m=1}^{n_i} s_{ij} s_{im}.$$

We define a cluster-adjusted variance by $V^* = \hat{V}^2 \cdot C$ and a cluster-adjusted standard error for $\hat{\beta}$ by $\sqrt{V^*}$.

If there are q covariates, then $\hat{\beta}$ is a vector, and the covariance matrix for $\hat{\beta}$ from the Cox model is given by V, and the standard errors (based on the standard Cox model assuming independent subjects) is the square root of the diagonal elements of V, i.e. $\text{se}\left(\hat{\beta}\right) = [\text{diag}(V)]^{1/2}$.

The score residuals $\underset{\sim}{s}_{ij}$ are 1 X q matrices, the quantity C is now a q by q matrix given by $C = \sum_{i=1}^{G} \sum_{j=1}^{n_i} \sum_{m=1}^{n_i} \underset{\sim}{s}_{ij}^{\prime} \underset{\sim}{s}_{im}$, and the cluster-adjusted covariance matrix is given by $V^* = \hat{V} C \hat{V}$. Since the empirical covariance matrix C is sandwiched between two copies of $\hat{V}$, this estimate is often referred to as the "sandwich" estimator. The adjusted standard errors of the parameter estimate $\hat{\beta}$ is given by $\mathrm{se}\left(\hat{\beta}\right) = [\mathrm{diag}(V^*)]^{1/2}$. Summaries of the theory may be found in Klein and Moeschberger [36] and in Hosmer, Lemeshow and May [32].

9.1.2 Frailty Survival Models

Let us begin with a review of how we set up a likelihood function for survival data, and we will write it in a form that allows us to generalize to clustered survival data. Suppose for now that we have independent survival data, with the ith observation given by (t_i, δ_i, z_i). The likelihood function may be written as follows, using the fact that $h(t) = f(t)/S(t)$ and the proportional hazards assumption $h(t, \beta) = h_0(t) e^{z_i \beta}$:

$$L(\beta; z_i) = \prod_{i=1}^{n} f(t_i, \beta)^{\delta_i} S(t_i, \beta)^{1-\delta_i} = \prod_{i=1}^{n} h(t_i, \beta)^{\delta_i} S(t_i, \beta).$$

This may be re-expressed as

$$L(\beta; z_i) = \prod_{i=1}^{n} \left[h_0(t_i) e^{z_i \beta} \right]^{\delta_i} \cdot e^{-H_0(t_i) e^{z_i \beta}}$$

where $H_0(t_i) = -\int_0^{t_i} h_0(v) dv$ is the baseline cumulative hazard.

Now suppose that the survival times are organized into clusters; common examples include families, schools, or other institutions. We suppose that subjects in the same cluster are more alike in their survival times than are subjects from different clusters. One way to accommodate such structure in the data is to assign each individual in a cluster a common factor known as a *frailty* or, alternatively, as a *random effect*. We denote the frailty for all individuals in the ith cluster by ω_i. Then we may express the hazard function for the jth subject in the ith cluster as follows:

$$h_{ij}(t_{ij}) = h_0(t_{ij}) \cdot \omega_i e^{z_{ij} \beta}.$$

The ω_i vary from one cluster to another, and a common model that governs this variability is a gamma distribution,

$$g(\omega, \theta) = \frac{\omega^{\frac{1}{\theta} - 1} e^{-\frac{\omega}{\theta}}}{\Gamma\left(\frac{1}{\theta}\right) \theta^{\frac{1}{\theta}}}.$$

Suppose that, in addition to the survival and censoring variables, we could somehow observe the frailties ω_i. Then the joint likelihood for the jth subject in the ith cluster would be

$$L_{ij}(\beta,\theta;\omega_i,t_{ij},\delta_{ij},z_{ij}) = g(\omega_i,\theta)\cdot\left[h_0(t_{ij})\omega_i e^{z_{ij}\beta}\right]^{\delta_{ij}}\cdot e^{-H_0(t_{ij})\omega_i e^{z_{ij}\beta}},$$

and the full likelihood would be

$$L(\beta,\theta) = \prod_{i=1}^{G}\prod_{j=1}^{n_i} L_{ij}(\beta,\theta;\omega_i,t_{ij},\delta_{ij},z_{ij})$$

Still assuming that we could observe the frailties ω_i, we could obtain maximum likelihood estimates for β and θ by numerically maximizing this function. In actuality, however, the frailties are latent variables, that is, variables that we presume to exist but which we cannot directly observe. Thus, to obtain estimates of β and θ we need to use a multistage procedure called the EM (expectation-maximization) algorithm. This algorithm alternates between finding expected values for the frailties based on current estimates of β and θ and using these expected values to find updated estimates for β and θ. The algorithm alternates between these two steps until convergence. If we use a parametric distribution for $f(t,\beta)$, setting up the EM algorithm is fairly direct. Generalizing this to the Cox proportional hazards model is more complex, in part because we also have to obtain updated estimates for the baseline hazard at each step. See Klein and Moeschberger [36] and Therneau and Grambsch [68] for details.

An alternative to using the gamma distribution to model the frailties is to write $\omega_i = e^{\sigma u_i}$, where u_i has a standard normal distribution. This alternative model puts the random effects and fixed effects on the same level,

$$h_{ij}(t_{ij}) = h_0(t_{ij})\cdot e^{z_{ij}\beta+u_i\sigma}.$$

The EM algorithm for estimation of β and σ proceeds as outlined above.

9.1.3 *Accounting for Family-Based Clusters in the “ashkenazi” Data*

Let us first consider the “ashkenazi” data in the “asaur” package. Here we fit a standard Cox proportional hazards model to predict the age of onset of breast cancer depending on the carrier status of the proband:

```
> result.coxph <- coxph(Surv(age, brcancer) ~ mutant,
+   data=ashkenazi)
> summary(result.coxph)

  n= 3920, number of events= 473

         coef exp(coef) se(coef)     z Pr(>|z|)
mutant 1.1907    3.2895   0.1984 6.002 1.95e-09 ***
```

The log partial likelihood from this model is obtained as follows:

```
result.coxph$loglik
[1] -3579.707 -3566.745
```

The first component is from the model with no covariates and the second from the model with "mutant" included as a predictor. The likelihood ratio test statistic is twice the difference, $G^2 = 2(-3566.745 + 3579.707) = 25.924$. This is compared to a chi-square distribution with 1 degree of freedom, resulting in a very small p-value, confirming the importance of including the mutational status of the proband in the model.

To accommodate the clustering using the Lin-Wei method [45], we use the "cluster" term in the model specification, as follows:

```
> result.coxph.cluster <-  coxph(Surv(age, brcancer) ~ mutant +
+                           cluster(famID), data=ashkenazi)
> summary(result.coxph.cluster)

  n= 3920, number of events= 473

         coef exp(coef) se(coef) robust se     z Pr(>|z|)
mutant 1.1907    3.2895   0.1984    0.2023 5.886 3.96e-09 ***
```

The parameter estimate and it's standard error are the same as in the ordinary Cox model above. Here, however, there is an additional estimate of the standard error, the "robust se", that is computed using the Lin-Wei method. This estimate is only slightly higher than the one from the standard Cox model, indicating that the effect of clustering within first-degree relatives is small. The p-value is higher, but still highly significant, indicating that having a first-degree relative who is a BRCA mutation carrier increases the hazard of developing breast cancer by a factor of 3.30.

As an alternative, we may account for the clustering using a gamma frailty model as follows:

```
> result.coxph.frail <-  coxph(Surv(age, brcancer) ~ mutant +
+                         frailty(famID), data=ashkenazi)
> summary(result.coxph.frail)

  n= 3920, number of events= 473

               coef  se(coef) se2    Chisq  DF    p
mutant         1.272 0.2317   0.2004  30.13   1.0 4.0e-08
frailty(famID)                       221.50 211.6 3.1e-01
```

Here we see that the coefficient for the "mutant" term is similar as before. The coxph function with the frailty option provides two estimates of the standard error. The first, "se(coef)" is generally the preferred estimate [66], while the "se2" estimate is an alternative estimate based on a variation of the sandwich estimator [26, 66]. By default "frailty" uses the gamma frailty distribution; to use a normally distributed frailty use "frailty(dist='normal')".

A newer facility for fitting random effects frailty models, which supersedes the "frailty" option, is the "coxme" package, which must be separately downloaded and installed. It may be used as follows:

```
 1  > library(coxme)
 2  > result.coxme <- coxme(Surv(age, brcancer) ~ mutant +
 3  +                     (1|famID), data=ashkenazi)
 4  > summary(result.coxme)
 5  Cox mixed-effects model fit by maximum likelihood
 6    Data: ashkenazi   events,
 7    n = 473, 3920
 8    Iterations= 10 63
 9                      NULL Integrated    Fitted
10  Log-likelihood -3579.707  -3564.622 -3411.522
11
12                     Chisq    df          p   AIC     BIC
13  Integrated loglik  30.17   2.0 2.8100e-07 26.17   17.85
14  Penalized loglik  336.37 150.1 2.2204e-16 36.16 -588.13
15
16  Model:  Surv(age, brcancer) ~ mutant + (1 | famID)
17  Fixed coefficients
18              coef exp(coef)  se(coef)    z       p
19  mutant 1.236609  3.443914 0.2205358 5.61 2.1e-08
20
21  Random effects
22   Group Variable  Std Dev   Variance
23   famID Intercept 0.5912135 0.3495334
```

The model statement " ~ mutant + (1 | famID)" states that we are fitting "mutant" as a fixed effect; the expression "(1 | famID)" indicates that family members, defined by the variable "famID" are nested within family. In the output, line 8 contains convergence information from the fitting process. Line 10 contains values of the partial log likelihood. The first ("NULL") value is the log partial likelihood from the model with no covariates, and is identical to the null value from the Cox model given at the beginning of this subsection; the second ("Integrated") is from the partial log-likelihood with the random effect terms integrated out. To find the statistical significance of including a random effect in the model, we compare the integrated log-likelihood value (−3564.622) to the value given earlier for the maximum (partial) log likelihood for the ordinary Cox model with "mutant" in the model (−3566.745). Twice the difference is $G^2 = 2(-3564.622 + 3566.745) = 4.246$. The p-value is thus 0.039, obtained as follows:

```
> pchisq(4.246,1,lower.tail=F)
[1] 0.03934289
```

The "Integrated loglik" chi-square statistic on Line 13, 30.17, is twice the difference between the integrated and null values of the log-likelihood on line 10. This is a combined test of the fixed effect "mutant" and the random effect, and is compared to a chi-square distribution with 2 degrees of freedom, yielding a highly significant p-value. The fixed effect parameter estimates (lines 17–19) give the coefficient estimate for "mutant", the estimated risk ratio (3.44) for those with a mutation-carrying relative compared to those without, and the Wald test p-value which, as we have seen, shows a highly significant association with risk of onset of breast cancer. Finally, the variance of the random effect (0.35) and it's square root, the standard deviation (0.59) are given in Line 23. Also shown are some additional terms, such as the fitted log-likelihood (Line 10) and the penalized log likelihood (Line 14), that have more specialized uses.

9.1.4 Accounting for Within-Person Pairing of Eye Observations in the Diabetes Data

We shall examine the effect of treatment on time-to-blindness in patients with diabetic retinopathy, and also the interaction of the effect of treatment with age of onset (early or adult onset), as defined by the "adult" indicator. The results, using the "coxme" function, are as follows:

```
 1  > result.coxme <- coxme(Surv(time, status) ~ treat +
 2  +  as.factor(adult) + treat*as.factor(adult) + (1 | id),
 3  +  data=diabetes)
 4  > summary(result.coxme)
 5  Cox mixed-effects model fit by maximum likelihood
 6   Data: diabetes   events,
 7   n = 155, 394
 8   Iterations= 7 39
 9                      NULL Integrated    Fitted
10  Log-likelihood -867.9511  -847.3837 -761.3231
11
12                     Chisq    df          p   AIC     BIC
13  Integrated loglik  41.13  4.00 2.5207e-08 33.13   20.96
14  Penalized loglik 213.26 77.99 1.6542e-14 57.28 -180.07
15
16  Model:  Surv(time, status) ~ treat + as.factor(adult) +
17                  treat * as.factor(adult) + (1 | id)
18  Fixed coefficients
19                             coef exp(coef)se(coef)     z      p
20  treat                   -0.4998 0.60667  0.22541 -2.22 0.0270
21  as.factor(adult)2        0.3995 1.49103  0.24568  1.63 0.1000
22  treat:as.factor(adult)2 -0.9681 0.37981  0.36164 -2.68 0.0074
23
24  Random effects
25   Group Variable  Std Dev   Variance
26   id    Intercept 0.9171924 0.8412418
```

Clearly, treatment increases time to blindness (Line 20), since the coefficient is negative (−0.5) with a p-value of 0.027. Those with adult-onset diabetes are at increased risk for early blindness (Line 21), but treatment has a greater beneficial effect for those with adult-onset diabetes than it does for those with juvenile-onset diabetes (Line 22). Several researchers have reported similar results with this data [41, 57].

9.2 Cause-Specific Hazards

Until now we have considered survival times with a single, well-defined outcome, such as death or some other event. In some applications, however, a patient may potentially experience multiple events, only the first-occurring of which can be observed. For example, we may be interested in time from diagnosis with prostate cancer until death from that disease (Cause 1) or death from some other cause (Cause 2), but for a particular patient we can only observe the time to the first event. Of course, as discussed in previous chapters, a patient may also be censored if he is still alive at the last follow-up time. If interest centers on a particular outcome, time to prostate cancer death, for example, a simplistic analysis method would be to treat death from other causes as a type of censoring. This approach has the advantage that implementing it is straightforward using the survival analysis methods we have discussed in previous chapters. However, a key assumption about censoring is that it is independent of the event in question. In most competing risk applications, this assumption may be questionable, and in some cases may be quite unrealistic. Furthermore, it is not possible to test the independence assumption using only the competing risks data. The only hope of evaluating the accuracy of the assumption would be to examine other data or appeal to theories concerning the etiology of the various death causes. Consequently, interpretation of survival analyses in the presence of competing risks will always be subject to at least some ambiguity due to uncertainty about the degree of dependence among the competing outcomes.

9.2.1 Kaplan-Meier Estimation with Competing Risks

We begin with estimating a survival curve in a single sample in the presence of competing events. The simplest method, as we have noted above, would be to in turn select each as the primary event, and to treat the other as a censoring event. However, to obtain unbiased estimates of survival curves, this simplistic method would require the usually false assumption that the two causes of death are independent. We may illustrate this problem be considering prostate cancer patients ages 80 and over diagnosed with stage T2 poorly differentiated prostate cancer. We define indicator variables “status.other” and “status.prost”, and then select the subset “prostateSurvival.highrisk” as follows, using the “prostate survival” data from Example 1.4 in Chap. 1:

```
> prostateSurvival <- within(prostateSurvival, {
+    status.prost <- as.numeric({status == 1})
+    status.other <- as.numeric({status == 2})})
> attach(prostateSurvival)
> prostateSurvival.highrisk <- prostateSurvival[{{grade ==
   "poor"} &
+  {stage=="T2"} & {ageGroup == "80+"}},]
> head(prostateSurvival.highrisk)
```

```
   grade stage ageGroup survTime status status.other status.prost
13  poor    T2      80+       21      0            0            0
38  poor    T2      80+      105      0            0            0
41  poor    T2      80+        2      1            0            1
47  poor    T2      80+       67      2            1            0
78  poor    T2      80+        2      0            0            0
93  poor    T2      80+       60      2            1            0
```

Let us consider two analyses, one with death due to other causes (status = 2) as censored, and the other with death due to prostate cancer (status = 1) as censored. We set these up as follows:

```
> status.prost <- {prostateSurvival.highrisk$status == 1}
> status.other <- {prostateSurvival.highrisk$status == 2}
```

The Kaplan-Meier estimates of survival defined as time to death from prostate cancer (with other causes of death considered as censored) is as follows:

```
> result.prostate.km <- survfit(Surv(survTime, event=status.
    prost) ~ 1,
+   data=prostateSurvival.highrisk)
```

Similarly, to estimate survival for time to death from other causes, we have

```
> result.other.km <- survfit(Surv(survTime, event=status.
    other) ~ 1,
+   data=prostateSurvival.highrisk)
```

To illustrate the problem with this analysis, let us first extract the Kaplan-Meier survival curve for death from other causes:

```
> surv.other.km <- result.other.km$surv
> time.km <- result.other.km$time/12
```

Now let's extract the corresponding survival curve for death from prostate cancer, and then express it as a cumulative incidence function, which is one minus the survival curve (also known as the cumulative distribution function):

```
> surv.prost.km <- result.prostate.km$surv
> cumDist.prost.km <- 1 - surv.prost.km
```

Now we may plot both on the same graph, using the plot option 'type = "s" ' to produce step functions:

```
> plot(cumDist.prost.km ~ time.km, type="s", ylim=c(0,1), lwd=2,
+   xlab="Years from prostate cancer diagnosis",  col="blue")
> lines(surv.other.km ~ time.km, type="s", col="green", lwd=2)
```

The result, shown in Fig. 9.1, shows that the two curves cross. At 10 years, for example, the probability of dying of prostate cancer is 0.46, and of other causes it is 0.88. The fact that the sum of these two probabilities exceeds one demonstrates that these estimates, viewed as probabilities that a particular patient would die of prostate cancer or something else, are severely biased. One might be tempted to view these curves as estimates of the probability of death from one cause if the other cause were eliminated as a possibility, but such an exercise would require the assumption that the causes be independent. This assumption cannot be tested from the data, and in any case the meaning of the resulting estimates would be purely hypothetical.

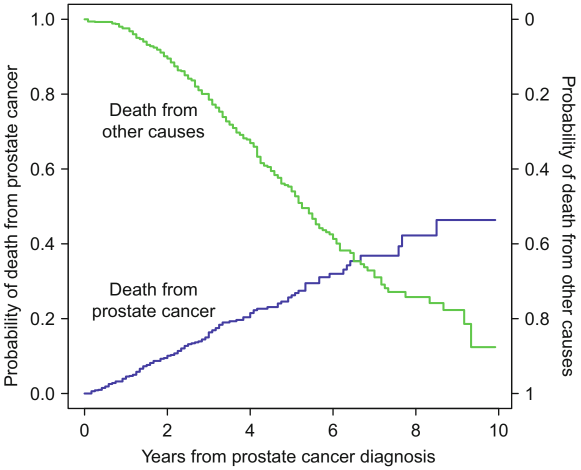


Fig. 9.1 Kaplan-Meier estimates of the probabilities of death from prostate cancer and from other causes

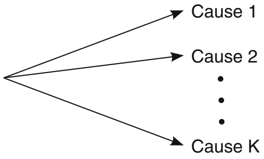


Fig. 9.2 Subject can die of only one of K causes

9.2.2 Cause-Specific Hazards and Cumulative Incidence Functions

To develop a formal model to accommodate competing risks, let us suppose that there are K distinct causes of death, which we may diagram as in Fig. 9.2.

The distinguishing feature of this competing causes framework is that each subject can experience at most one of the K causes of death; the times that the subject would have experienced the remaining causes is thus unknown. This framework can also accommodate applications with non-fatal events, as long as all of the events are mutually exclusive. For example, in individuals infected with HIV, one may be interested in the time to development of symptoms of AIDS, or the appearance of a condition called syncytium inducing (SI) HIV phenotype (Putter et al. [56]; Anderson et al. [4]). With competing risks, it is helpful to define, for each cause of interest, a function known as the cumulative risk function, also called the sub-distribution function. This is the cumulative probability that an individual dies from that particular cause by time t, and is given by

$$F_j(t) = \Pr(T \le t, C = j) = \int_0^t h_j(u)S(u)\,du.$$

This function is similar to the cumulative distribution function in that it is always increasing (or more precisely, non-decreasing). But unlike a cumulative distribution function, it goes, in the limit, to the probability of death from that particular cause, rather than to 1. Formally, we have

$$F_j(\infty) = \Pr(C = j)\,.$$

The cause-specific hazard is defined in a manner similar to the hazard function from Chap. 2, but now it is the probability that a specific event occurs at time t given that the individual survives that long:

$$h_j(t) = \lim_{\delta \to 0} \left(\frac{\Pr(t < T < t + \delta, C = j | T > t)}{\delta} \right).$$

If we add up all of the cause-specific hazards at a particular time, we get the hazard function of Chap. 2:

$$h(t) = \sum_{j=1}^{K} h_j(t).$$

That is, the risk of death at a particular time is the sum of the risks of all of the specific causes of death at that time.

Suppose now that we have D distinct ordered failure times t_1,t$_2$,..., t_D. We may estimate the hazard at the ith time t_i using $\hat{h}\,(t_i) = d_i/n_i$, as we have seen in previous chapters. The cause-specific hazard for the kth hazard may be written in a similar form as $\hat{h}_k\,(t_i) = d_{ik}/n_i$. This is just the number of events of type k at that time divided by the number at risk at that time. The sum over all cause-specific hazards is the overall hazard, $\hat{h}(t_i) = \left(\sum_k d_{ik}\right)/n_i$. The probability of failure from any cause at time t_i is the product of $\hat{S}(t_{i-1})$, the probability of being alive just before t_i, and $\hat{h}\,(t_i)$, the risk of dying at t_i. Similarly, the probability of failure *due to cause k* at that time is $\hat{S}(t_{i-1})\hat{h}_k(t_i)$. The sub-distribution function, or cumulative incidence function, is the probability of dying of cause k at time t_i. This is the sum of all probabilities of dying of this cause up to time t_i and is given by

$$\hat{F}_k(t) = \sum_{t_i \le t} \hat{S}(t_{i-1})\hat{h}_k(t_i).$$

That is, once we have an estimate of the overall survival function $\hat{S}(t)$, we can obtain the cumulative incidence function for a particular cause by summing over the product of this and the cause-specific hazards for that cause.

To illustrate this methodology, let us consider a simple hypothetical data set with six observations and two possible causes of death, displayed in Fig. 9.3.

Denoting the event types with the numbers 1 and 2, and the censored observations with the number 0, we may enter the data into R as follows:

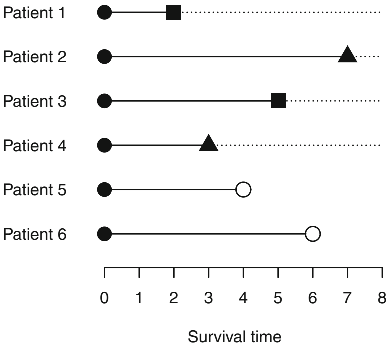


Fig. 9.3 Competing risk survival data. *Squares* represent failure from cause 1 and triangles from cause 2. *Open circles* represent censored observations

```
> tt <- c(2,7,5,3,4,6)
> status <- c(1,2,1,2,0,0)
```

We first compute the overall survival distribution,

```
> status.any <- as.numeric(status >= 1)
> result.any <- survfit(Surv(tt, status.any) ~ 1)
> result.any$surv
[1] 0.8333333 0.6666667 0.6666667 0.4444444 0.4444444 0.0000000
```

We compute the cumulative incidence functions as in the following table:

Time	n.risk	n.event.1	n.event.2	n.event.any	Survival	h.1	h.2	CI.1	CI.2
2	6	1	0	1	0.833	1/6	0	0.167	0.000
3	5	0	1	1	0.667	0	1/5	0.167	0.167
5	3	1	0	1	0.444	1/3	0	0.389	0.167
7	1	0	1	1	0.1000	0	1	0.389	0.611

For example, the probability of event type 1 at the first time ($t = 2$) is given by $1.000 \times \frac{1}{6} = 0.167$. This is also the estimate of the cumulative incidence function at this time. The probability of an event of this type at time $t = 5$ is $0.667 \times \frac{1}{3} = 0.222$. Then the cumulative incidence for this event at time $t = 5$ is $0.167 + 0.222 = 0.389$. These results may be more easily obtained using the "Cuminc" function in the "mstate" R package [14]:

```
> library(mstate)
> ci <- Cuminc(time=tt, status=status)
> ci
  time      Surv  CI.1  CI.2   seSurv seCI.1 seCI.2
1    0 1.00e+00 0.000 0.000 0.00e+00  0.000  0.000
2    2 8.33e-01 0.167 0.000 1.52e-01  0.152  0.000
3    3 6.67e-01 0.167 0.167 1.92e-01  0.152  0.152
4    5 4.44e-01 0.389 0.167 2.22e-01  0.219  0.152
5    7 4.93e-17 0.389 0.611 2.47e-17  0.219  0.219
```

The standard errors for the survival curve are computed using Greenwood's formula as discussed in Chap. 3. The standard errors for the cumulative incidence functions are computed in an analogous manner; see Putter et al. [56] for details.

9.2.3 Cumulative Incidence Functions for Prostate Cancer Data

Returning to the prostate cancer example of Fig. 9.1, we may now estimate the competing risks cumulative incidence functions using the "Cuminc" function in the R package "mstate" as follows:

```
> library(mstate)
> ci.prostate <- Cuminc(time=prostateSurvival.highrisk$survTime,
+  status=prostateSurvival.highrisk$status)
```

The first few lines of the resulting file "ci.prostate" are as follows:

```
> head(ci.prostate)
  time  Surv    CI.1    CI.2  seSurv  seCI.1  seCI.2
1    0 1.000 0.00000 0.00000 0.00000 0.00000 0.00000
2    1 0.994 0.00000 0.00592 0.00264 0.00000 0.00264
3    2 0.988 0.00602 0.00592 0.00376 0.00269 0.00264
4    3 0.984 0.00848 0.00715 0.00430 0.00319 0.00291
5    4 0.983 0.00973 0.00715 0.00447 0.00342 0.00291
6    5 0.978 0.01477 0.00715 0.00511 0.00423 0.00291
```

The first column, "time" is the time in months. The column "Surv" is the Kaplan-Meier survival estimate for time to death from any cause (prostate or something else). The next two columns are the cumulative incidence function estimates for causes 1 (prostate) and 2 (other). The remaining columns are standard errors of the respective estimates. We may plot the cause-specific cumulative incidence functions as follows:

```
ci1 <- ci.prostate$CI.1  # CI.1 is for prostate cancer
ci2 <- ci.prostate$CI.2  # CI.2 is for other causes
times <- ci.prostate$time/12  # convert months to years
Rci2 <- 1 - ci2
```

We may plot the cumulative incidence function for death from prostate cancer, and for death from other causes in solid green and blue, respectively, and the previous estimates with thin lines of the same (but lighter) colors,

```
> plot(Rci2 ~ times, type="s", ylim=c(0,1), lwd=2, col="green",
+   xlab="Time in years", ylab="Survival probability")
> lines(ci1 ~ times, type="s", lwd=2, col="blue")
> lines(surv.other.km ~ time.km, type="s",
    col="lightgreen", lwd=1)
> lines(cumDist.prost.km ~ time.km, type="s",
    col="lightblue", lwd=1)
```

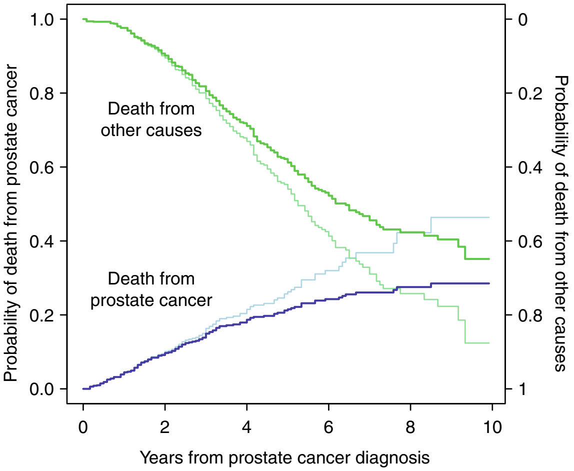


Fig. 9.4 Cumulative incidence of death from prostate cancer and from other causes, compared to the Kaplan-Meier estimates

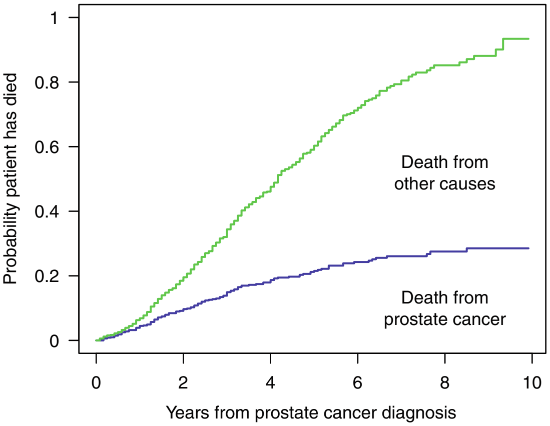


Fig. 9.5 Stacked cumulative incidence functions of death from prostate cancer and from other causes

Figure 9.4, analogous to one presented by Putter et al. [56] for a different data set, clearly illustrates the value of displaying competing risks cumulative incidence functions. These curves represent estimates of the actual probabilities that a patient will die of a particular cause, rather than hypothetical probabilities that he would die of one cause in the absence of the other.

A common way to display competing risk cumulative incidence curves is via a stacked plot, as shown in Fig. 9.5. The lower, blue curve represents the cumulative probability of death from prostate cancer, and the difference between the blue and upper, green curve represents the probability of death from other causes. The sum of the two probabilities of death, i.e. the upper, green curve, represents the cumulative probability of death from any cause, and is equal to one minus the Kaplan-Meier survival curve for death from any cause.

9.2.4 Regression Methods for Cause-Specific Hazards

When there is a single outcome of interest, the Cox proportional hazards model provides an elegant method for accommodating covariate information. However, modeling covariate information for competing risks data presents special challenges, since it is difficult to define precisely the hazard function on which the covariates should operate. The first method we will discuss, discussed in detail by Putter et al. [56] and de Wreede, Fiocco, and Geskus [14], is the most direct. We will illustrate using the prostate cancer data, this time restricting our attention (for now) to patients with stage T2 prostate cancer. Essentially, we will study the effects of the remaining covariates (grade and age) on prostate cancer death, treating other causes of death as censoring indicators, and vice versa for the effects of the covariates on other causes of death. We set up the data as follows:

```
prostateSurvival.T2 <- prostateSurvival[prostateSurvival$stage
   =="T2",]
attach(prostateSurvival.T2)
```

We then fit a standard Cox model for prostate cancer death as follows:

```
> result.prostate <- coxph(Surv(survTime, status.prost) ~ grade +
+           ageGroup)
> summary(result.prostate)


                  coef exp(coef) se(coef)      z Pr(>|z|)
gradepoor       1.2199    3.3867   0.1004 12.154  < 2e-16 ***
ageGroup70-74 -0.2860    0.7513   0.2595 -1.102   0.2704
ageGroup75-79  0.4027    1.4958   0.2257  1.784   0.0744 .
ageGroup80+    0.9728    2.6454   0.2148  4.529 5.92e-06 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

These results show that patients having poorly differentiated disease (grade = poor) have much worse prognosis than do patients with moderately differentiated disease (the reference group here), with a log-hazard ratio of 1.2199. These results also show that the hazard of dying from prostate cancer increases with increasing age of diagnosis (the reference is the youngest age group, 65–69).

Considering death from other causes as the event of interest, we have

```
> result.other <- coxph(Surv(survTime, status.other) ~ grade +
+  ageGroup)
> summary(result.other)


                  coef exp(coef) se(coef)     z Pr(>|z|)
gradepoor      0.28104   1.32451  0.05875 4.784 1.72e-06 ***
ageGroup70-74 0.09462   1.09924  0.12492 0.757  0.44879
ageGroup75-79 0.31330   1.36793  0.11709 2.676  0.00746 **
ageGroup80+    0.79012   2.20367  0.11204 7.052 1.76e-12 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Taken at face value, these results indicate that patients with poorly differentiated cancer have a higher risk of death from non-prostate-cancer related disease than do those with moderately differentiated disease. While the log hazard ratio is much smaller than with prostate cancer death as the outcome (0.28104 vs. 1.2199), one might expect that cancer grade wouldn't have any effect on death from non-prostate-cancer causes. These hazard ratios refer to hazard functions for death from prostate cancer and for death from other causes, and these are assumed to be operating independently. As we have discussed previously, these assumptions are highly suspect, and it is unclear to what extent the hazard functions that have been estimated correspond to actual (and unobservable) hazards.

To address this issue, Fine and Gray developed an alternative method for modeling covariate data with competing risks. Instead of defining the effects of covariates on the cause-specific hazards, they define a "sub-distribution hazard"

$$\bar{h}_k(t) = \lim_{\delta \to 0} \frac{pr(t < T_k < t + \delta | E)}{\delta}$$

where the conditional event is given by

$$E = \left\{ \{T_k > t\} \text{ or } \left\{ T_{k'} \leq t \, \text{and} \, k' \neq k \right\} \right\}.$$

That is, the sub-distribution hazard for cause k, like the definition of the ordinary hazard function given in Chap. 2, is essentially the probability that the failure time lies in a small interval at t conditional on an event E, divided by the length of that small interval. The difference is that, in addition to referring to the kth failure time, the conditioning set specifies not only that $T_k > t$ but also allows inclusion of events other than the kth event in question, in which case we must have $T_{k'} \leq t$. Thus, when computing these sub-distribution hazards, the risk set includes not only those currently alive and at risk for the kth event type, but also those who died earlier of other causes.

Consider, for example, for the data in Fig. 9.3, the risk set for death from Cause #2 (triangles) at time $t = 7$ consists not only of Patient 2, the sole patient still alive at that time, but also Patients 1 and 3, since they died of Cause #1 (squares) earlier. Patient 4 is not in the risk set for death from Cause #2 at time $t = 7$ since that person died earlier from Cause #2, the same cause as Patient 2. Patients 5 and 6 also are not in the risk set at this time since they were censored. The sub-distribution hazard may be written in a more compact equivalent form as

$$\bar{h}_k(t) = -\frac{\mathrm{d} \log(1 - F_k(t))}{\mathrm{d}t}.$$

The Fine and Gray method uses these sub-distribution hazards for modeling the effects of covariates on a specific cause of death analogously to the Cox model,

$$\bar{h}_k(t; z, \beta) = \bar{h}_{0k}(t) e^{z\beta}.$$

That is, the sub-distribution hazard for a subject with covariates z is proportional to a baseline sub-distribution function $\bar{h}_{0k}(t)$.

The Fine and Gray methods are implemented in the ”crr” function in the R package “cmprsk”. Before we can use the competing risk function “crr” in this package, we need to put the covariates into a model matrix using the “model.matrix” function. Using our attached data set “prostateSurvival.T2”, we do this as follows:

```
> cov.matrix <- model.matrix(~ grade + ageGroup)
> head(cov.matrix)
    (Intercept) gradepoor ageGroup70-74 ageGroup75-79 ageGroup80+
1         1         0           1             0            0
2         1         1           0             1            0
3         1         1           0             1            0
4         1         1           0             0            1
5         1         0           0             1            0
6         1         0           0             1            0

> cov.matrix.use <- cov.matrix[,-1] # drop the first column
```

We obtain estimates for the prostate cancer as follows, dropping the first (intercept) column of the covariate matrix:

```
> library(cmprsk)
> result.prostate.crr <- crr(survTime, status, cov1=cov.
    matrix[,-1],
+   failcode=1)

                 coef exp(coef) se(coef)      z p-value
gradepoor       1.132     3.102    0.101 11.20 0.00000
ageGroup70-74 -0.272     0.762    0.253 -1.08 0.28000
ageGroup75-79  0.367     1.443    0.219  1.67 0.09400
ageGroup80+    0.799     2.224    0.208  3.85 0.00012
```

The argument “failcode=1” refers to death from prostate cancer. For death from other causes, we use “failcode=2”,

```
> result.other.crr <- crr(survTime, status, cov1=cov.matrix[,-1],
+   failcode=2)
> summary(result.other.crr)

                coef exp(coef) se(coef)     z p-value
gradepoor      0.126      1.13   0.0584 2.154 3.1e-02
ageGroup70-74 0.103      1.11   0.1252 0.824 4.1e-01
ageGroup75-79 0.273      1.31   0.1176 2.323 2.0e-02
ageGroup80+   0.667      1.95   0.1128 5.917 3.3e-09
```

Again we see that poorly differentiated patients have higher risk for death from other causes (risk ratio = 0.126), but the effect size is smaller than we obtained from the Putter et al. method (risk ratio 0.281). The estimated effect on death from prostate cancer of having poorly differentiated disease is similar for both methods (risk ratio of 1.22 for Putter et al. vs. 1.132 for Fine and Gray).

9.2.5 Comparing the Effects of Covariates on Different Causes of Death

An advantage of the Putter et al. method over the Fine and Gray method is the ease with which we can compare the effects of a covariate on, for example, death from prostate cancer and death from other causes. For example, we know that the risk of both causes of death increase with age. But does the effect of age differ for these two causes? To answer this question, we first need to convert the data set from the original one where each patient has his own row in the data set into one where each patient's data is split into separate rows, one for each cause of death. In the prostate cancer case, we need to create, for each patient, two rows, one for death from prostate cancer and one for death from other causes. To simplify this process, we can use utilities in the "mstate" package. This package is capable of handling complex multistate survival models, but can also be used to set up competing risks as a special case. We begin by setting up a "transition" matrix using the function "trans.comprisk",

```
> tmat <- trans.comprisk(2, names = c("event-free", "prostate",
      "other"))
> tmat            to
from        event-free prostate other
  event-free        NA        1     2
  prostate          NA       NA    NA
  other             NA       NA    NA
```

The first argument is the number of specific outcomes, and the second argument ("names") gives the name of the censored outcome and the two other outcomes. The resulting matrix states that a patient's status can change from "event-free" to either "prostate" or "other", these latter two being causes of death. The other entries of the matrix simply state that once a patient dies of one cause, they cannot change to another cause or return to the "event-free" status. Next, we use the function "msprep" to create the new data set, and examine the first few rows:

```
> prostate.long <- msprep(time = cbind(NA, survTime, survTime),
+     status = cbind(NA, status.prost, status.other),
+     keep = data.frame(grade, ageGroup), trans = tmat)
> head(prostate.long)

  id from to trans Tstart Tstop time status grade ageGroup
1  1    1  2     1      0    27   27      0  mode    70-74
2  1    1  3     2      0    27   27      0  mode    70-74
3  2    1  2     1      0    38   38      0  poor    75-79
4  2    1  3     2      0    38   38      1  poor    75-79
5  3    1  2     1      0    13   13      0  poor    75-79
6  3    1  3     2      0    13   13      0  poor    75-79
```

In this "msprep" function, the argument "time" consists of three columns, each corresponding the states defined by the "tmat" transition matrix. The first "event-free" state is represented by a placeholder, "NA"; the second and third by the survival times for time to death from prostate cancer and from other causes. In our data set, both are represented by the "survTime" vector. The two times are distinguished in the next argument, “status”. This also has three columns. The first is a placeholder, “NA” as before; the second is the censoring indicator for prostate cancer (“status.prost”), and the third is for other causes (“status.other”). These latter two variables were defined earlier from the “status” column of the data frame “prostateSurvival.T2”. Finally, the transition matrix is defined by “trans = tmat”. Note that the variables “survTime”, “grade”, and “ageGroup” from the “prostateSurvival.T2” file are available for use to us because we have previously attached it.

The output file has twice as many rows as the original “prostateSurvival.T2” file. The first column, “id”, refers to the patient number in the original file; here, each is repeated twice. For our purposes, we can ignore the columns “from” and “two”. The column “trans” will be important, because it contains an indicator of the cause of death; here “1” refers to death from prostate cancer and “2” refers to death from other causes. The “Tstart” column contains all 0’s, since for our data, “time = 0” indicates the diagnosis with prostate cancer. We can ignore “Tstop”, and use the “time” column as the survival time and the “status” column as the censoring indicator. Note that for each patient, there are two entries for “status”. Both can be 0, or one can be 1 and the other 0; they can’t both be 1 because each patient can die of only one cause, not both. Finally, the last two columns are covariate columns we carried over from the original “prostateSurvival.T2” data frame. Each original value is doubled, since each patient has one covariate value, regardless of their cause of death.

We may obtain a summary of the numbers of events of each type as follows:

```
> events(prostate.long)
$Frequencies
            to
from         event-free prostate other no event total entering
event-free            0      410  1345     4165           5920
prostate              0        0     0        0              0
other                 0        0     0        0              0
```

These results indicate that there are 410 deaths due to prostate cancer, 1345 due to other causes, and 4165 censored observations, for 5920 total. (We may ignore the second two rows, which are relevant only for multistate models.)

To show how to use our newly expanded data set, we can use it to reproduce our analysis from the previous section. To obtain these estimates of the effects of covariates on prostate-specific and other death causes, we use separate commands, one for “trans = 1” (prostate cancer) and the other for “trans = 2” (other causes of death), as follows:

```
> summary(coxph(Surv(time, status) ~ grade + ageGroup,
+   data=prostate.long, subset={trans==1}))
> summary(coxph(Surv(time, status) ~ grade + ageGroup,
+   data=prostate.long, subset={trans==2}))
```

The results (not shown) are identical to what we obtained before.

If we stratify on cause of death using “strata(trans)” we get estimates of the effect of the covariates on cause of death under the assumption that they affect both causes of death equally,

```
> summary(coxph(Surv(time, status) ~ grade + ageGroup
   + strata(trans),
+   data=prostate.long))

  n= 11840, number of events= 1755

               coef exp(coef)se(coef)    z  Pr(>|z|)
gradepoor     0.515   1.673  0.050 10.372  < 2e-16 ***
ageGroup70-74 0.027   1.027  0.112  0.238  0.81210
ageGroup75-79 0.332   1.394  0.104  3.198  0.00139 **
ageGroup80+   0.833   2.301  0.099  8.396  < 2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

In this example, this model wouldn’t be appropriate, since we would expect that cancer grade affects prostate cancer death differently than it does death from other causes. To test this formally, we fit the following model:

```
> summary(coxph(Surv(time, status) ~ grade*factor(trans) +
+   ageGroup + strata(trans), data=prostate.long))

 n= 11840, number of events= 1755

                        coef exp(coef)se(coef)     z  Pr(>|z|)
gradepoor              1.239   3.451  0.100  12.391  < 2e-16 ***
factor(trans)2            NA      NA  0.000      NA       NA
ageGroup70-74          0.026   1.027  0.112   0.235  0.81431
ageGroup75-79          0.333   1.395  0.104   3.201  0.00137 **
ageGroup80+            0.833   2.301  0.099   8.394  < 2e-16 ***
gradepoor:
factor(trans)2        -0.963   0.382  0.116  -8.327  < 2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

The coefficient estimate 1.239 for “gradepoor” is the effect of grade on prostate cancer death, and is similar to the estimate we got earlier (1.220) for prostate cancer death alone. Here however, we also have an estimate in the last row for the difference between the effect on prostate cancer death and death from other causes. This is the interaction between a grade of “poor” and cause “2” (other death). The estimate, –0.963, which is highly statistically significant, represents the additional effect of poor grade on risk of death from other causes relative to its effect on prostate cancer death. Specifically, the hazard of death from other causes is exp(–0.963) = 0.381 times the hazard of death from prostate cancer.

We have determined that having a poor grade of prostate cancer strongly affects the risk of dying from prostate cancer, and this effect is much stronger on the risk of death from prostate cancer than on the risk of death from other causes. We may next ask how increasing age affects the risk of dying from prostate cancer and of other causes. Unsurprisingly, the trend is clear in both cases, as we have seen above. But is the effect any different on these two causes? We can answer this by examining the interaction between age group and cause of death as follows:

```
> summary(coxph(Surv(time, status) ~ (grade + ageGroup)*trans +
+    ageGroup + strata(trans), data=prostate.long))

  n= 11840, number of events= 1755

                       coef exp(coef se(coef)   z  Pr(>|z|)
gradepoor             1.220   3.387  0.100 12.154  < 2e-16 ***
ageGroup70-74        -0.286   0.751  0.260 -1.102   0.2704
ageGroup75-79         0.403   1.496  0.226  1.784   0.0744 .
ageGroup80+           0.973   2.645  0.215  4.529 5.92e-06 ***
trans2                   NA      NA  0.000     NA       NA
gradepoor:trans2     -0.939   0.391  0.116 -8.072 6.66e-16 ***
ageGroup70-74:trans2  0.380   1.463  0.288  1.322   0.1863
ageGroup75-79:trans2 -0.089   0.914  0.254 -0.351   0.7252
ageGroup80+:trans2   -0.183   0.833  0.242 -0.754   0.4508
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

The results are in the last three rows of parameter estimates. None of these differences are statistically significant, so we conclude that there is no difference in the effect of age on the two death causes, after adjusting for grade.

9.3 Additional Notes

9.1 The “ashkenazi” data may be used to estimate the age-of-onset distribution of carriers and non-carriers of the BRCA mutation, but doing this properly is quite involved. See for Struewing et al. [64], Moore et al. [49], and Chatterjee and Wacholder [8] for details.

9.2 The “mstate” package is capable of modeling complex multistage survival models with alternative pathways to an endpoint. The competing risks capabilities of this package are actually a special case of the more general multistage methods. See Putter et al. [56] and the package documentation for details.

Exercises

9.1. Using the “ashkenazi” data of Sect. 9.1, use “coxme” to fit a random effects model without the “mutant” fixed effect term. How does the estimate of the variance of the random effect from this model compare to that from the model that includes “mutant” as a fixed effect?

9.2. Using the “diabetes” data of Sect. 9.1, fit the interaction model using (1) the frailty option of “coxph”, using both the gamma and gaussian random effects options, and (2) using the “cluster” option in “coxph”. Compare the estimated parameters and standard errors to those from the “coxme” model.

9.3. Again using the “diabetes” data, use “coxme” to fit a model without the interaction term. Test for the importance of the interaction term using both a Wald test and a likelihood ratio test.

9.4. Repeat the calculations of the cumulative incidence functions for death from prostate cancer and from other causes from Sect. 9.2.3, but use the age group 75–84 instead of 85 and above.

Chapter 10
Parametric Models

10.1 Introduction

In biomedical applications, non-parametric (e.g. the product-limit survival curve estimator) and semi-parametric (e.g. the Cox proportional hazards model) methods play the most important role, since they have the flexibility to accommodate a wide range of hazard function forms. Still, parametric methods have a place in biomedical research, and may be appropriate when survival data can be shown to approximately follow a particular parametric form. Parametric models are often much easier to work with than the partial-likelihood-based models we have discussed in earlier chapters, since the former are defined by a small and fixed number of unknown parameters. This allows us to use standard likelihood theory for parameter estimation and inference. Furthermore, accommodating complex censoring and truncation patterns is much more direct with parametric models than with partial likelihood models. Of course, the validity of these techniques depends heavily on the appropriateness of the particular parametric model being used. In Chap. 2 we introduced the exponential, Weibull, and gamma distributions, and mentioned several others that could potentially serve as survival distribution models. In this chapter we will emphasize the exponential and Weibull distributions, since these are the most commonly used parametric distributions. We will also briefly discuss the use of some other parametric models in analyzing survival data.

10.2 The Exponential Distribution

The exponential distribution is the simplest distribution to work with. It has a constant hazard function, $h(t) = \lambda$, which gives it the memory-less property. That is, the risk of having the event of interest is the same at any point in time as it was at the beginning. The p.d.f. and survival functions are, as discussed in

D. Moore, *Applied Survival Analysis Using R*, Use R!,
DOI 10.1007/978-3-319-31245-3_10

Chap. 2, $f(t;\lambda) = \lambda e^{-\lambda t}$ and $S(t;\lambda) = e^{-\lambda t}$, respectively. To construct a likelihood function, we include the p.d.f. for each observed failure and the survival function for each (right) censored observation, as shown in Sect. 2.6. The simplicity of the exponential distribution makes it attractive for certain specialized applications, such as for power and sample size calculations, as we shall see in the next chapter. But for modeling survival data, we will need the additional flexibility afforded by the Weibull distribution, of which the exponential distribution is a special case. In fact, if a survival variable T has an exponential distribution with parameter λ, the transformed variable T^{α}, where α is an additional parameter, will have a Weibull distribution.

10.3 The Weibull Model

10.3.1 Assessing the Weibull Distribution as a Model for Survival Data in a Single Sample

The Weibull survival distribution, as expressed in Sect. 2.4, has hazard and survival functions $h(t) = \alpha\lambda^{\alpha}t^{\alpha-1}$and $S(t) = e^{-(\lambda t)^{\alpha}}$. Later, when we use the Weibull distribution to assess the effects of covariates on survival, we shall find it convenient to use the scale parameter $\sigma = 1/\alpha$, and the mean parameter $\mu = -\log\lambda$. Then

$$h(t) = \frac{1}{\sigma}e^{-\frac{\mu}{\sigma}}t^{\frac{1}{\sigma}-1}$$

and

$$S(t) = e^{-e^{-\mu/\sigma}t^{1/\sigma}}.$$

As discussed in Additional Note 2 in Chap. 2, the term “scale” parameter as we use it here has a different meaning than what is often used when defining the Weibull distribution.

In the special case where $\sigma = 1$ the Weibull distribution reduces to an exponential distribution with rate parameter λ. Taking a complementary log-log transformation $g(u) = \log\left[-\log(u)\right]$ of the Weibull survival function, we have

$$\log\left[-\log(S_i)\right] = \alpha\log(\lambda) + \alpha\log(t_i) = -\frac{\mu}{\sigma} + \frac{1}{\sigma}\log(t_i) \qquad (10.3.1)$$

where $S_i = S(t_i)$.

This result suggests a diagnostic tool to assess how well a set of survival data follow a Weibull distribution. We first compute the Kaplan-Meier estimate $\hat{S}$ of a survival distribution. Then, we define $y_i = \log\left\{-\log\left[\hat{S}(t_i)\right]\right\}$ and plot y_i versus

$\log(t_i)$. Finally, we fit through these points a straight line, with equation of the form $y = b + m \log t$ where $b = -\mu/\sigma$ and $m = 1/\sigma$ are, respectively, the y-intercept and slope of the line. If the plotted points fall along this fitted line, one may conclude that the survival data may be approximately modeled using a Weibull distribution. Furthermore, the slope of the fitted line will provide an estimate of $1/\sigma$, so that $\sigma = 1/m$, and the y-intercept is $b = -\mu/\sigma$, so that $\mu = -b/m$.

We first examine the gasticXelox data discussed in Chap. 3 to see if it follows a Weibull distribution. We first obtain a Kaplan-Meier estimate of the survival distribution,

```
timeMonths <- gastricXelox$timeWeeks*7/30.25
delta <- gastricXelox$delta
library(survival)
result.km <- survfit(Surv(timeMonths, delta) ~ 1)
```

Next we extract the survival estimates and time variables from "result.km" and transform the former with a complementary log-log transformation, and the latter with a log transformation,

```
survEst <- result.km$surv
survTime <- result.km$time
logLogSurvEst <- log(-log(survEst))
logSurvTime <- log(survTime)
```

Finally, we plot "logLogSurvEst" versus "logSurvTime" and fit a straight line through the points,

```
plot(logLogSurvEst ~ logSurvTime)
result.lm <- lm(logLogSurvEst ~ logSurvTime)
abline(result.lm)
```

The results, shown in Fig. 10.1, indicate that a Weibull distribution may not be appropriate for these data, since the points do not follow a linear relationship.

We now consider if a Weibull distribution is appropriate for the pharmacoSmoking data discussed in earlier chapters. Ignoring for now the covariate information, we may examine the survival times to assess the suitability of the Weibull distribution

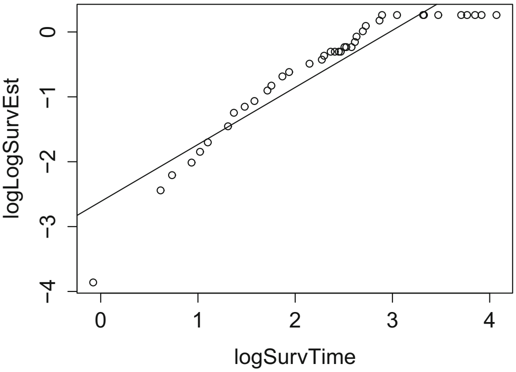


Fig. 10.1 Plot of the complementary log-log transformation of survival probability versus log survival time for the gastricXelox data

as a basis for modeling these data. Recall that the survival time is denoted “ttr” and indicates the time to relapse (or censoring), and “relapse” is the censoring variable. We first attach it and re-define survival times listed as zero to 0.5,

```
> attach(pharmacoSmoking)
> ttr[ttr == 0]  <- 0.5
```

We then fit a fit a Kaplan-Meier survival curve to the data, and extract the survival and corresponding event times,

```
> result.surv <- survfit(Surv(ttr, relapse) ~ 1)
> survEst <- result.surv$surv
> survTime <- result.surv$time
```

Then we compute a complementary log-log transformation of the survival times and a log transformation of the corresponding event times, fit a linear regression line through the points, and plot the points and the fitted line,

```
> logLogSurvEst <- log(-log(survEst))
> logSurvTime <- log(survTime)
> result.lm <- lm(logLogSurvEst ~ logSurvTime)
> result.lm

Coefficients:
(Intercept)  logSurvTime
    -2.0032       0.4385
```

We see that the slope is 0.4385 and the y-intercept is −2.0032. We plot the points and fitted line as follows:

```
plot(logLogSurvEst ~ logSurvTime)
abline(result.lm)
```

The resulting plot, shown in Fig. 10.2, shows a close agreement with a Weibull distribution.

Estimates of the scale and mean parameters are $\mu = -b/m = 2.0032/0.4385 = 4.568$ and $\sigma = 1/m = 1/0.4385 = 2.280$.

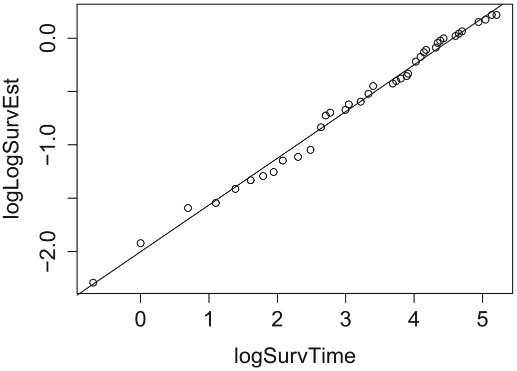


Fig. 10.2 Plot of the complementary log-log transformation of the Kaplan-Meier survival estimate for the pharmacoSmoking data versus the log event time. The *straight line* is the least squares regression line

10.3.2 Maximum Likelihood Estimation of Weibull Parameters for a Single Group of Survival Data

The log-likelihood function, following the notation in Sect. 2.6, is

$$l(\lambda, \alpha) = \sum_{i=1}^{n} \{\delta_i \log [h(t_i)] + \log [S(t_i)]\}$$

Substituting the expressions for $h(t_i)$ and $S(t_i)$, we get

$$l(\lambda, \alpha) = d \log \alpha + d\alpha \log \lambda + (\alpha - 1) \sum_{i=1}^{n} \delta_i \log t_i - \lambda^{\alpha} \sum_{i=1}^{n} t_i^{\alpha} \qquad (10.3.2)$$

where $d = \sum_{i=1}^{n} \delta_i$. Later, when we use the Weibull distribution to assess the effects of covariates on survival, we shall find it convenient to use the scale parameter $\sigma = 1/\alpha$, and the mean parameter $\mu = -\log \lambda$.

As an alternative, we may directly compute maximum likelihood estimates of these parameters. We may encode the log-likelihood of Eq. 10.3.2 in the following R function, which takes parameters μ and σ as the first and second elements of a vector "par". Within the function, we first re-parametrize in terms of $\alpha = 1/\sigma$ and $\lambda = e^{-\mu}$, and then compute the log-likelihood using Eq. 10.3.2.

```
logLikWeib <- function(par, tt, status) {
   mu <- par[1]
   sigma <- par[2]
   lambda.p <- exp(-mu)
   alpha.p <- 1/sigma

   dd <- sum(status)
   sum.t <- sum(status*log(tt))
   sum.t.alpha <- sum(tt^alpha.p)

   term.1 <- dd*log(alpha.p) + alpha.p*dd*log(lambda.p)
   term.2 <- (alpha.p - 1)*sum.t
   term.3 <- (lambda.p^alpha.p)*sum.t.alpha
   result <- term.1 + term.2 - term.3
   result
   }
```

The m.l.e may be obtained using the "optim" function, using as starting values the estimates of μ and σ from the linear regression,

```
result <- optim(par=c(4.568, 2.280), fn=logLikWeib, method=
    "L-BFGS-B",
    lower=c(0.001, 0.01), upper=c(5, 5),
    control=list(fnscale = -1),
    tt=ttr, status=relapse)
```

As always, we use the option "control=list(fnscale = -1)" to tell the optim function to find a maximum (rather than a minimum). The final m.l.e. is given by

```
> result$par
   [1] 4.656329 2.041061
```

The first element of "result$par" is $\hat{\mu}$ and the second element is $\hat{\sigma}$. A more practical way to obtain these estimates is by means of the function "survreg" in the "survival" package, which of course yields the same parameter estimates:

```
> result.survreg.0 <- survreg(Surv(ttr, relapse) ~ 1,
+  dist="weibull")
> summary(result.survreg.0)

            Value Std. Error     z         p
(Intercept) 4.656     0.2170 21.46 3.68e-102
Log(scale)  0.713     0.0919  7.76  8.26e-15

Scale= 2.04
```

The m.l.e. of the scale parameter, 2.04, is close to the value 2.28 from the linear regression approach. The "Intercept" m.l.e., 4.656, is approximately the value 4.57 we obtained from the linear regression. The estimate "Log(scale)" is, of course, the log of the scale parameter.

10.3.3 *Profile Weibull Likelihood*

Suppose a survival random variable T follows a Weibull distribution with parameters α and λ, as defined in Sect. 10.3.1. If the parameter α is fixed, then a new random variable $T^* = T^\alpha$ has an exponential distribution with parameter λ^α. It follows then, from results in Sect. 2.6 on the maximum likelihood estimate for an exponential distribution, that for a known value of α, we have $\hat{\lambda} = (d/V)^{1/\alpha}$, where $V = \sum t_i^\alpha$ and d is the total number of deaths. Since the m.l.e. $\hat{\lambda}(\alpha)$ for a fixed value of α can be obtained so easily, we can express the Weibull log-likelihood of Eq. 10.3.2 as $l^*(\alpha) = l(\hat{\lambda}(\alpha), \alpha)$, which is a function of a single parameter α. This form of the likelihood function is known as a *profile likelihood*, since one of the parameters (λ) is replaced with it's maximum likelihood estimate contingent on a particular value of the other parameter (α). So maximizing $l^*(\alpha)$ will yield the maximum likelihood estimate of α; the m.l.e for λ is then $\hat{\lambda} = (d/V)^{1/\hat{\alpha}}$. In R, we define the profile likelihood as follows:

```
logLikWeibProf <- function(par, tt, status) {
   # find log-likelihood for a particular sigma, using mle for mu
   sigma <- par
   alpha.p <- 1/sigma
   dd <- sum(status)
   sum.t <- sum(status*log(tt))
   sum.t.alpha <- sum(tt^alpha.p)
   lambda.p <- (dd/sum.t.alpha)^(1/alpha.p)
   term.1 <- dd*log(alpha.p) + alpha.p*dd*log(lambda.p)
   term.2 <- (alpha.p - 1)*sum.t
   term.3 <- (lambda.p^alpha.p)*sum.t.alpha
   result <- term.1 + term.2 - term.3
   result     }
```

This differs from the function “logLikWeib” of the previous section in that now “par” is a single number, sigma, and “lambda.p” is defined using a particular value of alpha.p = 1 / sigma. To obtain the m.l.e. for σ we find the maximum of the profile log-likelihood as follows:

```
> resultProf <- optim(par=c(2.280), fn=logLikWeibProf, method=
    "L-BFGS-B",
+    lower=c(0.01), upper=c(5), control=list(fnscale = -1),
+    tt=ttr, status=relapse)
> sigma.hat <- resultProf$par
> sigma.hat
[1] 2.041063
```

The resulting estimate, $\hat{\sigma} = 2.041063$, is the same as we obtained in the previous section. To obtain $\hat{\lambda}$and $\hat{\mu} = 1/\hat{\lambda}$, we do the following:

```
> dd <- sum(relapse)
> sigma <- resultProf$par
> alpha.p <- 1/sigma.hat
> sum.t.alpha <- sum(ttr^alpha.p)
> lambda.p <- (dd/sum.t.alpha)^(1/alpha.p)
> mu.hat <- -log(lambda.p)
> mu.hat
[1] 4.656329
```

The resulting estimate, $\hat{\mu}$ =4.656329, is also the same as we obtained in the previous section.

We may plot the profile likelihood in terms of sigma as follows, for a range of values of σ from 1.0 to 5.0:

```
sigma.list <- (100:500)/100
n.list <- length(sigma.list)
logLik.list <- rep(NA, n.list)
for (i in 1:n.list) {
  logLik.list[i] <- logLikWeibProf(par=sigma.list[i], ttr,
  relapse) }
plot(logLik.list ~ sigma.list, type="l", xlab="sigma",
   ylab="profile log-likelihood")
abline(v=sigma.hat, col="gray")
```

The profile log-likelihood is shown in Fig. 10.3.

10.3.4 Selecting a Weibull Distribution to Model Survival Data

We can fit a Weibull distribution to a set of data to obtain maximum likelihood estimates of the two parameters, as we have seen. In some cases, it may be desirable

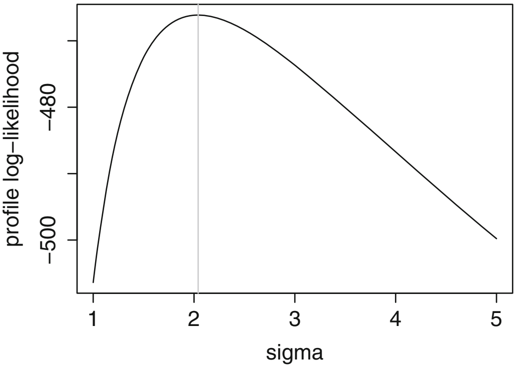


Fig. 10.3 Profile log-likelihood for a Weibull distribution fitted to the pharmcoSmoking data, as a function of sigma. The *vertical line* indicates the m.l.e. at $\sigma = 2.041$

to find a Weibull distribution that matches the survival data at two specified time points. Suppose the two time points are t_1 and t_2, and the estimated survival points (from the Kaplan-Meier survival curve) at these two points are s_1 and s_2, respectively. Let us define $y_1 = \log[-\log(s_1)]$ and $y_2 = \log[-\log(s_2)]$. Using Eq. 10.3.1, we have

$$y_1 = \alpha\log(\lambda) + \alpha\log(t_1)$$
$$y_2 = \alpha\log(\lambda) + \alpha\log(t_2)$$

Solving these two simultaneous linear equations, we get

$$\tilde{\alpha} = \frac{y_1 - y_2}{\log(t_1) - \log(t_2)}$$
$$\tilde{\lambda} = \exp\left\{\frac{y_2\log(t_1) - y_1\log(t_2)}{y_1 - y_2}\right\}.$$

To illustrate, consider the pharmacoSmoking data, and let's find a Weibull distribution that matches the Kaplan-Meier estimate of the survival distribution for the "patchOnly" group at 4 and 12 weeks (28 and 84 days). In R, we first find the Kaplan-Meier estimate, which is in "result.surv", and then find the survival estimates at times 28 and 84 days, which we put into "result.summ". Then we extract those two times ("t.vec") and the survival estimates ("s.vec"), and display them. In the following, we summarize the survival results at 28 and 84 days, that is, 4 and 12 weeks, respectively (assuming we have attached the pharmacoSmoking data):

```
> result.surv <- survfit(Surv(ttr, relapse) ~ 1,
+   subset={grp =="patchOnly"})
> result.summ <- summary(result.surv, time=c(28, 84))
> t.vec <- result.summ$time
> s.vec <- result.summ$surv
> data.frame(t.vec, s.vec)
  t.vec    s.vec
1    28 0.437500
2    84 0.265625
```

Next, we use the “Weibull2” function in F. Harrell’s “Hmisc” package to produce a Weibull function that matches these two points,

```
library(Hmisc)
pharmWeib <- Weibull2(t.vec, s.vec)
```

The function “pharmWeib” computes the Weibull survival estimates for a range of time values,

```
t.vals <- 1:200
s.vals <- pharmWeib(t.vals)
```

(The internal parametrization used by the “Weibull2” function is different from what we use in this book, but this doesn’t matter, since of course it produces the same survival estimates.)

Next, let us obtain the predicted Weibull survival curve based on maximum likelihood estimates of the Weibull parameters.

```
model.pharm.weib.basic <- survreg(Surv(ttr, relapse) ~ 1,
     dist="weibull", subset={grp =="patchOnly"} )
mu.hat <- model.pharm.weib.basic$coefficients
sigma.hat <- model.pharm.weib.basic$scale
lambda.hat <- exp(-mu.hat)      # " 1 / scale"
alpha.hat <- 1/sigma.hat        # "shape"
s.mle.vals <- 1 - pweibull(t.vals, shape=alpha.hat,
  scale=1/lambda.hat)
```

Finally, we plot the survival estimates in Fig. 10.4

```
plot(result.surv, conf.int=F, xlab="Days to relapse",
   ylab="Survival probability")
lines(s.mle.vals ~ t.vals, col="blue")
lines(s.vals ~ t.vals, col="red")
points(t.vec, s.vec, col="red")
```

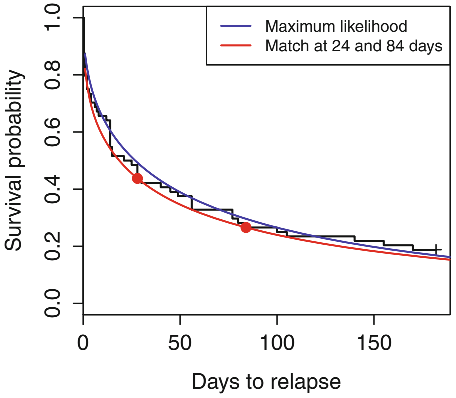


Fig. 10.4 Survival curve estimates for the “patch only” group in the pharmacoSmoking data. The step function is the Kaplan-Meier estimate. The *blue line* is the Weibull estimate of the survival curve based on maximum likelihood estimates of the parameters. The *red line* is the Weibull estimate that matches the Kaplan-Meier estimate at 24 and 84 days; the two matching points are indicated by *solid red circles*

In the next chapter we will use the estimated Weibull function for simulating survival data for computing the power to detect a difference in a randomized study.

10.3.5 Comparing Two Weibull Distributions Using the Accelerated Failure Time and Proportional Hazards Models

Suppose now that we have two groups of survival data, one of patients who received an experimental treatment and one of patients who received a control. In prior chapters the quantity we used to compare the two distributions was the hazard ratio e^{β} which, under the proportional hazards assumption, was assumed not to change over time. If the experimental treatment were effective in increasing survival, the hazard ratio would be *less* than one, and the log-hazard ratio β would thus be *negative*. An alternative way of comparing a treatment group to a control group, often used with parametric models, is called the *accelerated failure time* (AFT) model (sometimes referred to as the *accelerated life* model). In this model we assume that the survival time for a treated patient is a multiple e^{γ} of what the survival time would have been had the patient received the control treatment. A key property of the AFT model is this: if the treatment is effective, the accelerated time coefficient e^{γ} will be *greater* than one, and thus γ will be *positive*.

Formally, the survival distributions for the accelerated life model are given by $S_1(t) = S_0(e^{-\gamma}t)$ and the hazards are given by $h_1(t) = e^{-\gamma}h_0(e^{-\gamma}t)$. In the case of the Weibull distribution, we have

$$h_1(t) = e^{-\gamma}h_0(e^{-\gamma}t) = e^{-\gamma} \cdot \frac{1}{\sigma}e^{-\frac{\mu_0}{\sigma}}(e^{-\gamma}t)^{\frac{1}{\sigma}-1}$$

Rearranging, we have

$$h_1(t) = e^{-\frac{\gamma}{\sigma}} \cdot \frac{1}{\sigma} \cdot e^{-\frac{\mu_0}{\sigma}} t^{\frac{1}{\sigma}-1} = e^{-\frac{\gamma}{\sigma}}h_0(t)$$

That is, in the case of the Weibull distribution, the accelerated life model is equivalent to a proportional hazards model with proportionality factor $e^{\beta} = e^{-\frac{\gamma}{\sigma}}$. Thus, the proportional hazards model and the accelerated life model are equivalent in the case of a Weibull distribution, with $\beta = -\gamma/\sigma$. Furthermore, it is possible to show that the Weibull distribution is the *only* distribution with this property [11].

The pharmacoSmoking data, comparing the triple therapy treatment group to the patch treatment provides an illustration of these principles. The Weibull model may be used to compare the two groups as follows, using the "survreg" function in the survival package:

```
> result.survreg.grp <- survreg(Surv(ttr, relapse) ~ grp,
+   dist="weibull")
> summary(result.survreg.grp)

              Value Std. Error     z        p
(Intercept)   5.286     0.3320 15.92 4.59e-57
grppatchOnly -1.251     0.4348 -2.88 4.00e-03
Log(scale)    0.689     0.0911  7.56 3.97e-14

Scale= 1.99
```

We see that $\hat{\gamma} = -1.251$, indicating that the "patch only" treatment group has shorter times to relapse than the triple therapy group by a factor of $e^{\hat{\gamma}} = e^{-1.251} = 0.286$. The estimate of the scale parameter is $\hat{\sigma} = 1.99$. Thus, if we want to compare the patch group to the triple therapy group using a proportional hazards model, the log proportional hazards is given by $\hat{\beta} = -\hat{\gamma}/\hat{\sigma} = 1.251/1.99 = 0.629$. We may compare this to the results of fitting a Cox proportional hazards model as follows:

```
> result.coxph.grp <- coxph(Surv(ttr, relapse) ~ grp)
> summary(result.coxph.grp)

  n= 125, number of events= 89

               coef exp(coef) se(coef)   z Pr(>|z|)
grppatchOnly 0.6050    1.8313   0.2161 2.8  0.00511 **
```

The corresponding estimate of the log hazards ratio from the Cox model, 0.6050, is near (but not the same as) 0.629, the estimate from the Weibull model.

Notice that the Cox model output shows only one parameter estimate, that for the effect of the patch (as compared to the triple therapy). The Weibull model results in three parameter estimates, one of which is also a comparison of the patch to the triple therapy. The other two estimates represent the baseline Weibull distribution. As discussed in previous chapters, the Cox proportional hazards model does not produce an "intercept" term among the coefficient estimates; if there were an intercept term, it would cancel out of the partial likelihood just as the baseline hazard does. (Once a Cox model has been fitted, it is of course possible to obtain an estimate of the baseline hazard, as discussed in Sect. 5.5.) Parametric survival models, by contrast, include an intercept term, which can be used to determine the baseline hazard function. For the pharmacoSmoking data, with "grp" as a predictor, the baseline hazard function is a Weibull distribution with parameter estimates $\hat{\mu}_0 = 5.286$ and the scale parameter estimate, which is the same for both groups, is $\hat{\sigma} = 1.99$. To obtain the estimated survival curve for the triple-therapy group, which here is the baseline group, we compute the parameters of the baseline Weibull distribution, $\hat{\alpha} = 1/\hat{\sigma} = 1/1.99 = 0.502$ and $\hat{\lambda}_0 = e^{-\hat{\mu}} = e^{-5.286} = 0.00506$. The estimated baseline survival function is then

$$\hat{S}_0(t) = e^{-(\hat{\lambda} t)^{1/\hat{\sigma}}}$$

We may obtain the baseline Weibull coefficient estimates in R as follows:

```
mu0.hat <- result.survreg.grp$coef[1]
sigma.hat <- result.survreg.grp$scale
alpha.hat <- 1/sigma
lambda0.hat <- exp(-mu0.hat)
```

From these we compute the baseline survival function,

```
tt.vec <- 0:182
surv0.vec <- 1 - pweibull(tt.vec, shape=alpha,scale=1/
     lambda0.hat)
```

recalling the "scale" terminology for the "pweibull" function is quite different from the "scale" term in the "survreg" function.

To obtain the Weibull function for the comparison group (here the "patchOnly" group), we note that the proportional hazards constant is $e^{-\hat{\gamma}/\hat{\sigma}} = e^{-0.629} = 0.533$. That is, the hazard for the "patchOnly" group is 0.533 times the hazard for the "combination" group. The survival function for the combination group is $S_1(t) = \{S_0(t)\}^{e^{-\hat{\gamma}/\hat{\sigma}}}$. In R, $\hat{\gamma}$ is the coefficient for the "grp" term, and is the second element of "coef",

```
gamma.hat <- result.survreg.grp$coef[2]
surv1.vec <- surv0.vec^(exp(-gamma.hat/sigma.hat))
```

It is helpful to compare these survival estimates to those from the Cox proportional hazards model. The latter survival estimates are obtained as follows:

```
coxph.surv.est <- survfit(result.coxph.grp,
    newdata=data.frame(list(grp=c("combination","patchOnly"))))
```

In the call to "survfit", we have created a data frame for the "grp" variable, and use that data along with the results of the Cox proportional hazards model to obtain the predicted survival curves. We may plot the Cox-based survival curves and the Weibull-based survival curves on the same plot,

```
plot(coxph.surv.est, col=c("red", "black"))
lines(surv0.vec ~ tt.vec, col="red")
lines(surv1.vec ~ tt.vec)
```

The resulting plot, shown in Fig. 10.5, shows the Cox model estimates as step functions and the Weibull-based estimates as smooth curves.

10.3.6 A Regression Approach to the Weibull Model

An alternative way of looking at a Weibull accelerated failure time model comparing two groups is by modeling the log survival time as a location-scale model, as follows:

$$\log(t) = \mu + \gamma z + \sigma\epsilon^* \qquad (10.3.3)$$

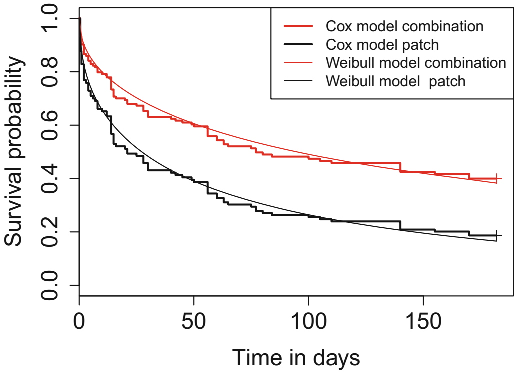


Fig. 10.5 Comparisons of combination therapy (*red*) vs. patch (*black*) for time to smoking relapse using the pharmacoSmoking data. The step functions are survival function estimates obtained using a Cox proportional hazards model, and the *smooth curves* are obtained using a Weibull model

where ϵ follows a unit exponential distribution, which leads to $\epsilon^* = \log \varepsilon$ having what is called an extreme value distribution. This formulation suggests that other choices for the distribution of ϵ can lead to other parametric survival models, as will be discussed in Sect. 10.4.

10.3.7 Using the Weibull Distribution to Model Survival Data with Multiple Covariates

We may use "survfit" to accommodate multiple covariates into a Weibull accelerated failure time model in a straightforward manner. For example, for the pharmacoSmoking data, we previously (in Chap. 7) settled on a Cox proportional hazards model with treatment group, age, and employment status as predictors. The output of that model is as follows:

```
> modelAll2.coxph <- coxph(Surv(ttr, relapse) ~ grp + age +
+   employment)
> summary(modelAll2.coxph)

  n= 125, number of events= 89
                     coef  exp(coef)  se(coef)      z  Pr(>|z|)
grppatchOnly      0.60788    1.83654   0.21837  2.784   0.00537 **
age              -0.03529    0.96533   0.01075 -3.282   0.00103 **
employmentother   0.70348    2.02077   0.26929  2.612   0.00899 **
employmentpt      0.65369    1.92262   0.32732  1.997   0.04581 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Here, a positive coefficient indicates higher hazard, and thus worse survival. For example, the coefficient for "patchOnly" is 0.608, which indicates that the hazard is higher for this treatment group than for the triple therapy group, by a constant factor of $e^{0.60788} = 1.83654$. We may include these covariates in a Weibull model as follows:

```
 1  > model.pharm.weib <- survreg(Surv(ttr, relapse) ~ grp + age +
 2  +   employment, dist="weibull")
 3  > summary(model.pharm.weib)
 4                       Value Std. Error     z        p
 5  (Intercept)         2.4024     0.9653  2.49 1.28e-02
 6  grppatchOnly       -1.1902     0.4133 -2.88 3.98e-03
 7  age                 0.0697     0.0203  3.43 6.02e-04
 8  employmentother -1.3890     0.5029 -2.76 5.74e-03
 9  employmentpt       -1.3143     0.6132 -2.14 3.21e-02
10  Log(scale)          0.6313     0.0900  7.02 2.26e-12
11
12  Scale= 1.88
13  Weibull distribution
14  Loglik(model)= -454.1   Loglik(intercept only)= -466.1
15          Chisq= 23.96 on 4 degrees of freedom, p= 8.2e-05
```

Even though we have fit a survival model to the same data using the same predictors, the output of “survreg” differs from that of “coxph” in two important ways. The first difference is that “coxph” produces estimates for the predictors only, whereas “survreg” produces not only those estimates but also two more, one for the “intercept” (line 5) and one for “Log(scale)” (line 10). These two parameters define the baseline Weibull survival model. The scale parameter estimate, 1.88, is also printed in line 12; the log of this, unsurprisingly, is 0.6313, and is printed in line 10.

The second important difference is that the parameter estimates from “survreg” are accelerated failure time constants. That is, for “patchOnly” (line 6), the estimate -1.1902 is negative, and indicates that patients receiving this treatment have shorter times to relapse than do the patients receiving triple therapy, and the “acceleration” factor is $e^{-1.1902} = 0.304$. (Since this factor is less than one, it might be more properly referred to as deceleration.)

Since for a Weibull distribution, the accelerated failure time model is equivalent to a proportional hazards model, we may convert the acceleration coefficients to proportional hazards estimates to better compare them to those obtained from the Cox partial likelihood model. As discussed earlier, if γ_j represents the jth parameter from the accelerated failure time model, then $\beta_j = -\gamma_j/\sigma$ represents the jth parameter from a proportional hazards model, where σ is the scale. Converting the output from “survreg” to proportional hazards is thus straightforward in principle, but the mechanics of doing in R are rather involved. First, we need to extract the coefficient estimates from “model.pharm.weib”, which is a vector of seven elements. Then select the coefficient estimates, which are elements 2 through 5,

```
weib.coef.all <- model.pharm.weib$coef
weib.coef <- weib.coef.all[2:5]
```

To get the proportional hazards estimates, we need to extract the estimate of the scale factor, “model.pharm.weib$scale”, and then switch the sign, and divide,

```
weib.coef.ph <- -weib.coef/model.pharm.weib$scale
```

The vector “weib.coef.ph” contains the proportional hazards parameter estimates from the Weibull model.

Extracting the coefficients from the Cox (partial likelihood) model is somewhat simpler,

```
coxph.coef <- model.pharm.coxph$coef
```

We may use the "data.frame" function to assemble the estimates and standard errors in a table as follows:

```
> data.frame(weib.coef.ph, coxph.coef)
                 weib.coef.ph  coxph.coef
grppatchOnly       0.63301278  0.60788405
age               -0.03708786 -0.03528934
employmentother    0.73878031  0.70347664
employmentpt       0.69903157  0.65369019
```

The parameter estimates from the two models are quite similar, differing by no more than 7 %.

10.3.8 Model Selection and Residual Analysis with Weibull Survival Data

Many of the facilities for model selection and residual analysis that we discussed in Chaps. 6 and 7 may also be used with Weibull modeling of survival data. For example, we may fit a model with all covariates as predictors, and then use backwards stepwise regression, using the AIC as a measure of goodness of fit, as follows:

```
modelAll.pharm.weib <- survreg(Surv(ttr, relapse) ~ grp + gender
                  + race + employment + yearsSmoking + level
                  Smoking + age + priorAttempts + longestNoSmoke,
                   dist="weibull")
model.step.pharm.weib <- step(modelAll.pharm.weib)
```

The resulting model, with "grp", "age", and "employment", is the same as we discussed in the previous section. We may also use the "residuals" function to compute deviance residuals and deletion residuals,

```
resid.deviance <- residuals(model.pharm.weib, type="deviance")
par(mfrow=c(2,2))
plot(resid.deviance ~ age)
smoothSEcurve(resid.deviance, age)
title("Deviance residuals\nversus age")

plot(resid.deviance ~ grp)
title("Deviance residuals\nversus treatment group")

plot(resid.deviance ~ employment)
title("Deviance residuals\nversus employment")
```

The results are shown in Fig. 10.6. Note that the function "residuals", when applied to a survreg object such as "modelAll.pharm.weib", recognizes the type of the object, and actually calls "residuals.survreg".

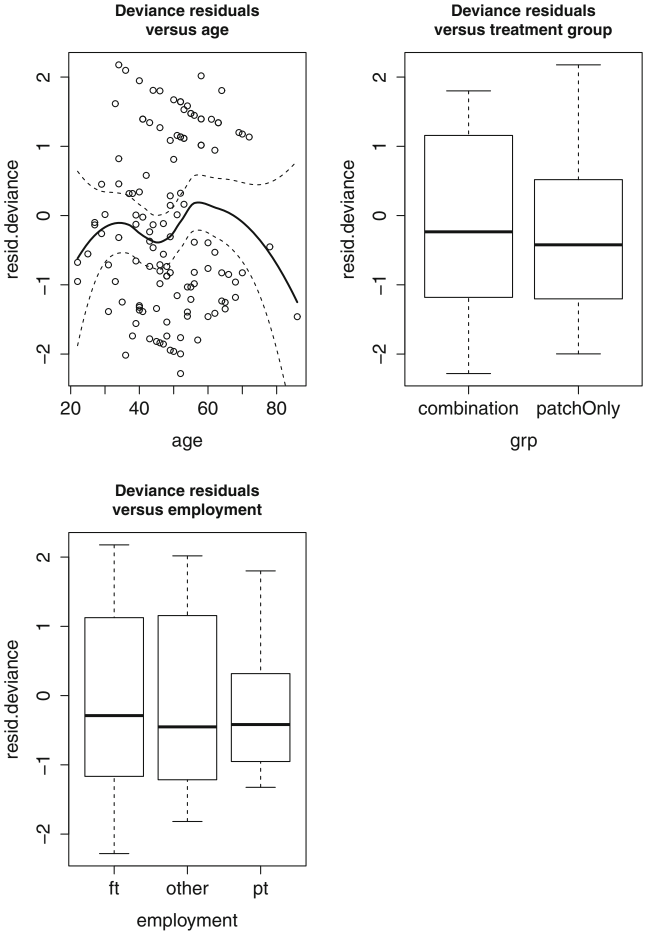


Fig. 10.6 Deviance residual plots from Weibull model fit to the pharmacoSmoking data

We see that the residual distributions of both “grp” and “employment” are reasonably comparable, indicating that these variables are modeled successfully. As for “age”, the distribution may be consistent with a linear model, when one considers the width of the 95 % confidence intervals. These results are similar to the diagnostics we saw with the Cox proportional hazards model, as shown in Fig. 7.2.

The effects of individual patients on the estimate of the coefficient for “age” may be computed as follows:

```
resid.dfbeta <- residuals(model.pharm.weib, type="dfbeta")
n.obs <- length(ttr)
index.obs <- 1:n.obs
plot(resid.dfbeta[,3] ~ index.obs, type="h",
   xlab="Observation", ylab="Change in coefficient",
   ylim=c(-0.0065, 0.004))
abline(h=0)
```

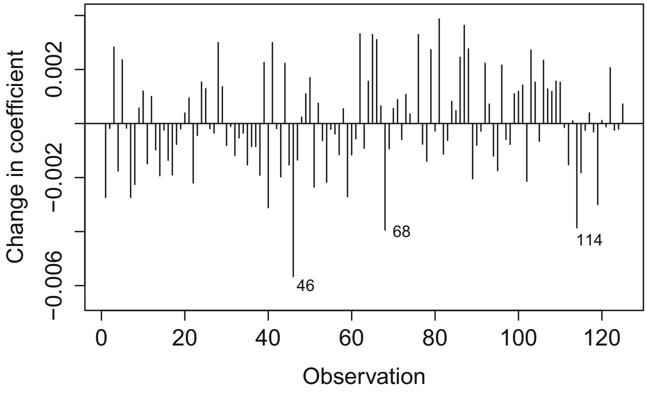


Fig. 10.7 "dfbeta" index plot for "age" in the Weibull model fit to the pharmacoSmoking data

The result is shown in Fig. 10.7. Compared to the corresponding plot for the Cox model (Fig. 7.3), we see that patients 46 and 68 are again influential, as is patient 114.

10.4 Other Parametric Survival Distributions

We may construct other accelerated failure time models by choosing other distributions for ϵ in Eq. 10.3.3. For example, if ϵ follows a standard normal distribution, the survival times T follow a log-normal distribution. We may fit this model as follows:

```
> model.pharm.lognormal <- survreg(Surv(ttr, relapse) ~ grp +
   age +
+   employment, dist="lognormal")
> summary(model.pharm.lognormal)
                  Value Std. Error     z        p
(Intercept)      1.6579     1.0084  1.64 1.00e-01
grppatchOnly    -1.2623     0.4523 -2.79 5.25e-03
age              0.0648     0.0203  3.20 1.39e-03
employmentother -1.1711     0.5316 -2.20 2.76e-02
employmentpt    -0.9543     0.7198 -1.33 1.85e-01
Log(scale)       0.8754     0.0796 10.99 4.15e-28
Scale= 2.4
Log Normal distribution
Loglik(model)= -451.5   Loglik(intercept only)= -461.7
        Chisq= 20.4 on 4 degrees of freedom, p= 0.00042
```

These parameter estimates are not from a proportional hazards model.

If ε has a logistic distribution, with survival distribution given by

$$S(u) = \frac{1}{1 + e^u},$$

then T has a *log-logistic* distribution. This model may be fitted using “survreg” as follows:

```
> model.pharm.loglogistic <- survreg(Surv(ttr, relapse) ~ grp +
+   age + employment, dist="loglogistic")
> summary(model.pharm.loglogistic)
                  Value Std. Error     z        p
(Intercept)      1.9150     0.9708  1.97 4.85e-02
grppatchOnly    -1.3260     0.4588 -2.89 3.85e-03
age              0.0617     0.0196  3.15 1.66e-03
employmentother -1.2605     0.5392 -2.34 1.94e-02
employmentpt    -1.0991     0.7050 -1.56 1.19e-01
Log(scale)       0.3565     0.0884  4.03 5.47e-05
Scale= 1.43
Log logistic distribution
Loglik(model)= -453.4   Loglik(intercept only)= -463.6
        Chisq= 20.47 on 4 degrees of freedom, p= 4e-04
```

With this distribution, the odds of survival are proportional,

$$\frac{S_1(t)}{1-S_1(t)} = e^{z\beta}\frac{S_0(t)}{1-S_0(t)}.$$

Just as the proportional hazards and accelerated lifetime models are equivalent for the Weibull distribution, the proportional odds and accelerated lifetime models are equivalent for the log-logistic distribution. The parameter estimates obtained from all “survreg” parametric procedures are for accelerated failure time models.

10.5 Additional Note

1. Many texts provide detailed discussions of the use of parametric models in survival analysis. Examples include Cox and Oakes [11], Kalbfleisch and Prentice [34], Klein and Moeschberger [36], and Tableman and Kim [65].

Exercises

10.1. Consider the “hepatoCellular” data in the “asaur” package. Use the method of Sect. 10.3.1 to assess how appropriate a Weibull distribution is for (a) overall survival, and (b) recurrence-free survival.

10.2. Test for the effect of CXCL17 on overall survival. Which of the three measures is the best predictor? Repeat for recurrence-free survival.

10.3. Using the covariates with complete data, use the “step” function to find a well-fitting model with low AIC for overall survival. Repeat for recurrence-free survival. Which covariates are included in both models?

10.4. Using the “ashkenazi” data in the “asaur” package, fit a Weibull distribution to the women with the “wild type” (non-mutant) BRCA genotype, matching the Kaplan-Meier survival curve at ages 45 and 65. Then predict the probability that a woman with the wild type BRCA genotype will develop breast cancer before the age of 70.

Chapter 11
Sample Size Determination for Survival Studies

Deciding how many subjects to include in a randomized clinical trial is a key component of its design. In the classical hypothesis testing framework, for any type of outcome, one must specify the effect change one is aiming for, the inherent variability in the test statistic, the significance level of the test, and the desired power of the test to detect the effect change. In survival analysis, there are additional factors that one must specify regarding the censoring mechanism and the particular survival distributions in the null and alternative hypotheses. First, one needs either to specify what parametric survival model one is using, or that the test will be semi-parametric, e.g., the log-rank test. This allows for determining the number of deaths (or events) required to meet the power and other design specifications. Second, one must, for administrative reasons, provide an estimate of the number of patients that need to be entered into the trial to produce the required number of deaths. We shall assume that the clinical trial is run as described in Chap. 1, where patients enter a trial over a certain accrual period of length a, and then followed for an additional period of time f known as the *follow-up time*. Patients still alive at the end of follow-up are censored. We will describe sample size methods for single arm clinical trials and then for two arm clinical trials.

11.1 Power and Sample Size for a Single Arm Study

In a study with a single arm, we assume for planning purposes that the survival times follow an exponential distribution with hazard $h(t; \lambda) = \lambda$ and survival distribution $S(t, \lambda) = e^{-\lambda t}$. We shall test $H_0 : \lambda = \lambda_0$ versus $H_A : \lambda = \lambda_A$. The null hypothesis mean, $\mu_0 = 1/\lambda_0$, would correspond to the mean survival one has observed in the past for the standard therapy, and the alternative hazard, $\mu_A = 1/\lambda_A$ is a (presumably) larger mean survival that we aim to find with a new, experimental therapy. Thus, the treatment ratio we would like to detect may

D. Moore, *Applied Survival Analysis Using R*, Use R!,
DOI 10.1007/978-3-319-31245-3_11

be written as $\Delta = \mu_A/\mu_0 = \lambda_0/\lambda_A$. After the trial is completed, we obtain a series of independent survival times $t_1, t_2, \ldots, t_n$ and censoring indicators $\delta_1, \delta_2, \ldots, \delta_n$, where n is the total number of subjects in the trial. Our ultimate goal is to determine how many patients we need to detect a certain hazard ratio Δ with a specified power and significance level. The derivation of the formula is similar to that for tests concerning the mean of normally distributed observations, but some adjustments are needed to account for the presence of censoring. The main adjustment is that our sample size formula will directly specify d, the number of *deaths* needed to achieve the desired power. Once we have d, we use a separate method to find the number of patients n needed to produce the required d.

In practice, the null and alternative hypotheses may be expressed in terms of either median survival or survival probability at a specified time t. A median survival may be converted into a hazard rate by re-expressing $0.5 = e^{-\lambda t}$ as $\lambda = [\log(2)]/t$. Similarly, a survival rate p at time t may be written as $p = e^{-\lambda t}$, which may be expressed as $\lambda = -[\log(p)]/t$. These comparisons are strictly valid only if the survival distribution is exponential. However, for other survival distributions, the conversions between median survival and survival rates may provide reasonable approximations when survival rates are in the neighborhood of 50 %.

The most direct way to derive a sample size formula is based on a Wald test, but the resulting formula varies depending on the parametrization of the likelihood. The simplest derivation uses the parameter $\theta = \log(\mu) = -\log(\lambda)$. Then we may express the log-likelihood function of Sect. 2.6 as follows:

$$l(\theta) = d\log\lambda - \lambda V = -\theta d - Ve^{-\theta} \tag{11.1.1}$$

By following the development in Sect. 2.6, the m.l.e. may be shown to be $\hat{\theta} = \log(V/d)$ and $\mathrm{var}(\hat{\theta}) \approx 1/d$, where $d = \sum \delta_i$ and $V = \sum t_i$ are the number of deaths and the total patient-time, respectively.

We will use $\hat{\theta}$ as our test statistic, and reject H_0 in favor of H_A if $\hat{\theta} > k$ for some constant k. The significance level of the test, or Type I error rate, is $\alpha = \Pr\left(\hat{\theta} > k | \theta = \theta_0\right)$. That is, the constant k is chosen so that the probability of rejecting the null hypothesis is α, assuming that the null hypothesis is true. Using a normalizing transformation,

$$Z = \frac{\hat{\theta} - \mu}{1/\sqrt{d}}$$

we have

$$\alpha = \Pr\left(Z > \frac{k - \theta_0}{1/\sqrt{d}}\right)$$

If z_α is the value of a standardized normal distribution that cuts off an area α to the right, then

$$z_\alpha = \frac{k - \theta_0}{1/\sqrt{d}}$$

and hence

$$k = \theta_0 + \frac{z_\alpha}{\sqrt{d}}$$

Now let us consider what happens under the alternative hypothesis, where $\theta = \theta_A$. The *power* of the test is given by

$$1 - \beta = \Pr\left(\hat{\theta} > k | \theta = \theta_A\right) = \Pr\left(Z > \frac{k - \theta_A}{1/\sqrt{d}}\right)$$

or equivalently,

$$z_{1-\beta} = -z_\beta = \sqrt{d}\,(k - \theta_A)$$

Substituting the value of k, we have

$$-z_\beta = \sqrt{d}\left(\theta_0 + \frac{z_\alpha}{\sqrt{d}} - \theta_A\right)$$

Solving for d we have

$$d = \frac{\left(z_\beta + z_\alpha\right)^2}{(\theta_A - \theta_0)^2} = \frac{\left(z_\beta + z_\alpha\right)^2}{(\log \Delta)^2} \tag{11.1.2}$$

since $\log(\Delta) = \log(\lambda_0) - \log(\lambda_A)$.

This gives us the number of *deaths* needed to achieve the specified power, not the number of patients. This derivation method is based on the parameter estimate normalized by its standard deviation, which produces the Wald test statistic. One aspect of this test is that different parametrizations lead to somewhat different formulas for the sample size. This dependence on parametrization may be largely avoided by working directly with the likelihood ratio statistic. Now, for fixed d, $V = \sum t_i$ has a gamma distribution with index d and scale parameter λ. Cox and Oakes [11], following Epstein and Sobel [16], show that

$$W = \frac{2d\lambda}{\hat{\lambda}} \sim \chi^2_{2d}$$

although this result is approximate for general censoring patterns. Under $H_0 : \lambda = \lambda_0$, we need to find a constant k such that

$$\alpha = \Pr\left(1/\hat{\lambda} > k|\lambda = \lambda_0\right) = \Pr\left(W > 2dk\lambda_0\right)$$

and hence $\chi^2_{2d,\alpha} = 2dk\lambda_0$. Finally,

$$k = \frac{\chi^2_{2d,\alpha}}{2d\lambda_0}. \tag{11.1.3}$$

The power of the test is given by

$$1 - \beta = \Pr\left(1/\hat{\lambda} > k|\lambda = \lambda_A\right) = \Pr\left(W > 2dk\lambda_A\right)$$

where

$$k = \frac{\chi^2_{2d,1-\beta}}{2d\lambda_A} \tag{11.1.4}$$

Equating Eqs. 11.1.3 and 11.1.4, we have

$$\Delta = \frac{\lambda_0}{\lambda_A} = \frac{\chi^2_{2d,\alpha}}{\chi^2_{2d,1-\beta}} \tag{11.1.5}$$

For specified α, power $1-\beta$, and ratio Δ, we may solve this for the required number of deaths, d.

In R, we may compute the number of deaths based on Eq. 11.1.2 using the following function:

```
expLogMeanDeaths <- function(Delta, alpha, pwr) {
        z.alpha <- qnorm(alpha, lower.tail=F)
        z.beta <- qnorm(1-pwr, lower.tail=F)
        num <- (z.alpha + z.beta)^2
        denom <- (log(Delta))^2
        dd <- num/denom
        dd   }
```

We use the "qnorm" function to compute z_α and z_β, and the final result is the number of deaths required to detect a hazard ratio Δ with significance level α and power $1-\beta$. To use the likelihood ratio method, we first compute the hazard ratio Δ given α, $1-\beta$, and a specified number of deaths d :

```
expLikeRatio <- function(d, alpha, pwr) {
        num <- qchisq(alpha, df=(2*d), lower.tail=F)
        denom <- qchisq(pwr, df=(2*d), lower.tail=F)
        Delta <- num/denom
        Delta }
```

Here we use the "qchisq" function to compute $\chi^2_{2d,\alpha}$ and $\chi^2_{2d,1-\beta}$. To get the number of deaths d for a specified Δ, we define a new function "LRD" internally which is zero at the required number of deaths:

```
expLRdeaths <- function(Delta, alpha, pwr) {
        LRD <- function(x, alpha, pwr)
             expLikeRatio(x, alpha, pwr) - Delta
        result <- uniroot(f=LRD, lower=1,upper=1000,
             alpha=alpha, pwr=pwr)
        result$root }
```

Suppose that we are designing a Phase II oncology trial where we plan a 5 % level (one-sided) test, and we need 80 % power to detect a hazard ratio of 1.5. Once the above functions have been entered into R, we can find the required number of deaths as follows:

```
> expLRdeaths(1.5, 0.05, 0.8)
[1] 36.33916

> expLogMeanDeaths(1.5, 0.05, 0.8)
[1] 37.60635
```

That is, we would need (rounding up) 38 deaths according to the log mean method (function "expLogMeanDeaths"), and 37 deaths according to the likelihood ratio method (function "expLikeRatio"). The two methods give similar results over a wide range of specifications (Narula and Li [53]).

11.2 Determining the Probability of Death in a Clinical Trial

In the previous section, we saw how to compute the required number of deaths to satisfy the power, significance, and survival difference design requirements of a trial. But as we have seen, in survival analysis, many subjects are still alive at the time of analysis. For administrative reasons, we usually need to specify how many patients need to be entered onto the trial, not how many will die. Thus, we need to provide an estimate of the proportion π of patients who will die by the time of analysis. If all patients entered at the same time, we would simply have $\pi = 1 - S(t, \lambda)$, where t is the follow-up time. However, patients actually enter over an accrual period of length a and then, after accrual to the trial has ended, they are followed for an additional time f. So a patient who enters at time $t = 0$ will have failure probability $\pi(0) = 1 - S(a + f, \lambda)$, since a patient who enters at time $t = 0$ will have the maximum possible follow-up time $a + f$. But a patient who enters at time a, which is the end of the accrual period, will have the minimum possible follow-up time f. Thus, that patient will have failure probability $\pi(a) = 1 - S(f, \lambda)$. This is illustrated in Fig. 11.1

If patients were entered in equal numbers at times 0 (curve A) or a (curve B) only, then the probability of death would be $(\pi(0) + \pi(a))/2$. A much more realistic scenario is that the patients enter uniformly between times 0 and a, so that the

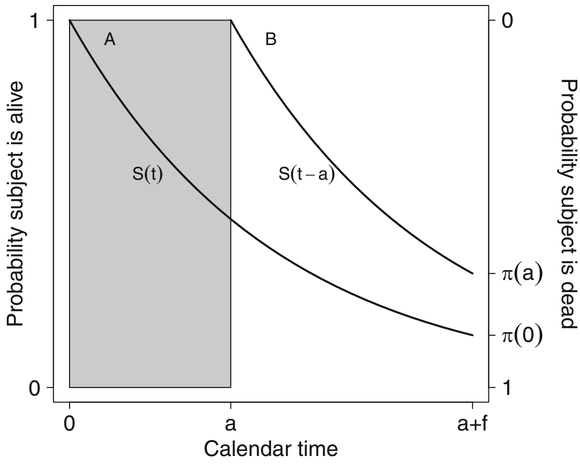


Fig. 11.1 Probability of death for subjects entering at the beginning (curve A) or the end (curve B) of the accrual period. The probability of death is given on the right axis

patient entry follows a Uniform$(0,a)$ distribution. Then the probability of death π is obtained by averaging over these times, so that a patient that enters at time t is followed for additional time $a+f-t$. This idea may be expressed by the following integral, which uses the fact that the probability of death given the patient enters at time t is $1-S(a+f-t;\lambda)$,

$$\pi = \int_0^a \frac{1}{a} \Pr\big(\text{death} \mid \text{enter at time } t\big)\, dt$$

or just

$$\pi = \int_0^a \frac{1}{a}\left(1 - S(a+f-t;\lambda)\right) dt \tag{11.2.1}$$

Using the variable transformation $u = a+f-t$ and re-arranging, we have

$$\pi = 1 - \frac{1}{a}\int_f^{a+f} S(u;\lambda) du \tag{11.2.2}$$

Since we are assuming an exponential distribution, we have $S(u;\lambda) = e^{-\lambda u}$, and we have, using basic integration,

$$\pi = 1 - \tfrac{1}{a}\int_f^{a+f} e^{-\lambda u} du = 1 - \tfrac{1}{a\lambda}\left\{e^{-\lambda f} - e^{-\lambda(a+f)}\right\}. \tag{11.2.3}$$

The following function uses this expression to compute the probability of death:

```
prob.death <- function(lambda, accrual, followup) {
  probDeath <- 1 - (1/(accrual*lambda))*
    (exp(-lambda*followup) - exp(-lambda*(accrual + followup)))
        probDeath
        }
```

Consider again our example in the previous section where we plan a single sample clinical trial with a 5 % significance level (one-sided) test, and we need 80 % power to detect a hazard ratio of 1.5. Suppose that the null hypothesis rate is $\lambda_0 = 0.15$, so that the alternative hypothesis hazard rate is $\lambda_1 = \lambda_0/\Delta = 0.15/1.5 = 0.10$. We suppose now that the accrual period is $a = 2$ years and that the follow-up period is an additional $f = 3$ years. Previously we found that 38 deaths were needed. To obtain an estimate of the number of patients needed to produce this number of deaths, we first compute the probability of death under $H_1 : \lambda = 0.10$.

```
> prob.death(lambda=0.10, accrual=2, followup=3)
[1] 0.3285622
```

Then the number of patients needed is approximately $38/0.3285622 = 115.6$, or 116 after rounding up. This estimate depends critically not only on the assumption of an exponential distribution, but also on the unknown value of the exponential parameter λ. Using a specific value, such as $\lambda_1 = 0.10$, is helpful in the planning stage of the trial, since administrators will need an estimate of the number of patients that will be needed. However, to maintain the integrity of the design, it would be preferable to tie the stopping rule for the trial to the number of deaths, 38, rather than to the estimated total number of patients, 116.

11.3 Sample Size for Comparing Two Exponential Survival Distributions

We now consider a comparative clinical trial, where an experimental regimen is being compared to a standard, control regimen. Suppose that we are going to test the null hypothesis $H_0 : S_0 \geq S_1$ versus the alternative $H_A : S_0 < S_1$ for all t, where S_0 and S_1 are exponential survival distributions with hazards λ_0 and λ_1, for the control and experimental regimens, respectively. To determine the required sample size, we consider the hazard ratio $\Delta = \lambda_0/\lambda_1$. Now λ_0 and λ_1 are the hazards for a control and experimental treatments, and we presume that the latter hazard is the smaller one, and that our test may be viewed as the one-sided test of $H_0 : \Delta = 1$ versus $H_A : \Delta > 1$. This well-known case was developed by Bernstein and Lagakos [6], Rubenstein et al. [58] and others. We let p denote the proportion of patients randomized to the control group. Typically one uses equal randomization, so that $p = 0.5$, but this is not required. We denote by n the total number of patients in the trial, and we have $n_0 = np$ and $n_1 = n(1 - p)$ control and experimental patients, respectively. When the trial has been completed, we will observe d_0 and d_1

deaths in the control and experimental groups, and total patient times of $V_0 = \sum t_{0i}$ and $V_1 = \sum t_{1i}$, respectively. We know from Sect. 2.6 that the maximum likelihood estimates of the hazards are $\hat{\lambda}_0 = d_0/V_0$ and $\hat{\lambda}_1 = d_1/V_1$. To compare the two distributions, it is more convenient to use $\delta = \log\Delta = \log\lambda_0 - \log\lambda_1$, since the log scale transformed value will be more symmetric. One may show that, based on maximum likelihood theory,

$$\mathrm{var}(\hat{\delta}) = \sigma^2 = \left(\frac{1}{E(d_0)} + \frac{1}{E(d_1)}\right) = \frac{1}{n_0\pi_0} + \frac{1}{n_1\pi_1} = \frac{1}{np(1-p)} \cdot \frac{p\pi_0 + (1-p)\pi_1}{\pi_0\pi_1}.$$

where π_0 and π_1 are the probabilities of death in the control and treatment groups, respectively. If we define a new parameter $\tilde{\pi}$ as follows,

$$\tilde{\pi} = \left(\frac{p\pi_0 + (1-p)\pi_1}{\pi_0\pi_1}\right)^{-1} = \left(\frac{p}{\pi_1} + \frac{1-p}{\pi_0}\right)^{-1} \tag{11.3.1}$$

we see that $\tilde{\pi}$ is a weighted harmonic mean of π_0 and π_1, and thus may be viewed as an average probability of death across the control and treatment groups. The harmonic mean $\tilde{\pi}$ is an approximation to the weighted mean $\bar{\pi} = p\pi_0 + (1-p)\pi_1$. Thus, we have

$$\mathrm{var}(\hat{\delta}) = \sigma^2 = \frac{1}{np(1-p)} \cdot \tilde{\pi}^{-1}$$

Expressing the test in terms of $\delta = \log\Delta$, we reject $H_0 : \delta = 0$ in favor of $H_A : \delta > 0$ if $\hat{\delta} > k$ for some constant k. For a one-sided test, we have, following an argument similar to that in Sect. 11.1,

$$\alpha = \Pr\left(\hat{\delta} > k|\delta = 0\right) = \Pr\left(Z > k/\sigma\right),$$

where σ is defined above. Then $k = z_\alpha\sigma$. The power is given by

$$1 - \beta = \Pr\left(\hat{\delta} > k|\delta\right) = \Pr\left(Z > \frac{k-\delta}{\sigma}\right).$$

So, $z_{1-\beta} = -z_\beta = \frac{k-\delta}{\sigma}$. Substituting the value of k, we get $z_\beta = \frac{\delta}{\sigma} - z_\alpha$, and finally

$$\frac{\delta^2}{\left(z_\alpha + z_\beta\right)^2} = \sigma^2 = \frac{1}{np(1-p)} \cdot \tilde{\pi}^{-1}.$$

Solving for n, we have

$$n = \frac{\left(z_\alpha + z_\beta\right)^2}{\delta^2 p(1-p)\tilde{\pi}}$$

This states that the required number of patients is the number of deaths,

$$d = \frac{\left(z_\alpha + z_\beta\right)^2}{\delta^2 p(1-p)} \tag{11.3.2}$$

divided by the probability of death, $\tilde{\pi}$, as in Sect. 11.2. The only difference here is that the probability of death is an average (actually the geometric mean) of the death probabilities in the control and treatment groups (Eq. 11.3.1). The harmonic mean effectively weights the smaller of the two death probabilities more heavily than does the sample mean $\tilde{\pi}_m = (\pi_1 + \pi_0)/2$. The consequence is that the harmonic mean estimate of the required sample size will be larger than the estimate one would obtain using the sample mean. We will see this in the worked examples.

11.4 Sample Size for Comparing Two Survival Distributions Using the Log-Rank Test

When the data from a completed comparative clinical trial are analyzed, typically one uses a log-rank test rather than a test based on the exponential distribution, for reasons discussed in Chap. 3. Thus, it would seem reasonable to develop a sample size formula based on the log-rank test. With this test, which is based on the proportional hazards assumption, the ratio $\Delta = \lambda_0(t)/\lambda_1(t)$ is constant, but the baseline hazard $\lambda_0(t)$ is unspecified. We then use the log-rank statistics U_0 and its variance, which for the ith failure time is given by

$$v_{0i} = \mathrm{var}(d_{0i}) = \frac{n_{0i}n_{1i}d_i(n_i - d_i)}{n_i^2(n_i - 1)}$$

As explained in Chap. 3, these are also the score function of the partial likelihood and its variance. Assuming that the number of deaths at each failure time is small compared to the number at risk, and that the proportion $p \approx n_{0i}/n_i$ of subjects assigned to the control group is constant over time, we have the following approximation,

$$v_{0i} = \mathrm{var}(d_{0i}) = \frac{n_{0i}n_{1i}d_i(n_i - d_i)}{n_i^2(n_i - 1)} \approx \frac{n_{0i}n_{1i}d_i}{n_i^2} \approx p(1-p)d_i$$

The variance of the log-rank statistic is then approximately

$$V_0 = \sum_{i=1}^{D} v_{0i} \approx p(1-p)\sum_{i=1}^{D} d_i = p(1-p)d$$

Then using a standard sample size and power calculation (see Collett [10] and Schoenfeld [60, 61] for details), we find that the number of events needed to detect a treatment difference $\delta = \log(\lambda_0(t)/\lambda_1(t)) = \log(\varDelta)$ with power $1-\beta$ and with a two-sided level α log-rank test, is identical to that given in Eq. 11.3.2.

$$d = \frac{\left(z_{\alpha/2} + z_\beta\right)^2}{p(1-p)\delta^2} \tag{11.4.1}$$

or, if the same number of patients are in both groups, by

$$d = \frac{\left(z_{\alpha/2} + z_\beta\right)^2}{\delta^2/4} \tag{11.4.2}$$

As in previous cases, this formula gives us the required number of *deaths (*or *events*), not the number of patients. Estimating the number of patients that are needed requires and estimate of the probability of death, as we saw in Sect. 11.3.

11.5 Determining the Probability of Death from a Non-parametric Survival Curve Estimate

One way of determining the number of patients needed to produce a particular number of deaths is to assume that patients enter uniformly over the accrual period, and that survival is governed by an exponential distribution. Then we can proceed as in Sect. 11.2. However, if a survival distribution estimate is available for the control group, say, from an earlier trial, then we can use that, along with the proportional hazards assumption, to estimate a probability of death without assuming that the survival distribution is exponential.

Typically, the survival function for the control group of a randomized trial is a Kaplan-Meier estimate $\hat{S}_0(t)$ obtained from a prior study. If we need to detect a hazard ratio Δ, the alternative hypothesis survival function will be, assuming proportional hazards,

$$\hat{S}_1(t) = \left[\hat{S}_0(t)\right]^{1/\Delta}. \tag{11.5.1}$$

Then to compute the expected number of deaths with accrual and followup times a and f, we use the weighted mean survival,

$$\hat{S}(t) = p\hat{S}_0(t) + (1-p)\hat{S}_1(t) \tag{11.5.2}$$

where p is the proportion of subjects randomly assigned to the control group.

Given a survival function $\hat{S}(t)$ there are a number of ways of evaluating the integral in Eq. 11.2.2. In most cases we may obtain a good approximation of the integral by evaluating $\hat{S}(t)$ for a patient entering at time 0, $a/2$, and a, and use some results from elementary integral calculus. One simple approach is the trapezoidal rule, which uses the areas under two trapezoids defined by the time points 0, $a/2$, and a and values that match the integrand at these points. This yields an estimate of the probability of death given by

$$\pi_t \approx 1 - \frac{1}{4}\left\{\hat{S}(a+f) + 2\hat{S}\left(\frac{a}{2}+f\right) + \hat{S}(f)\right\}.$$

Alternatively, we can use Simpson's rule, which uses the area under a quadratic polynomial that matches the integrand at these three points [10, 61]. The underlying algebra is a bit tedious, but the well-known end result is quite simple:

$$\pi_s \approx 1 - \frac{1}{6}\left\{\hat{S}(a+f) + 4\hat{S}\left(\frac{a}{2}+f\right) + \hat{S}(f)\right\}.$$

The most accurate method is to evaluate the integral numerically. Since the survival estimate $\hat{S}(t)$, a weighted mean of $\widehat{S_0}(t)$ and $\widehat{S_1}(t)$, is a step function, the integral may be written as a sum of areas of rectangles under each "step" at the failure times between a and $a+f$. To do this we denote all of the ordered failure times by $t_{(1)}, t_{(2)}, \ldots, t_{(n)}$. Then the integral in Eq. 11.2.2 may be estimated as follows:

$$\pi_r = \sum_{t_{(i)}: f < t_{(i)} \leq a+f} \left[\hat{S}(a+f-t_{(i)}) \cdot \left(t_{(i)} - t_{(i-1)}\right)\right].$$

We may illustrate estimating π using the data "gastricXelox" from Chap. 3. Figure 11.2 illustrates this.

The probability of death may be computed as follows. First, we extract the failure times and survival probabilities:

```
library(survival)
result.km <- survfit(Surv(timeMonths, delta) ~ 1,
     conf.type="log-log")
timesXe <- result.km$time
survXe <- result.km$surv
```

Next we set up the accrual and follow-up times, and select the portion of the failure times in the interval from f to $a+f$, taking care to include the time f for the first rectangle:

```
accrual <- 12
followup <- 6
times.use <- c(followup, timesXe[{timesXe >= followup} &
   {timesXe <= accrual + followup}])
surv.use <- summary(result.km, times=times.use)$surv
```

Finally, we use the "diff" function to get the widths of the rectangles, and "sum" to complete the evaluation of Eq. 11.2.2:

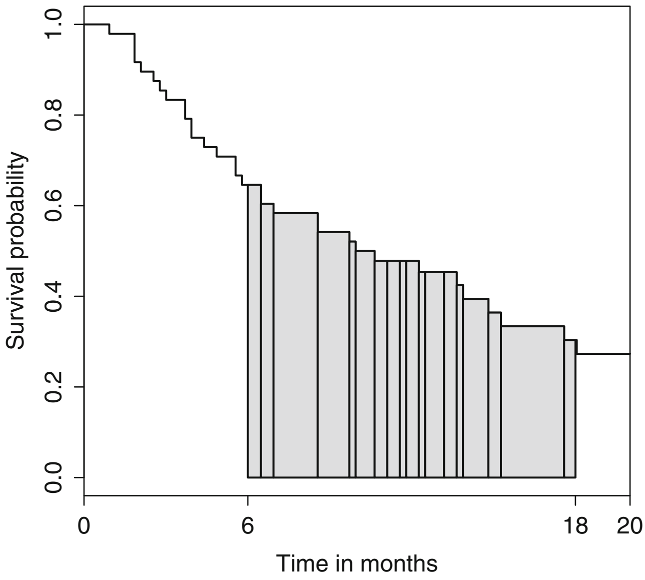


Fig. 11.2 Portion of the Kaplan-Meier plot (Fig. 3.6) showing the rectangles needed for the integral in Eq. 11.2.2

```
> times.diff <- diff(c(times.use, accrual + followup))
> pi.rec <- 1 - (1/accrual)*sum(times.diff*surv.use)
> pi.rec
[1] 0.5365161
```

Thus, we estimate, using the "rectangle" method, that $\pi_r = 0.536$. To do this using Simpson's method, we first evaluate the survival function at f, $a/2 + f$, and $a + f$:

```
> surv.simpson <- summary(result.km,
+     times=c(followup, accrual/2 + followup, accrual+followup))
   $surv
> surv.simpson
[1] 0.6458333 0.4782609 0.3034080
```

Then we evaluate the probability of death:

```
> pi.simpson <- 1 -
+   (1/6)*(surv.simpson[1] + 4*surv.simpson[2] + surv.simpson[3])
> pi.simpson
[1] 0.5229525
```

Thus, our estimate of the probability of death using this method is $\pi_s = 0.523$, which is close to the rectangle method estimate $\pi_e = 0.536$.

However we estimate the probability of death π, the number of patients needed for the trial will be estimated by $n = d/\pi$, where d is the required number of deaths.

11.6 Example: Calculating the Required Number of Patients for a Randomized Study of Advanced Gastric Cancer Patients

Suppose now that we need to design a two-arm randomized clinical trial to test the effect of a new therapy to Xelox in patients with advanced gastric cancer. For the control arm survival distribution we use the PFS survival curve in Fig. 3.3, and we wish to have 80 % power to detect an alternative hazard ratio $\Delta = 2$ (for an alternative experimental therapy) using a 2.5 % level one-sided log-rank test. We again assume that we will accrue patients for 12 months and follow them for an additional 6 months. We first determine the number of events that we require, using the following R function:

```
TwoArmDeaths <- function(Delta, p=0.5, alpha=0.025, pwr=0.8) {
  z.alpha <- qnorm(alpha, lower.tail=F)
  z.beta <- qnorm(1-pwr, lower.tail=F)
  num <- (z.alpha + z.beta)^2
  denom <- p*(1-p)*(log(Delta))^2
  dd <- num/denom
  dd   }
```

The number of deaths is as follows:

```
> TwoArmDeaths(Delta=2, p=0.5, alpha=0.025, pwr=0.8)
[1] 65.34566
```

That is, we need 66 events total.

To determine the probability of death under the alternative hypothesis, we take the survival function defined by "surv.use" in the previous section, and compute the average survival according to Eqs. 11.5.1 and 11.5.2:

```
Delta <- 2
surv.alt <- surv.use^(1/Delta)
surv.avg <- 0.5*surv.use + 0.5*surv.alt
```

The exact estimate of the probability of death is obtained as before, using "surv.avg" in place of "surv.use":

```
> pi.exact <- 1 - (1/accrual)*sum(times.diff*surv.avg)
> pi.exact
[1] 0.4298326
```

Thus, the probability of death is 0.430. Finally, the required number of patients is given by $65.346/0.430 = 152.0$. Thus, we would need to enroll 152 patients, 76 in each arm, to meet the design specifications.

If the full survival curve is unavailable, we may still estimate the sample size by specifying the null and alternative survival distributions in terms of median survival. We can then directly convert these into exponential parameters using $\lambda = \log 2/t_m$, where t_m is the median survival time. This approximation will be reasonable if the hazard at the median survival time is near the median of an exponential distribution. In the current example, we have $\lambda_0 = \log(2)/10.3 = 0.0673$. Thus, the hazard for the alternative hypothesis will be $\lambda_1 = 0.0673/2 = 0.0336$. As discussed earlier, the required number of events under the exponential assumption is 66, the same as we need using a log-rank test. The probability of death, however, is obtained using Eq. 11.2.3 and the harmonic mean of the probabilities in the control and treatment arms using Eq. 11.3.1. In R, using the function "prob.death" defined in Sect. 11.2,

```
> pi0 <- prob.death(lambda=0.0673, accrual=12, followup=6)
> pi1 <- prob.death(lambda=0.0336, accrual=12, followup=6)
> pi.harmonicMean <- 1/(0.5/pi0 + 0.5/pi1)
> pi.harmonicMean
[1] 0.408085
```

Thus, the probability of death is 0.408, as compared to 0.430 for the nonparametric estimate. The required number of patients under the exponential assumption is $65.346/0.408 = 160.2$, so we would need 162 patients all together. This is the result according to the method in Bernstein and Lagakos [6], as presented in Sect. 11.3. This estimate is somewhat higher than the value 152 we obtained using the nonparametric approach. To better understand the difference, suppose that, instead of the harmonic mean 0.408, we use the sample mean:

```
> pi.avg <- (pi0 + pi1)/2
> pi.avg
[1] 0.4345708
```

Then the number of subjects we would need according to this estimate of the probability of death would be $65.346/0.434 = 150.6$, or 152 in total. This is essentially the same as that obtained using the "nonparametric" approach, and smaller than using the harmonic mean approach. The upshot is that the estimate of the total number of patients is highly sensitive to the method of computing the probability of death.

11.7 Example: Calculating the Required Number of Patients for a Randomized Study of Patients with Metastatic Colorectal Cancer

Morse et al. [50] reported a Kaplan-Meier plot of overall survival probabilities of 161 patients with metastatic colon cancer who had undergone metastasectomy (surgical removal of cancerous growths that have spread from the colon) in Figure 5B of the paper. The survival probabilities at 24, 36, and 48 months (2, 3, and 4 years) are, respectively, 0.76, 0.59, and 0.49. Suppose that we plan a randomized phase III study comparing a new therapy to placebo for these patients. We plan to carry out a 0.025 level log-rank test, and wish to have 85 % power to detect an increase in the three-year survival probability from 0.59 (the current untreated rate) to 0.75. How many patients do we need?

First, we find the hazard ratio by solving $0.59 = 0.80^{\Delta}$. Taking logs, we have $\Delta = \log(0.59)/\log(0.75) = 1.834$. Then the number of deaths required is

```
> TwoArmDeaths(Delta=1.834, p=0.5, alpha=0.025, pwr=0.85)
[1] 97.63333
```

or about 98. A simple function to compute the probability of death using Simpson's rule is as follows:

```
pDeathSimpson <- function(aa, ff, S) {
 #Use Simpson's rule to approximate the probability of death
 #   assuming uniform accrual
 probDeath <- 1 - (1/6)*(S[1] + 4*S[2] + S[3])
 probDeath    }
```

where "aa" is the accrual period, "ff" is the follow-up period, and "S" is a vector with three elements corresponding to the survival probabilities at times "ff", "ff + 0.5*aa", and "ff + aa". We define the parameters as follows:

```
aa=2
ff=2
So <- c(0.76, 0.59, 0.49)
psi = 1.834
Sa <- So^(1/psi)
Sboth <- 0.5*(So + Sa)
```

The probabilities of death in the control arm, treatment arm, and the average, are as follows:

```
> pDeathControl <- pDeathSimpson(aa=2, ff=2, S=So)
> pDeathControl
[1] 0.4003333
> pDeathTreatment <- pDeathSimpson(aa=2, ff=2, S=Sa)
> pDeathTreatment
[1] 0.2449027
> pDeathAll <- 0.5*(pDeathControl + pDeathTreatment)
> pDeathAll
[1] 0.322618
```

Thus, the total number of patients necessary to produce and expected value of 97.63 deaths is $97.63/0.322618 = 302.6$, or $304/2 = 152$ patients per arm.

11.8 Using Simulations to Estimate Power

The previous sections described methods for computing sample size and power for specific situations for which explicit formulas are available. An alternative approach to estimating power is to simulate a large number of survival data sets from a particular distribution and accrual pattern, and empirically compute the power. Specifically, suppose that we are computing power for a two arm randomized clinical trial comparing a standard therapy to an experimental therapy. Based on past studies of the standard therapy, we may select a parametric distribution that approximates the survival of patients given the standard therapy. We also specify a hazard ratio that we would like to detect. Then, we model the entry of patients over a specified accrual period, randomization to either of the two arms, and follow them until death or, for those still alive at the end of a pre-specified follow-up period, until they are censored. We then compute a test statistic and p-value using, typically, a log-rank test. We repeat this process a large number of times, and observe the proportion of times we reject the null hypothesis of no treatment difference. This is the estimated power.

Example 11.1. Let us design a clinical trial to determine if an experimental agent can increase the time to death from prostate cancer among patients diagnosed with advanced localized prostate caner. The first step is to specify the eligibility criteria, and to use the data in "prostateSurvival" to determine the survival distribution (defined as time to death from prostate cancer) to select a Weibull distribution that matches the data. Specifically, we shall consider men aged 66 to 74 with newly diagnosed, poorly differentiated, stage T2 prostate cancer, and find a Weibull distribution that matches the survival proportions of these patients at four and eight years. In R, we first define the population of interest, "prost.66to74.poor", and then define a censoring variable "status.prost" that indicates death from prostate cancer. (Patients who die of other causes are here considered censored, as are patients still alive at the last time of follow-up.) We obtain a Kaplan-Meier survival distribution for these data, and find the survival probabilities at 48 and 96 months (that is, at four and eight years), as follows:

```
> attach(prostateSurvival)
> prost.66to74.poor <- prostateSurvival[{{grade == "poor"}  &
+   {{ageGroup == "70-74"} | {ageGroup == "66-69"}} &
    {stage == "T2"}},]
> library(survival)
> status.prost <- as.numeric(prost.66to74.poor$status == 1)
> result.prost.survfit <- survfit(Surv(survTime, status.
     prost) ~ 1,
+   data=prost.66to74.poor)
> summary(result.prost.survfit, time=c(48, 96))
 time n.risk n.event survival std.err lower 95% CI upper 95% CI
48    154      16     0.931   0.0171        0.898        0.965
96     26      17     0.717   0.0565        0.615        0.837
```

Next, we find a Weibull distribution that matches the survival probabilities at these two times. While we could use the methods described in Sect. 10.3.4, it will be more convenient to use some facilities in the R package "Hmisc" developed by Frank Harrell. This package must be downloaded and installed separately. Then we define the Weibull function using the "Weibull2" function as follows:

```
library(Hmisc)
Weib.p <- Weibull2(c(4,8),c(0.931,0.717))
```

This creates a function named "Weib.p" that computes, for any time, the survival probability based on the Weibull distribution that we have specified. We may take a peek at it as follows:

```
> Weib.p
function (times = NULL, alpha = 0.0033021237632906,
+  gamma = 2.21819823268731)  {
    exp(-alpha * (times^gamma))
        }
```

The parameters that “Weibull2” has computed are clearly indicated. (The parametrization used in the Hmisc package is somewhat different than what we have used in this text, but that doesn’t matter, since the associated survival functions in this package are consistent with this.)

Next, we define a set of two functions using the Hmisc function “Quantile2”, which allows us to specify the control survival function (“Weib.p”) and the hazard ratio. The hazard ratio here is assumed to be a constant, 0.75, which would indicate that the experimental agent would reduce the hazard of prostate cancer mortality by 25 %. (It is defined as a function because Quantile2 allows specification of more complex hazard ratio relationships. The result, the R object “ff”, specifies the control and experimental survival distributions. There is a plot “method” for “ff”, which means that the ordinary “plot” function will plot the survival distributions for both the control and experiment arms. Here is the R code:

```
ff <- Quantile2(Weib.p,hratio=function(x) 0.75)
plot(ff, xlim=c(0,8))
```

Before we can carry out the simulation, we need to specify names for the two survival distributions and extract them from “ff” (Fig. 11.3).

```
rcontrol <- function(n) ff(n, what='control')
rintervention  <- function(n) ff(n, what='intervention')
```

We also need to specify the censoring distribution, and to do this, we have to select the accrual and follow-up times. Here, let us accrue patients over three years, and follow them for an additional seven years. For now we shall assume that the accrual will follow a uniform distribution. This leads to the censoring distribution being uniform, with a minimum of five years (for a patient entering at the end of the accrual period) and a maximum of eight years (for a patient entering at the start of the trial). We specify the censoring distribution “rcens” using the R function “runif” as follows (using time in years):

```
rcens    <- function(n) runif(n, 5, 8)
```

We carry out the power simulation using the Hmisc function “spower”. Here we specify that there will be nc = 1500 patients enrolled in the control arm and

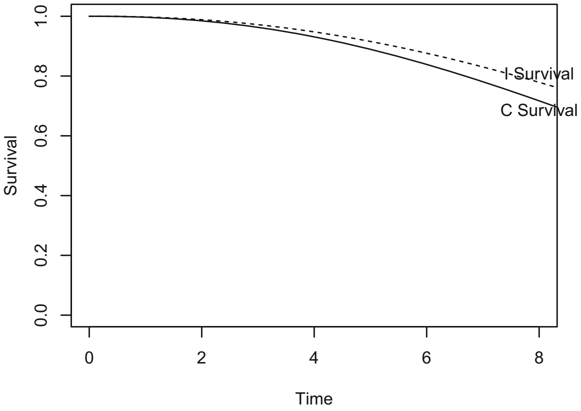


Fig. 11.3 Survival curves based on a Weibull distribution for times to death from prostate cancer, for use in a power simulation. The *solid curve* is for the control group and the *dashed curve* is for the intervention group

ni = 1500 in the intervention arm. We will simulate nc = 1000 data sets, and carry out a logrank test at the 0.025 significance level, as follows:

```
> spower(rcontrol, rintervention, rcens, nc=1500, ni=1500,
+        test=logrank, nsim=1000, alpha=0.025)
[1] 0.827
```

The result of the function, 0.827, is the power of the test. We can verify the significance level of the test by specifying "rcontrol" as both the control and intervention distribution,

```
> spower(rcontrol, rcontrol, rcens, nc=1500, ni=1500,
+        test=logrank, nsim=1000, alpha=0.025)
[1] 0.028
```

We see that the empirical Type I error rate is 2.8 %, which is consistent with a 2.5 % level test.

Computing power using simulations has the advantage that it can accommodate deviations from the usual assumptions of uniform accrual, proportional hazards, and perfect patient compliance. The "spower" function, combined with "Quantile2", can model a wide variety of such deviations. For example, suppose that we expect that 10 % of the patients on the intervention arm will be non-compliant, in that they to not take the experimental agent. We may include that noncompliance factor in the simulation via the "dropout" argument in the "Quantile2" function.

```
> ff.dropout <- Quantile2(Weib.p,hratio=function(x) 0.75,
+                         dropout=function(x) 0.10)
> rcontrol <- function(n) ff.dropout(n, what='control')
> rintervention  <- function(n) ff.dropout(n, what='intervention')
> spower(rcontrol, rintervention, rcens, nc=350, ni=350,
+               test=logrank, nsim=1000, alpha=0.025)
[1] 0.734
```

We see that the noncompliance has resulted in a loss of power, from 82.7 % to 73.4 %.

The "spower" suite of functions can accommodate a wide variety of deviations, including non-uniform accrual, non-proportional hazards, and noncompliance in either the control or intervention subjects. Details and examples may be found in the R help file for "spower" in the "Hmisc" package. If one needs to find the sample size or detectable hazard ratio for a specific power (80 % for example), one can use trial and error. Alternatively, one can define an R function that takes (for example) the hazard ratio as an argument and returns the power. Using that, combined with the R function "uniroot", one can solve to get the detectable hazard ratio.

11.9 Additional Notes

1. Freedman [20] derived an alternative to Eq. 11.4.2 for the number of deaths required using the log-rank test:

$$d = \frac{(z_{\alpha/2} + z_\beta)^2(\Delta + 1)^2}{(\Delta - 1)^2} \tag{11.9.1}$$

This is approximately equal to the number of deaths given in Eq. 11.4.2. To see this, let $\psi = (\Delta - 1)/(\Delta + 1)$. Using the first term of a standard logarithm series expansion, we have $\log(\Delta) \approx 2\psi$. Substituting into Eq. 11.9.1 we get Eq. 11.4.2. See [54] for more details.

2. Methods for testing for non-inferiority, bio-equivalence, adaptive sample size methods, and the use of alpha spending functions for interim analyses have been adapted to use survival data. See Shih and Aisner [62] for a review of these methods.

Exercises

11.1. For the colon cancer example in Sect. 11.7, compute the probability of death and required number of patients assuming an exponential distribution with a three-year survival probability of 0.59.

11.2. Using the exponential distribution from Exercise 11.1, find the survival probabilities at 2, 3, and 4 years, and use them in the Simpson's rule method to obtain the probability of death. Compare your answers to those given in Exercise 11.1.

11.3. Consider the prostate cancer clinical trial of Example 11.1, where there was a 10 % non-compliance rate. We found that we had 73.4 % power to detect a hazard ratio of 0.75. What would the hazard ratio need to be to be detactable with 80 % power?

Chapter 12
Additional Topics

12.1 Using Piecewise Constant Hazards to Model Survival Data

The exponential distribution, with its constant hazard assumption, is too inflexible to be useful in most lifetime data applications. The piecewise exponential model, by contrast, is a generalization of the exponential which can offer considerable flexibility for modeling. In Chap. 2 (Exercise 2.5) we saw a simple piecewise exponential model with two "pieces". That is, the survival time axis was divided into two intervals, with a constant hazard on each interval. Here we show how to generalize this model to accommodate multiple intervals on which the hazard is constant. An important feature of the piecewise exponential is that the likelihood is equivalent to a Poisson likelihood. Thus, we can use a Poisson model-fitting function in R to find maximum likelihood estimates of the hazard function and of parameters of a proportional hazards model.

The connection between the piecewise exponential and Poisson models is most easily seen with a single piece, which is just an ordinary exponential distribution with rate parameter λ. The likelihood, as we have seen in Chap. 2, is as follows,

$$L_e(\lambda) = \prod_{i=1}^{n} h(t_i)^{\delta_i} S(t_i) = \prod_{i=1}^{n} \lambda^{\delta_i} e^{-\lambda t_i} = \lambda^d e^{-\lambda V} \tag{12.1.1}$$

where, as usual, t_i denotes the failure time of the ith subject, and δ_i is that subject's censoring indicator. As in Chap. 2, $d = \sum \delta_i$ denotes the number of deaths and $V = \sum t_i$ denotes the total time at risk. If time is in years, V is the number of person-years at risk, for example. As we saw in Chap. 2, the maximum likelihood estimate is given by $\hat{\lambda} = d/V$, and we may interpret this estimate as the "crude" event rate per person-year (or, more generally, per time unit).

D. Moore, *Applied Survival Analysis Using R*, Use R!,
DOI 10.1007/978-3-319-31245-3_12

Now let us suppose that the random variable d has a Poisson distribution with mean μ, and that $\mu = V\lambda$. In this context, λ is a rate parameter. Again, if time is in years, then λ would denote the death rate per year. The likelihood function for a Poisson distribution is

$$L_p(\lambda) = (\lambda V)^d e^{-\lambda V} = \lambda^d e^{-\lambda V} \cdot V^d. \tag{12.1.2}$$

Clearly the Poisson likelihood L_p is proportional to the exponential likelihood L_e, the constant multiple being V^d. Thus, the maximum likelihood estimate of one is the same as the maximum likelihood estimate of the other, specifically, $\hat{\lambda} = d/V$. We are not claiming, though, that the number of deaths, d, follows a Poisson distribution. Rather, we are pointing out that the likelihoods are proportional, so that if we treat d as having a Poisson distribution, we can use an R function for finding the maximum likelihood estimate of a Poisson distribution, and that m.l.e. will also be the m.l.e. of the exponential distribution.

In this simple example, the exponential is simple enough that no practical benefit comes from this observation of the equivalence of the two m.l.e.'s. But in the case of a piecewise exponential distribution, the equivalence is of great value. Suppose that we divide the time axis into contiguous intervals using cut points $0, c_1, c_2, \ldots, c_k$. For each subject, say i, we denote the time spent in each interval by $t_{i1}, t_{i2,\ldots,}, t_{ik'}$, where $k^{'}$ denotes the time interval in which subject i dies, or the largest time interval in which subject i is still known to be alive. We also define δ_{ij} to be 0 for each interval j in which the ith subject is known to be alive, and 1 for interval k if the subject died in that interval. Then for patient i, we have the survival time of that patient given by $t_i = \sum_{j=1}^{k} t_{ij}$ and the censoring indicator by $\delta_i = \sum_{j=1}^{k} \delta_{ij}$.

We assume a proportional hazards model

$$\lambda_i(t_i, \beta) = \lambda_0(t_i) e^{z_i \beta},$$

where now the baseline hazard is a piecewise exponential, $\lambda_0(u) = \lambda_j, j$ being the jth interval, the one in which u falls. The full likelihood is then given by

$$L_{pe}(\lambda_1, \lambda_2, \ldots, \lambda_k, \beta) = \prod_{i=1}^{n} \prod_{j=1}^{k^{'}(i)} \lambda_{ij}^{\delta_{ij}} e^{-\lambda_{ij} t_{ij}} \tag{12.1.3}$$

where $\lambda_{ij} = \lambda_j e^{z_i \beta}$, which is the proportional hazards assumption for the piecewise exponential. As with the single exponential, we may treat the censoring indicators δ_{ij} as having a Poisson distribution with mean $\lambda_{ij} t_{ij}$, and the Poisson likelihood will be proportional to the piecewise exponential likelihood, so that the maximum likelihood estimates obtained from the Poisson model with

$$\log(\lambda_{ij}) = \log(\lambda_j) + z_i \beta = \alpha_j + z_i \beta$$

will also be the m.l.e.'s for the piecewise exponential proportional hazards model. See [40] and [31] for details.

To fit the Poisson model with response variable δ_{ij}, mean $\lambda_{ij}t_{ij}$ and covariates z_i, we write

$$\log\left(\lambda_{ij}t_{ij}\right) = \log(\lambda_j) + z_i\beta + \log(t_{ij}) = \alpha_j + z_i\beta + \log(t_{ij}). \tag{12.1.4}$$

The α_j are the logs of the baseline hazard coefficients and β is the log of the hazard ratio for z_i. The constant $\log(t_{ij})$ is called an *offset* in the language of generalized linear models. We may fit such a model using the "glm" function with a "Poisson" family.

If the coefficients z_j are categorical, we may collapse over all the covariates as well as over the time intervals to obtain a more compact representation of the data set. To see this, let us first consider the case with no covariates, and sum over the patients $i = 1, \ldots, n$ rather than over the time periods. The we have $d_j = \sum_{i=1}^{n} \delta_{ij}$, which denotes the total number of deaths that occurred in interval $(c_{j-1,} c_j]$, which is open on the left and closed on the right. Also, we have $V_j = \sum_{i=1}^{n} t_{ij}$, which denotes the total amount of time (person-years, for example) that subjects spent in interval j. Then by an argument similar to that shown above, we can show that, if we treat the $d_j$as Poisson-distributed variables with mean $\lambda_j V_j$, and use the likelihood based on this Poisson assumption to get maximum likelihood estimates $\hat{\lambda}_1, \hat{\lambda}_2, \ldots, \hat{\lambda}_k$, which are also maximum likelihood estimates of the hazard parameters of the piecewise exponential distribution. If we have a binary covariate, we compute the corresponding sums for each of the two levels of the covariate, and also for the k time intervals, resulting in a 2 by k table of event counts and person years. Once the data has been put into this summary form, we may use Eq. 12.1.4 to obtain estimates of the log baseline hazard parameters $\alpha_j$and of the effect parameter β.

To clarify this discussion, let us consider a simple synthetic example given earlier in Table 4.1. This data set consists of six survival times, three receiving a control and three receiving an experimental treatment. For the sake of discussion, we shall say that these times represent numbers of weeks. In the following output, we define those six survival times, censoring indicators, and treatment indicators, and put them together in a data frame called "simple":

```
> tt <- c(6, 7, 10, 15, 19, 25)
> delta <- c(1, 0, 1, 1, 0, 1)
> trt <- c(0, 0, 1, 0, 1, 1)
> id <- 1:6
> simple <- data.frame(id, tt, delta, trt)
> simple
  id tt delta trt
1  1  6     1   0
2  2  7     0   0
3  3 10     1   1
4  4 15     1   0
5  5 19     0   1
6  6 25     1   1
```

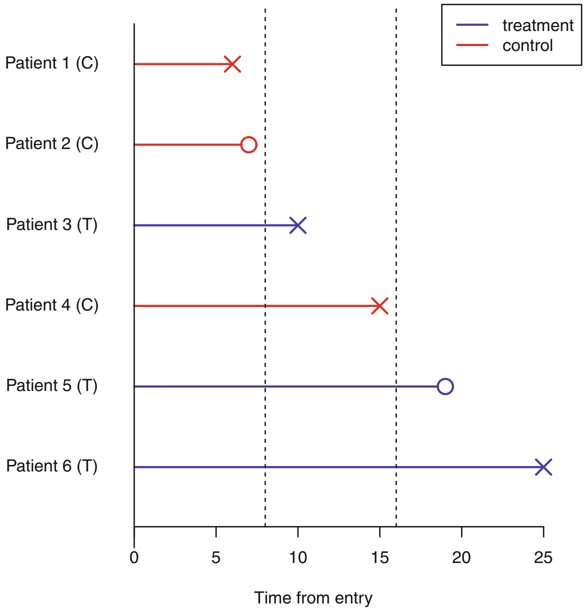


Fig. 12.1 Survival times of a synthetic data set. The intervals defined by the cut points tau are shown using *vertical dotted lines*

Next we define three intervals that we will use to define a piecewise exponential distribution:

```
tau.s <- c(0, 8, 16, 30)
```

We plot these data in Fig. 12.1.

To prepare the data to fit a piecewise exponential, we first partition the data into events and person-weeks per interval. The first patient, who died at 6 weeks, is recorded an event in the first interval, with 6 person-weeks of "experience" in that interval. The second patient was censored at 7 weeks. That patient is recorded as a non-event in the first interval, with 7 person-weeks of experience. The third patient's survival experience is divided into two portions. The first portion is a non-event in the first interval, with 8 person-weeks of exposure (8 weeks being the length of the first interval). The second portion is an event in the second interval, with $10-8=2$ person-weeks of experience in that interval. We continue in the same way with the remaining patients. Patient 4 generates two records in the final expanded data set, while Patients 5 and 6 each generate three records. In R, we may automate this procedure using the "survSplit" function (which is part of the survival package).

```
simple.split.s <- survSplit(data=simple, cut=tau.s, end="tt",
      start="t0", event="delta", episode="diagGrp")
```

In this function, "t0" is the name given to the start of the interval for the respective part of a patient's record, and "diagGrp" is the indicator for the interval that a particular record refers to. Next, we define the number of person-weeks per record:

```
simple.split.s$expo <- simple.split.s$tt - simple.split.s$t0
```

Finally, for convenience, we order the observations in "simple.split.s" to group the different parts of each patients records together in a data frame called "simple.split.ord", and examine its contents:

```
> ord <- order(simple.split.s$id)
> simple.split.ord <- simple.split.s[ord,]
> simple.split.ord
   id tt delta trt t0 diagGrp expo
7   1  6     1   0  0       1    6
8   2  7     0   0  0       1    7
9   3  8     0   1  0       1    8
15  3 10     1   1  8       2    2
10  4  8     0   0  0       1    8
16  4 15     1   0  8       2    7
11  5  8     0   1  0       1    8
17  5 16     0   1  8       2    8
23  5 19     0   1 16       3    3
12  6  8     0   1  0       1    8
18  6 16     0   1  8       2    8
24  6 25     1   1 16       3    9
```

This output confirms our previous computations for Patients 1, 2, and 3, and gives the results for the remaining patients. For example, Patient 3 (id $= 3$) consists of two records, one for interval ("diagGrp") 1, and one for interval 2. The column "expo" gives the number of person-weeks per record, which for this patient is 8 in the first interval and 2 in the second. Patient 6 (id $= 6$), to take another example, has three records, one for each of the three time intervals in which this patient spends time. There are zero events in time periods ("diagGrp") 1 and 2, and one event in time period 3. That patient spends the full interval of time in each of the first two intervals (8 weeks each, respectively), and the remaining time, $25 - 16 = 9$ weeks, in the third interval.

To obtain parameter estimates for the piecewise exponential distribution, we treat "delta" in "simple.split.ord" as if it had a Poisson distribution, the log of "expo" as the offset, and the mean as given in Eq. 12.1.4. The R code is as follows:

```
> result.simple.poisson <- glm(delta ~ -1 + factor(diagGrp)+trt +
+  offset(log(expo)), family=poisson, data=simple.split.ord)
> summary(result.simple.poisson)

Coefficients:                 Estimate Std. Error z value Pr
  (>|z|)
factor(diagGrp)1  -3.2942     1.0370  -3.177  0.00149 **
factor(diagGrp)2  -1.7463     0.8569  -2.038  0.04156 *
factor(diagGrp)3  -1.0912     1.5949  -0.684  0.49389
trt               -1.3937     1.2425  -1.122  0.26199
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

In the function call, "glm" is a "generalized linear model" with a Poisson family. The "-1" in the model definition tells R to fit a separate term for each of the three interval factors (rather than to consider the first level as a reference level). There are four output parameter estimates. The first three are estimates of α_1, α_2, and α_3, the log-transformed baseline hazard parameters, of Eq. 12.1.4. The fourth parameter estimate is an estimate of β, the treatment effect estimator. In this synthetic example, the negative estimate for "trt" would indicate that the experimental treatment is effective, but that it is not statistically significant. Remember, even though these are results from fitting a Poisson model to the extended data set in "simple.split.ord", the effect is actually of fitting a piecewise exponential distribution to the original data, "simple".

Since the covariate (treatment) is binary, we may further simplify the data set by collapsing over the three time intervals and two treatment groups. The R function "aggregate" facilities this, as follows:

```
> simple.tab <- aggregate(simple.split.ord[c("delta", "expo")],
 +         by=list(treat=simple.split.ord$trt,
 +         diagGrp=simple.split.ord$diagGrp), sum)
> simple.tab
  treat diagGrp delta expo
1     0       1     1   21
2     1       1     0   24
3     0       2     1    7
4     1       2     1   18
5     1       3     1   12
```

The first argument, "simple.split.ord[c("delta", "expo")]", selects the columns "delta" and "expo", representing the event indicator and person-weeks, respectively. The "by" argument indicates that the collapsing will be done over the two levels of "trt" and the three levels of "diagGrp". Finally, the "sum" argument indicates that we will add up all the components of "delta" and "expo" across the levels of "trt" and "diagGrp". The result is a 3-by-2 table, each cell of which gives the number of events (delta) out of the person-weeks (expo):

	trt	
diagGrp	0	1
1	1 / 21	0 / 24
2	1 / 7	1 / 18
3	1 / 12	- / -

While there are six cells, the lower right one is blank, since there are neither events nor person-years for the experimental treatment in the third time interval. As a result, the "simple.tab" data set has only five rows. We may fit the log-linear model to this compact data set as follows, and we get exactly the same parameter estimates and standard errors using the compact data set "simple.tab" as we did using the extended data set "simple.split.ord".

```
result.simple.tab.poisson <- glm(delta ~ -1 + factor(diagGrp) +
    treat + offset(log(expo)), family=poisson, data=simple.tab)
```

We may compare these results to those from fitting a Cox proportional hazards model, which we did in Sect. 5.3.3:

```
> coxph(Surv(tt, status) ~ grp)

     coef exp(coef) se(coef)     z    p
grp -1.33     0.266     1.25 -1.06 0.29
```

We see that the parameter estimate of the treatment effect, and its standard error, are similar to those resulting from the piecewise exponential model. Unlike the piecewise exponential model, the Cox model does not directly produce an estimate of the baseline hazard since, as discussed in Chap. 5, the baseline hazard cancels out of the partial likelihood.

The log-hazard estimates for the control group are the first three elements of the coefficient vector,

```
alpha0.hat <- as.numeric(result.simple.tab.poisson$coef[1:3])
```

The corresponding estimates for the treatment group are obtained by adding the constant estimate of the treatment effect, since the proportional hazards model implies that the log hazards for the two groups differ by a constant,

```
beta.hat <- result.simple.tab.poisson$coef[4]
alpha1.hat <- alpha0.hat + beta.hat
```

A plot of the piecewise exponential log hazard function estimates is given in Fig. 12.2.

To obtain the corresponding survival curve estimates, we use the "ppexp" function from the "msm" package, which must be separately downloaded and installed. The R code is as follows:

```
library(msm)
tt.vec <- (0:300)/10
piece.surv.0 <- ppexp(q=tt.vec, rate=exp(alpha0.hat),
   t=tau.s[1:3], lower.tail=F)
piece.surv.1 <- ppexp(q=tt.vec, rate=exp(alpha1.hat),
   t=tau.s[1:3], lower.tail=F)
```

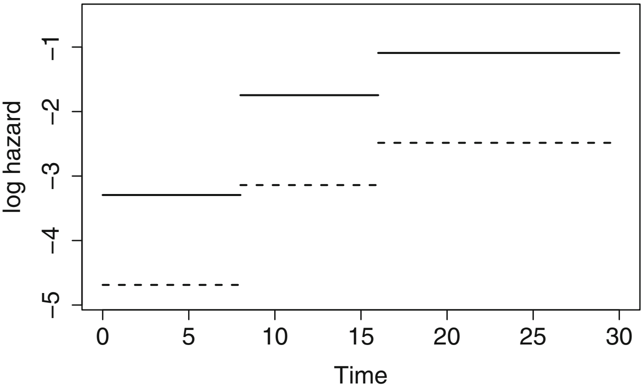


Fig. 12.2 Log hazard piecewise exponential plot for the synthetic data. The control log hazard is a *solid line*, and the treatment log hazard is a *dashed line*

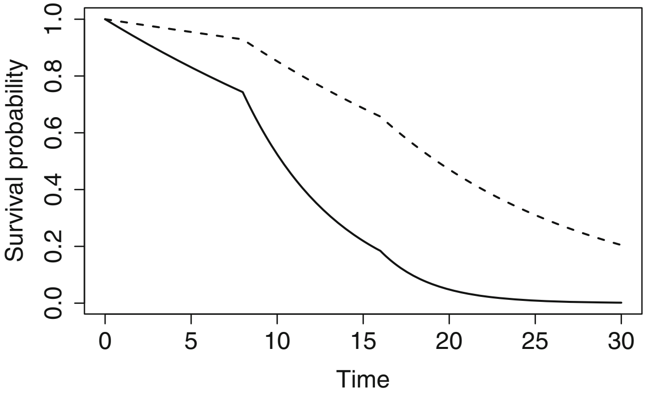


Fig. 12.3 Survival curve estimates for the piecewise exponential for the synthetic data. The control survival is a *solid line*, and the treatment survival is a *dashed line*

The "rate" argument is a vector of hazard rates, and the "t" argument is a vector of cut points. The "lower.tail=F" option causes the function to return a survival function (rather than a cumulative distribution function). The survival curves are in Fig. 12.3. Note that the survival curve estimates are continuous functions, but are not smooth at the two break points.

The piecewise exponential distribution is especially helpful when analyzing large data sets. Consider, for example, the "prostateSurvival" data, and let's select those patients with poorly differentiated prostate cancer (grade = "poor") who are 80 years old or above. What is the overall survival difference between men with stage T2 disease as compared to men with stage T1 disease? Since we are asking about overall survival (rather than prostate-specific survival), the outcome variables are "survTime" and "status". In the following code, we select the relevant subset. Also, since "status" is 1 for death from prostate cancer and 2 for death from other causes, we must define a new censoring variable "status.all" which is 1 if a patient died of any cause and 0 if alive at the last time of follow-up. There are 1640 patients in this subset.

```
> prost.80plus.poor <- prostateSurvival[{{grade == "poor"}  &
+   {ageGroup == "80+"}},]
> prost.80plus.poor$status.all <-
+    as.numeric(prost.80plus.poor$status >= 1)
> prost.80plus.poor$T2 <-
+    as.numeric(prost.80plus.poor$stage == "T2")
> prost.80plus.poor$id <- 1:nrow(prost.80plus.poor)
> head(prost.80plus.poor)
   grade stage ageGroup survTime status status.all T2 id
7   poor   T1c      80+       18      0          0  0  1
12  poor   T1c      80+       23      0          0  0  2
13  poor    T2      80+       21      0          0  1  3
14  poor  T1ab      80+       13      0          0  0  4
16  poor  T1ab      80+       30      0          0  0  5
```

```
36   poor    T1c       80+        6       0          0  0  6

> dim(prost.80plus.poor)
[1] 1640    8
```

Next, we define intervals on which the hazard is assumed to be constant (which, of course, is an approximation). Here we shall use 2-year (24 month) intervals.

```
> tau.s <- (0:5)*24
> tau.s
[1]   0  24  48  72  96 120
```

Now we use the "survSplit" procedure as before to compute events and person-months for each interval.

```
prost.split.s <- survSplit(data=prost.80plus.poor, cut=tau.s,
   end="survTime", start="t0", event="status.all", episode=
      "survGrp")
prost.split.s$expo <- prost.split.s$survTime - prost.split.s$t0
```

At this point we could fit a piecewise exponential distribution (via the Poisson likelihood of Eq. 12.1.4) using "prost.split.s". Instead, we order the records in "prost.split.s" so that subject id's are together, and then aggregate the data into a more compact form:

```
> prost.split.s <- survSplit(data=prost.80plus.poor,
+      cut=tau.s, end="survTime", start="t0",
+      event="status.all", episode="survGrp")
> prost.split.s$expo <- prost.split.s$survTime - prost.split.s$t0
> ord <- order(prost.split.s$id)
> prost.split.ord <- prost.split.s[ord,]
> prost.tab <- aggregate(prost.split.ord[c("status.all","expo")],
+        by=list(T2=prost.split.ord$T2,
+        survGrp=prost.split.ord$survGrp), sum)
> prost.tab
   T2 survGrp status.all  expo
1   0       1        145 14014
2   1       1        130 15388
3   0       2         86  7279
4   1       2        130  7429
5   0       3         66  3868
6   1       3         80  3149
7   0       4         22  1780
8   1       4         25   893
9   0       5          9   365
10  1       5          5   175
```

We see that the original data set of 1640 patients has been summarized in a compact data set "prost.tab" with only 10 rows. This compact data set allows us, if we like, to compute the crude event rates in each category. For example, from the first row, we see that the death rate for patients with Stage T1 disease (indicated by $T2 = 0$) and the first time interval (0 to 24 months) is $145/14014 = 0.0103$ events per person-month. We may also use the Poisson model to fit a piecewise exponential distribution,

```
> result.prost.tab.poisson <- glm(status.all ~ -1 +
+     factor(survGrp) + T2 + offset(log(expo)),
+     family=poisson, data=prost.tab)
> summary(result.prost.tab.poisson)

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
factor(survGrp)1 -4.77101    0.07428 -64.231   <2e-16 ***
factor(survGrp)2 -4.31653    0.07996 -53.985   <2e-16 ***
factor(survGrp)3 -3.95792    0.09094 -43.522   <2e-16 ***
factor(survGrp)4 -4.10510    0.14865 -27.615   <2e-16 ***
factor(survGrp)5 -3.71493    0.26871 -13.825   <2e-16 ***
T2                0.18128    0.07632   2.375   0.0175 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

The first five parameter estimates are the baseline (Stage T1), and the sixth is the log hazard ratio of the effect on overall survival of a patient having Stage T2 (as compared to Stage T1) disease. We see that someone with Stage T2 disease has hazard $\exp(0.18128) = 1.20$ times that of someone with Stage T1 disease. We may plot the fitted piecewise exponential survival distributions as we did previously:

```
> alpha0.hat <- as.numeric(result.prost.tab.poisson$coef[1:5])
> beta.hat <- result.prost.tab.poisson$coef[6]
> alpha1.hat <- alpha0.hat + beta.hat
> library(msm)
> tt.vec <- 0:120
> piece.surv.0 <- ppexp(q=tt.vec, rate=exp(alpha0.hat),
+   t=tau.s[1:5], lower.tail=F)
> piece.surv.1 <- ppexp(q=tt.vec, rate=exp(alpha1.hat),
+   t=tau.s[1:5], lower.tail=F)
> plot(piece.surv.0 ~ tt.vec, type="n", xlab="Time in months",
+   ylab="Survival probability")
> lines(piece.surv.0 ~ tt.vec, lwd=2)
> lines(piece.surv.1 ~ tt.vec, lwd=2, lty=2)
```

The plot is shown in Fig. 12.4. We see that, although the survival difference is statistically significant, this difference is not very large in practical terms. The probability that a patient in this group (80+ years old and poorly differentiated cancer at diagnosis) lives for 10 years (120 months) is only about 20 %, with only a slight survival advantage for those patients with T1 disease.

For comparison, let us look at the results of fitting a Cox proportional hazards model to the original data subset.

```
> summary(coxph(Surv(survTime, status.all) ~ T2,
+      data=prost.80plus.poor))

  n= 1640, number of events= 698

     coef exp(coef) se(coef)     z Pr(>|z|)
T2 0.1831    1.2009   0.0764 2.397   0.0165 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

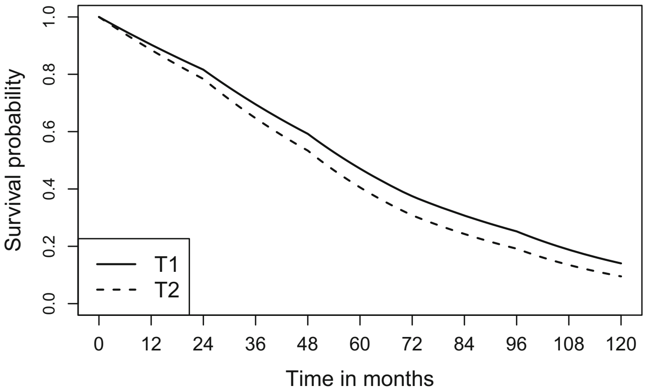


Fig. 12.4 Overall survival of men diagnosed at ages 80 and above with poorly differentiated prostate cancer. The fitted model is based on a piecewise exponential distribution for the baseline (T1), assuming proportional hazards

The parameter estimate for T2 (0.1831) is very close to the estimate given above using the piecewise exponential (0.1813). The standard errors are also similar.

The piecewise exponential model, which uses a small number of parameters to model the baseline hazard function, may be viewed as intermediate in flexibility between a parametric model and nonparametric methods such as the Kaplan-Meier survival curve estimator and the Cox partial-likelihood proportional hazards model. Choosing the number of "pieces" and the locations of the cut points is an important issue. In many practical applications, a small number of pre-specified intervals will work well. For a more formal way of selecting the number and location of the intervals, see Demarqui et al. [15]. The piecewise exponential model is also appropriate with heavily tied survival data, when the ties are a result of rounding. It is thus an alternative to the methods for handling tied survival times discussed in Sect. 5.6.

12.2 Interval Censoring

Until now the only type of censoring we have considered is right censoring. Right censoring occurs naturally in clinical trials, as we have discussed in Chap. 1 and elsewhere, and we can directly incorporate it into a survival distribution likelihood function.as But sometimes data are left- or interval-censored, and thus require specialized methods. The general likelihood function for right-censored data was presented in Eq. 2.6.1, which we now express in the following equivalent form:

$$L(\beta) = \prod_{i \in D} f(t_i) \cdot \prod_{i \in C} S(t_i) \tag{12.2.1}$$

where D represents the set of all subjects who fail and R the set of all who are right-censored. Now suppose that some of the observations are interval-censored, so that, say, patient k has an event that lies between times L_k and R_k

$$L(\beta) = \prod_{i \in D} f(t_i) \cdot \prod_{i \in C} [S(L_i) - S(R_i)] . \tag{12.2.2}$$

This more general formulation includes right censoring as a special case when $R_i = \infty$. In addition, if the event for subject i is known only to have occurred between the times L_i and R_i, that event is said to be interval-censored, and is directly incorporated into Eq. 12.2.1. Furthermore, if an event is only known to have occurred before a particular time R_i, this is known as left censoring, and may be accommodated by setting $L_i = 0$. Accommodating interval censoring with parametric survival distributions is usually straightforward, since maximization of Eq. 12.2.2 may be achieved using readily available optimization software. Non-parametric maximization of Eq. 12.2.2 is far more difficult with interval censoring since, unlike with right-censoring, there is no closed form solution which maximizes the likelihood, and there may be intervals on the time-scale where the form of the survival curve is ambiguous. A special optimization technique known as the Expectation-Maximization algorithm does allow one to find a solution, albeit at often great cost in computation time [73]. Following are two examples.

Example 12.1. Descent Times of Baboons
In this example, discussed by Ware and DeMets [75], researchers in the Amboseli Reserve in Kenya observed the descent times, in hours, of 152 baboons from their sleeping sites. Sometimes the observers arrived after the baboon had already completed its descent, in which case only an upper limit of the descent time is known; that is, the time was left-censored. The data are available in the data set "Baboons" in the R package "ssym". Here are a few cases:

```
> library(ssym)
> data(Baboons)
> Baboons[c(1,39,71,101, 150),]
            t cs
1    6.933333  0
39   8.983333  0
71   8.000000  1
101  8.983333  1
150 17.833333  1
```

Here, the variable "cs" is 0 for a time where the event is observed and 1 if it is left-censored. For example, animal 1 was observed to complete the descent in 6.93 h, whereas for animal 150, all we know is that it had completed its descent in fewer than 17.83 h. To work with the data in R, we shall need to add some extra variables. Specifically, we shall define a variable "delta" that has the more conventional coding of 0 for a censored observation and 1 for an uncensored one. Next, we define left and right intervals for every case. For uncensored observations, we define both "tt.L" and "tt.R" to equal "t"; for left-censored observations, we define "tt.L = 0.1" (a small value greater than 0)[1] and "tt.R = t". We define these variables within the "Baboons" data frame using the "within" function:

```
Baboons <- within(Baboons, {
  delta <- rep(0, length(cs))
  delta[cs == 0] <- 1
  tt.L <- t
  tt.R <- t
  tt.L[cs == 1] <- 0.1 })
```

The "Baboons" data set now looks like this:

```
> Baboons[c(1,39,71,101, 150),]
            t cs      tt.R      tt.L delta
1    6.933333  0  6.933333 6.933333     1
39   8.983333  0  8.983333 8.983333     1
71   8.000000  1  8.000000 0.100000     0
101  8.983333  1  8.983333 0.100000     0
150 17.833333  1 17.833333 0.100000     0
```

We may obtain a nonparametric estimate of the survival function using the packages "Icens" and "interval". The Icens package implements a procedure known as the "expectation-maximization" (EM) algorithm to obtain the survival curve estimate [17, 73]. The code is as follows[2]:

```
library(Icens)
library(interval)
result.icfit <- icfit(Surv(time=tt.L, time2=tt.R,
     type="interval2") ~ 1, conf.int=T, data=Baboons)
```

We may plot the estimated survival curve and 95 % confidence intervals as follows:

```
plot(result.icfit, XLAB="Time in hours",
    YLAB="Survival probability", estpar=list(col="blue", lwd=2),
    cipar=list(col="blue", lty=2))
```

For comparison, we may fit a Weibull model to the interval-censored data as follows:

```
baboon.survreg <- survreg(Surv(time=tt.L, time2=tt.R,
     type="interval2") ~ 1, dist="weibull", data=Baboons)
```

Unfortunately, we cannot directly use the results of this model in a "predict" statement, which is necessary for plotting the predicted values. To do this, we define a variable "ones" which is just a column of ones, and fit that as a covariate. Then we can obtain predicted values, as follow:

[1] The "icfit" function in the "interval" package does not require times to have values greater than zero, but the "survreg" function does.

[2] The "Icens" package is on bioconductor,http:www.bioconductor.org. In the R graphical interface window, be sure to select the drop down menu items "Packages", then "Repositories", and then include "BioC software" in addition to the default repository "CRAN".

```
ones <- rep(1,nrow(Baboons))
baboon.survreg <- survreg(
     Surv(time=tt.L, time2=tt.R, type="interval2") ~ ones,
     dist="weibull", data=Baboons)
pct <- 1:999/1000
ptime <- predict(baboon.survreg, type='quantile',
     newdata=data.frame(ones=1), p=pct, se=TRUE)
```

Finally, we may add the predicted Weibull survival curve (and 95 % confidence intervals) using the "matplot" function (which simultaneously plots the survival curve and upper and lower confidence intervals):

```
matlines(cbind(ptime$fit, ptime$fit + 2*ptime$se.fit,
        ptime$fit- 2*ptime$se.fit), 1-pct,
        xlab="Hours", ylab="Survival", type='l', lty=c(1,2,2),
        lwd=c(2,1,1), xlim=c(0,20), col="red")
```

The resulting plots are in Fig. 12.5.

Since all of the censored observations are left-censored, an alternative way to obtain the same survival curve estimate is to reverse time by negating all of the survival times. This process converts left-censoring to right-censoring, allowing us to compute a standard Kaplan-Meier survival curve in reverse time. This is the approach taken by Ware and DeMets [75]. To re-create the survival curve, we need to reverse the time on the x-axis ("xlim = c(0, −18) and plot a cumulative incidence function (ranging from 0 to 1) instead of the usual survival function ("fun = 'event' "):

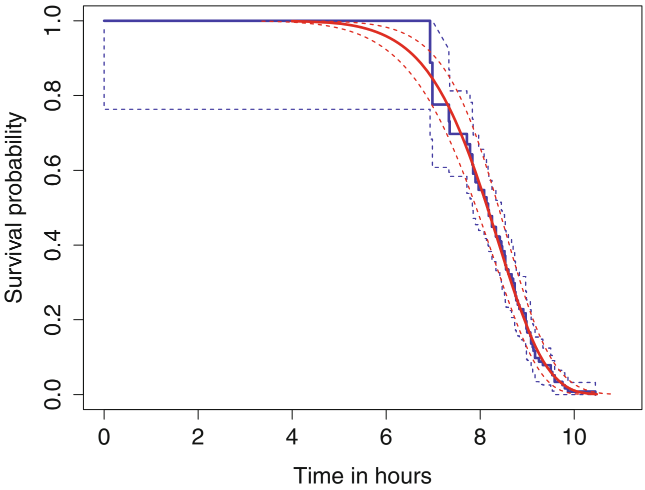


Fig. 12.5 Non-parametric (*blue*) and Weibull (*red*) survival distribution estimates of left-censored descent times of baboons. The *dashed lines* indicate 95 % pointwise confidence bands

```
result.surv.reverse <- survfit(Surv(-t, delta) ~ 1, conf.int=T,
         data=Baboons, conf.type="log-log")
plot(result.surv.reverse, xlim=c(0, -18), fun="event")
```

Example 12.2. Breast Cosmesis Study
Finkelstein [18] presented a method for fitting a proportional hazards model to interval-censored data, and illustrated it using data from a breast cosmesis study. In this study, 94 breast cancer patients treated with radiation therapy with or without adjuvant chemotherapy were followed to determine the time until cosmetic deterioration (specifically, the appearance breast retraction) of the treated breast. Since patients were evaluated at office visits separated by a number of months, the data were interval-censored. The data set, "bcos", is available in the "interval" package, and here are the first few observations:

```
> library(interval)
> data(bcos)
> bcos[c(1,33, 47, 62, 90),]
   left right treatment
1    45   Inf       Rad
33    0     5       Rad
47    8    12   RadChem
62   14    17   RadChem
90   16    60   RadChem
```

For example, patient 1 was treated with radiation alone, and had no event as of 45 months; the left end of the interval is thus 45, and the right end is infinite ("Inf"), meaning that the event (if it occurred at all) would have to have happened sometime after 45 months. Patient 47, who received radiation and adjuvant chemotherapy, had not had the event at an office visit at 8 months, but the breast retraction had been observed at the next visit four months later. Thus, the event took place sometime between 8 and 12 months. We may obtain maximum likelihood estimates of the survival distributions (assuming a proportional hazards model) and plot them as follows:

```
icout <- icfit(Surv(left,right,type="interval2")~treatment,
           data=bcos, conf.int=F)
plot(icout, XLAB="Time in months", YLAB="Survival probability",
        COL=c("lightblue", "pink"), LEGEND=F,
        estpar=list(col=c("blue", "red"), lwd=2, lty=1))
legend("bottomleft",
        legend=c("Radiation alone", "Radiation and chemo"),
        col=c("blue","red"), lwd=2)
```

We may also fit a Weibull proportional hazards model (also an accelerated failure time model) to the interval-censored data. First we must define modified left and right endpoints of the intervals, since the "survreg" R function will not accept survival times that are zero, and certainly not infinite survival time. In the latter case, we must set a maximum possible time, which depends on the experimental design. Here we choose the maximum to be 65 months:

```
> bcos <- within(bcos, {
+         left.alt <- left
+         left.alt[left == 0] <- 0.1
+         right.alt <- right
+         right.alt[is.infinite(right)] <- 65})
> bcos[c(1,33, 47, 62, 90),]
   left right treatment right.alt left.alt
1    45   Inf       Rad        65     45.0
33    0     5       Rad         5      0.1
47    8    12   RadChem        12      8.0
62   14    17   RadChem        17     14.0
90   16    60   RadChem        60     16.0
```

We may fit the Weibull model and add the fitted survival curves as follows:

```
bcos.survreg <-
  survreg(Surv(left.alt, right.alt, type="interval2") ~ treatment,
  dist="weibull", data=bcos)
pct <- 1:999/1000
ptime <- predict(bcos.survreg, type='quantile',
       newdata=data.frame(treatment=c("Rad", "RadChem")),
       p=pct, se=F)
lines(ptime[1,], 1-pct, xlab="Hours", ylab="Survival", type='l',
          lty=c(1,2,2), lwd=c(2,1,1), xlim=c(0,20), col="blue")
lines(ptime[2,], 1-pct, xlab="Hours", ylab="Survival", type='l',
          lty=c(1,2,2), lwd=c(2,1,1), xlim=c(0,20), col="red")
```

The results from both the proportional hazards (semi-parametric) model and the Weibull model are plotted in Fig. 12.6. The shaded rectangles in the figure represent gaps in the interval-censored data, where there is no information on the shape of the survival curve. The solid slanted lines, which connect the right ends of each such interval with the left ends of the next, represent only one possible shape for the curve; any monotone non-increasing connector line would also be a valid estimate. This ambiguity is a feature of interval-censored data.

12.3 The Lasso Method for Selecting Predictive Biomarkers

The primary purpose of the survival models we have discussed in this text has been to understand how covariates contribute to survival times. For example, in the "pharmacoSmoking" clinical trial, we wanted to know if the triple therapy was effective in increasing the time to relapse; in the "prostateSurvival" data set, we wanted to understand the extent to which age, stage, and grade affected prostate-specific and overall survival. In some applications, by contrast, our interest focuses on the predictive ability of a set of covariates. In studies with large numbers of genetic or other biomarker measurements, the focus may be on using those measurements to *predict* a patient's survival prospects, perhaps to aid in treatment decision making. In such studies, dozens (or even thousands) of predictors may be available. Many—or in some cases most—of these predictors may have nothing to

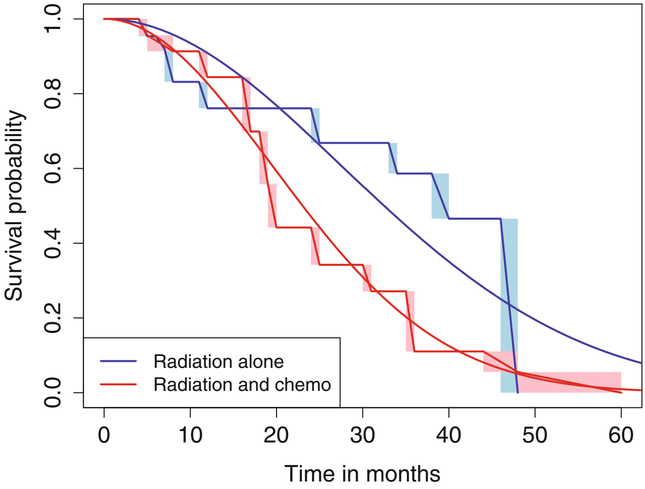


Fig. 12.6 Survival times (times to appearance of breast retractions) for breast cancer patients treated with radiation alone (*blue*) or radiation with adjuvant chemotherapy (*red*). The *smooth lines* are from a proportional hazards (also accelerated failure time) Weibull fit to the interval-censored data, whereas the step functions are non-parametric proportional hazards estimates. The *shaded areas* represent intervals on which the form of the survival curve estimate is ambiguous

with survival, and those that are associated with survival may be strongly correlated amongst themselves, complicating the prediction process. While one could use one of the model-search procedures discussed in Chap. 6, a wide range of procedures are known to be more effective when the primary aim of a study is to make survival predictions [29, 38]. An important such method is the "lasso" procedure developed originally by Tibshirani [71, 72]. Goeman [24] proposed a practical computational procedure which is implemented in the R package "penalized" [23]. This approach maximizes the partial likelihood function $l(\beta) = \log L(\beta)$ given in Eq. 5.4.1 in Sect. 5.4, but now with the additional stipulation that the L1 norm of the parameter estimates satisfies the constraint $\sum_{j=1}^{p} \left|\beta_j\right| \leq t$ for a constant t, where p is the number of parameters. This may be shown to be equivalent to maximizing the penalized likelihood which, for a pre-specified value of λ is given by

$$l_{pen}(\beta) = l(\beta) - \lambda \sum_{j=1}^{p} \left|\beta_j\right| \qquad (12.3.1)$$

Adding the constraint on the sum of the absolute values of the coefficient estimates shrinks them toward zero (as compared to the maximum partial likelihood parameter estimates without the constraint). Unlike ridge regression (a predecessor of the lasso), the lasso shrinks the estimates of the least predictive coefficients all the way to zero, effectively doing variable selection. A sufficiently large value of λ will result in no covariates at all in the model, and smaller λ values result in larger numbers of non-zero coefficient estimates. At the lower limit, where $\lambda = 0$, the penalized partial likelihood is just the ordinary partial likelihood, and all of the estimates are non-zero. A complication is that the function $l_{pen}(\beta)$ may not be strictly concave, and it may be only weakly concave or even flat at the maximum, causing convergence problems; see Goeman [24] for further discussion.

How do we select λ? Since the goal of the procedure is accurate prediction, we want to select the value of λ that maximizes the "predictive accuracy" of the procedure, where "predictive accuracy" is a measure of how well the prediction is doing. An effective and practical way to do this is through a procedure called cross validation. In one version of this procedure, we start with an initial value of λ and we randomly divide the data set into five subsets of approximately equal size. We select one of the subsets to be what we shall call the "validation" set, and combine the remaining subsets into what we shall call the "training" set. We use the training set, which comprises 80 % of the data, to construct the lasso model based on Eq. 12.3.1. We use this model to predict the survivals of patients in the validation set, which is the remaining 20 %. We use a partial-likelihood-based measure of goodness-of-fit to these data [24]. This is only the first step. We repeat this four more times, with each of the remaining four subsets in turn playing the role of the 20 % validation set and the others being the training set. The result is five sets of predictions, from which we may derive an average partial-likelihood goodness of fit.. This entire process is repeated for a range of values of λ, and we select that value that produces the optimum goodness of fit. See Goeman et al. [23] for a detailed description of the cross-validation process used by the "penalized" package.

We illustrate the use of this method using data from a study of patients with hepatocellular carcinoma on whom a range of clinical and biomarker covariates were taken [42, 43]. The data, which are in "hepatoCellular", contains 17 clinical and biomarker measurements on 227 patients, as well as overall survival and time to recurrence, both recorded in months. Of the 227 patients, 117 have levels of a variety of chemokines and other markers, some representing levels in the tumor itself and some outside the tumor. Constructing a predictive model that will be used in practice is a complex process that involves interplay between the known science about the predictors and the optimal predictive model, tethered by the realities of implementation; see for example Kuhn [38] for discussion of these issues. That said, for the purposes of illustration, we shall use 26 of these measurements (columns 23 to 48) as potential predictors of overall survival using a lasso model. We begin by selecting the 117 patients with complete data:

```
hepatoCellularNoMissing <-
    hepatoCellular[complete.cases(hepatoCellular),]
```

Here is a sample of the data and predictors we shall use in this illustration, including "OS" (overall survival), "Death" (censoring), and a few of the cytokine measurements, for patient numbers 1, 76, and 131 (which are numbers 1, 5, and 12 in the non-missing subset):

```
> hepatoCellularNoMissing[c(1,5,12),c(16,17, 23:27)]
    OS Death      CD4T     CD4N     CD8T   CD8N      CD20T
1   83     0  2.600000 0.000000 190.6000 126.80 20.950000
76  20     1 14.450000 2.758621   2.1500  38.95 26.100000
131 35     1  2.821133 8.294828   8.0064  62.64  2.821133
```

The R library "penalized", which must be separately downloaded and installed, contains a number of functions to support the fitting of lasso models. In the following example, we first attach the complete data subset "hepatoCellularNoMissing" so that "OS" and "Death" are available, and then attach the "penalized" library. Then we fit a simple lasso model using the 26 predictors (columns 23 to 48), and we fix the penalty at $\lambda = 10$. Also, since the biomarker ranges vary widely, we standardize them ("standardize = T") so that they all have a standard deviation of 1. Here are the results:

```
> attach(hepatoCellularNoMissing)
> library(penalized)
> hepato.pen <- penalized(Surv(OS, Death),
+   penalized=hepatoCellularNoMissing[,23:48],
    standardize=T, lambda1=10)
# nonzero coefficients: 7
```

The result of the model consists of seven non-zero coefficients. We may list their values using the "coef" function. For compactness, we round them to three decimal places.

```
> round(coef(hepato.pen, standardize=T), 3)
    CD8N  CD68T  CD4TR  CD8TR CD68TR   Ki67   CD34
   0.104  0.258 -0.035 -0.096  0.111  0.285 -0.013
```

The "penalized" function requires that we specify a value for λ, and we did this by rather arbitrarily selecting the value 7. As discussed earlier, we can use cross-validation to select a value that optimizes the predictive ability of the lasso model, as defined by maximizing the cross-validated partial log-likelihood (CVL). We can plot the CVL (using ten-fold cross-validation) as a function of lambda as follows:

```
set.seed(34)
hepato.prof <- profL1(Surv(OS, Death),
   penalized=hepatoCellularNoMissing[,23:48],
   standardize=T, fold=10, minlambda1=2, maxlambda1=12)
plot(hepato.prof$cvl ~ hepato.prof$lambda, type="l", log="x",
    xlab="lambda", ylab="Cross-validated log partial likelihood")
```

The purpose of "set.seed" is to set the random number seed so that we can reproduce this model fit exactly. The results, in Fig. 12.7, show a CVL with two local maxima.

To find the optimal value, we use "OptL1" with the same starting seed.

```
> set.seed(34)
> hepato.opt <- optL1(Surv(OS, Death),
+      penalized=hepatoCellularNoMissing[,23:48],  standardize=T,
    fold=10)
> hepato.opt$lambda
[1] 8.242321
> abline(v=hepato.opt$lambda, col="gray")
```

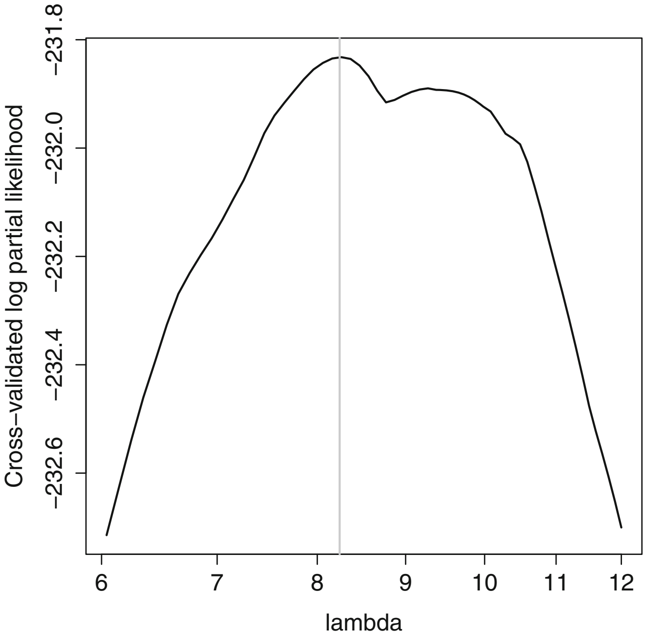


Fig. 12.7 Cross-validated log partial likelihood for a range of values of lambda for the hepatocellular data. The *vertical gray line* shows the global maximum at $\lambda = 8.24$, which was obtained using the "OptL1" function in the "penalized" package

The optimal value is 8.24, and this is added to the plot in figure using "abline". This example illustrates that it is important to plot the CVL versus lambda, since it may have local maxima, and we want to be sure that the numerical optimization procedure in "OptL1" has selected the global maximum.

The number and magnitude of the coefficient estimates depends on the value of λ. To illustrate this, we use the "penalized" function now with "steps=20". This computes the penalized CVL for a range of 20 values of λ, starting with the specified minimum ($\lambda = 5$) up to the maximum (the smallest value of λ for which there are no covariates in the lasso model). We then use the "plotpath" function to plot the coefficient profiles:

```
hepato.pen <- penalized(Surv(OS, Death),
   penalized=hepatoCellularNoMissing[,23:48], standardize=T,
   steps=20, lambda1=5)
plotpath(hepato.pen, labelsize=0.9, standardize=T, log="x",
  lwd=2)
abline(v=hepato.opt$lambda, col="gray", lwd=2)
```

The results are in Fig. 12.8.

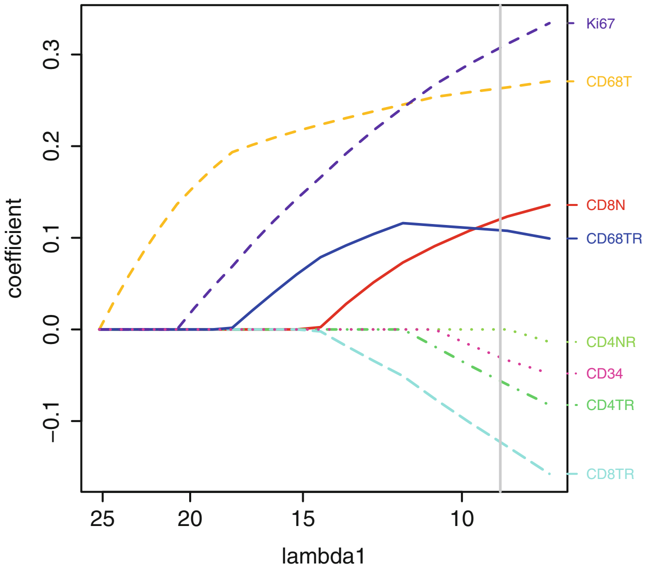


Fig. 12.8 Paths of the standardized coefficient estimates over a range of values of the lasso constraint λ. The *vertical gray line* is the optimal value of λ, the same value as in Fig. 12.7

The vertical gray line is at the optimum value $\lambda = 8.24$. The eight paths that it intersect are positive coefficients Ki67, CD68T, CD8N, and CD68TR, and negative coefficients CD4NR, CD34, CD4TR, and CD8TR. While a few of the labels are legible in the plot, we may examine them by using the "penalized" function evaluated at the optimal value we found for λ,

```
> hepato.pen <- penalized(Surv(OS, Death),
+   penalized=hepatoCellularNoMissing[,23:48],  standardize=T,
+   lambda1=hepato.opt$lambda)
# nonzero coefficients: 8
> round(coef(hepato.pen, standardize=T), 3)
  CD8N   CD68T   CD4NR  CD4TR  CD8TR  CD68TR   Ki67   CD34
  0.133  0.269 -0.009 -0.076 -0.149  0.102  0.328 -0.044
```

Unlike the partial likelihood models discussed in earlier chapters, the magnitude of these parameter estimates are not intended to be interpreted in terms of hazard ratios. For one thing, the lasso procedure has shrunken them, and for another, they are standardized to have standard deviation one. Rather, they are used to predict the survival profile for a new patient using that patient's array of biomarker measurements. For example, to show the predicted survival profiles for patients 1, 5, and 12, we use the "predict" function:

```
plot(predict(hepato.pen,
   penalized=hepatoCellularNoMissing[c(1, 5, 12),23:48]))
```

This plot function, unfortunately, does not identify which patient is which. To do this, we need to do something more complicated. First, we find the predicted models for each of these three patients in turn.

```
hepato.predict.1 <- predict(hepato.pen,
   penalized=hepatoCellularNoMissing[1,23:48])
hepato.predict.5 <- predict(hepato.pen,
   penalized=hepatoCellularNoMissing[5,23:48])
hepato.predict.12 <- predict(hepato.pen,
   penalized=hepatoCellularNoMissing[12,23:48])
```

The results of the "predict" function for the "penalized" package are of a different form than we have seen in the past. Most of our R functions produce what are called "S3" class R objects, with components that can be accessed via the "$" operator. The object "hepato.predict.1", as well as the next two, are "S4" class R objects. This means that, rather than components, they have "slots". We can see their names via the "slotNames" function:

```
> slotNames(hepato.predict.1)
[1] "time"   "curves"
```

We access these names using the "@" operator, e.g. "hepato.predict.1@time" for the survival times, and "hepato.predict.1@curves" for the values of the survival curves. To plot them, we need to convert them into step functions via the "stepfun" function. This function requires that we drop the first element of the "time" vector. The plots may be obtained as follows:

```
plot(stepfun(hepato.predict.1@time[-1], hepato.predict.1@curves),
  do.points=F, ylim=c(0,1),
  xlab="Time in months", ylab="Predicted survival probability")
plot(stepfun(hepato.predict.5@time[-1], hepato.predict.5@curves),
   do.points=F, add=T, col="blue", lwd=2)
plot(stepfun(hepato.predict.12@time[-1], hepato.predict.
  12@curves),
   do.points=F, add=T, col="red")
```

The results of the plot is given in Fig. 12.9.

The legend is added as follows:

```
legend("bottomleft", legend=c("Patient 1", "Patient 5",
   "Patient 12"), col=c("black", "blue", "red"))
```

Exercises

12.1. For the data subset discussed in Sect. 12.1, repeat the analysis using 12-month intervals instead of 24-month intervals. How do the parameter estimate for T2 and its standard error change?

12.2. Repeat Exercise 12.2, but use 1-month grouping for the piecewise exponential. Since the survival times are given to us as numbers of months, this method is a way of fitting a proportional hazards model with heavily tied survival times.

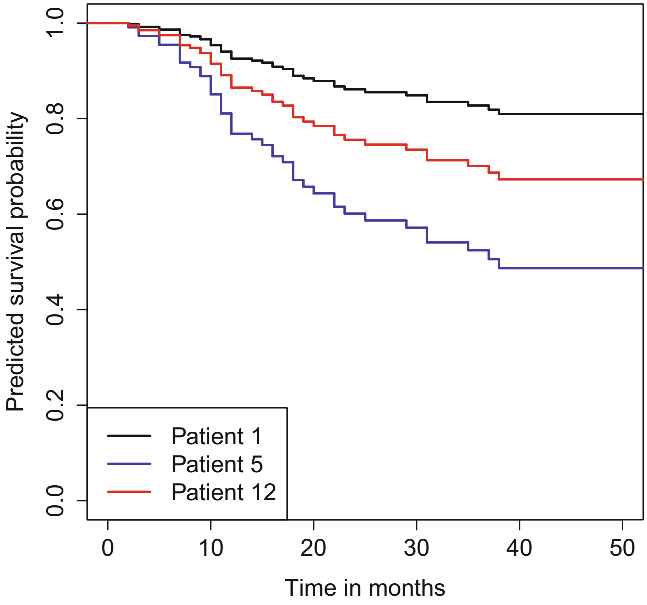


Fig. 12.9 Predicted survival curves for three patients using the lasso model "hepato.pen"

12.3. Consider the interval-censored data in the data set "cmv" in the "Icens" package. This data consists of times to shedding of cytomegalovirus (CMV) and to colonization of mycobacterium avium complex. Use the "icfit" function to estimate the survival curve for time to CMV shedding, and plot it. Identify the intervals on which the survival distribution estimate is ambiguous. Repeat for time to MAC colonization. See Betensky and Finkelstein [7].

12.4. Use the "penalized" package on the full set of predictors in the "hepatoCellular" data, including clinical predictors, following the procedure outlined in Sect. 12.3. How many of the predictors from the model with the optimal value of λ from Sect. 12.3 remain in the predictive model?

Appendix A
A Basic Guide to Using R for Survival Analysis

A.1 The R System

This first section of the appendix provides a brief but necessarily incomplete introduction to the R system. Readers with little prior exposure to R can start here, and then follow up with one of the many books or online guides to the R system. Succeeding sections cover specialized R topics relevant to using R for survival analysis.

The R statistical system, which henceforth we will refer to as just "R", is a programming language geared to doing statistical analyses. It was created by Ross Ihaka and Robert Gentleman at the University of Auckland in New Zealand and, since 1997, has been maintained by a core group of about twenty developers in diverse locations. More information about the R system and its maintenance may be found at the website http://cran.r-project.org. It is an interpretative language, meaning that it interprets and executes code as the user types it, or as it reads code from a file. It includes features for creating and manipulating variables, vectors and matrices. It can also work with the more advanced structures known as data frames, arrays and lists. It also has facilities for creating plots, and it has a special format for handling missing values, which are a common feature of data sets. Its facilities overlap with those of widely used commercial statistical software packages. Like them, it provides a wide range of statistical procedures, and includes facilities for manipulating statistical data. Unlike those packages, however, R is "open source," meaning essentially that the code is freely available and free to distribute. There are, however, certain licensing restrictions, a key one being that any code derived from existing R code must also be made freely available. The absence of any cost to downloading and installing R has led to widespread worldwide adoption of the system, and encouraged researchers who develop new statistical methods to make those methods available in R.

D. Moore, *Applied Survival Analysis Using R*, Use R!,
DOI 10.1007/978-3-319-31245-3

To install the system in Microsoft Windows, go to http://cran.r-project.org/bin/windows/base/ and follow the instructions. The Windows installer will create an icon on the desktop or the start menu. (On a 64-bit Windows installation, two icons will be created, one for a 32-bit version and one for a 64-bit version. The latter version will include "x64" in the icon name.) R is also available for Apple OS and Linux OS; see the main R site at http://cran.r-project.org for details.

To start R, click on the icon (for now, either the 32- or 64-bit version is fine) as you would with any standard Windows program. A command window will open up with a ">" prompt. The command window includes several dropdown menus for opening or creating files of R commands, for installing "packages" to provide additional functionality to the system, and for accessing the help system. While one can carry out any R procedure from the base system, more advanced users may prefer to use a separate editor for creating R programs. A free editor that works well with R is "Tinn-R", which may be downloaded from https://sourceforge.net/projects/tinn-r. Alternatively, one can use a full-fledged programming environment called Rstudio. When launched, R Studio automatically detects the most current version of R on your system, and opens that up in one of four window panes. Other panes include an editor for writing R code, a viewer that shows the names of objects you have created, and a plot viewer. Rstudio may be downloaded from http://www.rstudio.com/ide/download/desktop.

This guide provides a brief introduction to those aspects of R most useful for survival analysis. A guide to more complete treatments of the R language may be found at http://www.r-project.org/doc/bib/R-books.html.

A.1.1 A First R Session

The purpose of this session is to show how to start up R, carry out a few simple numerical operations, and then exit the system. First, start R from the start menu of Windows or from the Desktop. You will see the R Console, a window with a ">" prompt. This is the R window into which you type commands and receive responses. To get a feel for how R works, enter a numerical expression, such as this:

```
> 2 + 3
[1] 5
```

The ">" symbol is the prompt that R provides, and following that is "2 + 3", which the user types. The "[1]", which is perhaps superfluous here, just indicates that the printed result is the first (and in this simple case only) element of the result. (The purpose of the output format will become clear when you look at long vectors that stretch out over more than one line.)

Now try some other operations. Note that the "#" symbol indicates the start of a "comment", and is not interpreted by the computer.

```
> 2^3
[1] 8
> 2**3  # same thing
[1] 8

> x <- 3    # assign 3 to the variable x
> y <- 2
> y^x
[1] 8

> y**x
[1] 8

> q()   # end the R session
```

In addition to constants, R can work with vectors, as shown in the following code:

```
> x.vec <- c(1, 3.5, 7)
> y.vec <- c(2, 7, 8.6)
> x.vec
[1] 1.0 3.5 7.0
> y.vec
[1] 2.0 7.0 8.6
```

Here we have defined two vectors, "x.vec" and "y.vec", each of length 3. The ".vec" ending of the names is for the convenience of the user only; any name that consists of letters and numbers and certain separators such as "." or "_" can be use. Beginners should note that unlike in some other statistical systems, R is case sensitive; that is, upper- and lower-case letters are interpreted as distinct. Thus, for example, "X.vec" and "x.vec" are two different names.

Vectors may be added and multiplied, as long as they are the same length. Also, constants may multiply vectors.

```
> x.vec
[1] 1.0 3.5 7.0
> y.vec
[1] 2.0 7.0 8.6
> x.vec + y.vec
[1]  3.0 10.5 15.6

> z.vec <- 2*y.vec
> z.vec
[1]  4.0 14.0 17.2
```

The "c()" function may also be used to combine vectors,

```
> z.vec <- c(x.vec, y.vec)
> z.vec
[1] 1.0 3.5 7.0 2.0 7.0 8.6
```

and individual components of vectors may be accessed by index. For example, to list the first four elements of z.vec, we can use the ":" to get the indices from 1 to 4,

```
> z[1:4]
[1] 1.0 3.5 7.0 2.0
```

Vectors may also contain characters,

```
> w.vec <- c("a", "A", "aBc")
> w.vec
[1] "a"   "A"   "aBc"
```

In survival analysis, there is a special structure for right-censored survival data. To use this, one first must load the "survival" package, which is included in the main R distribution,

```
library(survival)
```

Next, define the survival times "tt" and the censoring indicator "status", where "status = 1" indicates that the time is an observed event, and "status = 0" indicates that it is censored. Then the "Surv" function binds them into a single object. In the following example, time 6 is right censored, while the others are observed event times,

```
> tt <- c(2, 5, 6, 7, 8)
> status <- c( 1, 1, 0, 1, 1)
> Surv(tt, status)
[1] 2  5  6+ 7  8
```

If you enter an R command that is syntactically incomplete, it will continue onto the next line with a "+" symbol, where you can complete the command. For example,

```
> tt <- c(2, 5, 6,
+   7, 8)
> tt
[1] 2 5 6 7 8
```

This feature is particularly convenient for long commands that will not fit on a single line.

The character "#" is used to introduce a comment; anything written after this character will be ignored by the R system. This is useful for annotating code, e.g.

```
> Surv(tt, status)  # Create a survival data structure
```

Finally, to quit the R session, use the "q()" function, with no arguments,

```
> q()
```

A.1.2 Scatterplots and Fitting Linear Regression Models

We use linear regression methods in survival analysis in a number of ways, so we introduce it along with the plot function here. To illustrate, let's create two vectors,

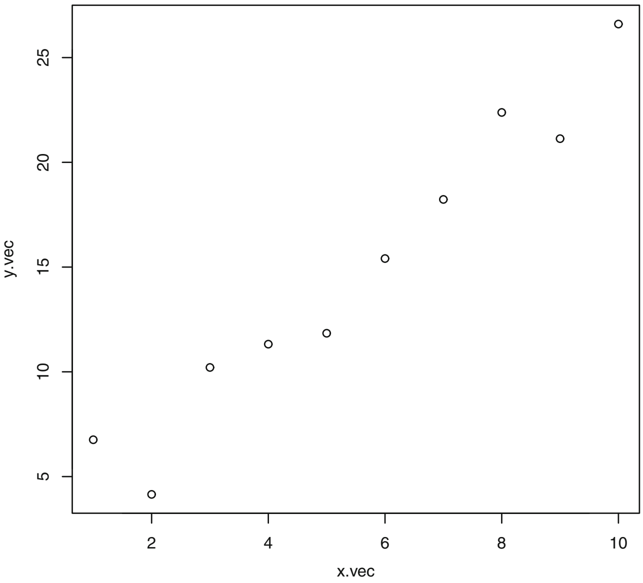
```
> x.vec <- 1:10
> x.vec
[1]  1  2  3  4  5  6  7  8  9 10
>
> y.vec <- 3 + 2*x.vec + rnorm(10, mean=0, sd=2)
> y.vec
[1]  6.758104  4.148936 10.208296 11.320387 11.840879 15.407382
    18.228854
[8] 22.381251 21.130111 26.600463
```

The vector “x.vec” was created using the “:” operator to obtain the integers from 1 to 10, and the vector “y.vec” is defined as

$$y = 3 + 2x + \varepsilon,$$

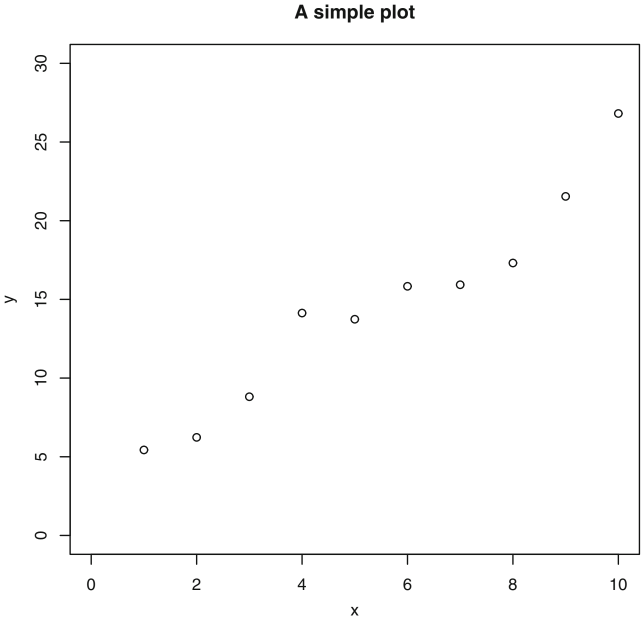
where ϵ is a standard normal random variable with mean 0 and standard deviation 2. Notice that when “y.vec” is printed, its values wrap onto the second line. the “[8]” on the second line indicates that this line begins with the 8th element of the vector. One may easily plot y.vec vs. x.vec,

```
plot(y.vec ~ x.vec)
```



The plot may be enhanced by specifying ranges for the x and y variables, and with specific labels for the axes,

```
> plot(y.vec ~ x.vec, xlim=c(0, 10), ylim=c(0, 30),
+   xlab="x", ylab="y")
> title("A simple plot")
```

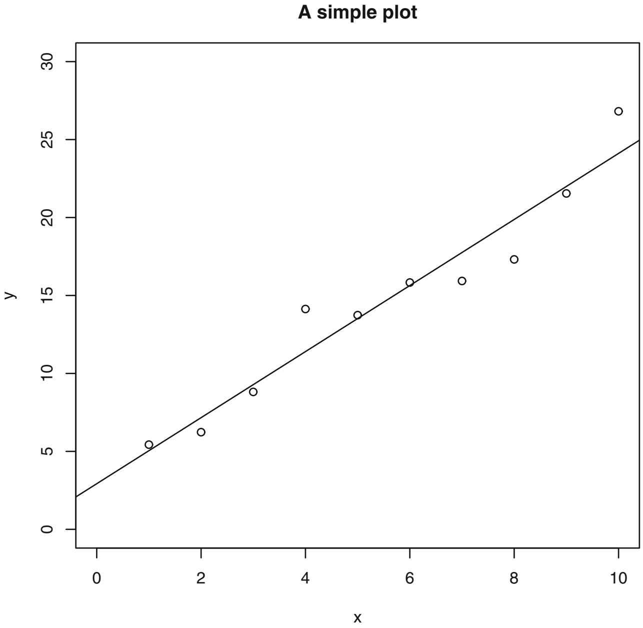


To fit a linear regression line through these points, use the "lm" function, with "y.vec" as the outcome variable, "x.vec" as the predictor variable, and "~" meaning "regressed on". In the following, the results of fitting a linear regression model are put into a "data structure" which we have chosen to call "result.lm". To print out a brief summary of the structure, just type its name,

```
> result.lm <- lm(y.vec ~ x.vec)
> result.lm
Call: lm(formula = y.vec ~ x.vec)
Coefficients:
(Intercept)       x.vec
      2.925       2.119
```

The output indicates that the fitted regression model is given by $y = 2.925 + 2.119x$. To plot this fitted line on the above scatterplot, use the "abline" function,

```
> abline(result.lm)
```



A more complete output of the linear regression, including standard errors and hypothesis tests, may be obtained by entering "summary(result.lm)".

A.1.3 Accommodating Non-linear Relationships

A non-linear relationship between y and x will require more sophisticated tools. To illustrate, let us suppose that the true relationship between y and x is given by $y = 2x^3 - 9x^2 + 5x + 6$. We may define this as an R function as follows:

```
ff <- function(x) {
  result <- 2*x^3 - 9*x^2 + 5*x + 6
  result
  }
```

This function, which we have named "ff", takes a value (or a vector of values), evaluates the defined polynomial at those values, and puts the result in an R object named "result". The last object in the function ("result") is the value that the function returns. For example, to evaluate the function at 0, 1, and 2, we can do the following:

```
> ff(x=c(0, 1, 2))
[1]  6  4 -4
```

We simulate points with this relationship, with error, by defining the relationship as a function, creating a series of x values, and then the y values, as follows:

```
> x.vec <- (-99:400)/100    # create 500 points between -1 and 4
> y.vec <- ff(x.vec) + 10*rnorm(500)   # fixed and random effects
```

We may plot these points, and the "true" functional relationship, shown as a red curve, as follows:

```
> plot(y.vec ~ x.vec, col="gray")
> curve(ff, from=-1, to=4, col="red", lwd=2, add=T)
```

A straightforward way to model such a non-linear relationship is to create quadratic and cubic forms of the x-values, and incorporate them into a linear model as follows:

```
> x2.vec <- x.vec^2
> x3.vec <- x.vec^3
> result.lm <- lm(y.vec ~ x.vec + x2.vec + x3.vec)
> summary(result.lm)

              Estimate Std. Error t value Pr(>|t|)
(Intercept)   6.9369     0.7911    8.769  < 2e-16 ***
x.vec         4.1090     0.9969    4.122  4.4e-05 ***
x2.vec       -9.1537     0.9079  -10.082  < 2e-16 ***
x3.vec        2.0945     0.1936   10.818  < 2e-16 ***
--- Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 10.23 on 496 degrees of freedom
```

We see that the coefficient estimates closely match the originals, as does the "Residual standard error", which matches $\sigma = 10$ in the original error function.

In this case, the "true" relationship was a cubic polynomial function. But if the relationship between y and x were some other function, not necessarily a polynomial one, what could we do? There is a technique called "locally weighted scatterplot smoothing", often abbreviated by "loess" for, presumably, "LOcal rESSion". This is implemented in R as the "loess" function. We may use this function as follows:

```
result.smooth <- loess(y.vec ~ x.vec)
smooth.estimates <- predict(result.smooth)
lines(smooth.estimates ~ x.vec, col="blue", lwd=2)
```

However, we shall find it helpful to plot not only the smooth function but also 95 % confidence intervals. To do this, we define a function that fits a loess curve and also 95 % confidence intervals for this curve, and plots them:

```
smoothSEcurve <- function(yy, xx) {
  # use after a call to "plot"
  # fit a lowess curve and 95% confidence interval curve

  # make list of x values
  xx.list <- min(xx) + ((0:100)/100)*(max(xx) - min(xx))

  # Then fit loess function through the points (xx, yy)
  #   at the listed values
  yy.xx <- predict(loess(yy ~ xx), se=T,
     newdata=data.frame(xx=xx.list))
```

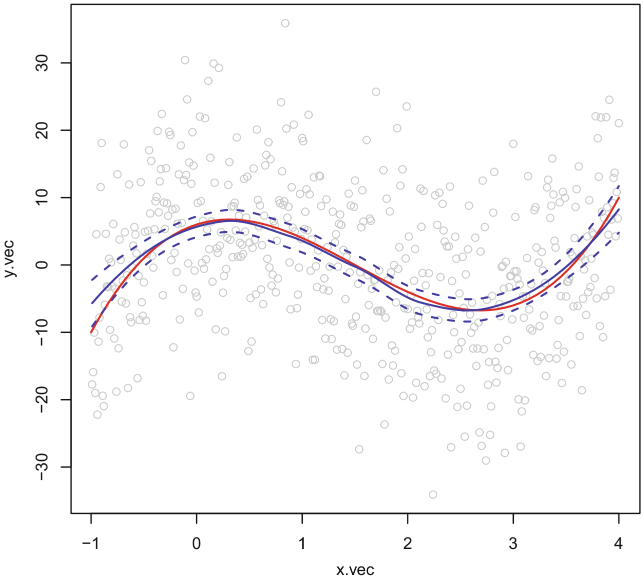


Fig. A.1 Smooth loess curve (*blue*) and true functional relationship (*red*)

```
  lines(yy.xx$fit ~ xx.list, lwd=2)
  lines(yy.xx$fit -
       qt(0.975, yy.xx$df)*yy.xx$se.fit ~ xx.list, lty=2)
  lines(yy.xx$fit +
       qt(0.975, yy.xx$df)*yy.xx$se.fit ~ xx.list, lty=2)
  }
```

We use this function to add the smooth curve and confidence limits as follows:

```
smoothSEcurve(y.vec, x.vec)
```

The plot is shown in Fig. A.1.

A.1.4 Data Frames and the Search Path for Variable Names

R provides a special data structure, the "data frame", to conveniently store variables for statistical data analysis. A data frame is a two-dimensional array with named columns. Like many R packages, the survival package includes data sets as in the form of data frames to use as examples. The data set "lung" contains survival data on 228 patients with advanced lung cancer; here is a subset of the first six rows and seven columns:

```
> lung[1:6,1:7]
  inst time status age sex ph.ecog ph.karno
1    3  306      2  74   1       1       90
2    3  455      2  68   1       0       90
3    3 1010      1  56   1       0       90
4    5  210      2  57   1       1       90
5    1  883      2  60   1       0      100
6   12 1022      1  74   1       1       50
```

The survival variables are "time" (days from enrollment until death or censoring), "status" (1 for censoring, 2 for dead), and a number of other possibly relevant covariates. A detailed description of the data set may be found by typing "?lung" at the R prompt. Individual columns of the data frame may be accessed in two ways: by column number or by column name. Here are examples of accessing the "time" column in these two ways; in each case, the first few components are shown:

```
> time.A <- lung[,2]
> time.B <- lung$time
> time.A[1:5] [1]  306  455 1010  210  883
> time.B[1:5] [1]  306  455 1010  210  883
```

Alternatively, one can "attach" the data frame, in which case all of the variable names can be accessed by name:

```
> attach(lung)
> time[1:5]
[1]  306  455 1010  210  883
```

To avoid errors in referencing variables, it is important to realize that the variable name "time" is placed on what is known as a "search path". When a user types a variable name such as "time", the R system first searches the current workspace for a definition of the variable; finding none, it then looks into any attached data frames; in this case, it is "lung" and there it finds the variable named "time". If we redefine the variable "time" in the workspace, it will take preference:

```
> time <- c(1,2,3,4)
> time
[1] 1 2 3 4
```

Now, the variable "time" has been defined in the workspace as the numbers from 1 to 4. If we remove (i.e. delete) this variable with the "rm" command, this version of the variable goes away, and the version of "time" in the attached "lung" data frame again becomes visible:

```
> rm(time)
> time[1:5]
[1]  306  455 1010  210  883
```

This example illustrates the importance of keeping track of the variable names visible in an attached data frame, and ensuring that no variables of the same name are defined in the user workspace.

A.1.5 Defining Variables Within a Data Frame

When one needs to re-define variables in a data frame, it is often helpful to carry out the necessary calculations within the data frame itself using the "within" function. For example, suppose for the "lung" data we want to define a new censoring variable "delta" which takes the values 0 for a censored variable and 1 for an event. We can do this as follows:

```
lung <- within(lung, {
     delta <- status - 1 })
```

We may see the new variable "delta" as follows:

```
> lung[1:6, c(1:7, 11)]
  inst time status age sex ph.ecog ph.karno delta
1    3  306      2  74   1       1       90     1
2    3  455      2  68   1       0       90     1
3    3 1010      1  56   1       0       90     0
4    5  210      2  57   1       1       90     1
5    1  883      2  60   1       0      100     1
6   12 1022      1  74   1       1       50     0
```

In this way, one can directly incorporate new variables into the data frame without creating new ones in the R workspace.

A.1.6 Importing and Exporting Data Frames

Data frames may be most easily imported and exported using "comma-separated" files. Such files, when saved with a ".csv" extension, will open in Windows as an Excel file by default. Such files can easily be imported into other statistical packages if needed, since the file contains no special-purpose non-printing characters. For example, suppose we need to export the "lung" data. We can export it to a directory, say, "C:\survival" as follows:

```
> setwd("c:\\survival")
> write.csv(lung, file="lung.csv", na=".", row.names=F)
```

Since R treats the backslash character "\" as an escape character (imparting special meaning to certain letters), it cannot be used alone when referring to a Windows directory. Rather, it has to be doubled to correctly reference the windows directory "C:\survival". In this example, the function "setwd" sets the "working directory" to the "C:\survival" Windows folder, assuming that this folder has been created previously outside of the R program. The command "write.csv" writes out the file in comma separated form into the file named "lung.csv". The option "na=" defines the outputted missing value code to be the specified value, here a dot, ".". Without this option, R will write out "NA" (the R missing value code) for missing values. Finally, the "row.names=F" option suppresses numbered row names, which are usually unnecessary for exported files. The first few rows of the resulting file, if viewed using a text editor, look like this:

```
"inst","time","status","age","sex","ph.ecog","ph.karno","meal.cal"
3,306,2,74,1,1,90,1175
3,455,2,68,1,0,90,1225
3,1010,1,56,1,0,90,.
5,210,2,57,1,1,90,1150
1,883,2,60,1,0,100,.
```

The first row is a list of column names, and the remaining rows contain the data. If this data set weren't already in R, and one needed to import it, one would do the following:

```
> setwd("c:\\survival")
> lung2 <- read.csv("lung.csv", na.strings=".", header=T)
> head(lung2)
  inst time status age sex ph.ecog ph.karno  meal.cal
1    3  306      2  74   1       1       90      1175
2    3  455      2  68   1       0       90      1225
3    3 1010      1  56   1       0       90        NA
4    5  210      2  57   1       1       90      1150
5    1  883      2  60   1       0      100        NA
6   12 1022      1  74   1       1       50       513
```

The option "na.strings=" tells R that, in the Windows file, the "." character indicates missing values, so that they are recognized as such during the import process, and represented using R's missing data indicator "NA". The "header=T" option tells R that the first row consists of column names.

A.2 Working with Dates in R

Often survival data come in the form of calendar dates. Typically we are given the date of entry into a trial, the date of death, and the date a patient was last seen, if still alive. We must then compute the intervening times, and determine if a final date represents a death or a censored observation. With medical data, time is measured in days, although that may be later converted to months or years for presentation purposes. The R package "date", which must be explicitly downloaded and installed, allows us to work with data in date format. To do this from the R window, click on the "Packages" tab to get a pull-down menu. Then click on "Install packages". You may be asked to select a "repository." In that case, choose your country and then a location near you. Then you will get a pop-up window that lists all available packages in alphabetical order. Highlight the "date" package and then "install". The package will then be installed on your R system, and will be available for use in this and future occasions when you use R.

A.2.1 *Dates and Leap Years*

R includes a special “date” format. When you examine a variable with dates in that format, you will see a listing that is given in “day-month-year” format. Internally, however, the date is stored as an integer that represents the number of days between the date of interest and the reference date, which is January 1, 1960. This reference date is arbitrary, but often used by computer packages; when you subtract date objects, the results will be the number of intervening days between the two dates. The date package is written to accommodate leap years, by including February 29 in calculations only when a leap year is involved. It also understands that the year 2000 was a leap year but that the year 1900 was not a leap year. This latter fact would become relevant if, for example, one works with birth dates of individuals born before 1900. (Leap years in the Gregorian calendar occur in years that are multiples of 4, except for years that are multiples of 100; a further exception is that years that are multiples of 400 *are* leap years, which is why the year 2000 was a leap year. See http://aa.usno.navy.mil/faq/docs/leap_years.php, maintained by the United States Naval Academy, for more details.)

A.2.2 *Using the “as.date” Function*

Once the date package has been installed, you may load it by typing “library(date)” at the R prompt. The “as.date” function can then be used to convert dates in character form into R dates. Here are some examples of two dates:

```
> date.1 <- as.date("8/31/1956")
> date.2 <- as.date("7/5/1957")
> date.1
[1] 31Aug56
> date.2
[1] 5Jul57
```

We may see that there are 308 days separating these two dates as follows:

```
> date.2 - date.1
[1] 308
```

We may “look inside” the dates to reveal the internally-stored number of days using the “as.numeric” function. For example,

```
> as.numeric(date.1)
[1] -1218
> as.numeric(as.date("1/1/1960"))
[1] 0
```

This shows that the first date, August 31, 1956, is 1,218 days before the reference date of January 1, 1960. Also, we see that the reference date itself is stored as 0.

We may illustrate the leap-year issue as follows:

```
> as.date("2/29/2000")
[1] 29Feb2000
> as.date("2/29/1900")
[1] <NA>
```

This shows that February 29, 2000 is a legitimate date, whereas February 29, 1900 does not exist. (The value "NA" is a missing value indicator in R.)

Dates may also be input in text format:

```
> as.date("January 30 2005")
[1] 30Jan2005
```

The default format is "month-day-year", but it is possible to specify dates in "day-month-year" format using the "order" option:

```
> as.date("30/1/2005", order="dmy")
[1] 30Jan2005
```

Dates may also be vectors. Here is a small example that illustrates what may arise in survival analysis:

```
> entry.dates <- c("9/20/2010", "9/30/2010", "11/2/2010",
                   "1/5/2011")
> death.dates <- c("5/4/2013", NA, "6/9/2013", "4/5/2012")
> lastSeen.dates <- c("5/4/2013", "8/21/2013", "6/9/2013",
                      "4/5/2012")
>
> entry <- as.date(entry.dates)
> death <- as.date(death.dates)
> lastSeen <- as.date(lastSeen.dates)
```

We have defined entry, death, and date last seen dates for four patients. The second patient was known to still be alive as of August 21, 2013, so that person's death date is denoted by the missing value "NA". We define survival and censoring times as follows:

```
> censor <- as.numeric(!is.na(death))
> censor
[1] 1 0 1 1
> survTime.temp <- death - entry
> survTime.temp
[1] 957  NA 950 456
```

We have defined the censoring variable to be 1 if a death is observed and 0 if the person is censored, i.e., still alive at the time the data are analysed. The survival times are defined for all but the second patient; we fix that up as follows:

```
> survTime <- survTime.temp
> survTime[censor == 0] <- lastSeen[censor == 0]
    - entry[censor == 0]
> survTime
[1]  957 1056  950  456
> censor
[1] 1 0 1 1
```

The variables “survTime” and “censor” are now fully-formed survival variable ready for analysis. We may combine them into a survival object using the “Surv” function in the “survival” package,

```
> library(survival)
> Surv(survTime, censor)
[1]  957  1056+  950   456
```

This form of a survival variable show that the second survival time is censored at 1056 days (the time to death, though unknown, *is* known to be larger than 1056 days), whereas the others represent numbers of days until death.

A.3 Presenting Coefficient Estimates Using Forest Plots

The results of fitting a statistical model are typically presented as a table of coefficient names, coefficient estimates, standard errors, Z values, and p-values. Consider for example the survival dataset “veteran” that is included in the survival package. This data set of lung cancer patients consists of survival variable “time”, censoring indicator “status”, and several covariates, including “trt” (treatment), which takes the values “standard” and “test”, and “celltype”, which can be either squamous, small cell, adeno, or large cell. In the output below, we re-define “treatment” as a factor with levels “standard” and “treatment” and then fit a Cox proportional hazards model with treatment and cell type as predictors. (See Chaps. 5 and 6 for a discussion of the Cox model and examples of model fitting.)

```
> library(survival)
> head(veteran)
  trt celltype time status karno diagtime age prior
1   1 squamous   72      1    60        7  69     0
2   1 squamous  411      1    70        5  64    10
3   1 squamous  228      1    60        3  38     0
4   1 squamous  126      1    60        9  63    10
5   1 squamous  118      1    70       11  65    10
6   1 squamous   10      1    20        5  49     0

> trt.f <- factor(trt, labels=c("standard", "test"))
> result <- coxph(Surv(time, status) ~ trt.f + celltype,
+   data=veteran)
> result

                   coef exp(coef) se(coef)    z       p
trt.ftest         0.198      1.22    0.197 1.00 3.1e-01
celltypesmallcell 1.096      2.99    0.272 4.02 5.7e-05
celltypeadeno     1.169      3.22    0.295 3.96 7.4e-05
celltypelarge     0.297      1.35    0.286 1.04 3.0e-01
```

The result of the Cox model is put into the data structure named "result". Typing "result" produces the parameter estimates. The coefficient "trt.ftest" is the result of comparing "test" to "standard" therapy. The next three coefficient estimates are for three cell types compared to the reference cell type, which is "squamous".

It is helpful to present these results in graphical form using a display tool called a "forest plot". This type of display was originally developed as a way to present the results of a meta-analysis, which is a type of study that summarizes the results of a large number of related studies. We adapt this display for our purposes, using the "forestplot" function in the package also named "forestplot" (which must be downloaded from CRAN and installed). We set up the parameter estimates and confidence limits as follows; we use "NA"s, empty strings, and extra spaces to control the format of the plot.

```
coef.est <- c(NA, NA, 0, 0.198, NA, NA, NA, 0, 1.096, 1.169, 0.297)
se.est <- c(NA, NA, 0, 0.197, NA, NA, NA, 0, 0.272, 0.295, 0.286)
lower <- coef.est - 1.96*se.est
upper <- coef.est + 1.96*se.est
label.factors <- matrix(c("Treatment Group", "", "  standard",
   "  test", "", "Cell Type", "", "  sqamous", "  smallcell",
   "  adeno", "  large"), ncol=1)
```

Finally, we produce the plot. We use constant box sizes, and the option "txt_gp" to control the label sizes.

```
library(forestplot)
forestplot(label.factors, coef.est,  lower=lower, upper=upper,
      boxsize=0.4, xticks=c(-0.5,0,0.5, 1, 1.5, 2),
      txt_gp=fpTxtGp(label=gpar(cex=1.5)))
```

The resulting plot is shown in Fig. A.2. We can see that the log hazard ratio for the test treatment is slightly positive, indicating a small (non-significant) deleterious

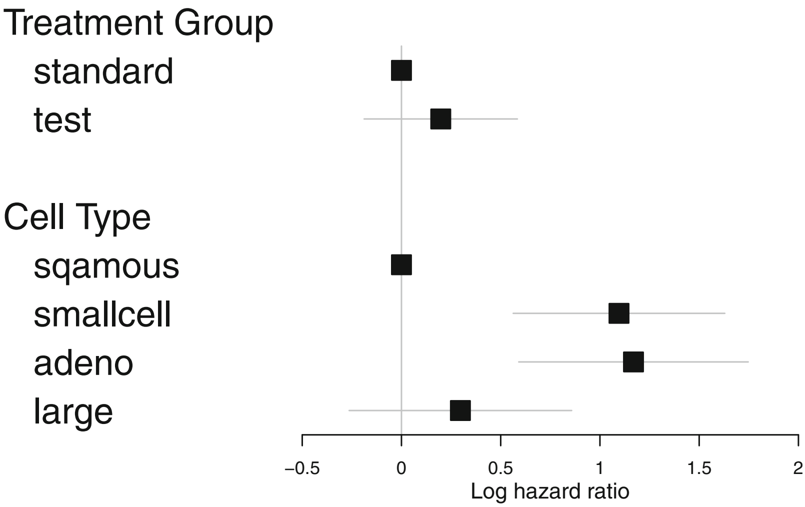


Fig. A.2 Forest plot of parameter estimates for the "veterans" dataset

effect of the test treatment as compared to the standard. We also see that the effect of small cell and adeno celltypes is similar, and larger than squamous (the reference level) and large cell. Of course, if additional covariates are included in the model, they can be included in the plot by direct extension of the code given above.

A.4 Extracting the Log Partial Likelihood and Coefficient Estimates from a coxph Object

As explained in Chap. 5, the log partial likelihood is a crucial component of a Cox model. With appropriate options, we may use the "coxph" function to evaluate the log partial likelihood at a particular value of the parameters. This specialized procedure is not necessary in ordinary data analysis, but is useful for specialized applications and for illustrating concepts, as in Sects. 5.3 and 5.4. We shall illustrate this by again using the "veteran" survival data. For simplicity, we shall define a new variable "treatInd" which is 1 for the test and 0 for the control treatments respectively. Then we fit a Cox model and examine the result:

```
> library(survival)
> attach(veteran)
> testInd <- trt - 1  # now 0 refers to standard, and 1 to test
> result <- coxph(Surv(time, status) ~ testInd)
> result
          coef exp(coef) se(coef)      z    p
testInd 0.0177      1.02    0.181 0.0982 0.92

Likelihood ratio test=0.01  on 1 df, p=0.922
```

We can explicitly evaluate the log partial likelihood at a particular value of the coefficient by specifying the initial value of the coefficient, and blocking iteration by setting the maximum number of iterations to 0:

```
> result.cox.0 <- coxph(Surv(time, status) ~ testInd,
+    init=0, control=list(iter.max=0))
> loglik.0 <- result.cox.0$loglik[2]
> loglik.0
[1] -505.4491
```

The log (partial) likelihood evaluated at $\beta = 0$ is the second element of the "loglik" component, specifically, −505.4491.

To get the log partial likelihood at the maximum, we use the m.p.l.e from the output or, to get a more precise value, we do the following:

```
> coef.mple <- as.numeric(result$coef)
> coef.mple
[1] 0.01774257
```

We may evaluate the log partial likelihood at the maximum and compute the likelihood ratio test statistic, $D = 2\left(l(\hat{\beta}) - l(0)\right)$ as follows:

```
> result.cox.max <- coxph(Surv(time, status) ~ testInd,
+     init=coef.mple, control=list(iter.max=0))
> loglik.max <- result.cox.max$loglik[2]
> 2*(loglik.max - loglik.0)
[1] 0.009643379
```

According to standard statistical theory, this is to be compared to a chi-square distibution with one degree of freedom. We evaluate this using the "pchisq" function:

```
> pchisq(0.009643379, 1, lower.tail=F)
[1] 0.9217729
```

We see that the p-value is approximately 0.92 (the same as given in the standard coxph output above), so that the treatment difference is not statistically significant.

References

1. Aalen, O.: Nonparametric inference for a family of counting processes. Ann. Stat. 701–726 (1978)
2. Agresti A.: Categorical Data Analysis, 3rd edn. Wiley, Hoboken (2012)
3. Andersen, P.K., Borgan, O., Gill, R.D., Keiding, N.: Statistical Models Based on Counting Processes, corrected edition. Springer, New York (1996)
4. Andersen, P.K., Geskus, R.B., de Witte, T., Putter, H.: Competing risks in epidemiology: possibilities and pitfalls. Int. J. Epidemiol. **41**(3), 861–870 (2012)
5. Barker, C.: The mean, median, and confidence intervals of the Kaplan-Meier survival estimate - computations and applications. Am. Stat. **63**(1), 78–80 (2009)
6. Bernstein, D., Lagakos, S.W.: Sample size and power determination for stratified clinical trials. J. Stat. Comput. Simul. **8**(1), 65–73 (1978)
7. Betensky, R.A., Finkelstein, D.M.: A non-parametric maximum likelihood estimator for bivariate interval censored data. Stat. Med. **18**(22), 3089–3100 (1999)
8. Chatterjee, N., Wacholder, S.: A marginal likelihood approach for estimating penetrance from kin-cohort designs. Biometrics **57**(1), 245–252 (2001)
9. Clark, T.G., Bradburn, M.J., Love, S.B., Altman, D.G.: Survival analysis part I: Basic concepts and first analyses. Br. J. Cancer **89**(2), 232–238 (2003)
10. Collett, D.: Modelling Survival Data in Medical Research, 3rd edn. Chapman and Hall/CRC, Boca Raton (2014)
11. Cox, D.R., Oakes, D.: Analysis of Survival Data. Chapman and Hall/CRC, London; New York (1984)
12. Cox, D.R.: Regression models and life-tables. J. R. Stat. Soc. Ser. B Methodol. 187–220 (1972)
13. De Boor, C.: A Practical Guide to Splines. Revised edition (1994)
14. de Wreede, L.C., Fiocco, M., Putter, H., et al.: mstate: an R package for the analysis of competing risks and multi-state models. J. Stat. Softw. **38**(7), 1–30 (2011)
15. Demarqui, F.N., Loschi, R.H., Colosimo, E.A.: Estimating the grid of time-points for the piecewise exponential model. Lifetime Data Anal. **14**(3), 333–356 (2008)
16. Epstein, B., Sobel, M.: Life testing. J. Am. Stat. Assoc. **48**(263), 486–502 (1953)

17. Fay, M.P.: Comparing several score tests for interval censored data. Stat. Med. **18**(3), 273–285 (1999)
18. Finkelstein, D.M.: A proportional hazards model for interval-censored failure time data. Biometrics **42**(4), 845–854 (1986)
19. Fleming, T.R., Harrington, D.P.: Counting Processes and Survival Analysis. Wiley, Hoboken (2011)
20. Freedman, L.S.: Tables of the number of patients required in clinical trials using the logrank test. Stat. Med. **1**(2), 121–129 (1982)
21. Gail, M.H.: Does cardiac transplantation prolong life? A reassessment. Ann. Intern. Med. **76**(5), 815–817 (1972)
22. Gasser, T., Muller, H.-G.: Kernel estimation of regression functions. In: Gasser, T., Rosenblatt, M. (eds.) Smoothing Techniques for Curve Estimation, vol. 757 in Lecture Notes in Mathematics, pp. 23–68. Springer, Berlin, Heidelberg (1979)
23. Goeman, J., Meijer, R., Chaturvedi, N.: L1 and L2 penalized regression models, R package Version 0.9-45, http://cran.r-project.org (2014)
24. Goeman, J.J.: L1 penalized estimation in the Cox proportional hazards model. Biom. J. **52**(1), 70–84 (2010)
25. Grambsch, P.M., Therneau, T.M.: Proportional hazards tests and diagnostics based on weighted residuals. Biometrika **81**(3), 515–526 (1994)
26. Gray, R.J.: Flexible methods for analyzing survival data using splines, with applications to breast cancer prognosis. J. Am. Stat. Assoc. **87**(420), 942–951 (1992)
27. Hall, W.J., Wellner, J.A.: Confidence bands for a survival curve from censored data. Biometrika **67**(1), 133–143 (1980)
28. Harrell, F.E.: Regression Modeling Strategies: With Applications to Linear Models, Logistic Regression, and Survival Analysis, 2nd edn. Springer Science & Business Media, New York (2015)
29. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd edn. Springer, New York (2009)
30. Hess, K.R., Serachitopol, D.M., Brown, B.W.: Hazard function estimators: a simulation study. Stat. Med. **18**(22), 3075–3088 (1999)
31. Holford, T.R.: The analysis of rates and of survivorship using log-linear models. Biometrics **36**, 299–305 (1980)
32. Hosmer, D.W., Jr., Lemeshow, S., May, S.: Applied Survival Analysis: Regression Modeling of Time to Event Data. Wiley, Hoboken (2008)
33. Huster, W.J., Brookmeyer, R., Self, S.G.: Modelling paired survival data with covariates. Biometrics **45**, 145–156 (1989)
34. Kalbfleisch, J.D., Prentice, R.L.: The Statistical Analysis of Failure Time Data, 2nd edn. Wiley, Hoboken (2002)
35. Kaplan, E.L., Meier, P.: Nonparametric estimation from incomplete observations. J. Am. Stat. Assoc. **53**(282), 457–481 (1958)
36. Klein, J.P., Moeschberger, M.L.: Survival Analysis: Techniques for Censored and Truncated Data, 2nd edn. Springer, New York (2005)
37. Kleinbaum, D.G., Klein, M.: Survival Analysis: A Self-Learning Text, 3rd edn. Springer, New York (2011)
38. Kuhn, M., Johnson, K.: Applied Predictive Modeling. Springer, New York (2013)
39. Lagakos, S.W., Barraj, L.M., De Gruttola, V.: Nonparametric analysis of truncated survival data, with application to aids. Biometrika **75**(3), 515–523 (1988)
40. Laird, N., Olivier, D.: Covariance analysis of censored survival data using log-linear analysis techniques. J. Am. Stat. Assoc. **76**(374), 231–240 (1981)
41. Lee, E.W., Wei, L.J., Amato, D.A., Leurgans, S.: Cox-type regression analysis for large numbers of small groups of correlated failure time observations. In: Klein, J.P., Goel, P. (eds.) Survival Analysis: State of the Art, pp. 237–247. Springer, New York (1992)

42. Li, L., Yan, J., Xu, J., Liu, C.-Q., Zhen, Z.-J., Chen, H.-W., Ji, Y., Wu, Z.-P., Hu, J.-Y., Zheng, L., et al.: CXCL17 expression predicts poor prognosis and correlates with adverse immune infiltration in hepatocellular carcinoma. PloS One **9**(10), e110064 (2014)
43. Li, L., Yan, J., Xu, J., Liu, C.-Q., Zhen, Z.-J., Chen, H.-W., Ji, Y., Wu, Z.-P., Hu, J.-Y., Zheng, L., et al.: Data from: CXCL17 expression predicts poor prognosis and correlates with adverse immune infiltration in hepatocellular carcidata. Dryad Digital Repository, http://datadryad.org (2014)
44. Lin, D.Y.: Cox regression analysis of multivariate failure time data: the marginal approach. Stat. Med. **13**(21), 2233–2247 (1994)
45. Lin, D.Y., Wei, L.J.: The robust inference for the cox proportional hazards model. J. Am. Stat. Assoc. **84**(408), 1074–1078 (1989)
46. Lu-Yao, G.L., Albertsen, P.C., Moore, D.F., et al.: Outcomes of localized prostate cancer following conservative management. J. Am. Med. Assoc. **302**(11), 1202–1209 (2009)
47. Matthews, D.: Exact nonparametric confidence bands for the survivor function. Int. J. Biostat. **9**(2), 185–204 (2013)
48. McCullagh, P., Nelder, J.A., McCullagh, P.: Generalized Linear Models, Vol. 2. Chapman and Hall London, (1989)
49. Moore, D.F., Chatterjee, N., Pee, D., Gail, M.H.: Pseudo-likelihood estimates of the cumulative risk of an autosomal dominant disease from a kin-cohort study. Genet. Epidemiol. **20**(2), 210–227 (2001)
50. Morse, M.A., Niedzwiecki, D., Marshall, J.L., Garrett, C., Chang, D.Z., Aklilu, M., Crocenzi, T.S., Cole, D.J., Dessureault, S., Hobeika, A.C., et al.: A randomized Phase II study of immunization with dendritic cells modified with poxvectors encoding CEA and MUC1 compared with the same poxvectors plus GM-CSF for resected metastatic colorectal cancer. Ann. Surg. **258**(6) (2013)
51. Moss, R.A., Moore, D., Mulcahy, M.F., Nahum, K., Saraiya, B., Eddy, S., Kleber, M., Poplin, E.A.: A multi-institutional phase 2 study of imatinib mesylate and gemcitabine for first-line treatment of advanced pancreatic cancer. Gastrointest. Cancer Res. **5**(3), 77–83 (2012)
52. Muller, H.-G., Wang, J.-L.: Hazard rate estimation under random censoring with varying kernels and bandwidths. Biometrics **50**(1), 61–76 (1994)
53. Narula, S.C., Li, F.S.: Sample size calculations in exponential life testing. Technometrics **17**(2), 229–231 (1975)
54. Piantadosi, S.: Clinical Trials: A Methodologic Perspective. Wiley, Hoboken (2013)
55. Preston, S.H., Heuveline, P., Guillot, M.: Demography: Measuring and Modeling Population Processes. Blackwell Malden, MA (2000)
56. Putter, H., Fiocco, M., Geskus, R.B.: Tutorial in biostatistics: competing risks and multi-state models. Stat. Med. **26**, 2389–2430 (2007)
57. Ross, E.A., Moore, D.: Modeling clustered, discrete, or grouped time survival data with covariates. Biometrics **55**(3), 813–819 (1999)
58. Rubinstein, L.V., Gail, M.H., Santner, T.J.: Planning the duration of a comparative clinical trial with loss to follow-up and a period of continued observation. J. Chronic Dis. **34**(9-10), 469–479 (1981)
59. Schemper, M., Smith, T.L.: A note on quantifying follow-up in studies of failure time. Control. Clin. Trials **17**(4), 343–346 (1996)
60. Schoenfeld, D.A.: The asymptotic properties of nonparametric tests for comparing survival distributions. Biometrika **68**, 316–319 (1981)
61. Schoenfeld, D.A.: Sample-size formula for the proportional-hazards regression model. Biometrics **39**(2), 499–503 (1983)
62. Shih, W.J., Aisner, J.: Statistical Design and Analysis of Clinical Trials: Principles and Methods. Chapman & Hall/CRC, (2016)
63. Steinberg, M.B., Greenhaus, S., Schmelzer, A.C., Bover, M.T., Foulds, J., Hoover, D.R., Carson, J.L.: Triple-combination pharmacotherapy for medically ill smokers: A randomized trial. Ann. Intern. Med. **150**(7), 447–454 (2009)

64. Struewing, J.P., Hartge, P., Wacholder, S., Baker, S.M., Berlin, M., McAdams, M., Timmerman, M.M., Brody, L.C., Tucker, M.A.: The risk of cancer associated with specific mutations of BRCA1 and BRCA2 among Ashkenazi Jews. N. Engl. J. Med. **336**(20), 1401–1408 (1997)
65. Tableman, M., Kim, J.S.: Survival Analysis Using S: Analysis of Time-to-Event Data. Chapman and Hall/CRC press, (2004)
66. Therneau, P.M., Grambsch, T.M., Shane Pankratz, V.: Penalized survival models and frailty. J. Comput. Graph. Stat. **12**(1), 156–175 (2003)
67. Therneau, T., Crowson, C.: Using time dependent covariates and time dependent coefficients in the cox model. Vignette for R survival package, http://cran.r-project.org/web/packages/survival, July 2015
68. Therneau, T.M., Grambsch, P.M.: Modeling Survival Data: Extending the Cox Model. Springer, New York (2000)
69. Therneau, T.M., Grambsch, P.M., Fleming, T.R.: Martingale-based residuals for survival models. Biometrika **77**(1), 147–160 (1990)
70. Therneau, T.M., Offord, J.: Expected survival based on hazard rates (update). Technical Report 63, Mayo Clinic Department of Health Science Research (1999)
71. Tibshirani, R.: Regression Shrinkage and Selection via the Lasso. J. R. Stat. Soc. Ser. B Methodol. **58**, 267–288 (1996)
72. Tibshirani, R.: The lasso method for variable selection in the Cox model. Stat. Med. **16**, 385–395 (1997)
73. Turnbull, B.W.: The empirical distribution function with arbitrarily grouped, censored and truncated data. J. R. Stat. Soc. Ser. B **38**, 290–295 (1976)
74. Wang, Y., Yu, Y.-y., Li, W., Feng, Y., Hou, J., Ji, Y., Sun, Y.-h., Shen, K.-t., Shen, Z.-b., Qin, X.-y., Liu, T.-s.: A phase II trial of xeloda and oxaliplatin (XELOX) neo-adjuvant chemotherapy followed by surgery for advanced gastric cancer patients with para-aortic lymph node metastasis. Cancer Chemother. Pharmacol. **73**(6), 1155–1161 (2014)
75. Ware, J.H., Demets, D.L.: Reanalysis of some baboon descent data. Biometrics 459–463 (1976)
76. Wei, L.J., Lin, D.Y., Weissfeld, L.: Regression analysis of multivariate incomplete failure time data by modeling marginal distributions. J. Am. Stat. Assoc. **84**(408), 1065–1073 (1989)
77. Weisberg, S.: Applied Linear Regression, 4th edn. Wiley, Hoboken (2014)

Index

D. Moore, *Applied Survival Analysis Using R*, Use R!,
DOI 10.1007/978-3-319-31245-3

R Package Index

D. Moore, *Applied Survival Analysis Using R*, Use R!,
DOI 10.1007/978-3-319-31245-3