Probation review report

Alessandro Gasparini

2017-08-10

Contents

Introduction

This report presents the work I have done during my first year as a PhD student at the Department of Health Sciences, University of Leicester, under the supervision of Dr. Michael Crowther and Prof. Keith Abrams.

I will begin by briefly introducing the topic of survival analysis in Chapter 1. Second, I will introduce survival models with random effects (e.g. frailties, in the simplest form) and joint models for longitudinal and time-to-event data in Chapters 2 and 3, respectively. Computational challenges that survival models with random effects and joint models pose are presented in Chapter 4. In Chapter 5, I will present a method for simulating survival data. I will then present the results of two simulation studies in Chapters 6 and 7; the first simulation study investigates the accuracy of quadrature methods when approximating analytically intractable terms, while the second simulation study investigates the impact of model misspecification in survival models with shared frailty terms. I will introduce an interactive tool I have been developing to aid the dissemination of results from simulation studies and motivated by the simulation studies of Chapter 6 and 7 in Chapter 8. Next, I will introduce the problem of informative visiting process in clinical research using healthcare consumption data in Chapter 9, and how I aim to evaluate and compare the different approaches that have been proposed and utilised in literature to tackle such problem in Chapter 10. Finally, I will briefly summarise the training and personal development activities I have participated to during the first year of my PhD in Chapter 11.

Chapter 1

Introduction to survival analysis

Survival analysis is a branch of statistics in which the main outcome consists in the time until the occurrence of a given event. Time could be years, months, weeks, or any amount of calendar time or even age time; event could be death, disease occurrence or relapse, or any other experience of interest. Survival analysis is also known as reliability theory in engineering, duration analysis in economics, and event history analysis in sociology. A broad overview of survival analysis is given in Kalbfleisch and Prentice (2011) and in Kleinbaum and Klein (2012).

Some examples of time to event data are:

- disease remission in leukemia patients. In this study, leukemia patients are followed over several weeks to study how long they stay in remission status;
- heart disease occurrence. In this study, healthy subjects are followed over several years until occurrence of heart disease, or end of the study;
- renal failure. In this study, individuals with kidney disease are followed until renal failure, or end of the study;
- reliability of complex technical installations. For instance, studies assessing failure

rates of components such as bulbs and valves.

In this Chapter I will define survival data and its peculiarities in Section 1.1 and 1.2. Terminology and notation used throughout this report will be introduced in Section 1.3. I will introduce common non-parametric and parametric methods in survival analysis in Sections 1.4 and 1.5. I will introduce the widely used semi-parametric Cox model in Section 1.6. Finally, I will provide a brief overview on advances in survival analysis in Section 1.7.

1.1 Survival data

Survival data generally consists - as previously mentioned - in an event of interest and time until its occurrence. In the leukemia remission example, time to event would be how many weeks it takes before a given patient experiences disease relapse and the event would be whether the individual relapsed or not before the end of the study. Nevertheless, in certain situations we may have some information about the survival time but the actual survival time may be unknown. This problem is know as censoring and it is presented in Section 1.2.

1.2 Censoring

Censoring is a mechanisms that causes survival times to be unobserved. There are many reasons why censoring may occur; among others:

1. a person does not experience the event before the end of the study;
2. a person drops out of the study before the occurrence of the event of interest;
3. a person experience a competing event that impedes the occurrence of the event of interest (e.g.: death, when death is not the study outcome).

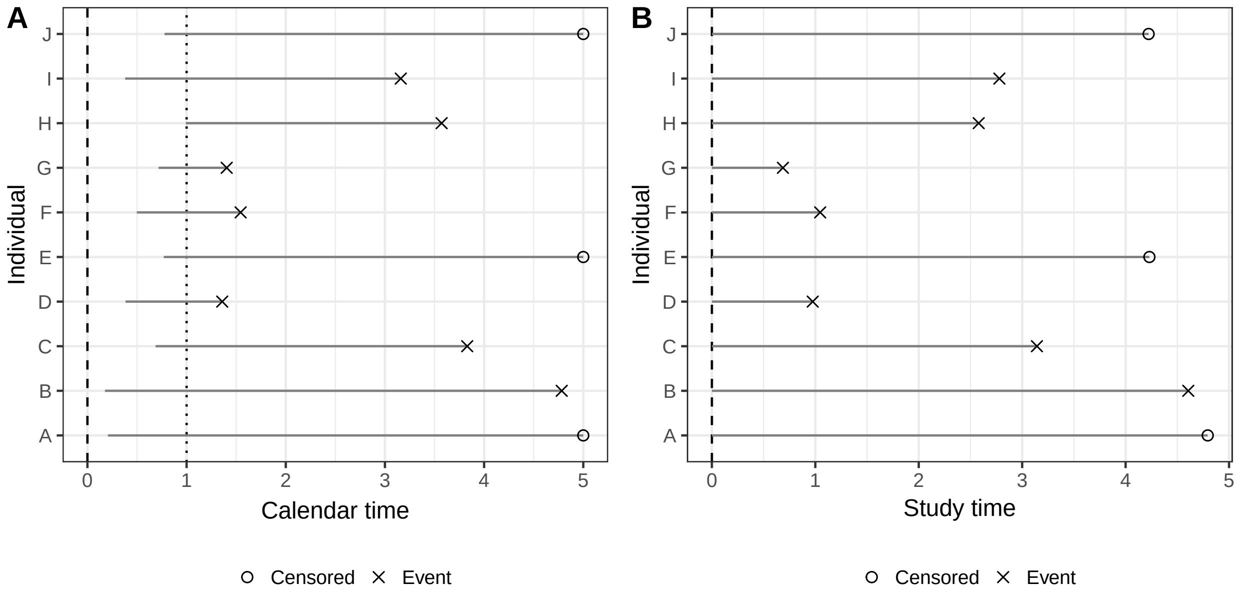

Figure 1.1: Simulated right censored survival data, plotted by their calendar time in panel A and by their study time in panel B.

I simulated survival data for illustration purposes: I assumed a clinical trial with 10 individuals enrolled during a recruitment window of 1 year, and followed for up to 5 years. Not all individuals experience the event of interest during the study period, and are therefore censored after five years from the start of the study. The observation time for each individual is depicted in Figure 1.1 with a solid dark grey line, a cross represents the occurrence of the study event, and a circle represents censoring. Individuals `A`, `E`, and `J` all have censored survival time: I know that they were still event-free at the end of follow-up, i.e. their real survival time is greater than the observed one, but the former is unknown. The simulated data is presented in Figure 1.1: in panel A, survival data is plotted against the calendar time; conversely, in panel B, survival data is plotted against the study time, e.g. each individual is assigned a *time zero* corresponding to their enrollment in the study, and survival time is counted from there.

This example represents a particular form of censoring: *right censoring*. The defining characteristic of right-censored data is that it is censored (or incomplete) at the right side of the follow-up time, hence the true survival time is greater than the observed time. This example represents *administrative censoring* as well, as individuals are censored at the end of the study to artificially restrict follow-up time (e.g. for financial reasons).

It is also possible to encounter data that is *left censored* or *interval censored*. In the former case, the true survival time is shorter that the observed one, e.g. I know that the event occurred before the observation time, but I do not know when - imagine onset of a viral infection, which can be detected only at a visit time. In the latter, I know that the event occurred within a certain interval of time but I do not know when; using the same example of infection onset, if infection was detected at a visit date but the individual was known to be infection-free at the previous visit, the true infection onset time is unknown and the event time is said to be interval censored.

Finally, another important concept related to right censoring is that of *left truncation* (or *delayed entry*). Left truncation occurs when an individual enrolls in the study some time after the inclusion criteria are satisfied; individuals that die (or emigrate, ...) before the start of observation time will never enter the study, and inclusion time may differ between individuals. Data arising from such phenomenon is therefore said to be left truncated.

1.3 Terminology and notation

I denote the random variable for an individual's survival time with S; since it denotes time, S can assume any non-negative value. The lower-case s represent a specific value of interest drawn from S for a given individual. In the case of right censoring, I denote with C the random variable representing censoring time, and c its realisation. The observed time is denoted with $T = \min(S, C)$, and its realisation is t. Finally, I denote with $D = I(S \leq C)$ the random variable indicating either occurrence of the event of interest or censorship; analogously as before, its realisation is lower-case d.

Next, I defined two quantities of interest in survival analysis, the *survival function* and the *hazard function*. They are both functions of the observed time t and are denoted by $S(t)$ and $h(t)$, respectively.

The survival function is the complement of the cumulative distribution function of the observed time T and represent the probability that a given individual survives[1] longer than a specified time t:

$$S(t) = 1 - F_T(t) = 1 - P(T \leq t) = P(T > t)$$

t ranges (theoretically) between 0 and infinity, hence the survival function can be plotted as a smooth, continuous function that tends to 0 as t goes to infinity. In practice, though, the survival function appears as a step function as (1) individuals can be observed at discrete times only and (2) not all individuals may experience the event before the end of the study. Figure 1.2 depicts this difference: in panel A I plotted a theoretical survival function, restricted to 15 years of follow-up for comparison purposes, while in panel B I plotted the survival function relative to the survival data simulated in Section 1.2. The former is a smooth function of time, and should we extend the x-axis to infinity the function would eventually reach zero. Conversely, the latter is a step function with steps at each event time, and should we extend the x-axis to infinity the function would remain flat after the last observed event.

The hazard function $h(t)$ is the limit of the probability of the survival time T laying within an interval $[t, t + \Delta(t))$ given that an individual survived up to time t divided by the length of the interval $\Delta(t)$, for $\Delta(t)$ approaching zero:

$$h(t) = \lim_{\Delta(t) \to 0} \frac{P(t \leq T < t + \Delta(t) | T \geq t)}{\Delta(t)}$$

It represent the instantaneous potential (e.g. risk) for the event to occur within the interval

[1] I use the term *survives* loosely speaking, for conciseness - formally, I refer to *not experiencing the event of interest*.

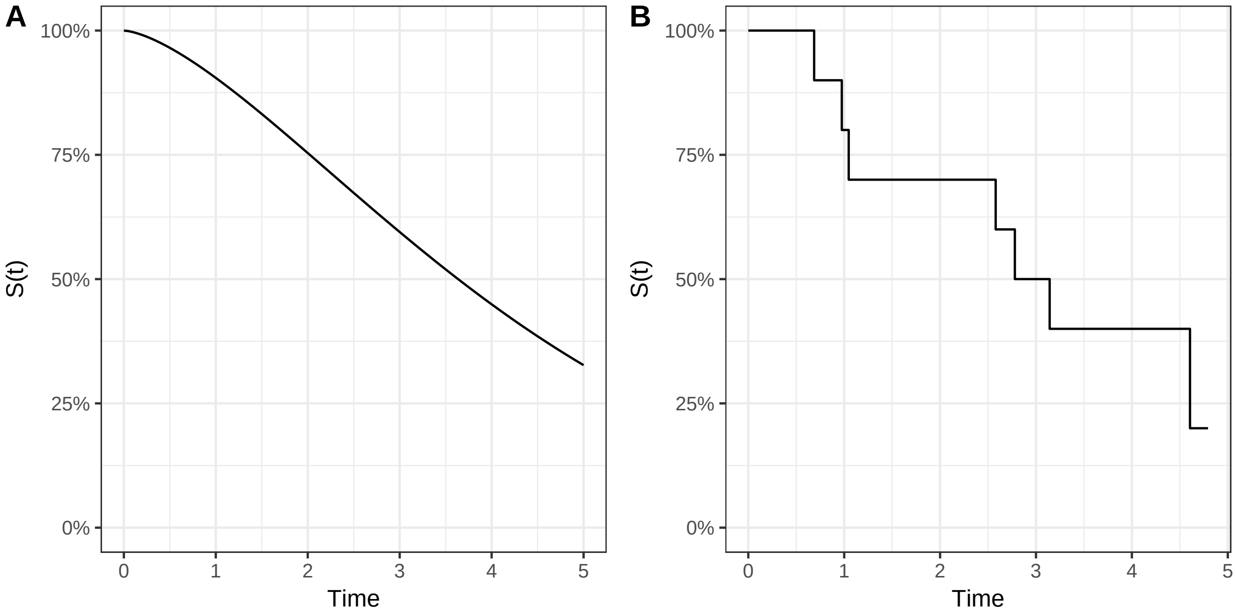

Figure 1.2: Theoretical survival function (A) and observed survival function for simulated data (B).

$[t, t + \Delta(t))$ (with $\Delta(t) \rightarrow 0$), given that the individual survived up to time t. The hazard function is always non-negative, it can assume different shapes over time, and it has no upper bound. In Figure 1.3 I present a simple hazard function; it increases over time, which means that the instantaneous risk of event increases over time.

The survival function from Figure 1.2, panel A, and the hazard function from Figure 1.3 are strictly related. In fact, there is a clearly defined mathematical relationship between the survival and the hazard function: it is possible to derive the form of $S(t)$ when knowing the form of $h(t)$, and vice versa. Formally:

$$S(t) = \exp\left[-\int_0^t h(u)\ du\right]$$

$$h(t) = -\left[\frac{dS(t)/dt}{S(t)}\right]$$

Finally, a third quantity of interest in survival analysis that is strictly related to the survival and hazard functions is the cumulative hazard function $H(t)$. The cumulative hazard

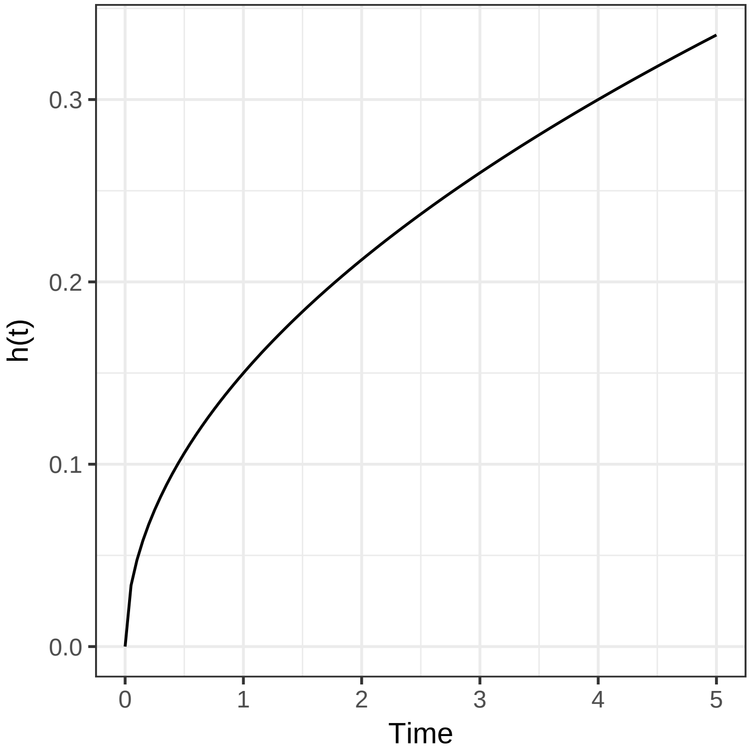

Figure 1.3: Example of hazard function.

function represents the accumulation of hazard (e.g. $h(t)$) over time, and can be defined as

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

it can conveniently be expressed in terms of survival function via the relationship $H(t) = -\log S(t)$, or alternatively with $S(t) = \exp(-H(t))$.

1.4 Estimation of the survival function

The survival function presented in Figure 1.2, panel B, is a non-parametric estimate of the true survival function based on the data only. The estimator employed is this case is the Kaplan-Meier estimator of the survival function (Kaplan and Meier, 1958), with which the estimated survival probabilities are obtained using a product limit formula. The general form for the Kaplan-Meier estimator at time $t_{(i)}$ is

$$\hat{S}(t_{(i)}) = \hat{S}(t_{(i-1)}) \times \hat{P}(T > t_{(i)} | T \geq t_{(i)}),$$

with $t_{(i)}$ being the i^{th} ordered failure time. The interpretation is straightforward: it is the product of the probability of surviving past the previous event-time ($\hat{S}(t_{(i-1)})$) times the conditional probability of surviving past the current time $t_{(i)}$ given survival to at least the current time ($\hat{P}(T > t_{(i)} | T \geq t_{(i)})$). The product limit formula is:

$$\hat{S}(t_{(i)}) = \prod_{j=1}^{i} \hat{P}(T > t_{(j)} | T \geq t_{(j)})$$

The conditional probability in the product limit formula can be estimated from the observed data as:

$$\hat{P}(T > t_{(i)} | T \geq t_{(i)}) = \frac{r_{(i)} - e_{(i)}}{r_{(i)}},$$

where $r_{(i)}$ and $e_{(i)}$ are the number of individuals at risk and the number of events at time $t_{(i)}$, respectively.

The Kaplan-Meier estimator can be computed using R and the function `survfit` from the `survival` package. An example using the simulated data from Section 1.2 (stored in a data frame named `data`):

```
# library(survival)
fit = survfit(Surv(time = t0, event = d) ~ 1, data = data)
summary(fit)
```

```
## Call: survfit(formula = Surv(time = t0, event = d) ~ 1, data = data)
##
##   time n.risk n.event survival std.err lower 95% CI upper 95% CI
##  0.687     10       1      0.9  0.0949       0.7320        1.000
##   0.975      9       1      0.8  0.1265      0.5868      1.000
##   1.047      8       1      0.7  0.1449      0.4665      1.000
##   2.580      7       1      0.6  0.1549      0.3617      0.995
##   2.780      6       1      0.5  0.1581      0.2690      0.929
##   3.143      5       1      0.4  0.1549      0.1872      0.855
##   4.606      2       1      0.2  0.1612      0.0412      0.971
```

By doing so, I obtain an estimate of the survival function (column `survival`) at each distinct failure time (column `time`). For instance, the survival probability at $t = 1.047$ is 0.700, with 95% confidence interval (0.467 - 1.000).

Finally, plotting the estimated survival curve I obtain Figure 1.4, which is exactly the same survival curve presented in panel B of Figure 1.2.

```
# library(ggfortify)
autoplot(fit, conf.int = FALSE, censor = FALSE) +
  theme_bw() +
  coord_cartesian(ylim = c(0, 1)) +
  labs(x = "Time", y = expression(hat(S)[KM](t)))
```

An alternative way of estimating the survival function is to use the *Nelson-Aalen* estimator for the cumulative hazard

$$\hat{H}(t) = \sum_{t_i < t} \frac{e_i}{r_i} = \sum_{t_i < t} \hat{h}_i,$$

and then use the relationship presented in Section 1.3 to obtain the survival function.

1.5 Parametric survival models

In applied settings it is often of interest to assess the association between observed covariates and the survival time of interest. For instance, it may be of interest to study whether a

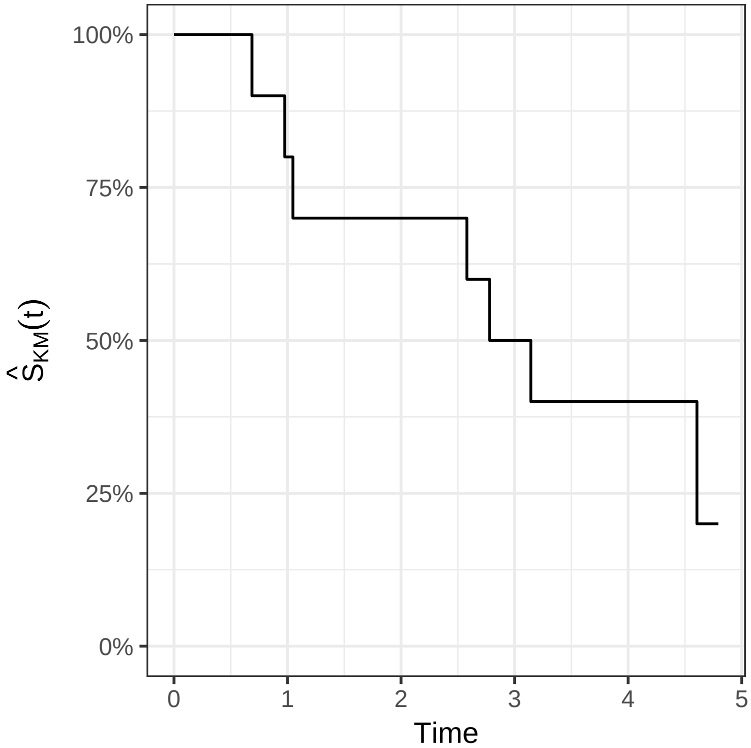

Figure 1.4: Estimated survival function using the Kaplan-Meier estimator on the simulated data.

treatment is effective in slowing disease relapse (e.g. relapse of leukemia), whether there are difference between genders or age categories. A common way of assessing the effect of covariates on a time to event outcome, while adjusting for potentially confounding factors at the same time, consists in using a regression model.

In the context of survival data, two models are commonly used: the *accelerated failure time* model (AFT), and the *proportional hazards* (PH) model. In the former, the natural logarithm of the observed survival time $\log t$ is expressed as a linear function of the covariates X:

$$\log t = X\beta + \epsilon,$$

with β a vector of regression coefficients and ϵ a vector of residual error terms. Assuming a parametric distribution for ϵ determines the regression model: log-normal, log-logistic, Weibull, etc. In the AFT model, a positive association of the covariates with survival time implies an increased expected time to event. In the PH model, the covariates have a multiplicative effect on the hazard function:

$$h(t; X) = h_0(t)g(X),$$

for some $h_0(t)$ and $g(X)$, with $g(\cdot))$ a non-negative function of the covariates. A popular choice for the latter is $g(X) = \exp(X\beta)$; conversely, is is possible to either left the former unspecified, or assume a parametric distribution. The focus of this Section is on specifying a parametric distribution for $h_0(t)$, yielding the so-called *parametric survival regression models*; I will present commonly assumed parametric distributions in Section 1.5.1, the estimation procedure in Section 1.5.2, and an example using data from the International Stroke Trial (IST) (International Stroke Trial Collaborative Group, 1997; Sandercock et al., 2011) in Section 1.5.3. Leaving $h_0(t)$ unspecified yields the semi-parametric Cox model, that I will present in Section 1.6.

From now on I will focus on the proportional hazards formulation of the survival model.

1.5.1 Failure time distributions

I mentioned in Section 1 that the random variable representing the survival time is non-negative; hence, we can choose any non-negative distribution to assign to $h_0(t)$. Commonly used distribution are the Exponential, Weibull, log-Normal, and Gompertz distributions; other possible distributions are the inverse Weibull, the inverse Gamma, the positive stable, the log-skew-Normal, the log-logistic, and complex mixture distributions (such as the two components mixture Weibull distribution, McLachlan and McGiffin (1994)). Each distribution yields a different shape for the Survival and hazard functions; in particular, focusing on the distribution I will be utilising in the rest of this report:

- Exponential distribution:
  - $h_0(t) = \lambda$
  - $S(t) = \exp(-\lambda t)$
  - $\lambda > 0$
- Weibull distribution:
  - $h_0(t) = \lambda p t^{p-1}$
  - $S(t) = \exp(-\lambda t^p)$
  - $\lambda, p > 0$
- log-Normal distribution:
  - $h_0(t) = \frac{\phi(\frac{\log t - \mu}{\sigma})}{\sigma t \left(1 - \Phi\left[\frac{\log t - mu}{\sigma}\right]\right)}$
  - $S(t) = 1 - \Phi\left(\frac{\log t - \mu}{\sigma}\right)$
  - $\mu \in R; \sigma > 0$
- Gompertz distribution:
  - $h_0(t) = \lambda \exp(\gamma t)$
  - $S(t) = \exp\left[-\frac{\lambda}{\gamma}(\exp(\gamma t) - 1)\right]$

  - $\lambda, \gamma > 0$

- Two components mixture Weibull distribution:
  - $h_0(t) = \frac{\lambda_1 p_1 t^{p_1-1}\pi \exp(-\lambda_1 t^{p_1})+\lambda_2 p_2 t^{p_2-1}(1-\pi)\exp(-\lambda_2 t^{p_2})}{\pi \exp(-\lambda_1 t^{p_1})+(1-\pi)\exp(-\lambda_2 t^{p_2})}$
  - $S(t) = \pi \exp(-\lambda_1 t^{p_1}) + (1-\pi)\exp(-\lambda_2 t^{p_2})$
  - $\lambda_1, \lambda_2, p_1, p_2 > 0; \pi \in [0, 1]$

1.5.2 Estimation procedure

Assume n observations with the bivariate response (t_i, d_i), with $i = 1, \ldots, n$. For a given survival function $S(t)$ the density function is given by

$$f(t) = -\frac{dS(t)}{dt},$$

and the hazard function by

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

The parameters of the parametric proportional hazards survival model presented in Section 1.5 can be estimated via the maximum likelihood method. A subject that experiences the event of interest at time t_i contributes to the likelihood the density at time t_i, i.e. $f(t_i)$; conversely, a censored observation know to survive until tile t_i contributes $S(t_i)$ to the likelihood. The individual contribution to the likelihood L_i can therefore be written as

$$L_i = h(t_i)^{d}_{i} S(t_i),$$

where d_i is the event indicator variable. The overall likelihood is the product of the individual contributions:

$$L = \prod_{i=1}^{n} L_i.$$

Taking the natural logarithm of the likelihood for ease of computation:

$$\begin{aligned}\log L = \sum_{i=1}^{n} \left[d_i \log f_i(t_i) + (1 - d_i) \log S_i(t_i)\right] = \\ = \sum_{i=1}^{n} \left[d_i \log h_i(t_i) + \log S_i(t_i)\right]\end{aligned}$$

Implicit in the above log-likelihood are the regression parameters β and the parameters of the parametric distribution of choice for $h_0(t)$.

The log-likelihood function $\log L$ has a closed-form; maximum likelihood estimates for β and the distribution parameters can hence be obtained by maximising $\log L$, e.g. using one of the many general purpose optimisers available in R (`optim`, `nlm`, …).

1.5.3 Data analysis example

The International Stroke Trial (IST) was a large, prospective, randomised controlled trial conducted between 1991 and 1996. The aim of the trial was to assess whether early administration of aspirin, heparin, both or neither influenced clinical outcomes in patients with acute ischaemic stroke (International Stroke Trial Collaborative Group, 1997; Sandercock et al., 2011).

As illustration, I will evaluate the association between tretment with aspirin and/or heparin and survival after acute ischaemic stroke. I will start by reading the data, stored in the `ist.csv` file. This file is a subset of the full IST dataset containing information on age, gender, Country, treatment, and survival; further, individuals with missing values and individuals with a survival time of zero were dropped.

```r
# library(readr)
ist = read_csv("data/ist.csv",
  col_names = c("gender", "age", "rxasp", "rxhep", "country", "d", "t"),
  col_types = "cicccii", skip = 1)
attr(ist, "spec") = NULL # removing "spec" attribute


# turn treatments into factors
ist$rxasp = factor(ist$rxasp, levels = c("N", "Y"))
ist$rxhep = factor(ist$rxhep, levels = c("N", "L", "H"))
```

I fit first a parametric survival model assuming a Weibull distribution for $h_0(t)$. The hazard function, including covariates and the imposing proportional hazards, has the form

$$h(t; X) = \lambda p t^{p-1} \exp(X\beta),$$

while the survival function has the form

$$S(t; X) = \exp(-\lambda t^p \exp(X\beta)).$$

X is the model design matrix, and β is the vector of regression coefficients. The log-likelihood has the form

$$\log L = \sum_{i=1}^{n} [d_i \log h_i(t_i) + \log S_i(t_i)]$$

First, I code a function with the model log-likelihood. The function depends on (1) the model parameters β, λ, and p (`pars` argument), (2) the model design matrix X (`X` argument), and (3) survival time t and event indicator d (`t` and `d` arguments):

```
ll = function(pars, X, t, d) {
  lambda = exp(pars[1])
  p = exp(pars[2])
  beta = pars[-(1:2)]
  log_hi = log(lambda) + log(p) + (p - 1) * log(t) + c(X %*% beta)
  log_Si = -lambda * t ^ p * exp(c(X %*% beta))
  ll = sum(d * log_hi + log_Si + log(t))
  # + sum(log(t)) is the same adjustment that Stata
  # does to remove the time units from log L
  return(-ll)
}
```

The function `ll()` returns the negative log-likelihood as most optimisers minimise a target function (and so does `optim`); however, minimising the negative log-likelihood function is equivalent to maximising the log-likelihood.

I define the model matrix X for a model with aspirin treatment, heparin treatment, and their interactions. The first column is removed to avoid collinearity:

```
X = with(ist, model.matrix(t ~ rxasp * rxhep - 1))[,-1]
```

Next, I define the starting values for the optmisation routine. I choose the value 1 for the parameters of the Weibull distribution and the value 0 for the regression coefficients:

```
start = c(1, 1, rep(0, ncol(X)))
names(start) = c("lambda", "p", colnames(X))
```

The value of the log-likelihood function at the starting values is -92861140101.392. Finally, I use the robust-variance modification of the Marquard algorithm, which is more efficient than Gauss-Newton-like algorithms when starting from points very far from the optimum (Marquardt, 1963; Commenges et al., 2006):

```
# library(marqLevAlg)
fit = marqLevAlg(b = start,
  fn = function(x) ll(x, X = X, t = ist$t, d = ist$d))
```

```
##
## Be patient. The program is computing ...
## The program took 11.26 seconds
```

Assess convergency:

```
fit$istop
```

```
## [1] 1
```

The convergence status indicator is equal to 1, hence the convergence criteria were satisfied. The log-likelihood at the maximum likelihood estimates is 61566.567. The optimising routine returns the upper triangle matrix of variance-covariance estimates at the stopping point, which can be used to obtain standard errors of the estimated coefficients:

```
fit$vcov = matrix(0,
  nrow = length(fit$b),
  ncol = length(fit$b))
fit$vcov[upper.tri(fit$vcov, diag = TRUE)] = fit$v
fit$vcov[lower.tri(fit$vcov)] = t(fit$vcov)[lower.tri(fit$vcov)]
```

Finally, I build a table of results:

```
res = data.frame(
  coef = fit$b,
  hr = exp(fit$b),
  se = sqrt(diag(fit$vcov)))
res$z = res$coef / res$se
res$p = 2 * pmin(pnorm(-abs(res$z)), 1 - pnorm(abs(res$z)))
kable(res,
  digits = 3,
  align = "rrrrr",
  booktabs = TRUE,
  col.names = c("Beta", "Hazard ratio", "SE (Beta)", "Z", "P > |Z|"),
```

Table 1.1: Results from a parametric Weibull model.

	Beta	Hazard ratio	SE (Beta)	Z	P > |Z|
lambda	-3.852	0.021	0.047	-82.665	0.000
p	-0.759	0.468	0.015	-52.050	0.000
rxaspY	-0.037	0.964	0.044	-0.850	0.395
rxhepL	0.085	1.088	0.052	1.640	0.101
rxhepH	0.044	1.045	0.052	0.843	0.399
rxaspY:rxhepL	-0.095	0.910	0.075	-1.269	0.204
rxaspY:rxhepH	0.037	1.038	0.074	0.503	0.615

```
  linesep = "",
  caption = "Results from a parametric Weibull model.")
```

I test the interaction term using the Wald test to assess whether combining aspirin and heparin alter their association with time to event. I use the Wald χ^2 test statistic as the sample size is big enough for it to be equivalent to its F counterpart:

```
# identify the interaction terms
idx = grepl(":", names(fit$b))

# compute the W statistic
W = t(fit$b[idx]) %*% solve(fit$vcov[idx, idx]) %*% fit$b[idx]

# produce the test
c(W = W, df = sum(idx), `p-value` = 1 - pchisq(W, sum(idx)))

##         W        df   p-value
## 2.6104650 2.0000000 0.2711095
```

The interaction terms seem to be not statistically significant. We can conclude the association of aspirin and heparin treatments with survival are not dependent on one another.

I can then re-fit the model excluding the interaction terms:

```
X = with(ist, model.matrix(t ~ rxasp + rxhep - 1))[,-1]
start = c(1, 1, rep(0, ncol(X)))
names(start) = c("lambda", "p", colnames(X))
re_fit = marqLevAlg(b = start,
  fn = function(x) ll(x, X = X, t = ist$t, d = ist$d))
```

```
##
## Be patient. The program is computing ...
## The program took 5.85 seconds
```

```
re_fit$istop
```

```
## [1] 1
```

The routine converged. I produce the variance-covariance matrix:

```
re_fit$vcov = matrix(0,
  nrow = length(re_fit$b),
  ncol = length(re_fit$b))
re_fit$vcov[upper.tri(re_fit$vcov, diag = TRUE)] = re_fit$v
re_fit$vcov[lower.tri(re_fit$vcov)] = t(re_fit$vcov)[lower.tri(re_fit$vcov)]
```

Finally, I build a new table of results:

```
re_res = data.frame(
  coef = re_fit$b,
  hr = exp(re_fit$b),
  se = sqrt(diag(re_fit$vcov)))
re_res$z = re_res$coef / re_res$se
re_res$p = 2 * pmin(pnorm(-abs(re_res$z)), 1 - pnorm(abs(re_res$z)))
kable(re_res,
  digits = 3,
```

Table 1.2: Results from a parametric Weibull model with no interactions.

	Beta	Hazard ratio	SE (Beta)	Z	P > |Z|
lambda	-3.845	0.021	0.044	-87.441	0.000
p	-0.759	0.468	0.015	-52.056	0.000
rxaspY	-0.051	0.950	0.030	-1.689	0.091
rxhepL	0.039	1.040	0.037	1.043	0.297
rxhepH	0.062	1.064	0.037	1.684	0.092

```
  align = "rrrrr",
  col.names = c("Beta", "Hazard ratio", "SE (Beta)", "Z", "P > |Z|"),
  booktabs = TRUE,
  linesep = "",
  caption = "Results from a parametric Weibull model with no interactions.")
```

I now test the significance of the two coefficients related to heparin treatment jointly:

```
idx = grepl("^rxhep", names(re_fit$b))
W = t(re_fit$b[idx]) %*% solve(re_fit$vcov[idx, idx]) %*% re_fit$b[idx]
c(W = W, df = sum(idx), `p-value` = 1 - pchisq(W, sum(idx)))
```

```
##       W      df p-value
## 3.08001 2.00000 0.21438
```

Heparin treatment seems to be not statistically significantly associated with time to death in acute ischaemic stroke patients; the effect size is small, with a 4% and 7% increased risk for the L and H heparin treatment modalities versus no heparin treatment, respectively.

Finally, the treatment with aspirin is also not statistically significantly associated with the outcome; effect size is small as well, approximately a 5% risk reduction for aspirin treatment compared to no treatment with aspirin (Table 1.2).

This is a simple application of parametric survival models; a fully developed analysis should take further aspects into account, such as considering different hazard distributions. It is possible to estimate various models and compare their fit to a specific distribution using information criteria such as the Akaike information criterion (AIC) and the Bayesian information criterion (BIC).

1.6 The Cox proportional hazards model

The parametric survival models of Section 1.5 could have both the accelerated failure time form and the proportional hazards form. Recall that the latter is formulated in terms of the hazard function:

$$h(t; X) = h_0(t) \exp(X\beta)$$

As I mentioned before, this model requires specifying the baseline hazard function $h_0(t)$ (e.g. using one of the parametric distributions of Section 1.5.1) and by leaving it unspecified I obtain the Cox proportional hazards model. Such model is also called *semi-parametric* as it is formed by a non-parametric component (the baseline hazard left unspecified) and a parametric component (the modelling assumption for the functional form of $g(\cdot)$, the usual $\exp(X\beta)$ in this case). The survival function for a Cox model can be written as:

$$S(t; X) = \exp\left[-\int_0^t h_0(u) \exp(X\beta)\ du\right]$$

The main problems when fitting a Cox model are related to estimation of the regression coefficients β and of the survival function $S(t)$. The former is tackled in Section 1.6.1, the latter in Section 1.6.2.

1.6.1 Estimation procedure

The main method for estimating the regression coefficients is the method of partial likelihood, proposed and discussed in detail in Cox (1972) and Cox (1975). In brief, the observed data are assumed to have density function $f(t;\theta,\beta)$ in which β is the vector of regression coefficients of interest and θ can be considered a vector of nuisance parameters. In particular, θ represents the unspecified function $h_0(t)$. It can be showed that is is possible to factorise the density into two terms, one of which only depends on β: this term is called *partial likelihood*. Ignoring the term that depends on θ, and even if the partial likelihood is not directly interpretable as a likelihood in the ordinary sense, it can be used like an ordinary likelihood for estimation purposes as the usual asymptotic properties formulas and properties associated with the likelihood function and likelihood estimation apply. The partial likelihood applies directly to the relative risk model $h(t;X)$, assuming independent right censoring. The individual contribution to the likelihood has the form

$$L_i(\beta) = \frac{h(t_i;x_i)\Delta t_i}{\sum_{l\in R(t_i)} h(t_i;x_l)\Delta t_i},$$

and provides information on failures occurrence in the interval $[t_i, t_i + \Delta t_i)$; $R(t_i)$ is the risk set of individuals at risk of failing at time t_i^-, right before t_i. Under the relative risk model, the baseline hazard in $h(t;X)$ cancels out in the numerator and denominator; the product over i gives the partial likelihood for β:

$$L(\beta) = \prod_{i=1}^{n} \frac{\exp(x_i\beta)}{\sum_{l\in R(t_i)} \exp(x_l\beta)}.$$

The values of β that maximise the partial likelihood $\hat{\beta}$ can be obtained by using a Newton-Raphson-like algorithm; asymptotics are fully analogous to a parametric likelihood. A caveat of the partial likelihood method is that it assumes continuous failure times: in practice, that is unrealistic and there will be tied failure times (e.g. due to rounding). In that case, several methods have been proposed to adjust the partial likelihood in order to handle ties; see for instance Peto (1972), Breslow (1974), and Efron (1977)

1.6.2 Estimating the survival function

Consider deriving an estimator for the survival function from a Cox model: the form of $h_0(t)$ is unspecified, hence it is not possible to directly estimate the parameters of the distribution as in fully parametric survival models. Under a Cox model, the survival function has the form

$$S(t;X) = S_0(t)^{\exp(X\beta)}$$

The coefficients β are estimated using the penalised likelihood procedure, and the baseline survival function $S_0(t)$ is estimated by assuming that the baseline hazard function is constant between each pair of consecutive observed failure times. The resulting estimator, known as the Breslow estimator, estimates the cumulative baseline hazard function as

$$\hat{H}_0(t) = \sum_{t(i)\leq t} \frac{e_{(i)}}{\sum_{l\in R(t_{(i)})} \exp(x_l\hat{\beta})},$$

with $e_{(i)}$ the number of events at time $t_{(i)}$. The baseline survival function follows as

$$\hat{S}_0(t) = \exp\left[-\hat{H}_0(t)\right],$$

and the survival function as

$$\hat{S}(t;X) = \hat{S}_0(t)^{\exp(X\hat{\beta})}.$$

An alternative estimator based on approximating the baseline survival function as a step function and consequently solving k simultaneous equations has been proposed by Kalbfleisch and Prentice (2011), and is omitted here.

1.6.3 Model assumptions

The Cox model relies on two main assumptions. First, the assumption of non-informative censoring, e.g. the censoring process must be independent of any covariate, observed and not. Second, the proportional hazards assumption requires hazards to be proportional across time, e.g. the hazard ratios must be constant. There are several ways of testing the proportional hazards assumption, both analytical and graphical; see Chapter 4 of Kleinbaum and Klein (2012) for further details.

1.6.4 Data analysis example

In this section I re-analyse the IST data of Section 1.5.3 using a semi-parametric Cox model. I first read the dataset:

```
# library(readr)
ist = read_csv("data/ist.csv",
  col_names = c("gender", "age", "rxasp", "rxhep", "country", "d", "t"),
  col_types = "cicccii", skip = 1)
attr(ist, "spec") = NULL # removing "spec" attribute

# turn treatments into factors
ist$rxasp = factor(ist$rxasp, levels = c("N", "Y"))
ist$rxhep = factor(ist$rxhep, levels = c("N", "L", "H"))
```

I fit the Cox model using the `coxph()` function from the `survival` package:

```
# library(survival)
fit = coxph(Surv(t, d) ~ rxasp * rxhep, data = ist)
summary(fit)
```

```
## Call:
## coxph(formula = Surv(t, d) ~ rxasp * rxhep, data = ist)
##
##   n= 19378, number of events= 4315
##
##                   coef exp(coef) se(coef)      z Pr(>|z|)
## rxaspY        -0.03710   0.96358  0.04356 -0.852    0.394
## rxhepL         0.08017   1.08347  0.05161  1.553    0.120
## rxhepH         0.04215   1.04305  0.05221  0.807    0.419
## rxaspY:rxhepL -0.09049   0.91348  0.07458 -1.213    0.225
## rxaspY:rxhepH  0.03595   1.03661  0.07415  0.485    0.628
##
##               exp(coef) exp(-coef) lower .95 upper .95
## rxaspY           0.9636     1.0378    0.8847     1.049
## rxhepL           1.0835     0.9230    0.9792     1.199
## rxhepH           1.0431     0.9587    0.9416     1.155
## rxaspY:rxhepL    0.9135     1.0947    0.7893     1.057
## rxaspY:rxhepH    1.0366     0.9647    0.8964     1.199
##
## Concordance= 0.513  (se = 0.004 )
## Rsquare= 0   (max possible= 0.987 )
## Likelihood ratio test= 7.98  on 5 df,   p=0.1574
## Wald test            = 8.02  on 5 df,   p=0.1554
## Score (logrank) test = 8.02  on 5 df,   p=0.1551
```

I test again the joint significancy of the interaction terms using the Wald test:

```
idx = grepl(":", names(coef(fit)))
```

```
W = t(coef(fit)[idx]) %*% solve(vcov(fit)[idx, idx]) %*% coef(fit)[idx]
c(W = W, df = sum(idx), `p-value` = 1 - pchisq(W, sum(idx)))
```

```
##         W        df   p-value
## 2.3929136 2.0000000 0.3022633
```

Analogously as before, the interaction is not significantly different than zero. I re-fit the model without the interaction term:

```
# library(survival)
re_fit = coxph(Surv(t, d) ~ rxasp + rxhep, data = ist)
summary(re_fit)
```

```
## Call:
## coxph(formula = Surv(t, d) ~ rxasp + rxhep, data = ist)
##
##   n= 19378, number of events= 4315
##
##             coef exp(coef) se(coef)      z Pr(>|z|)
## rxaspY -0.05079   0.95048  0.03046 -1.668   0.0954 .
## rxhepL  0.03641   1.03708  0.03725  0.978   0.3283
## rxhepH  0.05991   1.06174  0.03707  1.616   0.1061
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##        exp(coef) exp(-coef) lower .95 upper .95
## rxaspY    0.9505     1.0521    0.8954     1.009
## rxhepL    1.0371     0.9642    0.9641     1.116
## rxhepH    1.0617     0.9419    0.9873     1.142
##
```

```
## Concordance= 0.511  (se = 0.004 )
## Rsquare= 0   (max possible= 0.987 )
## Likelihood ratio test= 5.58  on 3 df,   p=0.1337
## Wald test            = 5.59  on 3 df,   p=0.1333
## Score (logrank) test = 5.59  on 3 df,   p=0.1332
```

Testing the significancy of the heparin treatment using the Wald test:

```
idx = grepl("^rxhep", names(coef(fit)))
W = t(coef(fit)[idx]) %*% solve(vcov(fit)[idx, idx]) %*% coef(fit)[idx]
c(W = W, df = sum(idx), `p-value` = 1 - pchisq(W, sum(idx)))
```

```
##         W        df   p-value
## 2.4963250 2.0000000 0.2870317
```

Treatment with heparin is not statistically significantly associated with risk of death; besides that, the effect size of heparin treatment is small: approximately 4% and 6% risk increase for heparin treatment modalities `L` and `H` compared to no heparin treatment, respectively.

The treatment with aspirin is also barely significantly different than zero, assuming a significancy level $\alpha = 0.10$, with a p-value of 0.0954. The effect size is comparable to the estimated effect size obtained with the Weibull model, approximately 5% risk reduction for treatment with aspirin compared to no aspirin treatment.

1.7 Advances in survival analysis

There are several extensions of the statistical methods presented in this Chapter: I will briefly introduce some of them in this Section, without going into great detail as that would be beyond the scope of this report.

The proportional hazards models have been extended to include time-dependent covariates and time-dependent covariate effects; additionally, the Cox model has been extended to allow stratification by a given factor (details in Kalbfleisch and Prentice (2011) and Kleinbaum and Klein (2012)). Parametric survival models have been generalised to allow combinations of linear predictors and penalised smoothers for the effect of time and covariates, both in the proportional hazards and proportional odds framework (Liu et al., 2016). More generally, models have been developed to account for competing events and multi-state diseases, even with intermediate states, and for modelling a wide range of multivariate survival data (Geskus, 2015; Crowder, 2016). Finally, the main advances that I will discuss further are models with random effects and joint models for longitudinal and survival data, in Chapters 2 and 3 respectively.

Chapter 2

Survival models with random effects

Random effects models are a kind of hierarchical model in which the data is assumed to have some sort of hierarchical structure: imagine having individual patients data clustered into families, cities, regions, and so on. It is also assumed that individuals are homogeneous within hierarchical unit, heterogeneous between different units. In comparison, fixed effects models do not take into account any hierarchy in the data. In biostatistics, also, the terms *fixed effects* and *random effects* have a special meaning, referring to the *population-average* and *subject-specific* effects, respectively, with the latter generally assumed to be unknown, unobserved variables.

Random effects models are generally used to analyse hierarchical data with a continuous, normally distributed outcome (e.g. hemoglobin levels, inflammation markers, ...); such models are referred to as *linear mixed-effects models*, as they can incorporate both fixed and random effects, and generalise the linear regression model. Additionally, when data consists in multiple observations for a given individual over time (and therefore the first level of clustering consists in the individual himself) the term *longitudinal data* is used. It is possible to encounter hierarchical data originating from a variety of ditribution from the exponential family such as the Poisson, Gamma, and Binomial distribution. Linear mixed-effects models can be generalised to include such data, and these models are generally referred to as *generalised linear mixed-effects models*. Practically speaking, the generalisation is analogous to generalising linear models to generalised linear models. It is also possible to relax the normality assumption for continuous, hierarchical data and model the median (or any quantile, really) rather than the mean. Such models are called *linear quantile mixed-effects models*, and generalise the linear mixed-effects models as quantile regression is generalising the linear model. Survival data can present a hierarchical structure too; for instance, data could be clustered in geographical areas, institutions, or patients themselves. Meta-analysis of individual-patient data are a common example of survival data (when the outcome is time to event, of course) with some hierarchical structure; another example is given by repeated-events data, such as infections or acute recurrent events, in which the first level of the hierarchical structure consists in the patient. Another example of survival data with biological cluster is given by twin data, in which siblings share some genetic factors. This heterogeneity structure often leads to violation of the implicit assumption that populations are homogeneous: sometimes it is impossible to include all relevant risk factors, or maybe such risk factors are not known at all. The result is unobserved heterogeneity. The simplest survival model with random effects is the *univariate frailty model*, in which a random effect - named frailty - is included in the model to account for the unobserved heterogeneity. The univariate frailty model can be generalised by allowing the frailty term to be shared between observations belonging to the same cluster of data. The resulting models are named *shared frailty model*. The frailty term generally acts multiplicatively on the baseline hazard, and it is modelled on the hazard scale; it is possible to alternatively formulate the model in terms of random effects rather than frailties, by including the frailty as an additive term on the log-hazard scale.

I will introduce the univariate frailty model in Section 2.1, and generalise it to allow shared frailty terms in Section 2.2. Finally, I will present the alternative formulation in terms of random effects rather than frailties in Section 2.3. A comprehensive treatment of frailty models in survival analysis is given in Hougaard (2000) and Wienke (2010).

2.1 Univariate frailty models

In those settings where risk factors are not measured, their relevance is unknown, or it is not known whether such risk factor exist at all or not, it is useful to consider two sources of variability in survival analysis: variability accounted for by observable risk factors included in the model and heterogeneity caused by unknown covariates. The unobserved heterogeneity is described by the frailty term, which is assumed to follow some distribution. Formally:

$$h(t|\alpha) = \alpha h_0(t),$$

where α is a non-observed frailty effect and $h_0(t)$ is the baseline hazard function. The random variable α, the frailty term, is chosen to have a distribution $f(\alpha)$ with expectation $E(\alpha) = 1$ and variance $V(\alpha) = \sigma^2$. $V(\alpha)$ is interpretable as a measure of heterogeneity across the population in baseline risk: as σ^2 increases the values of α are more dispersed, with greater heterogeneity in $\alpha h_0(t)$. Underlying assumptions are: the frailty is time independent, and it acts multiplicatively on the underlying baseline hazard function.

Introducing observed covariates into the model:

$$h(t|X, \alpha) = \alpha h_0(t) \exp(X\beta) = \alpha h(t|X),$$

with X and β covariates and regression coefficients, respectively. Given the relationship between hazard and survival function, it can be showed that the individual survival function conditional on the frailty is $S(t|\alpha) = S(t)^{\alpha}$. The population (i.e. marginal, or unconditional) survival function is obtained by integrating out the frailty from the conditional survival function:

$$S(t) = \int_0^{+\infty} [S(t)]^\alpha f(\alpha)\ d\alpha$$

The individual contribution to the likelihood (assuming no delayed entry) is conditional on the unobserved frailty α

$$L_i = \prod_{i=1}^{n} (\alpha h_0(t_i) \exp(X_i\beta))^{d_i} \exp(-\alpha H_0(t_i) \exp(X_i\beta)),$$

with d_i event indicator variable, $H_0(t_i)$ cumulative baseline hazard, and t_i observed survival time - all relative to the i-th individual.

Different choices for the frailty distribution are possible. Assigning a probability distribution implies that the frailty can be integrated out of the likelihood function. After this integration, the likelihood can be maximized in the usual way if an explicit form of it exists. Otherwise, more sophisticated approaches like numerical integration or Markov Chain Monte Carlo methods need to be applied. The most often used frailty distributions are the gamma and the log-normal distribution; the positive stable and the inverse Gaussian distribution are also common.

Assuming that the frailty α has a Gamma distribution is convenient: it has the appropriate range $(0, \infty)$ and it is mathematically tractable. A Gamma distribution with parameters a and b has density

$$f(x) = \frac{x^{a-1}\exp(-x/b)}{\Gamma(a)b^a};$$

by choosing $a = 1/\theta$ and $b = \theta$ the resulting distribution has expectation 1 and finite variance θ. In these settings, the model is analytically tractable: the population survival function has the form

$$S(t) = (1 - \theta \log(S(t)))^{-1/\theta};$$

the likelihood follows by substitution. Estimating such model becomes therefore straightforward, which likely contributed to the popularity of Gamma frailty models.

Together with the Gamma distribution, the log-normal distribution is the most commonly used frailty distribution, given its strong ties to random effect models; more on that in Section 2.3. Hence, assuming a log-normal distribution with a single parameter $\theta > 0$ (for comparison with the mathematically tractable Gamma frailty model), with density

$$f(x) = (2\pi\theta)^{-\frac{1}{2}} x^{-1} \exp\left(-\frac{(\log x)^2}{2\theta}\right),$$

the resulting model has a frailty whose expectation is finite. Nevertheless, this frailty distribution cannot be integrated out of the survival function analytically to obtain the population survival function, and therefore requires more complex estimation procedures involving numerical integration (taking the maximum likelihood approach), mathematical approximations (such as the Laplace approximation), or Markov Chain Monte Carlo methods (Clayton, 1991; Sinha et al., 2008). Further details in Chapter 4.

2.2 Shared frailty models

Further generalising the model presented in Section 2.1, it is possible for the frailty effect α to be shared between clusters of study subjects. Specifically, for the j-th observation in the i-th cluster:

$$h_{ij}(t|\alpha_i) = \alpha_i h(t|X_{ij}).$$

The conditional survival function is:

$$S_{ij}(t|\alpha_i) = S_{ij}(t)^{\alpha_i}.$$

In this setting, the cluster-specific contribution to the likelihood is obtained by calculating the cluster-specific likelihood conditional on the frailty, consequently integrating out the frailty itself:

$$L_i = \int_A L_i(\alpha_i) f(\alpha_i)\, d\alpha,$$

with $f(\alpha)$ the distribution of the frailty, A its domain, and $L_i(\alpha_i)$ the cluster-specific contribution to the likelihood, conditional on the frailty. The cluster-specific contribution to the likelihood is

$$L_i(\alpha_i) = \alpha_i^{D_i} \prod_{j=1}^{n_i} S_{ij}(t_{ij})^{\alpha_i} h_{ij}(t_{ij})^{d_{ij}},$$

with $D_i = \sum_{j=1}^{n_i} d_{ij}$. Analogously as before, analytical formulae can be obtained when α_i follows a Gamma distribution:

$$L_i = \left[\prod_{j=1}^{n_i} h_{ij}(t_{ij})^{d_{ij}} \right] \frac{\Gamma(1/\theta + D_i)}{\Gamma(1/\theta)} \left[1 - \theta \sum_{j=1}^{n_i} \log S_{ij}(t_{ij}) \right]^{-1/\theta - D_i};$$

further details in Gutierrez (2002). As in the univariate frailty model, assuming a log-normal distribution requires some numerical approximation to be performed, being the resulting model analytically intractable.

2.3 Alternative formulation

I mentioned briefly in Section 2.1 that the log-normal distribution for the frailty term has strong ties to random-effects models. Recall the formulation for a log-normal shared frailty model:

$$h_{ij}(t|\alpha_i) = \alpha_i h(t|X_{ij}) = \alpha_i h_0(t) \exp(X_{ij}\beta),$$

with α_i following a log-normal distribution. It is possible to formulate the same model on the log-hazard scale as

$$h_{ij}(t|\alpha_i) = h_0(t) \exp(X_{ij}\beta + \eta_i),$$

with $\eta_i = \log \alpha_i$. η_i results being normally distributed with parameters μ and σ^2 related to those of the log-normal distribution by the relationship

$$E(\alpha_i) = \exp(\mu + \sigma^2/2)$$

and

$$Var(\alpha_i) = \exp(2\mu + \sigma^2)(\exp(\sigma^2) - 1)$$

By formulating the model on the log-hazard scale, the frailty term has a direct interpretation as a random intercept in the model; that is, the heterogeneity is modelled by allowing the model intercept to vary between clusters. Consequently, it is possible to further extend this model by allowing random covariates effects, potentially ranging over multiple levels of clustering. Using the usual mixed-effects models notation:

$$h_{ij}(t|b_i) = h_0(t) \exp(X_{ij}\beta + Z_i b_i),$$

with X_{ij} representing the design matrix for the fixed effects β and Z_i representing the design matrix for the random effects b_i. Any distribution or functional form can be assumed for $h_0(t)$ (Crowther et al., 2014), or it is possible to leave it unspecified altogether yielding a Cox model with random effects (Ripatti and Palmgren, 2000; Therneau et al., 2003).

Chapter 3

Joint models for longitudinal and survival data

It is increasingly common for observational studies and trials to follow participants over time, recording abundant data on clinical features throughout the duration of the study. Moreover, routinely collected healthcare consumption data and population registries are being used more and more for research purpose, after being linked with other data sources (and each other). As a consequence, applied researchers often encounter longitudinally recorded covariates to account for when studying the clinical outcome of interest (e.g. time to event, that is what I will focus on). Researchers then face two options: (1) select only one of the multiple values per individual and analyse as such, ignoring part of the available data, or (2) take into account the potential dependency and association between the repeatedly measured covariates and the outcome interest. The latter is usually the sensible choice, as the longitudinal data can be important predictors or surrogates of the time to event outcome. A powerful tool to jointly model longitudinal and time-to-event data is joint models for longitudinal and time to event data, in which the longitudinal and survival processes are modelled jointly into a single model allowing to infer their association. The development of such models was motivated by HIV/AIDS clinical trials, in which immune response (in terms of CD4 lymphocite cells count) was recorded over the duration of the trial and the association with survival was of interest. Seminal works on the topic are the papers by Wulfsohn and Tsiatis (1997), Tsiatis and Davidian (2004), Henderson et al. (2000), Pawitan and Self (1993); a more recent tractation of the topic is in Ibrahim et al. (2010), Rizopoulos (2012), Gould et al. (2015).

Previous attempts to tackle this problem consisted in (1) fitting a time-dependent Cox model (Cox, 1972) by splitting individual rows every time a new observation from the longitudinal covariate becomes available, and (2) by using two-stages methods in which the longitudinal and survival data were modelled separately (Tsiatis et al., 1995). Nevertheless, it has been showed that joint modeling increases efficiency and reduces bias (Hogan and Laird, 1998), while improving predictions at the same time (Rizopoulos et al., 2014).

In this Chapter I will focus on the basic joint model for longitudinal and survival data, with a single longitudinal process. I will present its formulation in Section 3.1, and the estimation process in Section 3.2. However, several extensions of the basic joint model presented in this Chapter have been proposed during the years, as the topic has received considerable attention. A review on the state of the art in joint models with a single longitudinal process is given by Gould et al. (2015). Of course, the joint model has been extended to allow incorporating multiple longitudinal processes at one, measured intermittently and not necessarily at the same time or with the same structure of the association with the survival outcome: a review on recent developments, software, and persisting issues is given by Hickey et al. (2016).

3.1 Model formulation

A joint model for longitudinal and survival data consists of two components: a model for the longitudinal part (I will be assuming a single longitudinal trajectory from now on for simplicity) and a model for the survival part. These two components will then share a set of parameters that will describe the association between the two processes. In literature, the dominant approach seems to be allowing the two components to share random effects; I will follow this approach.

Building on the notation from Section 1.3, let $y_{ij} = \{y_{ij}(t_{ij}) \; \forall \; j = 1, \ldots, n_i\}$ be the observed longitudinal response for the i^{th} subject, with $y_{ij}(t_{ij})$ the observed response at time t_{ij} and n_i the number of longitudinal observations.

The longitudinal component of the joint model is modelled within the mixed-effects framework (Diggle et al., 2013), as longitudinal data is likely measured intermittently and with error. Therefore:

$$y_i(t) = m_i(t) + \epsilon_i(t), \;\; \epsilon_i(t) \sim N(0, \sigma^2)$$

and

$$m_i(t) = X_i(t)\beta + Z_i(t)b, \;\; b \sim N(0, \Sigma)$$

with $X_i(t)$ and $Z_i(t)$ the time-dependent design matrices for the fixed and random effects, respectively. $y_i(t)$ represents the observed longitudinal trajectory at time t, which could be decomposed into the true longitudinal trajectory $m_i(t)$ plus the measurement error $\epsilon_i(t)$.

The survival component of the joint model is modelled using a proportional hazards time to event model, given the true unobserved longitudinal trajectory up to time t, i.e. $M_i(t) = \{m_i(s) \; \forall \; 0 \leq s \leq t\}$:

$$h(t|M_i(t)) = h_0(t) \exp(W\psi + \alpha m_i(t)),$$

where $h_0(t)$ is the baseline hazard function and W is a vector of time-fixed covariates with their regression parameters ψ. α is the association parameter that links the longitudinal component and the survival component of the joint model; it can be intepreted as the change in log-hazard ratio for a unit increase in the true longitudinal trajectory $m_i(t)$, at time t. This specific form of the association parameter is also known as the *current value* parametrisation;

additional association structures are available, allowing for instance interactions, association with the slope of the trajectory or its cumulative effect, and so on. Further details in Rizopoulos (2012).

The survival function follows as

$$S(t|M_i(t)) = \exp\left(-\int_0^t h_0(u)\exp(W\psi + \alpha m_i(u))\ du\right)$$

Finally, regarding $h_0(t)$: the choice of the baseline hazard function follows the usual rationale. It can be left unspecified, therefore resulting in a Cox model for the survival component of the joint model, or it can be specified using a parametric distribution (e.g. a distribution from Section 1.5.1) or some flexible alternative (Crowther et al., 2012). Nevertheless, Hsieh et al. (2006) showed that choosing the Cox model for the survival component yields standard errors that are underestimated; consequently, bootstrap is required to obtain correct standard errors in that situation.

3.2 Estimation process

Estimation of a joint model for longitudinal and survival data is a non-trivial task. The complexity of jointly modelling the longitudinal component and the survival component motivated using a two-stages procedure as mentioned in Section 3. With that approach, the longitudinal component is modelled and estimated separately; consequently, subject-specific predictions from the longitudinal model are produced and plugged into the survival model as time-varying covariates. Despite the simplicity of this approach, though, it has been showed that it produces substantial bias and poor coverage (Tsiatis and Davidian, 2001; Sweeting and Thompson, 2011). Therefore, an approach that models both processes jointly is required. in particular, two approaches are predominant: a full likelihood approach, and a Bayesian approach; both have appealing characteristics, but they share the feature of being computationally intensive.

Focusing on the full likelihood approach, it is possible to formulate the joint likelihood (Rizopoulos, 2012) for the overall parameter vector $\theta = \{\theta_t, \theta_y, \theta_b\}$, formed by the parameters of the survival component, the parameters of the longitudinal component, and the elements of the variance-covariance matrix of the random effects, respectively. The joint distribution of the survival time T_i, the event indicator d_i, and the longitudinal response y_i, conditional on the random effects b_i, can be expressed as:

$$P(T_i, d_i, y_i | b_i, \theta) = P(T_i, d_i | b_i, \theta) P(y_i | b_i, \theta),$$

with

$$P(y_i | b_i, \theta) = \prod_{j=1}^{n_i} P(y_i(t_{ij}) | b_i, \theta).$$

It follows that the contribution to the log-likelihood for the i^{th} patient is

$$\begin{aligned}
\log L(\theta) &= \log \int_{-\infty}^{+\infty} P(T_i, d_i, y_i, b_i; \theta)\ db_i \\
&= \log \int_{-\infty}^{+\infty} P(T_i, d_i | b_i, \theta_t) \left[\prod_{j=1}^{n_i} P(y_i(t_{ij}) | b_i, \theta_y) \right] P(b_i | \theta_b)\ db_i
\end{aligned}$$

with $P(T_i, d_i | b_i, \theta_t)$ the likelihood relative to the survival component of the model:

$$\begin{aligned}
P(T_i, d_i | b_i, \theta_t) &= h_i(T_i | M_i(T_i), \theta_t)^{d_i} S_i(T_i | M_i(T_i), \theta_t) \\
&= [h_0(T_i) \exp(W\psi + \alpha m_i(T_i))]^{d_i} \exp\left[- \int_0^{T_i} h_0(u) \exp(W\psi + \alpha m_i(u))\ du \right],
\end{aligned}$$

$P(y_i(t_{ij}) | b_i, \theta_y)$ the likelihood of the longitudinal process at time t_{ij}:

$$P(y_i(t_{ij}) | b_i, \theta_y) = (2\pi\sigma^2)^{-1/2} \exp\left[- \frac{(y_i(t_{ij}) - m_i(t_{ij}))^2}{2\sigma^2} \right],$$

and $P(b_i|\theta_b)$ the density of the random effects:

$$P(b_i|\theta_b) = (2\pi)^{-q_b/2}|\Sigma|^{-1/2} \exp\left[-\frac{b_i^T\Sigma^{-1}b_i}{2}\right]$$

q_b being the dimension of the random effects.

Historically the full joint likelihood has been maximised using the Expectation-Maximisation algorithm (Dempster et al., 1977); alternatively, it is possible to use general purpose optimisers to maximise the full joint likelihood via algorithms such as the Newton-Raphson algorithm. Nevertheless, significant computational challenges persist; I will discuss them further in Chapter 4.

Chapter 4

Computational challenges in survival models with random effects

The models I presented in Chapter 2 and 3 present significant computational challenges during the estimation process. I showed how frailty models with a Gamma frailty are analytically tractable, as it is possible to obtain closed-form expressions for the marginal survival function and therefore the likelihood; conversely, including a log-normal frailty (or, correspondingly, random effects) in a survival model yields a survival function - and likelihood - that does not have a closed form. Recall the i^{th}-custer-specific contribution to the likelihood for a shared frailty model:

$$L_i(\alpha_i) = \alpha_i^{D_i} \prod_{j=1}^{n_i} S_{ij}(t_{ij})^{\alpha_i} h_{ij}(t_{ij})^{d_{ij}}$$

The marginal survival function has the form

$$S_{ij}(t_{ij}) = \int_0^{+\infty} [S_{ij}(t_{ij})]_i^{\alpha} \, f(\alpha_i) \, d\alpha_i$$

with $f(\alpha_i)$ a log-normal density function. This integral has no closed form, hence it is necessary to approximate it in order to obtain (1) marginal survival and (2) the likelihood.

Analogously, recall the joint likelihood in joint models for longitudinal and survival data:

$$\log L(\theta) = \log \int_{-\infty}^{+\infty} P(T_i, d_i, y_i, b_i; \theta) \ db_i.$$

Evaluating this likelihood requires evaluating an analytically intractable integral over a possibly multi-dimentional integral over the infinite domain; it is therefore necessary to use some method to approximate it numerically.

Methods for approximating intractable integrals form the majority of this Chapter, with more details in Section 4.1. I will conclude with additional considerations on numerical methods in Section 4.2.

4.1 Numerical integration

The term *numerical integration* implies the approximation of the integral of a function; generally, it aims to use the minimum number of function evaluations possible as it tends to be numerically expensive. There is a variety of methods being proposed in literature to perform numerical integration; throughout this Section, I will focus on *quadrature rules*, i.e. any method that evaluates the function to be integrated at some points over the integration domain and combines the resulting values to obtain an approximation of the integral. Quadrature rules vary in complexity and accuracy, and generally accuracy improves as rules get more complex. Additionally, integration of functions in few dimensions is generally not too problematic; the task becomes more difficult when integrating over many dimensions as obtaining an acceptable level of accuracy often requires an unfeasible number of function evaluations.

4.1.1 Unidimensional functions

The simplest method to approximate the integral of a unidimensional function numerically is given by the *Riemann sum*. A Riemann sum is an approximation of the integral of a continuous function $f(x)$ over an integration domain $[a, b]$ by a finite sum, defined as:

$$\int_a^b f(x)\ dx \approx \sum_{i=1}^{N} f(x_i^*)\Delta(x_i),$$

with $P = \{[x_0, x_1], [x_1, x_2], \dots, [x_{N-1}, x_N]\}$ a partition of $[a, b]$ such that $a = x_0 < x_1 < x_2 < \cdots < x_{N-1} < x_N = b$, $\Delta(x_i) = x_i - x_{i-1}$, and $x_i^* \in [x_{i-1}, x_i]$. x_i^* can be defined in many ways: it could be the left extremity of $\Delta(x_i)$, the right extremity, the midpoint, or many more. In particular, when choosing x_i^* as the midpoint of the interval, I obtain the so called *midpoint rule*; it approximates the integral of a continuous function $f(x)$ by the area under a set of N step functions, with the midpoint of each matching f:

$$\int_a^b f(x)\ dx \approx \frac{b-a}{N} \sum_{i=1}^{N} f(a + (i - 0.5)(b - a)/N)$$

An alternative to the midpoint rule is given by the *trapezoidal rule*, which approximates the area under a continuous function $f(x)$ as a trapezoid and then computes its area:

$$\int_a^b f(x)\ dx \approx (b-a)\left[\frac{f(a) + f(b)}{2}\right]$$

it works best when partitioning the integration area into many subinterval, applying the trapezoidal rule to all of them, and then sum the results:

$$\int_a^b f(x)\ dx \approx \sum_{i=1}^{N} \frac{f(x_{k-1}) + f(x)}{2}\Delta(x_k),$$

with x_k a partition of $[a, b]$ such that $a = x_0 < x_1 < x_2 < \cdots < X_{N-1} < x_N = b$ and $\Delta(x_k) = x_k - x_{k-1}$ the length of the k^{th} subinterval.

Accuracy of the midpoint and trapezoidal rules depends on the number of steps (subintervals) N used to approximate the function, but so does complexity (computationally speaking). The only requirement for applying these rules is that one needs to be able to evaluate the function $f(x)$ at a given point over its domain. If $f(x)$ is cheap to evaluate, than the midpoint and trapezoidal rules may be just fine; otherwise, it would be better to move onto more complicated methods that yield more accurate results.

A first method that is only slightly more complicated but yields better results is the *Simpson's rule*. It works analogously to the midpoint and trapezoidal rule, but using a smooth quadratic interpolant which takes the same values as $f(x)$ at the extremities of the integration interval $[a, b]$ and at the midpoint $m = (a + b)/2$:

$$\int_a^b f(x)\ dx \approx \frac{b-a}{6}\left[f(a) + 4f((a+b)/2) + f(b)\right]$$

Analogously as the trapezoidal rule, it is possible to obtain greater accuracy by splitting the integration interval into many subintervals, applying the Simpson's rule to each subinterval, and sum the results.

Second, it is possible to show that by choosing carefully the points at which to evaluate $f(x)$ and the weights assigned to each point it is possible to obtain an exact approximation of the integral of any polynomial of degree $2N - 1$ or less with N function evaluations (proof in Monahan (2011)). Let $f(x)$ be a function of order $2N - 1$ or less to integrate over a domain $[a, b]$; let $w(x)$ be a weight function. The quadrature formula is defined as:

$$\int_a^b f(x)w(x)\ dx = \sum_{i=1}^{N} w_i f(x_i)$$

Depending on the choice of the weighting function $w(x)$, different Gaussian quadrature rules can be obtained. When $w(x) = 1$, the associated polynomials are Legendre polynomials, the quadrature rule is then named -Gauss-Legendre_ quadrature rule, and it allows integrating over the interval $[-1, 1]$. The integration points are then obtained as the the N roots of the Legendre polynomials: $x = \{x_1, x_2, \ldots, x_N\}$. When choosing the weight function $\exp(-x)$ the associated polynomials are Laguerre polynomials, the quadrature rule is named Gauss-Laguerre quadrature rule, and the integration domain is $[0, +\infty)$. Finally, when choosing the weight function $\exp(-x^2)$ the associated polynomials are Hermite polynomials, the quadrature rule is named Gauss-Hermite quadrature rule, and the integration domain is $(-\infty, +\infty)$. Interestingly, the Gauss-Hermite quadrature can be re-formulated using a normal density kernel with mean μ and standard deviation σ as weighting function:

$$\int_{-\infty}^{+\infty} f(x)\phi(x|\mu, \sigma^2)\ dx = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{+\infty} f(x) \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]\ dx$$

By applying the change of variable $x = \mu + \sigma\sqrt{2}r$, the integral to approximate becomes

$$\int_{-\infty}^{+\infty} f(x)\phi(x|\mu, \sigma^2)\ dx = \frac{\sqrt{2}\sigma}{\sqrt{2\pi}\sigma} \int_{-\infty}^{+\infty} f(\mu + \sigma\sqrt{2}r) \exp(-r^2)\ dr,$$

which can then be approximated by the quadrature rule

$$\frac{\sqrt{2}\sigma}{\sqrt{2\pi}\sigma} \int_{-\infty}^{+\infty} f(\mu + \sigma\sqrt{2}r) \exp(-r^2)\ dr \approx \sum_{i=1}^{N} f(\mu + \sigma\sqrt{2}r)\frac{w_i}{\sqrt{\pi}}.$$

That is, a quadrature rule based on the normal kernel as weight function with nodes $\mu + \sigma\sqrt{2}x_i$ and weights $w_i/\sqrt{\pi}$ (x_i and w_i being the nodes and weights of the corresponding N-points Gauss-Hermite quadrature rule based on the usual weighting function).

A slightly more complicated version of Gaussian quadrature is given by the *Gauss–Kronrod* quadrature formula. In the Gauss-Kronrod quadrature rule the evaluation points are chosen dynamically so that an accurate approximation can be computed by re-using the information produced by the computation of a less accurate approximation. In practice, integration points from previous iterations can be reused as part of the new set of points, whereas usual Gaussian quadrature would require recomputation of all abscissas at each iteration.

This is particularly important when some specified degree of accuracy is needed but the number of points needed to achieve this accuracy is not known ahead of time. Despite this, the quadrature rule is the same as before, i.e. $\int_a^b f(x)\ dx \approx \sum_{i=1}^n w_i f(x_i)$. Gauss-Kronrod quadrature rule is implemented in R as the `integrate()` function.

4.1.2 Multidimensional functions

Finally, all the methods I presented so far only only apply to the integration of unidimensional functions. It is of course possible to extend quadrature rules to multidimensional settings, by recursively applying unidimensional quadrature rules. Say I want to approximate the integral of a bidimensional function $f(x, y)$; the bidimensional Gaussian quadrature rule has the form:

$$\int_X \int_Y f(x, y)\ dx\ dy \approx \sum_j \sum_i w_j w_i f(x_j, y_i)$$

This can be extended to any number of dimensions d, but it gets very computationally expensive very quickly as a N-points rule requires N^d function evaluations.

A better option when the number of dimensions d to integrate over is high is given by *Monte Carlo* integration. Consider integrating a multidimensional function $f(x)$ over some region Ω of volume $V(\Omega)$:

$$I_\Omega = \int_\Omega f(x)\ dx = E[f(U)]V(\Omega),$$

with $U \sim$ uniform over Ω. Drawing N uniform random vectors u_i an estimator for I_Ω is

$$\hat{I}_\Omega = \frac{V(\Omega)}{N} \sum_{i=1}^N f(u_i),$$

and this defines Monte Carlo integration. The variance of the estimated integral $\hat{I}_\Omega$ follows, assuming the u_i are independenent, as $var(\hat{I}_\Omega) = \frac{V(\Omega)^2}{N^2} N var(f(u_i))$. More details in Monahan (2011).

Luckily, both Gaussian quadrature and Monte Carlo integration can be tweaked to improve accuracy and convergence rates: two appealing options are, respectively, adaptive Gaussian quadrature and importance sampling. Adaptive Gaussian quadrature works best when using the Gauss-Hermite rule with the normal density kernel as weighting function; in a multivariate setting, using an iterative algorithm, it is possible to update the mean vector M and variance-covariance matrix Σ of the multivariate normal density at each step (e.g. using empirical Bayes estimates of M, Σ) to better adapt the grid of quadrature points to the actual shape of the integral to approximate. Conversely, Monte Carlo integration works best when it is possible to draw a sample from the target distribution (i.e. the distribution of the integral to approximate); unfortunately, that is rarely the case in practice. The idea of importance sampling consists then in drawing a sample from a proposal distribution and then re-weight the estimated integral using importance weights to better adapt to the target distribution.

4.2 Other considerations

4.2.1 Cancellation error, precision, and arithmetic over- and under-flow

One of the problems when doing calculations on a computer is *cancellation error* (or *round-off error*). That occurs as a side effect of performing finite-precision arithmetic, as computers can store numbers in memory using a finite number of digits. Cancellation error causes the number of significant digits in the result to be reduced unacceptably; when a sequence of calculations is performed, cancellation errors add up significantly, altering the final result. Cancellation error can be easily reproduced:

```
a <- 1e16

b <- 1e16 + pi

b - a
```

```
## [1] 4
```

```
pi
```

```
## [1] 3.141593
```

`b - a` should be π, instead it is 4. Analogously, *arithmetic over- and under-flow* is a condition that happens when the result of a calculation is, respectively, bigger or smaller than the minimum or maximum value that a given machine can store in memory. On the laptop used to produce this report, the smallest (and largest) floating-point number that the machine can represent are:

```
.Machine$double.xmin
```

```
## [1] 2.225074e-308
```

```
.Machine$double.xmax
```

```
## [1] 1.797693e+308
```

Next, precision. Machines can only distinguish numbers that they can represent as different. For instance:

```
a = 1

b = (.Machine$double.eps ^ 2)

c = (.Machine$double.neg.eps ^ 2)

d = a + b

e = a - c


# The following equalities should be FALSE
```

```
a == d
```

```
## [1] TRUE
```

```
a == e
```

```
## [1] TRUE
```

```
# Check that b, c are not 0
b == 0
```

```
## [1] FALSE
```

```
c == 0
```

```
## [1] FALSE
```

In this case, `.Machine$double.eps` and `.Machine$double.neg.eps` are the smallest positive floating-point numbers x such that $1 + x \neq 1$ and $1 - x \neq 1$, respectively.

It is necessary to keep this potential problems in mind when doing numerical calculation using finite-precision arithmetic; for instance, a common situation where we may incur in arithmetic over- or under-flow is when maximising a likelihood. That is, the product of the individual contributions to the likelihood may be a number so large (or so small) that the computer cannot distinguish it from $\pm\infty$, or rounding error may seriously affect the results. This specific example is easy to fix by using the log-likelihood instead, as the sum of the logarithm of the individual contributions behaves much better; nevertheless, this problem may not be always evident nor as easy to solve.

4.2.2 Numerical differentiation

Numerical differentiation is a series of algorithms to numerically estimate the derivative of a function. They tend to be computationally less demanding than numerical integration methods, but they are more sensitive to cancellation error.

The easiest method for approximating the derivative of a function is to use finite difference approximation. Say I want to estimate the first derivative of a function $f(\cdot)$ at x; the finite difference approximation of the derivative $f'(x)$ is calculated as

$$f'(x) \approx \frac{f(x+h) - f(x)}{h},$$

for a small h. This formula is affected by both truncation error (as it derives from a truncated Taylor series expansion of $f(x)$) and cancellation error (as a machine works with finite-precision arithmetic). It is necessary to choose a value h that gives a good balance between the two errors: it can be showed that a good choice in most cases is $h = \sqrt{\epsilon}$, with ϵ being the machine precision.

The formula I presented for finite difference approximation is also known as *forward differencing*; alternatively, it is possible to use methods such as *central differencing* ($[f(x+h) - f(x-h)]/2h$, more accurate but more computationally expensive) and *backward differencing* ($[f(x) - f(x-h)]/h$). Other methods are the *complex method*, which requires the function to be able to handle complex values and it is extremely powerful but with limited applicability, and the *Richardson's extrapolation method*, which is more accurate but slower than finite differencing. All these methods are implemented in R in the `numDeriv` package, which sets the standard for numerical differentiation.

4.2.3 Numerical root finding

Root-finding algorithms are algorithms for finding the values x such that $f(x) = 0$, for a given continuous function $f(\cdot)$. Such values x are named roots (or zeros) of a function. Most root-finding algorithms are based on the intermediate value theorem, which states that if a continuous function has values of opposite sign at the end points of an interval then the function has at least one root in the interval.

For instance, the easiest root-finding method is the *bisection method*: let $f(x)$ be a continuous function, for which one knows an interval $[a, b]$ such that $f(a)$ and $f(b)$ have opposite sign. Let $c = (a + b)/2$ be the midpoint the bisect the interval: now, either $f(a)$ and $f(c)$ or $f(c)$ and $f(b)$ have opposite sign, and one has in fact divided by two the size of the interval. One can iterate this method until the difference between the extremities of the interval is small enough (e.g. $< 1 \times 10^{-8}$).

Another well established method is the *secant method*: it uses a succession of roots of secant lines to approximate the root of a function $f(x)$. Starting with values x_0 and x_1, a line is constructed between $(x_0, f(x_0))$ and $(x_1, f(x_1))$:

$$y = \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_1) + f(x_1)$$

The root of this line is

$$x = x_1 - f(x_1)\frac{x_1 - x_0}{f(x_1) - f(x_0)}$$

Now, we set $x_2 = x$ and we iterate this method until the difference between the extremities of the interval is small enough (e.g. $< 1 \times 10^{-8}$).

The secant method is also known as a *linear interpolation* method; it is also possible to use higher order interpolation, specifically *quadratic interpolation*, to find the root of a function using the same rationale presented for the secant method. Specifically, starting with three starting values x_0, x_1, x_2 and their function values $f(x_0)$, $f(x_1)$, $f(x_2)$, applying the Lagrange interpolation formula to interpolate the inverse of $f(x)$ yields the equation

$$f^{-1}(y) = \frac{(y - f(x_1))(y - f(x_2))}{(f(x_0) - f(x_1))(f(x_0) - f(x_2))}x_0 + \frac{(y - f(x_0))(y - f(x_2))}{(f(x_1) - f(x_0))(f(x_1) - f(x_2))}x_1 + \frac{(y - f(x_0))(y - f(x_1))}{(f(x_2) - f(x_0))(f(x_2) - f(x_1))}x_2$$

Substituting $y = 0$ in the above equation yields the recursion formula, to be iterated until a desired precision is reached.

Finally, a well-established and robust method is the *Brent-Dekker* method, implemented in R with the `uniroot()` function. In combined the three methods presented before, trying to use the secant or quadratic interpolation method first - as they tend to converge faster to a solution - but falling back to the bisection method if necessary, for its robustness properties. More details on the Brent-Dekker method in Brent (1973).

Chapter 5

Simulating survival data

In this Chapter I will present a flexible and efficient method to simulate survival data from a variety of parametric distributions, first introduced by Bender et al. (2005). Then, I will present an extension that allows simulating from a variety of complex distributions proposed by Crowther and Lambert (2013).

Let $h(t) = h_0(t)\exp(X\beta)$ be the hazard function of a proportional hazards model, with $h_0(t)$ baseline hazard function and X a matrix of covariates with regression coefficients β. Let $H_0(t) = H_0(t)\exp(X\beta)$ be the corresponding cumulative hazard function, with $H_0(t) = \int_0^t h_0(u)\ du$. The survival function $S(t)$ and cumulative distribution function $F(t)$ follow naturally: $S(t) = \exp(-H(t))$ and $F(t) = 1 - S(t) = 1 - \exp(X\beta)$.

Bender et al. (2005) showed that by letting

$$F(\tau) = u,\ u \sim U(0, 1)$$

and denoting the simulated survival time with τ, it is possible to derive τ analytically by inverting $H_0(t)$ if $h_0(\tau) > 0$:

$$\tau = H_0^{-1}(-\log(u)\exp(X\beta))$$

The only requirement is for $H_0(t)$ to be directly invertible, which happens to be the case when simulating from an exponential, Weibull, or Gompertz distribution for the baseline hazard $h_0(t)$. The algorithm for simulating m survival times is as follows:

1. draw a vector u of m observations from a $U(0, 1)$ distribution;
2. simulate X (e.g. a binary treatment from a Bernoulli distribution) and fix β;
3. the survival times can be obtained directly by applying the formula $H_0^{-1}(-\log(u)\exp(X\beta))$.

Bender et al. (2005) derived the closed-form version of $H_0^{-1}(t)$ for the exponential, Weibull, and Gompertz distributions, presented in Table 5.1.

Table 5.1: Closed-form formulas for simulating survival data from an exponential, Weibull, or Gompertz distribution.

	Exponential	Weibull	Gompertz
Hazard function $h_0(t)$	λ	$\lambda p t^{p-1}$	$\exp(\gamma t)$
Cumulative hazard function $H_0(t)$	λt	λt^p	$(\lambda/\gamma)(\exp(\gamma t) - 1)$
Inverse cumulative hazard function $H_0^{-1}(t)$	$\lambda^{-1} t$	$(\lambda^{-1} t)^{1/p}$	$(1/\gamma)\log((\gamma/\lambda)t + 1)$
Survival time τ	$-\frac{\log(u)}{\lambda\exp(X\beta)}$	$\left[-\frac{\log(u)}{\lambda\exp(X\beta)}\right]^{1/p}$	$(1/\gamma)\log\left[1 - \frac{\gamma\log(u)}{\lambda\exp(X\beta)}\right]$

The requirement requirement for $H_0(t)$ to be directly invertible impedes the use of distributions other than the exponential, Weibull, or Gompertz - which could be appropriate in some setting but too restricting in others. Crowther and Lambert (2013) generalised this method in order to accommodate more complex distributions, even with turning points, under a proportional hazards model. In brief, when the cumulative baseline hazard function is not invertible it is not possible to solve the equation $F(\tau) = u$ for τ; assuming it is possible to write $H_0(t)$ analytically - a broader assumption compared to assuming $H_0(t)$ is invertible - it is possible to use root-finding methods to solve for τ numerically (Section 4.2.3). Formally, the survival time τ can be simulated as the root of the equation $S(\tau) - u = 0$.

Crowther and Lambert (2013) present an example of their method by simulating from a two-components mixture distribution (McLachlan and McGiffin, 1994), defined by additive components on the survival scale:

$$S_0(t) = \pi S_1(t) + (1 - \pi) S_2(t),$$

with $\pi \in [0, 1]$ mixing parameter. $S_i(t)$ can be any standard parametric distribution. Choosing two Weibull components for the mixture distribution, it can be showed that a proportional hazards model has the form

$$h(t) = \frac{\lambda_1 p_1 t^{p_1 - 1} \pi \exp(-\lambda_1 t^{p_1}) + \lambda_2 p_2 t^{p_2 - 1} (1 - \pi) \exp(-\lambda_2 t^{p_2})}{\pi \exp(-\lambda_1 t^{p_1}) + (1 - \pi) \exp(-\lambda_2 t^{p_2})} \exp(X\beta);$$

the survival function can be obtained directly from $h(t)$ in closed-form, plugged into the equation $S(\tau) - u = 0$, and numerically solved for τ.

Chapter 6

Simulation study: accuracy of Gaussian quadrature

6.1 Aim

6.2 Data-generating mechanisms

6.3 Methods

6.4 Estimands

6.5 Performance measures

6.6 Results

Chapter 7

Simulation study: impact of misspecification in survival models with shared frailty terms

7.1 Aim

7.2 Data-generating mechanisms

7.3 Methods

7.4 Estimands

7.5 Performance measures

7.6 Results

Chapter 8

Exploring results from simulation studies interactively

Chapter 9

Informative visiting process

Chapter 10

Future research developments

Chapter 11

Personal development

In this chapter I will introduce and briefly discuss the personal development activities I carried out during the first year of my PhD. In particular, I will present the supervisory meetings, training courses, and conferences I attended.

11.1 Supervisory meetings

I have been having frequent meetings with my supervisors, formally and informally. Formal supervisory meetings, recorded on PROSE (https://prose.le.ac.uk), have been held on average every other week, with summaries produced and shared between us. A comprehensive list is available on PROSE. Additionally, we held informal meetings to discuss developments and more urgent matters more often, whenever it was needed and without scheduling it.

11.2 Training and courses

I have attended a wide variety of courses during my first year, both externally and internally to the University of Leicester. The external courses I attended are:

- *Efficient R Programming*, on November 8th 2016, organised by the Royal Statistical Society in London. The instructor was Dr. Colin Gillespie, from the University of Newcastle, United Kingdom, and Jumping Rivers. The course covered how to program efficiently with R; in particular, it covered common pitfalls when writing R code, code profiling, RCpp, and parallel programming. General hints and tips were provided.

- *Introduction to causal inference*, on April 25th and 26th 2017, organised by the Biostatistics Research Group at the University of Leicester and delivered by Dr. Arvid Sjölander from Karolinska Institutet, Stockholm, Sweden. The course provided foundational concepts of causal inference such as the difference between association and causation, the counterfactual framework, exchangeability, directed acyclic graphs, methods for estimating a causal effect, etc. Additionally, it provided an introduction to more advanced methods such as intrumental variables and Mendelian randomisation.

- *Using simulation studies to evaluate statistical methods*, on May 22nd 2017, organised by University College London. The course was delivered by Dr. Tim Morris, Prof. Ian White and Dr. Michael Crowther, and it covered the rationale for using simulation studies, important concepts to keep in mind when planning a simulation study, computational tools, estimates of uncertainty, and tools for improving reporting and dissemination.

- Workshop on *Joint modelling of longitudinal and time-to-event data with R*, on July 5th, 2017, organised by the Department of Biostatistics of the University of Liverpool. The course was delivered by Dr. Graeme Hickey, and provided an introduction to joint models of longitudinal and survival data, including extensions to incorporate competing risks and multiple longitudinal processes and a practical session using R.

I have attended a few courses within the University and not offered on PROSE; specifically, I attended a course on *Time series analysis with R* (November 10th, 2016), a course on *Data visualisation* (November 15th, 2016), and a course on *High performance computing at*

Leicester (February 8th, 2017). The latter was particularly important, as it allowed me to make better use of the high-performance computing facilities offered by the University. I also attended the *Preparing to teach in higher education* workshop, strand A (July 24th and 27th 2017).

Additionally, I have attended the following PROSE training sessions to develop personal and communication skills in research settings. These are listed below:

- *Planning your literature search*, October 21st 2016;
- *Conducting your literature search*, October 25th 2016 ;
- *Assertiveness*, November 14th 2016;
- *Introduction to critical thinking*, December 15th 2016;
- *Presentations A: Fundamentals of an effective presentation*, January 30th 2017;
- *Communication in research and other work settings*, January 31st 2017;
- *Enhancing your digital profile*, February 2nd 2017;
- *Saying it with your abstract*, February 10th 2017;
- *Designing a poster*, February 27th 2017;
- *Leadership in research and other work environments*, February 28th 2017;
- *Preparing for the probation review (Physical natural and medical sciences)*, May 30th 2017.

11.3 Conferences

I have attended a number of conferences during this year, in which I delivered the following oral presentations:

- Survival Analysis for Junior Researchers conference, held in Leicester, UK, on April 5th and 6th 2017. I delivered a talk titled *Direct likelihood maximisation using numerical quadrature to approximate intractable terms*;
- Statistical Analysis of Multi-Outcome Data (SAM) conference, held in Liverpool, UK, on July 3rd and 4th 2017. I delivered a talk titled *Impact of model misspecification in survival models with frailties*;
- Annual Conference of the International Society for Clinical Biostatistics conference, held in Vigo, Spain, on July 9th to July 13th 2017. I delivered two talks: a titled *Impact of model misspecification in survival models with frailties* during the main conference, and a talk titled *Exploring results from simulation studies interactively* during the Students' Day organised on July 13th.

Additionally, I delivered an oral presentation on previous work external to my PhD project during the 54th ERA-EDTA Congress held in Madrid, Spain, between June 3rd and June 6th. The ERA-EDTA Congress is the main conference in the field of Nephrology in Europe, with approximately 10,000 participants in 2017. I delivered my presentation, titled *Inappropriate prescription of nephrotoxic drugs to individuals with chronic kidney disease*, to an audience of clinicians, epidemiologists, clinical researchers, and other stakeholders.

Appendix A

Slides

Appendix B

Manuscript

Appendix C

R Session

```
sessionInfo()
```

```
## R version 3.4.1 (2017-06-30)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 7 x64 (build 7601) Service Pack 1
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United Kingdom.1252
## [2] LC_CTYPE=English_United Kingdom.1252
## [3] LC_MONETARY=English_United Kingdom.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United Kingdom.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
```

```
##
## other attached packages:
##  [1] bookdown_0.4    marqLevAlg_1.1  readr_1.1.1     ggfortify_0.4.1
##  [5] dplyr_0.7.2     cowplot_0.8.0   survival_2.41-3 ggplot2_2.2.1
##  [9] pacman_0.4.6    knitr_1.17
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.12     highr_0.6        compiler_3.4.1   plyr_1.8.4
##  [5] bindr_0.1        tools_3.4.1      digest_0.6.12    evaluate_0.10.1
##  [9] tibble_1.3.3     gtable_0.2.0     lattice_0.20-35  pkgconfig_2.0.1
## [13] rlang_0.1.2      Matrix_1.2-10    rstudioapi_0.6   yaml_2.1.14
## [17] bindrcpp_0.2     gridExtra_2.2.1  stringr_1.2.0    hms_0.3
## [21] rprojroot_1.2    grid_3.4.1       glue_1.1.1       R6_2.2.2
## [25] rmarkdown_1.6    tidyr_0.6.3      magrittr_1.5     backports_1.1.0
## [29] scales_0.4.1     htmltools_0.3.6  splines_3.4.1    assertthat_0.2.0
## [33] colorspace_1.3-2 labeling_0.3     stringi_1.1.5    lazyeval_0.2.0
## [37] munsell_0.4.3
```

Bibliography

Bender, R., Augustin, T., and Blettner, M. (2005). Generating survival times to simulate cox proportional hazards models. *Statistics in Medicine*, 24:1713–1723.

Brent, R. P. (1973). *Algorithms for minimization without derivatives.* Prentice-Hall.

Breslow, N. (1974). Covariance analysis of censored survival data. *Biometrics*, 30(1):89–99.

Clayton, D. G. (1991). A Monte Carlo method for Bayesian inference in frailty models. *Biometrics*, 47(2):467.

Commenges, D., Jacqmin-Gadda, H., Proust, C., and Guedj, J. (2006). A newton-like algorithm for likelihood maximization: the robust-variance scoring algorithm. *ArXiv Mathematics e-prints.*

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

Cox, D. R. (1975). Partial likelihood. *Biometrika*, 62(2):269–276.

Crowder, M. (2016). *Multivariate survival analysis and competing risks.* Text in Statistical Sciences. Chapman & Hall / CRC.

Crowther, M. J., Abrams, K. R., and Lambert, P. C. (2012). Flexible parametric joint modelling of longitudinal and survival data. *Statistics in Medicine*, 31(30):4456–4471.

Crowther, M. J. and Lambert, P. C. (2013). Simulating biologically plausible complex survival data. *Statistics in Medicine*, 32(23):4118–4134.

Crowther, M. J., Look, M. P., and Riley, R. D. (2014). Multilevel mixed effects parametric survival models using adaptive gauss-hermite quadrature with application to recurrent events and individual participant data meta-analysis. *Statistics in Medicine*, 33(22):3844–3858.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 39(1):1–38.

Diggle, P. J., Heagerty, P., Liang, K., and Zeger, S. L. (2013). *Analysis of longitudinal data*. Oxford Statistical Science Series. OUP Oxford, 2 edition.

Efron, B. (1977). The efficiency of Cox's likelihood function for censored data. *Journal of the American Statistical Association*, 72(359):557–565.

Geskus, R. B. (2015). *Data analysis with competing risks and intermediate states*. The Biostatistics Series. Chapman & Hall / CRC.

Gould, A. L., Boye, M. E., Crowther, M. J., Ibrahim, J. G., Quartey, G., Micallef, S., and Bois, F. Y. (2015). Joint modeling of survival and longitudinal non-survival data: current methods and issues. report of the DIA Bayesian joint modeling working group. *Statistics in Medicine*, 34(14):2181–2195.

Gutierrez, R. G. (2002). Parametric frailty and shared frailty survival models. *The Stata Journal*, 2(1):22 – 44.

Henderson, R., Diggle, P., and Dobson, A. (2000). Joint modelling of longitudinal measurements and event time data. *Biostatistics*, 1(4):465–480.

Hickey, G. L., Philipson, P., Jorgensen, A., and Kolamunnage-Dona, R. (2016). Joint modelling of time-to-event and multivariate longitudinal outcomes: recent developments and issues. *BMC Medical Research Methodology*, 16(1).

Hogan, J. W. and Laird, N. M. (1998). Increasing efficiency from censored survival data by using random effects to model longitudinal covariates. *Statistical Methods in Medical Research*, 7(1):28–48.

Hougaard, P. (2000). *Analysis of multivariate survival data.* Springer New York.

Hsieh, F., Tseng, Y., and Wang, J. (2006). Joint modeling of survival and longitudinal data: likelihood approach revisited. *Biometrics*, 62:1037–1043.

Ibrahim, J. G., Chu, H., and Chen, L. M. (2010). Basic concepts and methods for joint models of longitudinal and survival data. *Journal of Clinical Oncology*, 28(16):2796–2801.

International Stroke Trial Collaborative Group (1997). The international stroke trial (ist): a randomised trial of aspirin, subcutaneous heparin, both, or neither among 19 435 patients with acute ischaemic stroke. *The Lancet*, 349(9065):1569–1581.

Kalbfleisch, J. D. and Prentice, R. L. (2011). *The statistical analysis of failure time data.* Wiley Series in Probability and Statistics. John Wiley & Sons, Inc, 2 edition.

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

Kleinbaum, D. G. and Klein, M. (2012). *Survival analysis: A self-learning text.* Springer-Verlag New York, 3 edition.

Liu, X., Pawitan, Y., and Clements, M. (2016). Parametric and penalized generalized survival models. *Statistical Methods in Medical Research.*

Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441.

McLachlan, G. and McGiffin, D. (1994). On the role of finite mixture models in survival analysis. *Statistical Methods in Medical Research*, 3(3):221–226.

Monahan, J. F. (2011). *Numerical methods of statistics*. Statistical and Probabilistic Mathematics. Cambridge University Press, 2 edition.

Pawitan, Y. and Self, S. (1993). Modeling disease marker processes in aids. *Journal of the American Statistical Association*, 88(423):719–726.

Peto, R. (1972). Contribution to discussion of paper by D.R. Cox. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 34:205–207.

Ripatti, S. and Palmgren, J. (2000). Estimation of multivariate frailty models using penalized partial likelihood. *Biometrics*, 56(4):1016–1022.

Rizopoulos, D. (2012). *Joint models for longitudinal and time-to-event data: with applications in R*. Biostatistics. Chapman & Hall / CRC.

Rizopoulos, D., Hatfield, L. A., Carlin, B. P., and Takkenberg, J. J. (2014). Combining dynamic predictions from joint models for longitudinal and time-to-event data using bayesian model averaging. *Journal of the American Statistical Association*, 109(508):1385–1397.

Sandercock, P., Niewada, M., and Czlonkowska, A. (2011). International stroke trial database (version 2). Technical report, University of Edinburgh, Department of Clinical Neuroscience.

Sinha, D., Maiti, T., Ibrahim, J. G., and Ouyang, B. (2008). Current methods for recurrent events data with dependent termination. *Journal of the American Statistical Association*, 103(482):866–878.

Sweeting, M. J. and Thompson, S. G. (2011). Joint modelling of longitudinal and time-to-event data with application to predicting abdominal aortic aneurysm growth and rupture. *Biometrical Journal*, 53(5):750–763.

Therneau, T. M., Grambsch, P. M., and Pankratz, V. S. (2003). Penalized survival models and frailty. *Journal of Computational and Graphical Statistics*, 12(1):156–175.

Tsiatis, A. A. and Davidian, M. (2001). A semiparametric estimator for the proportional hazards model with longitudinal covariates measured with error. *Biometrika*, 88(2):447–458.

Tsiatis, A. A. and Davidian, M. (2004). Joint modeling of longitudinal and time-to-event data: an overview. *Statistica Sinica*, 14:809–834.

Tsiatis, A. A., Degruttola, V., and Wulfsohn, M. S. (1995). Modeling the relationship of survival to longitudinal data measured with error. applications to survival and CD4 counts in patients with AIDS. *Journal of the American Statistical Association*, 90(429):27–37.

Wienke, A. (2010). *Frailty models in survival analysis*. Chapman & Hall / CRC.

Wulfsohn, M. S. and Tsiatis, A. A. (1997). A joint model for survival and longitudinal data measured with error. *Biometrics*, 53(1):330–339.

Xie, Y. (2016). *bookdown: Authoring books and technical documents with R Markdown*. The R Series. Chapman & Hall / CRC.