Survival, mortality, expected lifetime and competing risks

Computer practicals

EDEG 2025 / Umeå University
17 May 2025
http://bendixcarstensen.com/AdvCoh/courses/Um-2025/
Version 1, November 2024

Compiled Friday 22nd November, 2024, 13:08
from: C:\Bendix\teach\AdvCoh\courses\Um-2025\pracs/pracs.tex

Bendix Carstensen Steno Diabetes Center Copenhagen, Herlev, Denmark
& Department of Biostatistics, University of Copenhagen
bendix.carstensen@regionh.dk
bcar0029@regionh.dk b@bxc.dk
http://BendixCarstensen.com

Contents

0.1 Preface

This workshop will provide an overview of the concepts in the title with a special view to generating results (numbers and graphs) using R.

By the title of the workshop it will (hopefully) be relevant for persons that are working with follow-up data over time, be that clinical trials, cohort studies or register-based studies.

- The *target audience* is (young) statisticians and epidemiologists working in (diabetes) epidemiological research
- The *prerequisites* are
    1. a basic knowledge of R,
    2. a working installation of R(latest version, 4.4.1)
    3. a working installation of the latest version of the `Epi` package (2.57)
    4. a working installation of the latest version of the `popEpi` package (0.4.12)
    5. some epidemiological practice
    6. a basic knowledge of elementary biostatistics (you should know what a confidence interval is)
- The *format* of the workshop will be short lectures closely aligned with the topics in the exercises. The exercises will be run in chunks between the short lectures.
- The *mood* of the workshop will be relaxed, encouraging participants to ask questions and bring forward problems they consider relevant for the workshop. Fortunately, there will be the rest of the EDEG to interact.

Exercises are given including substantial parts of the solutions. You can get the exercise code chunks from the workshop website
http://bendixcarstensen.com/AdvCoh/courses/Um-2025

This workshop draws to some extent on the content of the book "Epidemiology with R" [**?**], (http://bendixcarstensen.com/EwR), but in particular on the draft book (which by no means is sure ever to appear as a book) "Practical multistate modeling with R and `Epi:Lexis`". The former is available through Oxford University Press, the latter as a draft (updated at unpredictable times) as http://bendixcarstensen.com/MSbook.pdf (200+ pages...).

Program of workshop

Each item on the program is a short(ish) lecture followed by a computer practical in R. The timing of the items is approximate.

The purpose of the workshop is to provide a hands-on experience of computing some the most common quantities in follow-up studies (and some not so common).

Saturday 17 May 2025	
09:00–09:10	Welcome and introdiction
09:10–10:00	Survival and mortality rates Kaplan-Meier survival, mortality function, parametric survival function
10:00–11:00	Expected lifetime (RMST)
11:10–11:40	Cause specific rates
11:40–12:20	Lunch
12:20–13:45	Competing risks
13:45–14:00	Wrap-up and questions

Chapter 1

Survival and mortality

```
> options(width = 90,
+         show.signif.stars = FALSE)
> par(mar = c(3, 3, 1, 1),
+     mgp = c(3, 1, 0) / 1.6,
+     las = 1,
+    lend = "butt",
+     bty = "n")
```

```
> library(Epi)
> library(popEpi)
> library(survival)
> clear()
```

```
> # just until 2.57 comes out
> glmLexis <- glm.Lexis
```

1.1 Survival

Survival analyses is about evaluation of variables that influence the survival of persons. Most likely age or disease duration.

As it happens, the data that underlies survival analysis will normally be derived from a set of dates from registers, trials or cohort studies. For example date of diabetes, date of death and date of last follow-up (not all persons die in the study period).

So what we see is *how long* persons have been at risk of dying, and if they die outcome: (d, y) of event (death, $d \in \{0, 1\}$), and time at risk (y).

1.2 Diabetes data

Now take a look at the date set `DMlate`, a random sample of diabetes patients diagnosed between 1995 and 2010 (incl.)

```
> data(DMlate)
> hist(DMlate$dodm, breaks = seq(1995, 2010, 1/4), col = 1)
> abline(v = 1995:2010, col = "red")
```

For convenience take a random subsample of 1000:

```
> set.seed(19540803)
> DMlate <- DMlate[sample(1:nrow(DMlate), 1000), ]
> str(DMlate)
'data.frame':        1000 obs. of  7 variables:
 $ sex  : Factor w/ 2 levels "M","F": 2 1 1 2 1 1 1 2 1 2 ...
 $ dobth: num  1969 1960 1923 1922 1953 ...
 $ dodm : num  2003 2005 2007 1998 2008 ...
 $ dodth: num  NA NA NA 1999 2009 ...
 $ dooad: num  NA 2005 2007 NA NA ...
 $ doins: num  2003 NA NA NA NA ...
 $ dox  : num  2010 2010 2010 1999 2009 ...
```

The dates in `DMlate` are coded as fractional years, so for example 1 January 2005 is coded 2005.0 and 1 July 2007 is coded 2007.495. It's more convenient to have person-time in units of years, and located at an understandable place on the time axis.

1.3 Survival probability

If we want to see how diabetes patients survived after diagnosis of diabetes we need the survival time (risk time), `dox` − `dodm` and the indicator of whether a person died at exit, `!is.na(dodth)`.

From this we will want the *probability* that a person survives a given time t as a function of t, say:

$$S(t) = \mathrm{P}\{\text{alive at time } t\}$$

We can directly estimate the survival as a function of time since diagnosis of diabetes. Note that the term "survival" only makes sense if we define an `origin`, and we then measure the risk time as the time since that origin. So here we defined the origin as the date of diabetes diagnosis, and t means "time since the origin".

The survival funtion can be estimated by the Kaplan-Meier estimator using `survfit` from the survival package. Here we use the time at diagnosis of diabetes as the origin:

```
> sf <- survfit(Surv(dox - dodm, !is.na(dodth)) ~ 1, data = DMlate)
> sf
Call: survfit(formula = Surv(dox - dodm, !is.na(dodth)) ~ 1, data = DMlate)

        n events median 0.95LCL 0.95UCL
[1,] 1000    231     NA      NA      NA
> plot(sf)
> abline(h = 0.5, lty = 3)
```

The curve is a step-curve, although it would seem a more natural assumption that $S(t)$ was a smooth curve.

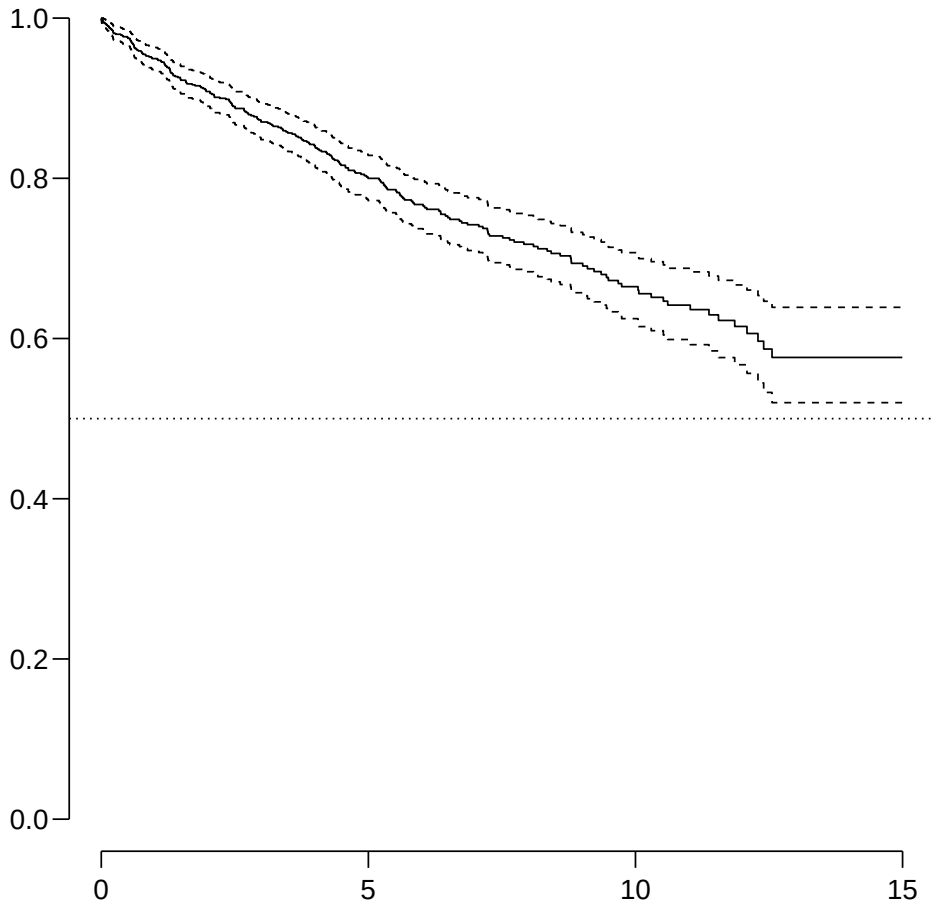

Figure 1.1: *Kaplan-Meier estimator of survival function for Danish diabetes patients.*
`../graph/surv0-KM`

1.4 Mortality

The data we get from a register, cohort or clinical trial is really mortality data: events (numerator) and risk time (denominator), so it would seem more natural to start with estimation of mortality rates:

$$\lambda(t) = \lim_{h \to 0} \mathrm{P}\{\text{death in } (t, t+h) | \text{alive at } t\}/h$$

It is a bit more complicated to estimate this because it requires statistical modeling of the effect of t on the mortality rate. But let's begin with the simplest case:

1.4.1 Constant mortality

If we make the assumption that the mortality is constant, the estimator of the mortality $\lambda(t) = \lambda$ is just the ratio of the number of deaths to the amount of risk time:

```
> (mort <- with(DMlate, sum(!is.na(dodth)) / sum(dox - dodm)))
[1] 0.04279325
```

in this case measured in cases per year, or 4.6 % per year. The relationship between survival since some origin and the mortality rate is

$$S(t) = \exp\Big(-\int_0^t \lambda(u)\,\mathrm{d}u\Big)$$

which, if $\lambda(u) = \lambda$ becomes

$$S(t) = \exp(-\lambda t)$$

so we can put this on top of the Kaplan-Meier curve:

```
> plot(sf)
> abline(h = 0.5, col = "gray")
> t = seq(0, 15, 0.1)
> lines(t, exp(-mort * t), col = "red", lwd =2)
```

We see that it is not a good approximation to the KM estimator to assume constant mortality, we obviously need to allow mortality to depend on time.

1.4.2 Time-dependent mortality

Note that when we allow mortality to depend on time, we are using time (that is time since diagnosis) as an explanatory variable.

1.4.2.1 Subdividing follow-up

This can be modeled by dividing the timescale in very small intervals, so that the assunption of constant mortality *within* each interval is tenable. And then assume that mortality varies smoothly *between* the intervals.

A simple way of doing this is to use the `Lexis` machinery in the `Epi` package:

```
> Lx <- Lexis(exit = list(tfd = dox - dodm),
+      exit.status = factor(!is.na(dodth),
+                           labels = c("DM", "Dead")),
+             data = DMlate)
NOTE: entry.status has been set to "DM" for all.
NOTE: entry is assumed to be 0 on the tfd timescale.
```

CODE EXPLAINED: The `Lexis` function takes the input data set (here `DMlate`) and adds information in the form of a few variables and attributes. In the `Epi` package is a vignette explaining all the features in detail, try `vignette(package = "Epi")`.

The `exit` argument defines the time of exit from the study on some time scale, in this case time from diabetes, `tfd`, assuming that entry is at 0 on this timescale.

`exit.status` defines the status of each person at exit, in this case `DM` if no date of death (`dodth`) is available otherwise `Dead`. The `labels=` refer to the order of the values of `!is.na(dodth)` where R defines `FALSE` < `TRUE`.

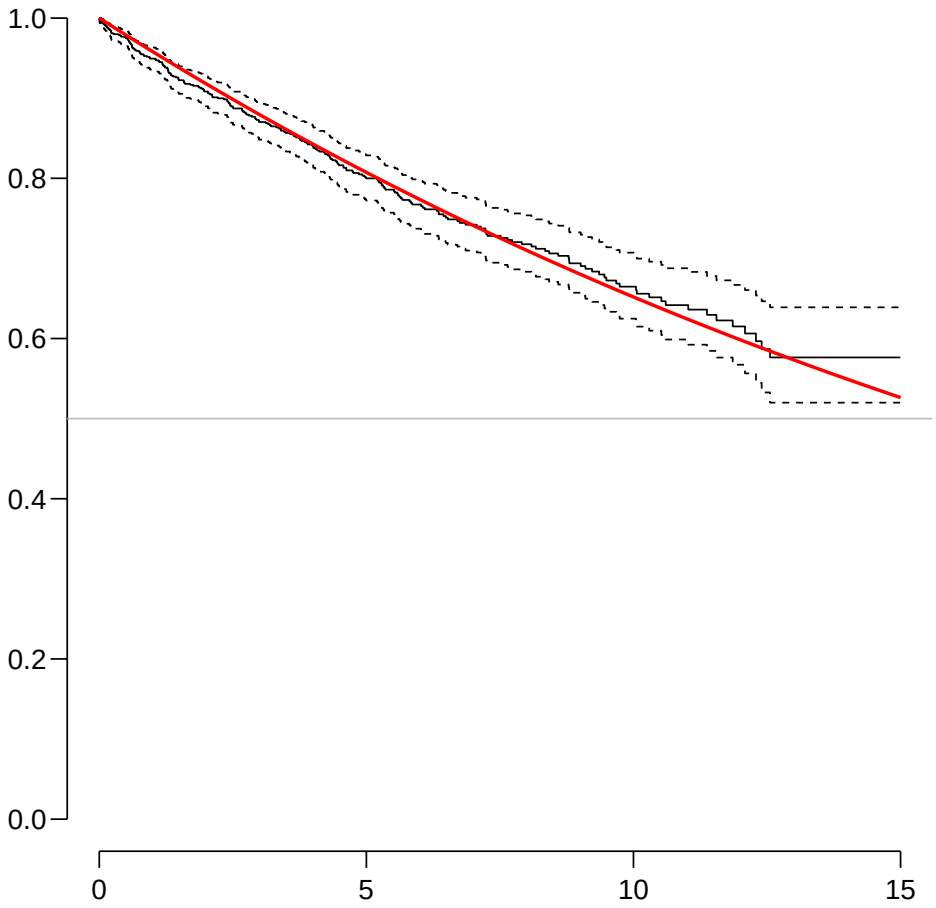

Figure 1.2: *Kaplan-Meier estimator of survival of Danish diabetes patients. The red curve is from the model with constant mortality (exponentially distributed survival times).*

../graph/surv0-KMexp

`Lexis` is talkative and tells about the assumptions made.

We can get a summary of the status and follow-up in the `Lexis` object:

```
> summary(Lx)
Transitions:
     To
From  DM Dead  Records:  Events: Risk time:  Persons:
  DM 769  231      1000      231    5398.05      1000
```

and get a nice print of the 5 first persons

```
> print(Lx[1:5,], nd = 2)
 lex.id tfd lex.dur lex.Cst lex.Xst sex   dobth    dodm   dodth   dooad   doins     dox
      1   0    7.08      DM      DM   F 1969.22 2002.92      NA      NA 2003.01 2010.00
      2   0    4.79      DM      DM   M 1959.55 2005.21      NA 2005.22      NA 2010.00
      3   0    3.00      DM      DM   M 1923.07 2007.00      NA 2007.00      NA 2010.00
```

```
      4   0    1.05      DM    Dead   F 1921.69 1998.24 1999.29      NA      NA 1999.29
      5   0    0.89      DM    Dead   M 1952.51 2008.12 2009.01      NA      NA 2009.01
```

The follow-up is still with one record per person, the time at the beginning of the interval is `tfd = 0` for all persons, and the length of the follow-up intervals is in `lex.dur`:

```
> sL <- splitLexis(Lx, seq(0, 15, 0.5))
> subset(Lx, lex.id %in% c(2,4))
 lex.id tfd lex.dur lex.Cst lex.Xst sex   dobth    dodm   dodth   dooad doins     dox
      2   0    4.79      DM      DM   M 1959.55 2005.21      NA 2005.22    NA 2010.00
      4   0    1.05      DM    Dead   F 1921.69 1998.24 1999.29      NA    NA 1999.29
> subset(sL, lex.id %in% c(2,4))
 lex.id tfd lex.dur lex.Cst lex.Xst sex   dobth    dodm   dodth   dooad doins     dox
      2 0.0    0.50      DM      DM   M 1959.55 2005.21      NA 2005.22    NA 2010.00
      2 0.5    0.50      DM      DM   M 1959.55 2005.21      NA 2005.22    NA 2010.00
      2 1.0    0.50      DM      DM   M 1959.55 2005.21      NA 2005.22    NA 2010.00
      2 1.5    0.50      DM      DM   M 1959.55 2005.21      NA 2005.22    NA 2010.00
      2 2.0    0.50      DM      DM   M 1959.55 2005.21      NA 2005.22    NA 2010.00
      2 2.5    0.50      DM      DM   M 1959.55 2005.21      NA 2005.22    NA 2010.00
      2 3.0    0.50      DM      DM   M 1959.55 2005.21      NA 2005.22    NA 2010.00
      2 3.5    0.50      DM      DM   M 1959.55 2005.21      NA 2005.22    NA 2010.00
      2 4.0    0.50      DM      DM   M 1959.55 2005.21      NA 2005.22    NA 2010.00
      2 4.5    0.29      DM      DM   M 1959.55 2005.21      NA 2005.22    NA 2010.00
      4 0.0    0.50      DM      DM   F 1921.69 1998.24 1999.29      NA    NA 1999.29
      4 0.5    0.50      DM      DM   F 1921.69 1998.24 1999.29      NA    NA 1999.29
      4 1.0    0.05      DM    Dead   F 1921.69 1998.24 1999.29      NA    NA 1999.29
```

We see that we now have more intervals for each person, but the total follow-up time for each person (in `lex.dur`) is the same. Also the number of deaths (`lex.Xst==TRUE`) is the same for each person (namely 0 for person 2, and 1 for person 4).

1.4.2.2 Modeling mortality

We can now model the mortality rates as a function of the timescale `tfd`, that is the time where the follow-up is. This is done by a spline function:

```
> m0 <- glmLexis(sL, ~ Ns(tfd, knots = c(0,1,2,4,7,10)))
stats::glm Poisson analysis of Lexis object sL with log link:
Rates for the transition:
DM->Dead
```

We now have a model for the mortality rates as a function of `tfd`, so we can make a *prediction* of the mortality at any set of values of `tfd`. `ci.pred` also produces confidence intervals:

```
> nd <- data.frame(tfd = seq(0, 15, 0.1))
> pm <- ci.pred(m0, nd)
> head(cbind(nd$tfd, pm))
        Estimate       2.5%      97.5%
1 0.0 0.05157788 0.03569002 0.07453843
2 0.1 0.05193715 0.03778322 0.07139327
3 0.2 0.05227126 0.03959602 0.06900403
4 0.3 0.05255191 0.04090424 0.06751632
5 0.4 0.05275030 0.04153939 0.06698687
6 0.5 0.05283753 0.04149627 0.06727847
```

These are mortality rates per 1 year (because `lex.dur` is in years), but we would want them in units per 100 person-years (% per year), so we multiply by 100:

```
> matshade(nd$tfd, pm * 100, plot = TRUE,
+          ylim = c(0, 10), yaxs = "i",
+          xlab = "Time since diabetes (years)",
+          ylab = "Mortality rate (% / year)")
```

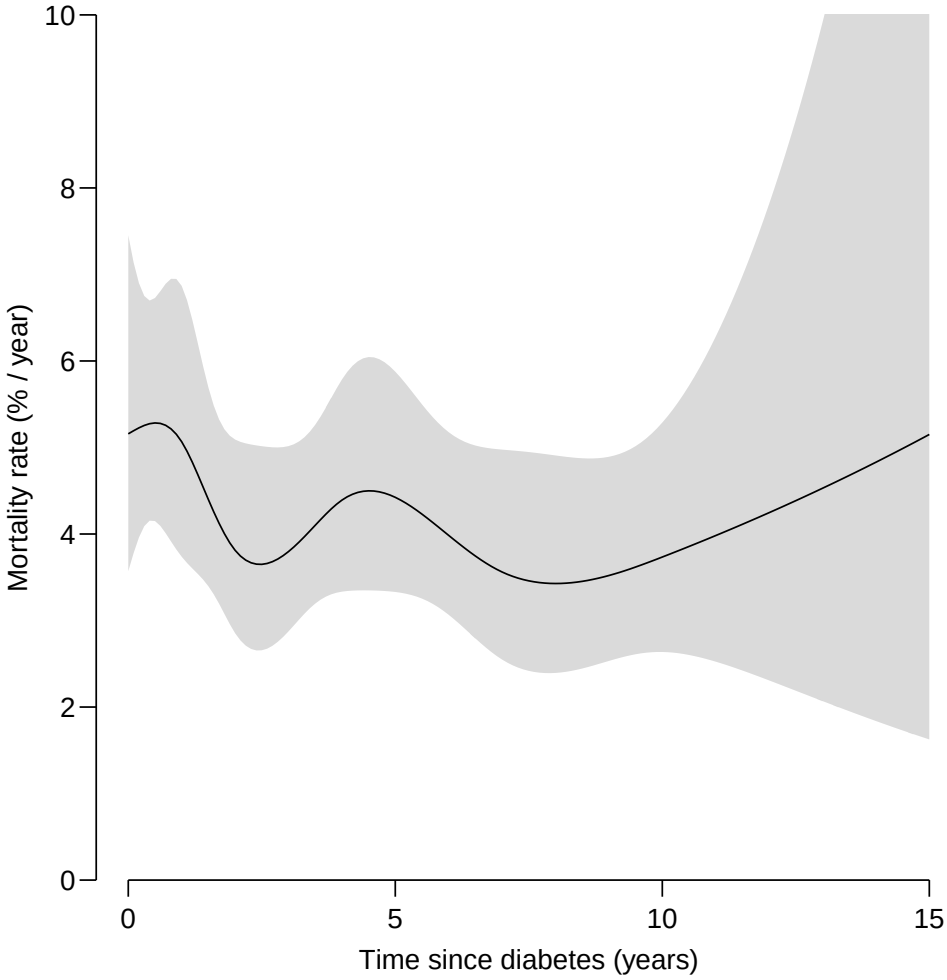

Figure 1.3: *Spline model for mortality among Danish diabetes patients.*

`../graph/surv0-mort`

1.4.2.3 Parametric survival function

We would also want to see the transformation to the survival function, which is straight-forward. Confidence limits for it is however not so easy, so there is a function that does this, `ci.surv`

```
> plot(sf, yaxs = "i")
> abline(h = 0.5)
> ps <- ci.surv(m0, nd)
NOTE: interval length chosen from  as tfd[2] - tfd[1]
> matshade(nd$tfd, ps, col = "red", lwd = 2,
+          ylim = c(0, 1), yaxs = "i",
+          xlab = "Time since diabetes (years)",
+          ylab = "Survival probability")
```

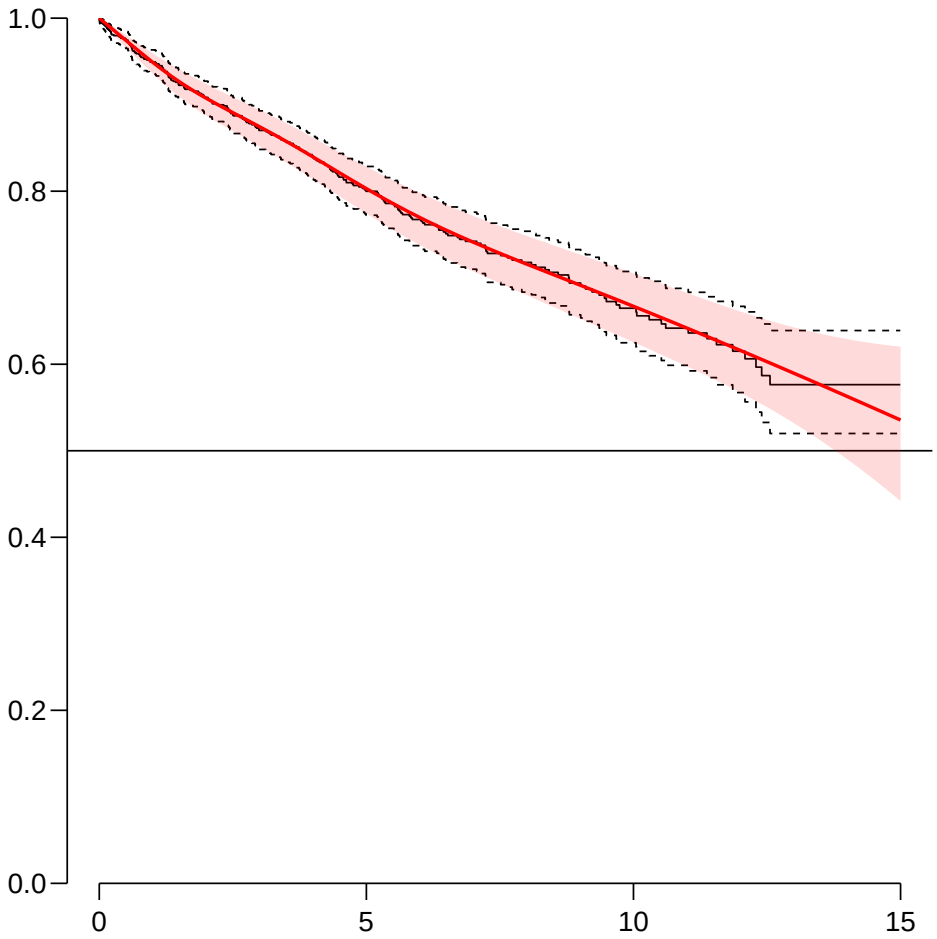

Figure 1.4: *Kaplan-Meier estimator of (black) overlaid with a parametric estimator based on a spline model for mortality. W* ../graph/surv0-KMmod

e now see a much closer agreement between the curve, but also a biologically more credible curve. Obtained by:

- define follow-up as `Lexis`
- split follow-up in small intervals
- use `glmLexis` to model mortality as function of time

- use `ci.pred` to get mortality rates—not available using K-M
- use `ci.surv` to get survival function—close to K-M, but biologically credible (continuous function)

Chapter 2

Expected survival time

Chapter 3

Cause-specific rates

Chapter 4

Competing risks

References