Survival models, Cox regression and follow-up data

SDCC
December 2017
http://bendixcarstensen.com/Epi/Courses/IDEG2017/data/
Version 2

Compiled Sunday 3rd December, 2017, 16:22
from: /home/bendix/sdc/conf/IDEG2017/teach/pracs/SurvFU.tex

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

Contents

Introduction

This is a brief introduction to basic concepts in survival data, and more general handling of follow-up data.

The first chapter introduces survival analysis in its simplest form, the second chapter generalizes the handling of follow-up data in the `Epi` package.

It is only the first chapter that contains proper exercises, time limitation of the workshop prevents full exploration of all aspects of follow-up data.

Acknowledgments

The chapter on follow-up data builds heavily on notes written jointly with Michael Hills (retired, Highgate, London) and Martyn Plummer (Senior statistician, International Agency for Research on Cancer, Lyon) — both of whom I owe thanks for many correcting remarks on my teaching.

Chapter 1

Survival analysis

1.1 Simple analysis of Estonian stroke data

We will need the `Epi` package, so we load this first:

```
library(Epi)
```

The file `stroke.csv` contains information on all registered cases of stroke in Tartu, Estonia during 1991–1993. The data consists of the following variables:

-

`age`	age in years (at entry)
`sex`	sex (1 = male, 0 = female)
`dstr`	date of stroke
`died`	date of death
`dgn`	specific diagnosis, type of stroke (ID - unidentified)
`coma`	indicator, whether patient was in a coma after the stroke
`minf`	history of myocardial infarction of the patient
`diab`	history of diabetes
`han`	history of hypertension

The follow-up was stopped at 1996-01-01. Subjects with missing value of the variable `died` is missing were alive on this date (but not vice versa!).

1. First, read in the data using the `read.csv2()` command. The `read.csv2` reads comma-separated files written in a continental locale, that is with semicolon as field separator and comma as decimal separator:

   Calculate an id variable in the dataframe, and get an overview using `str()`.

   ```
   stroke <- read.csv2(
       "http://bendixcarstensen.com/Epi/Courses/IDEG2017/data/stroke.csv",
                       na.strings="." )
   stroke$id <- 1:nrow(stroke)
   str( stroke )
   head( stroke )
   ```

2. Convert the dates read in as character (and converted to factors) to proper dates (and subsequently to fractions of calendar years — note that applying `cal.yr` to a data frame converts all date variables in the dataframe):

```
stroke <- transform( stroke, dstr=as.Date(dstr,format="%d.%m.%Y"),
                             died=as.Date(died,format="%d.%m.%Y") )
str( stroke )
stroke <- cal.yr(stroke)
```

3. Calculate the last day of follow-up as the smaller of the date of death (`died`) and 1 January 1996.

   Explain why death dates after 1 January 1996 cannot be used as endpoints in the analysis.

   How many deaths occurred after 1 January 1996?

4. Compute the failure indicator (indicator of death) as the existence of a death date *prior to 1 January 1996*. Note the use of a logical statement to generate a variable with values `FALSE` or `TRUE`:

   ```
   stroke <- transform( stroke, dox = pmin( died, 1996, na.rm=TRUE ) )
   subset( stroke, died>1996 )
   with( stroke, table( died>1996 ) )
   stroke <- transform( stroke, D = ( dox < 1996 ) )
   ```

   You have been using `transform`, `subset` and `with`. Look at the help pages for these functions so that you are familiar with what they do.

5. Plot the Kaplan-Meier estimates of overall survival. You will need to attach the `survival` library in order to have access to the function you need:

   ```
   library( survival )
   sst <- with( stroke, Surv( dox-dstr, D ) ~ 1 )
   survfit( sst )
   plot( survfit( sst ) )
   ```

6. Some persons have died on the same day as they had their stroke. Discuss what it means to include them in the study. Try to plot the Kaplan-Meier estimator after excluding these from the data.

   ```
   plot( survfit( sst ) )
   sst0 <- with( subset(stroke,dox>dstr), Surv( dox-dstr, D ) ~ 1 )
   lines( survfit( sst0 ), col="red" )
   ```

   The focus in this study is the survival of patients who actually pull through the stroke (i.e. more than the first day), so we would exclude the patients who die on the same day as the stroke:

   ```
   stroke <- subset( stroke, dox>dstr )
   ```

7. Compute the survival function for each of the 4 diagnoses (as in the variable `dgn`). Also find the median survival for each of the diagnoses? Do the medians exist? Why (not)?

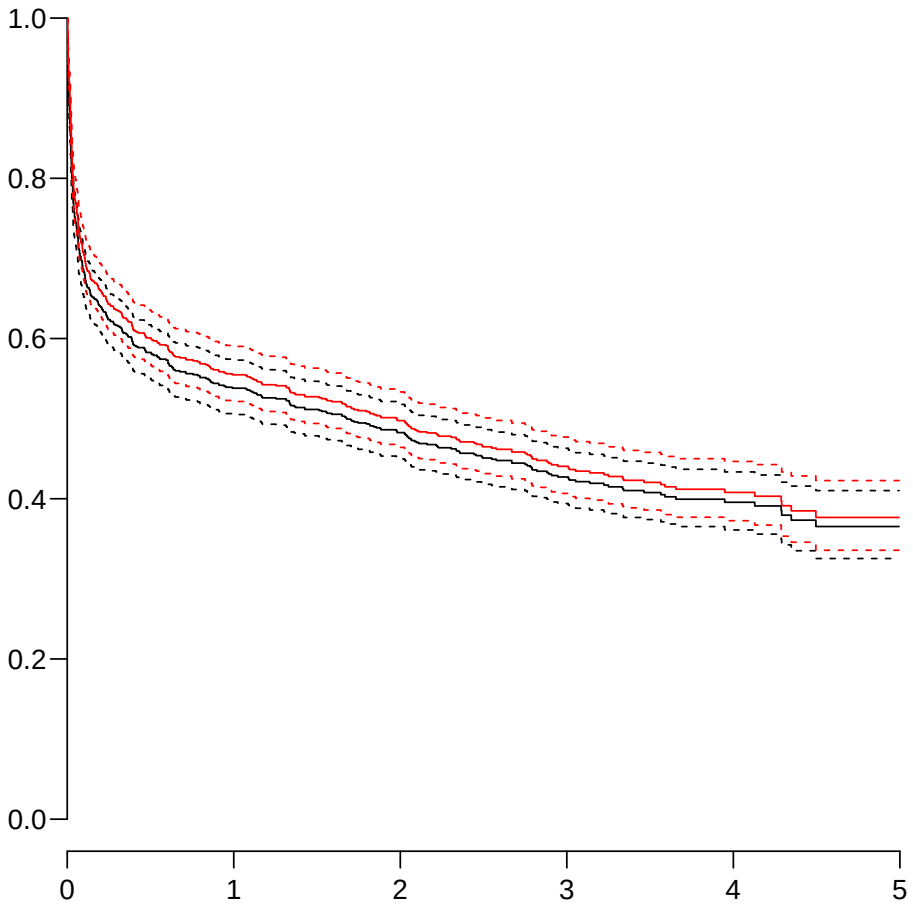

Figure 1.1: *The Kaplan-Meier estimator with (black) and without (red) the 0-survivors, i.e. the persons who die at the same time as their stroke.* `../graph/stroke-KM0`

```
with( stroke, table( dgn, D ) )
( sdiag <- survfit(  Surv( dox-dstr, D ) ~ dgn, data=stroke ) )
```

8. Plot the result as 4 curves.

```
plot( sdiag, col=1:4, lwd=3, mark.time=F )
legend( "bottomleft", legend=levels(stroke$dgn),
         col=1:4, lwd=3, bty="n", text.col=1:4 )
```

9. Plot the log-cumulative hazards for different diagnoses. You will need to use the `fun="cloglog"` argument to `plot.survfit`.

   Do the hazards look proportional?

   Do the same for diabetes history (`diab`) and sex.

```
par( mfrow=c(1,3), mar=c(3,3,1,1) )
plot( survfit( Surv(dox-dstr,D) ~ dgn , data=stroke ), col=1:4, fun="cloglog",
      xlim=c(0.002,5.5),ylim=c(-3.5,0.5),lwd=2)
```

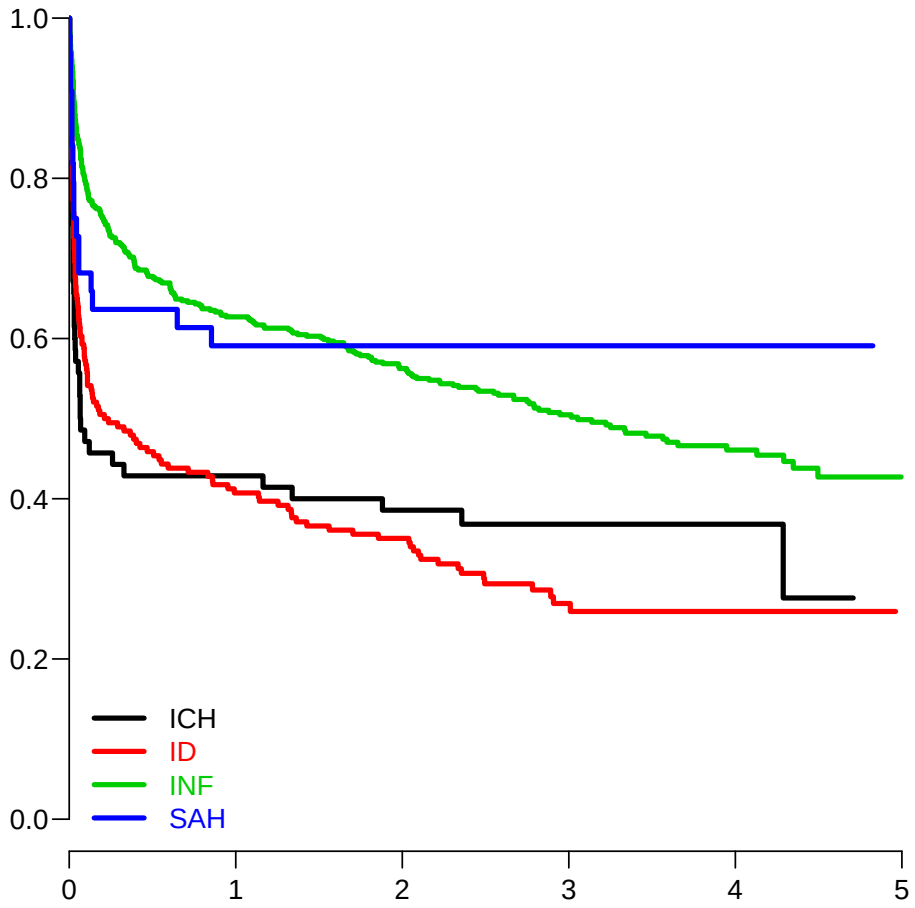

Figure 1.2: *Kaplan-Meier plot for the Estonian stroke data, subdivided by diagnosis.* `../graph/stroke-KM-dgn`

```
legend("bottomright", legend=levels(stroke$dgn), col=1:4, lty=1, lwd=3, bty="n" )
plot( survfit( Surv(dox-dstr,D) ~ diab, data=stroke ), col=1:2, fun="cloglog",
      xlim=c(0.002,5.5),ylim=c(-3.5,0.5),lwd=2)
legend("bottomright", legend=levels(factor(stroke$diab)),col=1:2,lty=1,lwd=3,bty="n"
plot( survfit( Surv(dox-dstr,D) ~ sex, data=stroke), col=c("red","blue"), fun="cloglo
      xlim=c(0.002,5.5), ylim=c(-3.5,0.5), lwd=2 )
legend("bottomright", legend=c("F","M"), col=c("red","blue"), lty=1, lwd=3, bty="n" )
```

10. Plot the Kaplan-Meier estimates of survival function separately for men and woman. Also test the difference using the logrank test:

```
plot(survfit( Surv(dox-dstr,D) ~ sex, data=stroke),
              col=c("red","blue") )
survdiff( Surv(dox-dstr,D) ~ sex, data=stroke)
```

    What do you conclude?

11. Now use `Lexis` to define the survival information, i.e. create a `Lexis` object.

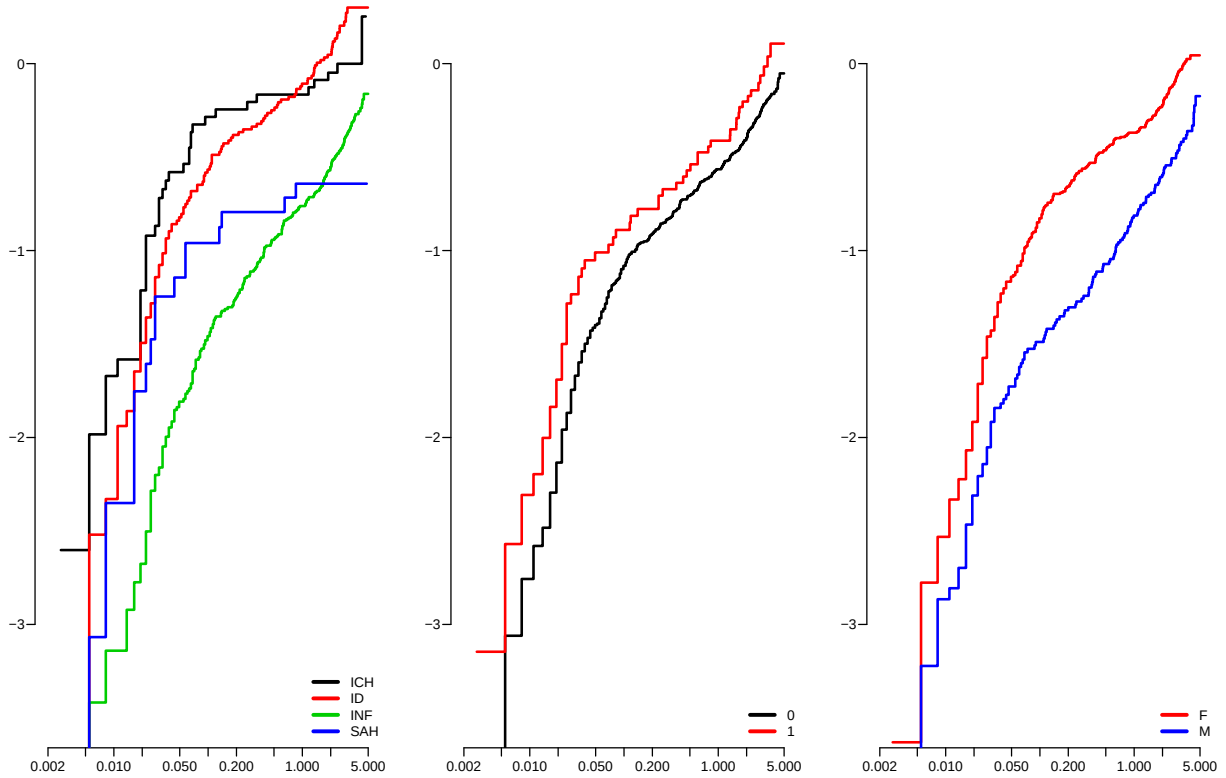

Figure 1.3: *Log-cumulative hazards by diagnosis, diabetes status and sex, respectively, for the Estonian stroke data.* `../graph/stroke-lch`

To do this you must specify date of entry, date of exit on one time scale and entry (or exit) on other timescales that you may be interested in:

```
Lst <- Lexis( data=stroke, entry=list(Per=dstr,Age=age,Tfs=dstr-dstr),
                           exit=list(Per=dox),
                 exit.status=factor( stroke$D, labels=c("Alive","Dead")) )
head( Lst )
```

Explain the variables that have been generated by `Lexis`.

Once you have set this up, you can get a compact overview using `summary` on the object:

```
summary( Lst )
```

12. Since the relevant time-scale is time since stroke (`Tfs`), and since all patients are represented by exactly one record, we can do the survival analysis (Kaplan-Meier estimator) particularly simple based on the `Lexis` object, try:

```
with( stroke, survfit( Surv( dox-dstr, D ) ~ sex ) )
with( Lst, survfit( Surv( lex.dur, lex.Xst=="Dead" ) ~ sex ) )
```

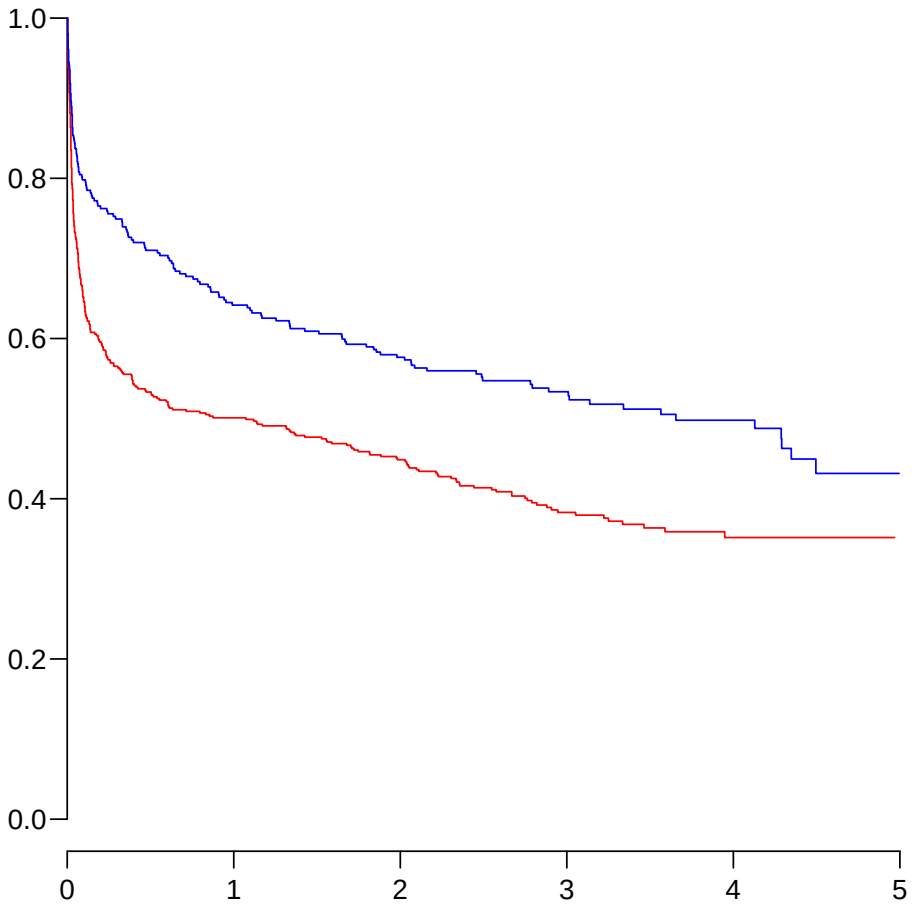

Figure 1.4: *Kaplan-Meier plot for the Estonian stroke data, subdivided by sex.* `../graph/stroke-KM-sex`

13. What is the time-scale we are using in this survival analysis?

14. Finally, save the datasets `stroke` and `Lst` for use in the next exercise (otherwise you are facing the the data processing one again):

```
save( stroke, Lst, file="./stroke.Rda" )
```

1.2 Cox model and time-splitting using Estonian stroke data

```
library(Epi)
library(survival)
```

Reload the Estonian stroke data as you saved them from the first exercise, and make sure that they are still of class `Lexis`:

```
load( file="./stroke.Rda" )
str( Lst )
```

Alternatively you must read the data afresh, transform etc.

15. Fit a Cox model with sex as a covariate. Interpret the hazard ratio and its confidence interval. Fit the model using both the `stroke` data and the data stored as a `Lexis` object (`Lst`).

    ```
    mc <- coxph( Surv(dox-dstr,D) ~ sex, data=stroke )
    summary( mc )
    mL <- coxph( Surv(lex.dur,lex.Xst=="Dead") ~ sex, data=Lst )
    summary( mL )
    ```

    What is the underlying time scale used here?

    What is the rate ratio of death between men and women?

16. Fit a Cox model with sex and age as covariates.

    ```
    mLa <- coxph( Surv(lex.dur,lex.Xst=="Dead") ~ sex + age, data=Lst )
    summary( mLa )
    ```

    What is the most likely reason for change in the effect of sex?

17. Plot the Kaplan-Meier estimate of the survival function for males and females under 75 and those over 75 — i.e. 4 curves. Try it first simple, then more elaborate:

    ```
    plot( survfit( Surv(dox-dstr,as.numeric(D)) ~ interaction(sex,age<75), data=stroke )
    ```

    ```
    plot( survfit( Surv(lex.dur,lex.Xst=="Dead") ~ interaction(sex,age<75),
                   data=Lst ),
                   col=c("red","blue"), lwd=3 )
    ```

    How can you be sure the coloring of curves is correct? (Hint: Try to write `levels(interaction(sex,age<75))`, and remember the recycling rule. Alternatively you can can do:

    ```
    with( Lst, table( interaction(sex,age<75) ) )
    ```

18. Use the `splitLexis` command to split the time-scale every 0.05 years, which is almost at all follow-up times.

    ```
    length( unique(Lst$lex.dur[Lst$lex.Xst==1]) )
    sLst <- splitLexis( Lst, breaks=seq(0,10,0.05), "Tfs" )
    summary( Lst )
    summary( sLst )
    ```

    How is the number of 1) records 2) deaths and 3) person-years in the two datasets?

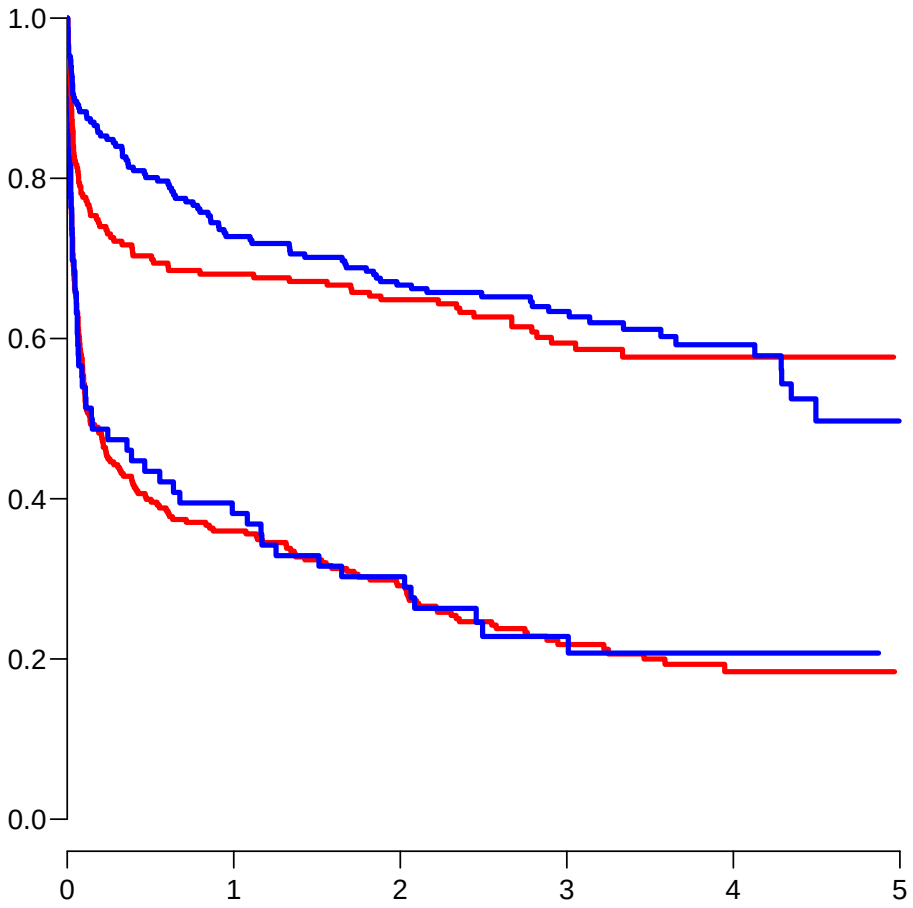

Figure 1.5: *Kaplan-Meier curves for males (blue) and females (red) over and under 75 years at stroke.* `../graph/stroke-75-KM`

19. Try to list the data for the persons with `lex.id` in the range 54:55 from the two datasets to see how the time-splitting has expanded the data:

```
subset( Lst, lex.id %in% 54:55 )
subset( sLst, lex.id %in% 54:55 )
```

20. Fit a Cox model with age and sex as covariates to the split dataset. Check that the parameter estimate are identical to the previous Cox model.

```
mCs <- coxph( Surv(lex.dur,lex.Xst=="Dead") ~ sex + age, data=Lst )
ci.exp( mLa )
mC <- coxph( Surv(Tfs,Tfs+lex.dur,lex.Xst=="Dead") ~ sex + age, data=sLst )
ci.exp( mC )
```

21. Now use Poisson regression with an indicator variable for each interval. Enclose the call in a `system.time()`, which will tell you how long it took on your computer.

```
system.time(
mP <- glm( (lex.Xst=="Dead") ~ factor( Tfs ) + sex + age,
            offset = log(lex.dur),
            family = poisson,
              data = sLst )
           )
```

Take a look at the estimated coefficients:

```
coef( mP )
```

So you may be interested in extracting only the relevant subset of them, and compare with the estimates from the Cox-model:

```
ci.lin( mP, subset=c("sex","age"), Exp=TRUE )
ci.lin( mC, Exp=TRUE )
```

Are there any major differences in estimates from the two approaches?

22. Now use a parametric function for the baseline hazard in the Poisson model. We will use restricted cubic splines (natural splines) with knots at 0.05, 0.1, 0.7, 1.5 and 4 years. These locations are rather arbitrary but not too far from the quantiles in the distribution of the death times:

```
with( subset(sLst, lex.Xst=="Dead"),
      quantile(Tfs+lex.dur,probs=(1:5-0.5)/5 ) )
kn <- c(0.02,0.05,0.1,0.7,1.5,4)
mS <- glm( (lex.Xst=="Dead") ~ Ns( Tfs, knots=kn ) + sex + age,
           offset = log(lex.dur) ,
           family = poisson,
             data = sLst )
```

Compare the parameter estimates with the previous models:

```
round( ci.lin( mC ), 4 )
round( ci.lin( mP, subset=c("sex","age") ), 4 )
round( ci.lin( mS, subset=c("sex","age") ), 4 )
```

Are there any differences?

23. Obtain an estimate of the baseline hazard function for a female aged 60. You will need to generate a prediction data frame with all covariates used in the model — note that `lex.dur` is a covaraite too, albeit one with a coefficient fixed at 1. Whne we set `lex.dur` to 1000 we get predictions as mortality rates per 1000 PY:

```
nd <- data.frame( Tfs = seq(0,5,0.01),
                  sex = 0,
                  age = 60,
              lex.dur = 1000 )
hz <- ci.pred( mS, newdata=nd )
matplot( nd$Tfs, hz, type="l", lwd=c(3,1,1), lty=1, col="black", log="y",
         xlab="Time since stroke (years)",
         ylab="Mortality rate per 1000 PY")
```

24. Obtain an estimate of the *survival function* for a female aged 60. You can reuse the `nd` from before to create the relevant contrast matrix — consult the help page for `ci.cum` first. (For some odd reason, model.matrix)

```
CM <- model.matrix(mS, data=cbind(nd,lex.Xst="Dead") )
Hz <- ci.cum( mS, ctr.mat=CM, intl=0.01 )
matplot( nd$Tfs, exp(-Hz)[,-4],
         type="l", lwd=c(3,1,1), lty=1, col="black",
         yaxs="i", ylim=c(0,1),
         xlab="Time since stroke (years)",
         ylab="Survival")
```

25. Compute the estimated survival function for a similar person from the Cox-model and plot in the same frame.

```
matplot( nd$Tfs, exp(-Hz)[,-4],
         type="l", lwd=c(3,1,1), lty=1, col="black",
         yaxs="i", ylim=c(0,1),
         xlab="Time since stroke (years)",
         ylab="Survival")
lines( survfit(mC, newdata = nd), conf.int=TRUE, col="red" )
# overplot the estimate with a thicker line:
lines( survfit(mC, newdata = nd), conf.int=FALSE, col="red", lwd=3 )
```

One morale of this exercise is that it is immaterial whether a Cox-model or a Poisson-model is used for estimation of covariate effects. But the assumptions behind the Poisson-model (continuous effect of time) seems more reasonable.

The other morale is that it requires some care to model the hazard correctly in the beginning (or rather in parts of the timescale where mortality is changing rapidly), if it has to be used for survival function construction.

The following things should be taken care of where hazards is changing rapidly:

- Time should be split finely.
- The effect of time should be modelled detailed.

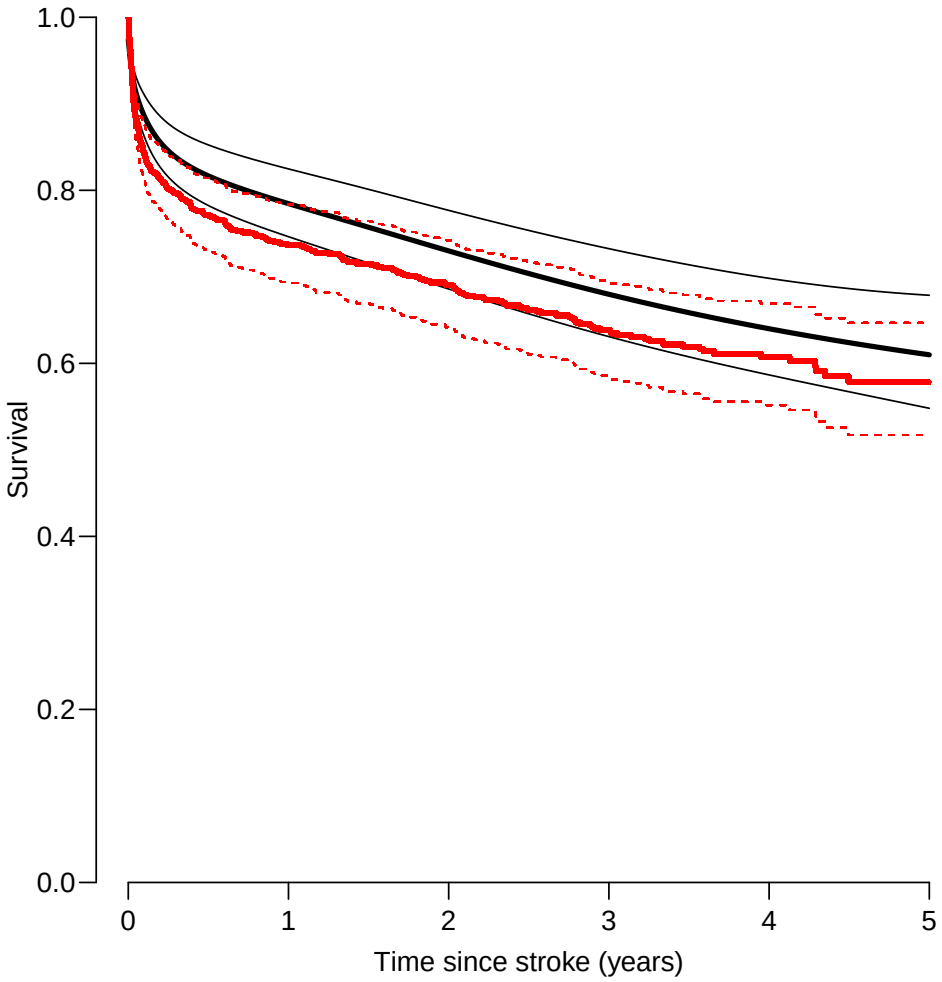

Figure 1.6: *Estimated survival curve for a female 75-year at stroke, computed by the Breslow-setimator from the Cox-model, and by using the approximation from the Poisson model.* `../graph/stroke-Cox-Pois2`

Chapter 2

Follow-up data in the Epi package

```
R version 3.4.2 (2017-09-28)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 14.04.5 LTS

Matrix products: default
BLAS: /usr/lib/openblas-base/libopenblas.so.0
LAPACK: /usr/lib/lapack/liblapack.so.3.0

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C                LC_TIME=en_DK.UTF-8
 [4] LC_COLLATE=en_US.UTF-8     LC_MONETARY=en_US.UTF-8     LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C                   LC_ADDRESS=C
[10] LC_TELEPHONE=C             LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] utils     datasets  graphics  grDevices stats     methods   base

other attached packages:
[1] Epi_2.21

loaded via a namespace (and not attached):
 [1] cmprsk_2.2-7      zoo_1.8-0         MASS_7.3-47       compiler_3.4.2
 [5] Matrix_1.2-11     plyr_1.8.4        parallel_3.4.2    survival_2.41-3
 [9] etm_0.6-2         Rcpp_0.12.12      splines_3.4.2     grid_3.4.2
[13] numDeriv_2016.8-1 lattice_0.20-35
```

In the `Epi`-package, follow-up data is in general represented by adding some extra variables to a data frame. The names of these are used to keep track of the followup. Such a data frame is called a `Lexis` object. The tools for handling follow-up data then use the structure of this for special plots, tabulations etc.

Follow-up data basically consists of a time of entry, a time of exit and an indication of the status at exit (normally either "Alive" or "Dead"). Implicitly is also assumed a status *during* the follow-up (usually "Alive").

2.1 Timescales

A timescale is a variable that varies deterministically *within* each person during follow-up, *e.g.*:

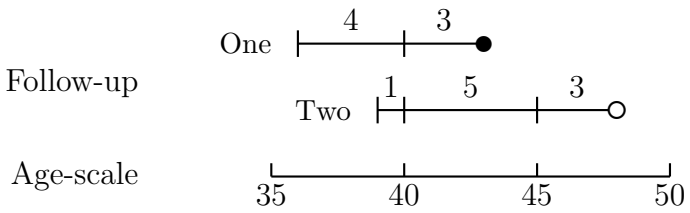

Figure 2.1: *Follow-up of two persons on the age-scale. Follow-up is allocated in small age-bands, whereby we can keep track of the persons'* current *age (a.k.a.* attained *age).*

- Age
- Calendar time
- Time since treatment
- Time since relapse

All timescales advance at the same pace, so the time followed is the same on all timescales. Therefore, it suffices to use only the entry point on each of the time scale, for example:

- Age at entry.
- Date of entry.
- Time since treatment (*at* treatment this is 0).
- Time since relapse (*at* relapse this is 0)..

In the `Epi` package, follow-up in a cohort is represented in a `Lexis` object. A `Lexis` object is a data frame with a bit of extra structure representing the follow-up. For the `nickel` data we would construct a `Lexis` object by:

```
> data( nickel )
> nicL <- Lexis( entry = list( per=agein+dob,
+                              age=agein,
+                              tfh=agein-age1st ),
+                exit = list( age=ageout ),
+          exit.status = ( icd %in% c(162,163) )*1,
+                data = nickel )
```

The `entry` argument is a *named* list with the entry points on each of the timescales we want to use. It defines the names of the timescales and the entry points. The `exit` argument gives the exit time on *one* of the timescales, so the name of the element in this list must match one of the names of the `entry` list. This is sufficient, because the follow-up time on all time scales is the same, in this case `ageout - agein`. Now take a look at the result:

```
> str( nickel )
'data.frame':        679 obs. of  7 variables:
 $ id      : num  3 4 6 8 9 10 15 16 17 18 ...
 $ icd     : num  0 162 163 527 150 163 334 160 420 12 ...
 $ exposure: num  5 5 10 9 0 2 0 0.5 0 0 ...
 $ dob     : num  1889 1886 1881 1886 1880 ...
 $ age1st  : num  17.5 23.2 25.2 24.7 30 ...
 $ agein   : num  45.2 48.3 53 47.9 54.7 ...
 $ ageout  : num  93 63.3 54.2 69.7 76.8 ...
```

```
> str( nicL )
Classes 'Lexis' and 'data.frame':       679 obs. of  14 variables:
 $ per     : num  1934 1934 1934 1934 1934 ...
 $ age     : num  45.2 48.3 53 47.9 54.7 ...
 $ tfh     : num  27.7 25.1 27.7 23.2 24.8 ...
 $ lex.dur : num  47.75 15 1.17 21.77 22.1 ...
 $ lex.Cst : num  0 0 0 0 0 0 0 0 0 0 ...
 $ lex.Xst : num  0 1 1 0 0 1 0 0 0 0 ...
 $ lex.id  : int  1 2 3 4 5 6 7 8 9 10 ...
 $ id      : num  3 4 6 8 9 10 15 16 17 18 ...
 $ icd     : num  0 162 163 527 150 163 334 160 420 12 ...
 $ exposure: num  5 5 10 9 0 2 0 0.5 0 0 ...
 $ dob     : num  1889 1886 1881 1886 1880 ...
 $ age1st  : num  17.5 23.2 25.2 24.7 30 ...
 $ agein   : num  45.2 48.3 53 47.9 54.7 ...
 $ ageout  : num  93 63.3 54.2 69.7 76.8 ...
 - attr(*, "time.scales")= chr  "per" "age" "tfh"
 - attr(*, "time.since")= chr  "" "" ""
 - attr(*, "breaks")=List of 3
  ..$ per: NULL
  ..$ age: NULL
  ..$ tfh: NULL
> head( nicL )
       per     age     tfh lex.dur lex.Cst lex.Xst lex.id id icd exposure      dob
1 1934.246 45.2273 27.7465 47.7535       0       0      1  3   0        5 1889.019
2 1934.246 48.2684 25.0820 15.0028       0       1      2  4 162        5 1885.978
3 1934.246 52.9917 27.7465  1.1727       0       1      3  6 163       10 1881.255
4 1934.246 47.9067 23.1861 21.7727       0       0      4  8 527        9 1886.340
5 1934.246 54.7465 24.7890 22.0977       0       0      5  9 150        0 1879.500
6 1934.246 44.3314 23.0437 18.2099       0       1      6 10 163        2 1889.915
   age1st   agein  ageout
1 17.4808 45.2273 92.9808
2 23.1864 48.2684 63.2712
3 25.2452 52.9917 54.1644
4 24.7206 47.9067 69.6794
5 29.9575 54.7465 76.8442
6 21.2877 44.3314 62.5413
```

The `Lexis` object `nicL` has a variable for each timescale which is the entry point on this timescale. The follow-up time is in the variable `lex.dur` (**dur**ation).

We defined the exit status to be death from lung cancer (ICD7 162,163), i.e. this variable is 1 if follow-up ended with a death from this cause. If follow-up ended alive or by death from another cause, the exit status is coded 0, i.e. as a censoring.

Note that the exit status is in the variable `lex.Xst` (e**X**it **st**atus. The variable `lex.Cst` is the state where the follow-up takes place (**C**urrent **st**atus), in this case 0 (alive).

It is possible to get a visualization of the follow-up along the timescales chosen by using the `plot` method for `Lexis` objects. `nicL` is an object of *class* `Lexis`, so using the function `plot()` on it means that R will look for the function `plot.Lexis` and use this function.

```
> plot( nicL )
```

The function allows a lot of control over the output, and a `points.Lexis` function allows plotting of the endpoints of follow-up.

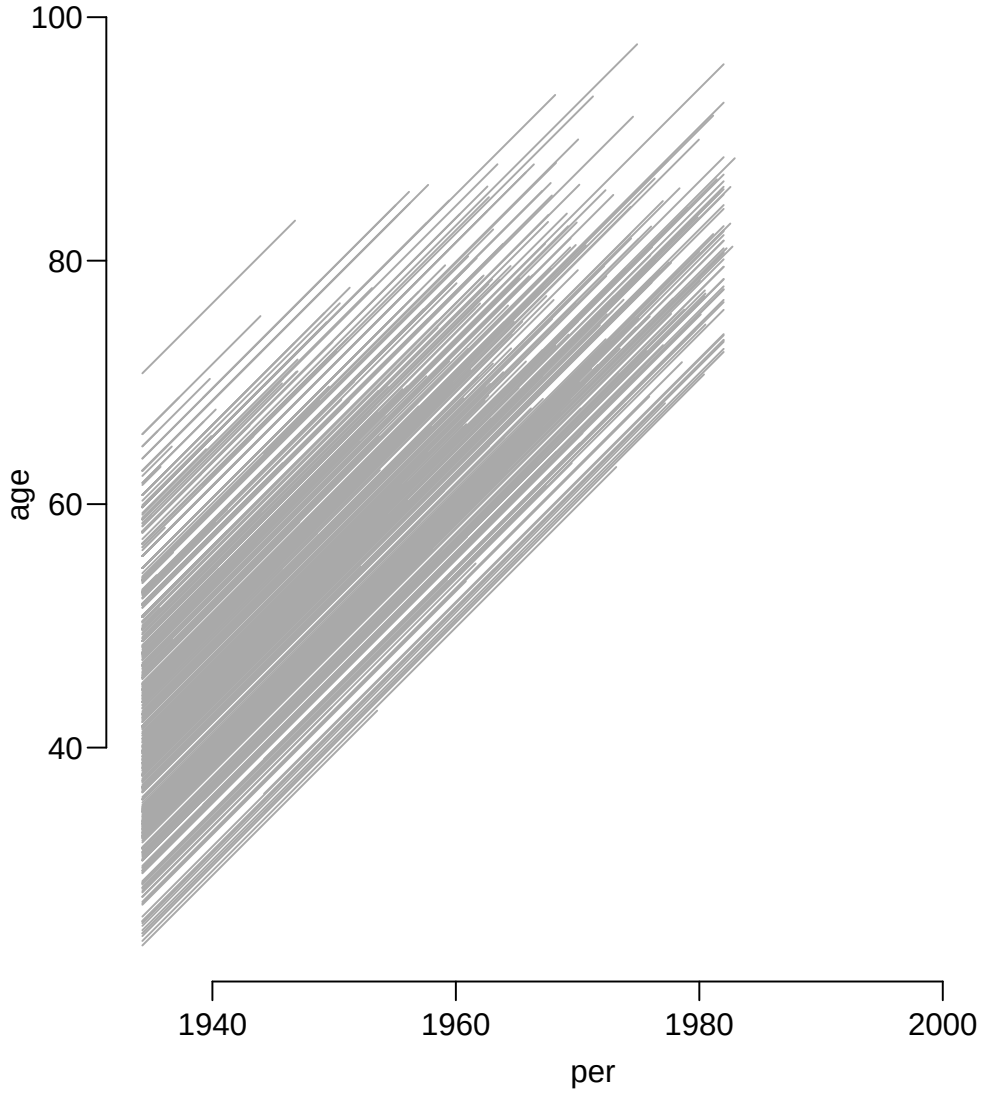

Figure 2.2: *Lexis diagram of the* **`nickel`** *dataset.* `./flup-nicL`

```
> par( mar=c(3,3,1,1), mgp=c(3,1,0)/1.6 )
> plot( nicL, 1:2, lwd=1, col=c("blue","red")[(nicL$exp>0)+1],
+       grid=TRUE, lty.grid=1, col.grid=gray(0.7),
+       xlim=1900+c(0,90), xaxs="i",
+       ylim=  10+c(0,90), yaxs="i", las=1 )
> points( nicL, 1:2, pch=c(NA,3)[nicL$lex.Xst+1],
+         col="lightgray", lwd=3, cex=1.5 )
> points( nicL, 1:2, pch=c(NA,3)[nicL$lex.Xst+1],
+         col=c("blue","red")[(nicL$exp>0)+1], lwd=1, cex=1.5 )
```

2.2 Splitting the follow-up time along a timescale

The follow-up time in a cohort can be subdivided by for example current age. This is achieved by the `splitLexis` (note that it is *not* called `split.Lexis`). This requires that the timescale and the breakpoints on this timescale are supplied. Try:

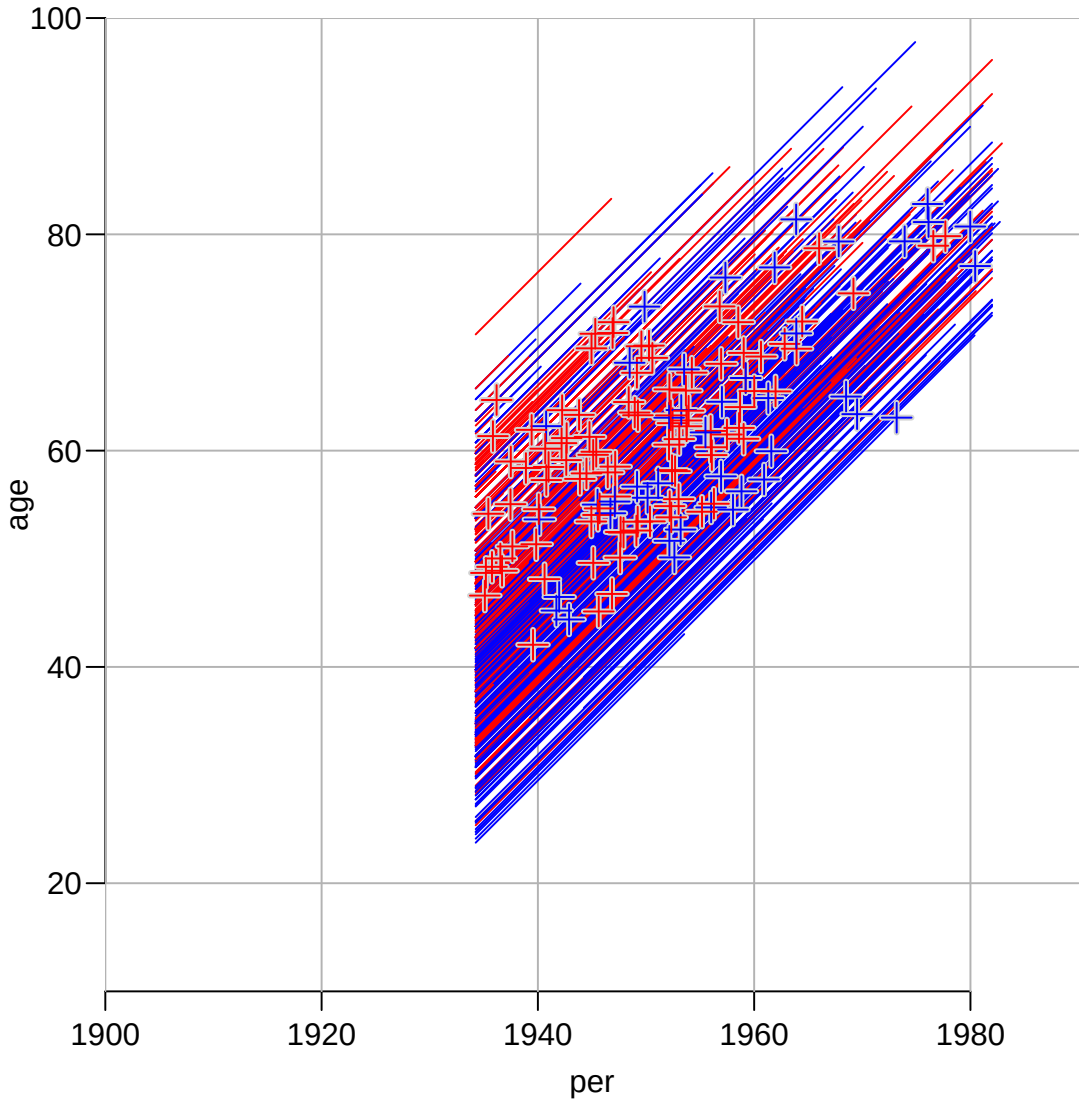

Figure 2.3: *Lexis diagram of the* `nickel` *dataset, with bells and whistles. The red lines are for persons with exposure*> 0*, so it is pretty evident that the oldest ones are the exposed part of the cohort.* `./flup-nicL2`

```
> nicS1 <- splitLexis( nicL, "age", breaks=seq(0,100,10) )
> str( nicL )
Classes 'Lexis' and 'data.frame':       679 obs. of  14 variables:
 $ per     : num  1934 1934 1934 1934 1934 ...
 $ age     : num  45.2 48.3 53 47.9 54.7 ...
 $ tfh     : num  27.7 25.1 27.7 23.2 24.8 ...
 $ lex.dur : num  47.75 15 1.17 21.77 22.1 ...
 $ lex.Cst : num  0 0 0 0 0 0 0 0 0 0 ...
 $ lex.Xst : num  0 1 1 0 0 1 0 0 0 0 ...
 $ lex.id  : int  1 2 3 4 5 6 7 8 9 10 ...
 $ id      : num  3 4 6 8 9 10 15 16 17 18 ...
 $ icd     : num  0 162 163 527 150 163 334 160 420 12 ...
 $ exposure: num  5 5 10 9 0 2 0 0.5 0 0 ...
 $ dob     : num  1889 1886 1881 1886 1880 ...
 $ age1st  : num  17.5 23.2 25.2 24.7 30 ...
 $ agein   : num  45.2 48.3 53 47.9 54.7 ...
 $ ageout  : num  93 63.3 54.2 69.7 76.8 ...
 - attr(*, "time.scales")= chr  "per" "age" "tfh"
 - attr(*, "time.since")= chr  "" "" ""
 - attr(*, "breaks")=List of 3
  ..$ per: NULL
  ..$ age: NULL
  ..$ tfh: NULL
> str( nicS1 )
Classes 'Lexis' and 'data.frame':       2210 obs. of  14 variables:
 $ lex.id  : int  1 1 1 1 1 1 2 2 2 3 ...
 $ per     : num  1934 1939 1949 1959 1969 ...
 $ age     : num  45.2 50 60 70 80 ...
 $ tfh     : num  27.7 32.5 42.5 52.5 62.5 ...
 $ lex.dur : num  4.77 10 10 10 10 ...
 $ lex.Cst : num  0 0 0 0 0 0 0 0 0 0 ...
 $ lex.Xst : num  0 0 0 0 0 0 0 0 1 1 ...
 $ id      : num  3 3 3 3 3 3 4 4 4 6 ...
 $ icd     : num  0 0 0 0 0 0 162 162 162 163 ...
 $ exposure: num  5 5 5 5 5 5 5 5 5 10 ...
 $ dob     : num  1889 1889 1889 1889 1889 ...
 $ age1st  : num  17.5 17.5 17.5 17.5 17.5 ...
 $ agein   : num  45.2 45.2 45.2 45.2 45.2 ...
 $ ageout  : num  93 93 93 93 93 ...
 - attr(*, "breaks")=List of 3
  ..$ per: NULL
  ..$ age: num  0 10 20 30 40 50 60 70 80 90 ...
  ..$ tfh: NULL
 - attr(*, "time.scales")= chr  "per" "age" "tfh"
 - attr(*, "time.since")= chr  "" "" ""
> round( subset( nicS1, id %in% 8:10 ), 2 )
   lex.id     per   age   tfh lex.dur lex.Cst lex.Xst id icd exposure     dob age1st
11      4 1934.25 47.91 23.19    2.09       0       0  8 527        9 1886.34  24.72
12      4 1936.34 50.00 25.28   10.00       0       0  8 527        9 1886.34  24.72
13      4 1946.34 60.00 35.28    9.68       0       0  8 527        9 1886.34  24.72
14      5 1934.25 54.75 24.79    5.25       0       0  9 150        0 1879.50  29.96
15      5 1939.50 60.00 30.04   10.00       0       0  9 150        0 1879.50  29.96
16      5 1949.50 70.00 40.04    6.84       0       0  9 150        0 1879.50  29.96
17      6 1934.25 44.33 23.04    5.67       0       0 10 163        2 1889.91  21.29
18      6 1939.91 50.00 28.71   10.00       0       0 10 163        2 1889.91  21.29
19      6 1949.91 60.00 38.71    2.54       0       1 10 163        2 1889.91  21.29
```

```
   agein ageout
11 47.91  69.68
12 47.91  69.68
13 47.91  69.68
14 54.75  76.84
15 54.75  76.84
16 54.75  76.84
17 44.33  62.54
18 44.33  62.54
19 44.33  62.54
```

The resulting object is again a `Lexis` object, and so follow-up may be split further along another timescale. Try this and list the result for individuals 4 and 6:

```
> system.time(
+ nicS2 <- splitLexis( nicS1, "tfh", breaks=c(0,1,5,10,20,30,100) ) )
   user  system elapsed
  0.118   0.013   0.130
> round( subset( nicS2, id %in% 8:10 ), 2 )
   lex.id     per   age   tfh lex.dur lex.Cst lex.Xst id icd exposure     dob age1st
13      4 1934.25 47.91 23.19    2.09       0       0  8 527        9 1886.34  24.72
14      4 1936.34 50.00 25.28    4.72       0       0  8 527        9 1886.34  24.72
15      4 1941.06 54.72 30.00    5.28       0       0  8 527        9 1886.34  24.72
16      4 1946.34 60.00 35.28    9.68       0       0  8 527        9 1886.34  24.72
17      5 1934.25 54.75 24.79    5.21       0       0  9 150        0 1879.50  29.96
18      5 1939.46 59.96 30.00    0.04       0       0  9 150        0 1879.50  29.96
19      5 1939.50 60.00 30.04   10.00       0       0  9 150        0 1879.50  29.96
20      5 1949.50 70.00 40.04    6.84       0       0  9 150        0 1879.50  29.96
21      6 1934.25 44.33 23.04    5.67       0       0 10 163        2 1889.91  21.29
22      6 1939.91 50.00 28.71    1.29       0       0 10 163        2 1889.91  21.29
23      6 1941.20 51.29 30.00    8.71       0       0 10 163        2 1889.91  21.29
24      6 1949.91 60.00 38.71    2.54       0       1 10 163        2 1889.91  21.29
   agein ageout
13 47.91  69.68
14 47.91  69.68
15 47.91  69.68
16 47.91  69.68
17 54.75  76.84
18 54.75  76.84
19 54.75  76.84
20 54.75  76.84
21 44.33  62.54
22 44.33  62.54
23 44.33  62.54
24 44.33  62.54
```

2.2.1 The `popEpi` package

With large data sets `splitLexis` may be slow; a faster function, `splitMulti` with slightly different syntax exist in the `popEpi` package. With this it is possible to split as above on two time-scale in one go, and much quicker:

```
> library( popEpi )
> system.time(
+ nicM2 <- splitMulti( nicL, age = seq(0,100,10),
+                            tfh = c(0,1,5,10,20,30,100) ) )
```

```
   user  system elapsed
  0.078   0.005   0.055
```

So you see the split along two timescales is much quicker than the `splitLexis` on only one. Yet the result is the same:

```
> summary( nicM2 )
Transitions:
     To
From    0   1  Records:  Events: Risk time:  Persons:
   0 2992 137      3129      137   15348.06       679
> summary( nicS2 )
Transitions:
     To
From    0   1  Records:  Events: Risk time:  Persons:
   0 2992 137      3129      137   15348.06       679
```

2.3 Cutting follow-up time at a specific date

If we have a recording of the date of a specific event as for example recovery or relapse, we may classify follow-up time as being before of after this intermediate event. This is what is usually termed a `time-dependent` covariate — it has different values during different parts of the followup for a single person.

This is achieved with the function `cutLexis`, which takes three arguments: the time point, the timescale, and the value of the (new) state following the date.

Now we define the age for the nickel workers where the cumulative exposure exceeds 50 exposure years:

```
> subset( nicL, id %in% 8:10 )
       per     age     tfh lex.dur lex.Cst lex.Xst lex.id id icd exposure      dob
4 1934.246 47.9067 23.1861 21.7727       0       0      4  8 527        9 1886.340
5 1934.246 54.7465 24.7890 22.0977       0       0      5  9 150        0 1879.500
6 1934.246 44.3314 23.0437 18.2099       0       1      6 10 163        2 1889.915
   age1st   agein  ageout
4 24.7206 47.9067 69.6794
5 29.9575 54.7465 76.8442
6 21.2877 44.3314 62.5413
> agehi <- nicL$age1st + 50 / nicL$exposure
> nicC <- cutLexis( data = nicL,
+                   cut = agehi,
+              timescale = "age",
+              new.state = 2,
+       precursor.states = 0 )
> subset( nicC, id %in% 8:10 )
         per     age     tfh lex.dur lex.Cst lex.Xst lex.id id icd exposure      dob
683 1934.246 47.9067 23.1861 21.7727       2       2      4  8 527        9 1886.340
5   1934.246 54.7465 24.7890 22.0977       0       0      5  9 150        0 1879.500
6   1934.246 44.3314 23.0437  1.9563       0       2      6 10 163        2 1889.915
685 1936.203 46.2877 25.0000 16.2536       2       1      6 10 163        2 1889.915
     age1st   agein  ageout
683 24.7206 47.9067 69.6794
5   29.9575 54.7465 76.8442
6   21.2877 44.3314 62.5413
685 21.2877 44.3314 62.5413
```

(The `precursor.states=` argument is explained below). Note that individual 6 has had his follow-up split at age 25 where 50 exposure-years were attained. This could also have been achieved in the split dataset `nicS2` instead of `nicL`, try:

```
> subset( nicS2, id %in% 8:10 )
   lex.id      per     age     tfh lex.dur lex.Cst lex.Xst id icd exposure      dob
13      4 1934.246 47.9067 23.1861  2.0933       0       0  8 527        9 1886.340
14      4 1936.340 50.0000 25.2794  4.7206       0       0  8 527        9 1886.340
15      4 1941.060 54.7206 30.0000  5.2794       0       0  8 527        9 1886.340
16      4 1946.340 60.0000 35.2794  9.6794       0       0  8 527        9 1886.340
17      5 1934.246 54.7465 24.7890  5.2110       0       0  9 150        0 1879.500
18      5 1939.457 59.9575 30.0000  0.0425       0       0  9 150        0 1879.500
19      5 1939.500 60.0000 30.0425 10.0000       0       0  9 150        0 1879.500
20      5 1949.500 70.0000 40.0425  6.8442       0       0  9 150        0 1879.500
21      6 1934.246 44.3314 23.0437  5.6686       0       0 10 163        2 1889.915
22      6 1939.915 50.0000 28.7123  1.2877       0       0 10 163        2 1889.915
23      6 1941.203 51.2877 30.0000  8.7123       0       0 10 163        2 1889.915
24      6 1949.915 60.0000 38.7123  2.5413       0       1 10 163        2 1889.915
    age1st   agein  ageout
13 24.7206 47.9067 69.6794
14 24.7206 47.9067 69.6794
15 24.7206 47.9067 69.6794
16 24.7206 47.9067 69.6794
17 29.9575 54.7465 76.8442
18 29.9575 54.7465 76.8442
19 29.9575 54.7465 76.8442
20 29.9575 54.7465 76.8442
21 21.2877 44.3314 62.5413
22 21.2877 44.3314 62.5413
23 21.2877 44.3314 62.5413
24 21.2877 44.3314 62.5413
```

```
> agehi <- nicS2$age1st + 50 / nicS2$exposure
> nicS2C <- cutLexis( data=nicS2, cut=agehi, timescale="age",
+                     new.state=2, precursor.states=0 )
> subset( nicS2C, id %in% 8:10 )
     lex.id      per     age     tfh lex.dur lex.Cst lex.Xst id icd exposure      dob
3142      4 1934.246 47.9067 23.1861  2.0933       2       2  8 527        9 1886.340
3143      4 1936.340 50.0000 25.2794  4.7206       2       2  8 527        9 1886.340
3144      4 1941.060 54.7206 30.0000  5.2794       2       2  8 527        9 1886.340
3145      4 1946.340 60.0000 35.2794  9.6794       2       2  8 527        9 1886.340
17        5 1934.246 54.7465 24.7890  5.2110       0       0  9 150        0 1879.500
18        5 1939.457 59.9575 30.0000  0.0425       0       0  9 150        0 1879.500
19        5 1939.500 60.0000 30.0425 10.0000       0       0  9 150        0 1879.500
20        5 1949.500 70.0000 40.0425  6.8442       0       0  9 150        0 1879.500
21        6 1934.246 44.3314 23.0437  1.9563       0       2 10 163        2 1889.915
3150      6 1936.203 46.2877 25.0000  3.7123       2       2 10 163        2 1889.915
3151      6 1939.915 50.0000 28.7123  1.2877       2       2 10 163        2 1889.915
3152      6 1941.203 51.2877 30.0000  8.7123       2       2 10 163        2 1889.915
3153      6 1949.915 60.0000 38.7123  2.5413       2       1 10 163        2 1889.915
      age1st   agein  ageout
3142 24.7206 47.9067 69.6794
3143 24.7206 47.9067 69.6794
3144 24.7206 47.9067 69.6794
3145 24.7206 47.9067 69.6794
17   29.9575 54.7465 76.8442
18   29.9575 54.7465 76.8442
```

```
19   29.9575 54.7465 76.8442
20   29.9575 54.7465 76.8442
21   21.2877 44.3314 62.5413
3150 21.2877 44.3314 62.5413
3151 21.2877 44.3314 62.5413
3152 21.2877 44.3314 62.5413
3153 21.2877 44.3314 62.5413
```

Note that follow-up subsequent to the event is classified as being in state 2, but that the final transition to state 1 (death from lung cancer) is preserved. This is the point of the `precursor.states=` argument. It names the states (in this case 0, "Alive") that will be over-written by `new.state` (in this case 2, "High exposure"). Clearly, state 1 ("Dead") should not be updated even if it is after the time where the persons moves to state 2. On other words, only state 0 is a precursor to state 2, state 1 is always subsequent to state 2.

Note if the intermediate event is to be used as a time-dependent variable in a Cox-model, then `lex.Cst` should be used as the time-dependent variable, and `lex.Xst==1` as the event.

2.4 Competing risks — multiple types of events

If we want to consider death from lung cancer and death from other causes as separate events we can code these as for example 1 and 2.

```
> data( nickel )
> nicL <- Lexis( entry = list( per=agein+dob,
+                              age=agein,
+                              tfh=agein-age1st ),
+                exit = list( age=ageout ),
+          exit.status = ( icd > 0 ) + ( icd %in% c(162,163) ),
+                data = nickel )
> str( nicL )
Classes 'Lexis' and 'data.frame':        679 obs. of  14 variables:
 $ per     : num  1934 1934 1934 1934 1934 ...
 $ age     : num  45.2 48.3 53 47.9 54.7 ...
 $ tfh     : num  27.7 25.1 27.7 23.2 24.8 ...
 $ lex.dur : num  47.75 15 1.17 21.77 22.1 ...
 $ lex.Cst : num  0 0 0 0 0 0 0 0 0 0 ...
 $ lex.Xst : int  0 2 2 1 1 2 1 1 1 1 ...
 $ lex.id  : int  1 2 3 4 5 6 7 8 9 10 ...
 $ id      : num  3 4 6 8 9 10 15 16 17 18 ...
 $ icd     : num  0 162 163 527 150 163 334 160 420 12 ...
 $ exposure: num  5 5 10 9 0 2 0 0.5 0 0 ...
 $ dob     : num  1889 1886 1881 1886 1880 ...
 $ age1st  : num  17.5 23.2 25.2 24.7 30 ...
 $ agein   : num  45.2 48.3 53 47.9 54.7 ...
 $ ageout  : num  93 63.3 54.2 69.7 76.8 ...
 - attr(*, "time.scales")= chr  "per" "age" "tfh"
 - attr(*, "time.since")= chr  "" "" ""
 - attr(*, "breaks")=List of 3
  ..$ per: NULL
  ..$ age: NULL
  ..$ tfh: NULL
> head( nicL )
```

```
       per     age     tfh lex.dur lex.Cst lex.Xst lex.id id icd exposure      dob
1 1934.246 45.2273 27.7465 47.7535       0       0      1  3   0        5 1889.019
2 1934.246 48.2684 25.0820 15.0028       0       2      2  4 162        5 1885.978
3 1934.246 52.9917 27.7465  1.1727       0       2      3  6 163       10 1881.255
4 1934.246 47.9067 23.1861 21.7727       0       1      4  8 527        9 1886.340
5 1934.246 54.7465 24.7890 22.0977       0       1      5  9 150        0 1879.500
6 1934.246 44.3314 23.0437 18.2099       0       2      6 10 163        2 1889.915
   age1st   agein  ageout
1 17.4808 45.2273 92.9808
2 23.1864 48.2684 63.2712
3 25.2452 52.9917 54.1644
4 24.7206 47.9067 69.6794
5 29.9575 54.7465 76.8442
6 21.2877 44.3314 62.5413
> subset( nicL, id %in% 8:10 )
       per     age     tfh lex.dur lex.Cst lex.Xst lex.id id icd exposure      dob
4 1934.246 47.9067 23.1861 21.7727       0       1      4  8 527        9 1886.340
5 1934.246 54.7465 24.7890 22.0977       0       1      5  9 150        0 1879.500
6 1934.246 44.3314 23.0437 18.2099       0       2      6 10 163        2 1889.915
   age1st   agein  ageout
4 24.7206 47.9067 69.6794
5 29.9575 54.7465 76.8442
6 21.2877 44.3314 62.5413
```

If we want to label the states, we can enter the names of these in the `states` parameter, try for example:

```
> nicL <- Lexis( entry = list( per=agein+dob,
+                              age=agein,
+                              tfh=agein-age1st ),
+                exit = list( age=ageout ),
+          exit.status = ( icd > 0 ) + ( icd %in% c(162,163) ),
+                data = nickel,
+              states = c("Alive","D.oth","D.lung") )
> str( nicL )
Classes 'Lexis' and 'data.frame':       679 obs. of  14 variables:
 $ per     : num  1934 1934 1934 1934 1934 ...
 $ age     : num  45.2 48.3 53 47.9 54.7 ...
 $ tfh     : num  27.7 25.1 27.7 23.2 24.8 ...
 $ lex.dur : num  47.75 15 1.17 21.77 22.1 ...
 $ lex.Cst : Factor w/ 3 levels "Alive","D.oth",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ lex.Xst : Factor w/ 3 levels "Alive","D.oth",..: 1 3 3 2 2 3 2 2 2 2 ...
 $ lex.id  : int  1 2 3 4 5 6 7 8 9 10 ...
 $ id      : num  3 4 6 8 9 10 15 16 17 18 ...
 $ icd     : num  0 162 163 527 150 163 334 160 420 12 ...
 $ exposure: num  5 5 10 9 0 2 0 0.5 0 0 ...
 $ dob     : num  1889 1886 1881 1886 1880 ...
 $ age1st  : num  17.5 23.2 25.2 24.7 30 ...
 $ agein   : num  45.2 48.3 53 47.9 54.7 ...
 $ ageout  : num  93 63.3 54.2 69.7 76.8 ...
 - attr(*, "time.scales")= chr  "per" "age" "tfh"
 - attr(*, "time.since")= chr  "" "" ""
 - attr(*, "breaks")=List of 3
  ..$ per: NULL
  ..$ age: NULL
  ..$ tfh: NULL
```

You can get an overview of the number of records by state and transitions between states as well as the person-years in each state by using `summary.Lexis`. Try to compute the rates per 1000 person-years, and print them:

```
> summary( nicL, scale=1000 )
Transitions:
     To
From    Alive D.oth D.lung  Records:  Events: Risk time:  Persons:
  Alive    47   495    137        679      632      15.35        679
```

More illustratively you can show the transitions:

```
> boxes( nicL, boxpos=TRUE, show.BE=TRUE, scale.R=1000 )
```

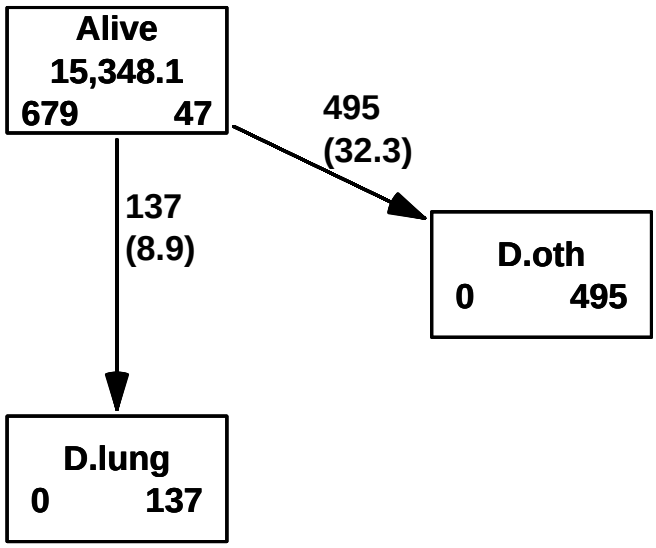

Figure 2.4: *Transitions from alive to different causes of death. Numbers in the middle of the* `Alive` *box is no.of person-years, number at the bottom of the boxes re the number of persons who start, resp. end their follow-up in the box (from* `show.BE=TRUE` *). The numbers on the arrows are no. of transitions and the average transition rates per 1000 PY (from* `scale.R=1000`*).* `./flup-nicB`

When we cut at a date as in this case, the date where cumulative exposure exceeds 50 exposure-years, we get the follow-up *after* the date classified as being in the new state if the exit (`lex.Xst`) was to a state we defined as one of the `precursor.states`:

```
> nicL$agehi <- nicL$age1st + 50 / nicL$exposure
> nicC <- cutLexis( data=nicL, cut=nicL$agehi, "age",
+                   new.state="HiExp", precursor.states="Alive" )
> subset( nicC, id %in% 8:10 )
         per     age     tfh lex.dur lex.Cst lex.Xst lex.id id icd exposure      dob
683 1934.246 47.9067 23.1861 21.7727   HiExp   D.oth      4  8 527        9 1886.340
```

```
5   1934.246 54.7465 24.7890 22.0977   Alive   D.oth      5  9 150        0 1879.500
6   1934.246 44.3314 23.0437  1.9563   Alive   HiExp      6 10 163        2 1889.915
685 1936.203 46.2877 25.0000 16.2536   HiExp  D.lung      6 10 163        2 1889.915
     age1st   agein  ageout    agehi
683 24.7206 47.9067 69.6794 30.27616
5   29.9575 54.7465 76.8442      Inf
6   21.2877 44.3314 62.5413 46.28770
685 21.2877 44.3314 62.5413 46.28770
```

```
> summary( nicC, scale=1000 )
```

```
Transitions:
     To
From    Alive HiExp D.oth D.lung  Records:  Events: Risk time:  Persons:
  Alive    39    83   279     65       466      427      10.77       466
  HiExp     0     8   216     72       296      288       4.58       296
  Sum      39    91   495    137       762      715      15.35       679
```

Note that the persons-years is the same, but that the number of events has changed. This is because events are now defined as any transition, including the transitions to `HiExp`.

Particularly when the number of states increase it is much more illustrative to show the different states in a diagram:

```
> boxes( nicC, boxpos=TRUE, show.BE=TRUE, scale.R=1000, pos.arr=0.7 )
```

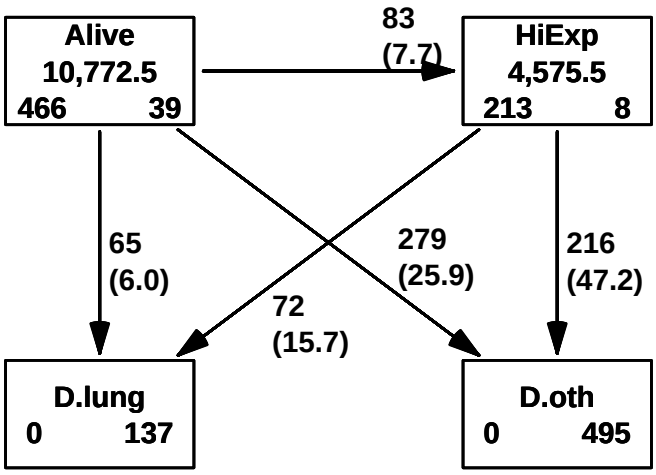

Figure 2.5: *Transitions between states of exposure and death. The number of events and rates are always shown on the* left *hand side of the arrows. You should consult the help page for* `boxes.Lexis` *to explore how the figure can be fine-tuned.* `./flup-nicC`

Chapter 3

Fundamental relations in survival and follow-up studies

The following is a summary of relations between various quantities used in analysis of follow-up studies. They are ubiquitous in the analysis and reporting of results. Hence it is important to be familiar with all of them and the relation between them.

3.1 Probability

Survival function:

$$\begin{aligned} S(t) &= \mathrm{P}\{\text{survival at least till } t\} \\ &= \mathrm{P}\{T > t\} = 1 - \mathrm{P}\{T \leq t\} = 1 - F(t) \end{aligned}$$

where T is the varaibel "time of death"

Conditional survival function:

$$\begin{aligned} S(t|t_{\text{entry}}) &= \mathrm{P}\{\text{survival at least till } t| \text{ alive at } t_{\text{entry}}\} \\ &= S(t)/S(t_{\text{entry}}) \end{aligned}$$

Cumulative distribution function of death times (cumulative risk):

$$\begin{aligned} F(t) &= \mathrm{P}\{\text{death before } t\} \\ &= \mathrm{P}\{T \leq t\} = 1 - S(t) \end{aligned}$$

Density function of death times:

$$f(t) = \lim_{h \to 0} \mathrm{P}\{\text{death in } (t, t+h)\}/h = \lim_{h \to 0} \frac{F(t+h) - F(t)}{h} = F'(t)$$

Intensity:

$$\begin{aligned}\lambda(t) &= \lim_{h\to 0} \mathrm{P}\{\text{event in } (t, t+h] \mid \text{alive at } t\}/h \\ &= \lim_{h\to 0} \frac{F(t+h)-F(t)}{S(t)h} = \frac{f(t)}{S(t)} \\ &= \lim_{h\to 0} -\frac{S(t+h)-S(t)}{S(t)h} = -\frac{\mathrm{d}\log S(t)}{\mathrm{d}t}\end{aligned}$$

The intensity is also known as the hazard function, hazard rate, mortality/morbidity rate or simply "rate".

Note that f and λ are *scaled* quantities, they have dimension time^{-1}.

Relationships between terms:

$$\begin{aligned}-\frac{\mathrm{d}\log S(t)}{\mathrm{d}t} &= \lambda(t) \\ &\Updownarrow \\ S(t) &= \exp\left(-\int_0^t \lambda(u)\,\mathrm{d}u\right) = \exp\bigl(-\Lambda(t)\bigr)\end{aligned}$$

The quantity $\Lambda(t) = \int_0^t \lambda(s)\,\mathrm{d}s$ is called the *integrated intensity* or the **cumulative rate**. It is *not* an intensity (rate), it is dimensionless, despite its name.

$$\lambda(t) = -\frac{\mathrm{d}\log(S(t))}{\mathrm{d}t} = -\frac{S'(t)}{S(t)} = \frac{F'(t)}{1-F(t)} = \frac{f(t)}{S(t)}$$

The cumulative *risk* of an event (to time t) is:

$$F(t) = \mathrm{P}\{\text{Event before time } t\} = \int_0^t \lambda(u)S(u)\,\mathrm{d}u = 1 - S(t) = 1 - \mathrm{e}^{-\Lambda(t)}$$

For small $|x|$ (< 0.05), we have that $1 - \mathrm{e}^{-x} \approx x$, so for small values of the integrated intensity:

$$\text{Cumulative risk to time } t \approx \Lambda(t) = \text{Cumulative rate}$$

3.2 Statistics

Likelihood contribution from follow up of one person:

The likelihood from a number of small pieces of follow-up from one individual is a product of conditional probabilities:

$$\begin{aligned}\mathrm{P}\{\text{event at } t_4 | \text{entry at } t_0\} &= \mathrm{P}\{\text{survive } (t_0,t_1) | \text{ alive at } t_0\} \times \\ &\quad \mathrm{P}\{\text{survive } (t_1,t_2) | \text{ alive at } t_1\} \times \\ &\quad \mathrm{P}\{\text{survive } (t_2,t_3) | \text{ alive at } t_2\} \times \\ &\quad \mathrm{P}\{\text{event at } t_4 | \text{ alive at } t_3\}\end{aligned}$$

Each term in this expression corresponds to one *empirical rate*[1]
$(d, y) = (\#\text{deaths}, \#\text{risk time})$, i.e. the data obtained from the follow-up of one person in the interval of length y. Each person can contribute many empirical rates, most with $d = 0$; d can only be 1 for the *last* empirical rate for a person.

Log-likelihood for one empirical rate (d, y):

$$\ell(\lambda) = \log\big(\mathrm{P}\{d \text{ events in } y \text{ follow-up time}\}\big) = d\log(\lambda) - \lambda y$$

This is under the assumption that the rate (λ) is constant over the interval that the empirical rate refers to.

Log-likelihood for several persons. Adding log-likelihoods from a group of persons (only contributions with identical rates) gives:

$$D\log(\lambda) - \lambda Y,$$

where Y is the total follow-up time, and D is the total number of failures.

Note: The Poisson log-likelihood for an observation D with mean λY is:

$$D\log(\lambda Y) - \lambda Y = D\log(\lambda) + D\log(Y) - \lambda Y$$

The term $D\log(Y)$ does not involve the parameter λ, so the likelihood for an observed rate can be maximized by pretending that the no. of cases D is Poisson with mean λY. But this does *not* imply that D follows a Poisson-distribution. It is entirely a likelihood based computational convenience. Anything that is not likelihood based is not justified.

A linear model for the log-rate, $\log(\lambda) = X\beta$ implies that

$$\lambda Y = \exp\big(\log(\lambda) + \log(Y)\big) = \exp\big(X\beta + \log(Y)\big)$$

Therefore, in order to get a linear model for $\log(\lambda)$ we must require that $\log(Y)$ appear as a variable in the model for $D \sim (\lambda Y)$ with the regression coefficient fixed to 1, a so-called *offset*-term in the linear predictor.

3.3 Competing risks

Competing risks: If there is more than one, say 3, causes of death, occurring with (cause-specific) rates λ_1, λ_2, λ_3, that is:

$$\lambda_c(a) = \lim_{h\to 0} \mathrm{P}\{\text{death from cause } c \text{ in } (a, a+h] \mid \text{alive at } a\}\,/h, \quad c = 1, 2, 3$$

The survival function is then:

$$S(a) = \exp\left(-\int_0^a \lambda_1(u) + \lambda_2(u) + \lambda_3(u)\,\mathrm{d}u\right)$$

[1] This is a concept coined by BxC, and so is not necessarily generally recognized.

because you have to escape all 3 causes of death. The probability of dying from cause 1 before age a (the cause-specific cumulative risk) is:

$$F_1(a) = \mathrm{P}\{\text{dead from cause 1 at } a\} = \int_0^a \lambda_1(u)S(u)\,\mathrm{d}u \neq 1 - \exp\left(-\int_0^a \lambda_1(u)\,\mathrm{d}u\right)$$

The term $\exp(-\int_0^a \lambda_1(u)\,\mathrm{d}u)$ is sometimes referred to as the "cause-specific survival", but it does not have any probabilistic interpretation in the real world. It is the survival under the assumption that only cause 1 existed and that the mortality rate from this cause was the same as when the other causes were present too.

Together with the survival function, the cause-specific cumulative risks represent a classification of the population at any time in those alive and those dead from causes 1, 2 and 3 respectively:

$$1 = S(a) + \int_0^a \lambda_1(u)S(u)\,\mathrm{d}u + \int_0^a \lambda_2(u)S(u)\,\mathrm{d}u + \int_0^a \lambda_3(u)S(u)\,\mathrm{d}u, \quad \forall a$$

Subdistribution hazard Fine and Gray defined models for the so-called subdistribution hazard, $\tilde{\lambda}_i(a)$. Recall the relationship between between the hazard (λ) and the cumulative risk (F):

$$\lambda(a) = -\frac{\mathrm{d}\log\big(S(a)\big)}{\mathrm{d}a} = -\frac{\mathrm{d}\log\big(1-F(a)\big)}{\mathrm{d}a}$$

When more competing causes of death are present the Fine and Gray idea is to use this transformation to the cause-specific cumulative risk for cause 1, say:

$$\tilde{\lambda}_1(a) = -\frac{\mathrm{d}\log\big(1-F_1(a)\big)}{\mathrm{d}a}$$

This is what is called the subdistribution hazard; as a function of $F_1(a)$ it depends on the survival function S, which depends on *all* the cause-specific hazards:

$$F_1(a) = \mathrm{P}\{\text{dead from cause 1 at } a\} = \int_0^a \lambda_1(u)S(u)\,\mathrm{d}u$$

The subdistribution hazard is merely a transformation of the cause-specific cumulative risk. Namely the same transformation which in the single-cause case transforms the cumulative risk to the hazard.

3.4 Demography

Expected residual lifetime: The expected lifetime (at birth) is simply the variable age (a) integrated with respect to the distribution of age at death:

$$\mathrm{EL} = \int_0^\infty a f(a)\,\mathrm{d}a$$

where f is the density of the distribution of lifetime (age at death).

The relation between the density f and the survival function S is $f(a) = -S'(a)$, so integration by parts gives:

$$\mathrm{EL} = \int_0^\infty a\big(-S'(a)\big)\,\mathrm{d}a = -\Big[aS(a)\Big]_0^\infty + \int_0^\infty S(a)\,\mathrm{d}a$$

The first of the resulting terms is 0 because $S(a)$ is 0 at the upper limit and a by definition is 0 at the lower limit.

Hence the expected lifetime can be computed as the integral of the survival function.

The expected *residual* lifetime at age a is calculated as the integral of the *conditional* survival function for a person aged a:

$$\mathrm{EL}(a) = \int_a^\infty S(u)/S(a)\,\mathrm{d}u$$

Lifetime lost due to a disease is the difference between the expected residual lifetime for a diseased person and a non-diseased (well) person at the same age. So all that is needed is a(n estimate of the) survival function in each of the two groups.

$$\mathrm{LL}(a) = \int_a^\infty S_{\mathrm{Well}}(u)/S_{\mathrm{Well}}(a) - S_{\mathrm{Diseased}}(u)/S_{\mathrm{Diseased}}(a)\,\mathrm{d}u$$

Note that the definition of the survival function for a non-diseased person requires a decision as to whether one will consider non-diseased persons immune to the disease in question or not. That is whether we will include the possibility of a well person getting ill and subsequently die. This does not show up in the formulae, but is a decision required in order to devise an estimate of S_{Well}.

Lifetime lost by cause of death is using the fact that the difference between the survival probabilities is the same as the difference between the death probabilities. If several causes of death (3, say) are considered then:

$$\begin{aligned} S(a) = 1 &- \mathrm{P}\{\text{dead from cause 1 at } a\} \\ &- \mathrm{P}\{\text{dead from cause 2 at } a\} \\ &- \mathrm{P}\{\text{dead from cause 3 at } a\} \end{aligned}$$

and hence:

$$\begin{aligned} S_{\mathrm{Well}}(a) - S_{\mathrm{Diseased}}(a) = &\ \mathrm{P}\{\text{dead from cause 1 at } a|\text{Diseased}\} \\ &+ \mathrm{P}\{\text{dead from cause 2 at } a|\text{Diseased}\} \\ &+ \mathrm{P}\{\text{dead from cause 3 at } a|\text{Diseased}\} \\ &- \mathrm{P}\{\text{dead from cause 1 at } a|\text{Well}\} \\ &- \mathrm{P}\{\text{dead from cause 2 at } a|\text{Well}\} \\ &- \mathrm{P}\{\text{dead from cause 3 at } a|\text{Well}\} \end{aligned}$$

So we can conveniently define the lifetime lost due to cause 2, say, by:

$$\begin{aligned} \mathrm{LL}_2(a) = \int_a^\infty &\ \mathrm{P}\{\text{dead from cause 2 at } u|\text{Diseased \& alive at } a\} \\ &- \mathrm{P}\{\text{dead from cause 2 at } u|\text{Well \& alive at } a\}\,\mathrm{d}u \end{aligned}$$

These quantities have the property that their sum is the total years of life lost due to the disease:

$$\mathrm{LL}(a) = \mathrm{LL}_1(a) + \mathrm{LL}_2(a) + \mathrm{LL}_3(a)$$

The terms in the integral are computed as (see the section on competing risks):

$$\mathrm{P}\{\text{dead from cause 2 at } x|\text{Diseased \& alive at } a\} = \int_a^x \lambda_{2,\mathrm{Dis}}(u) S_{\mathrm{Dis}}(u)/S_{\mathrm{Dis}}(a)\,\mathrm{d}u$$

$$\mathrm{P}\{\text{dead from cause 2 at } x|\text{Well \& alive at } a\} = \int_a^x \lambda_{2,\mathrm{Well}}(u) S_{\mathrm{Well}}(u)/S_{\mathrm{Well}}(a)\,\mathrm{d}u$$