The missing intensities in "Analysis of time-to-event for observational studies: Guidance to the use of intensity models"

or: Whatever became of the intercept

SDCC
http://bendixcarstensen.com/
April 2022
Version 2

Compiled Friday 11th March, 2022, 10:43
from: /home/bendix/teach/AdvCoh/art/STRATOS/commSTRATOS.tex

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

Contents

1 Introduction

The STRATOS topic group 8 has issued a paper with guidance on use of intensity models [1], "Analysis of time-to-event for observational studies: Guidance to the use of intensity models" (hereafter the "Intensity guide"). One remarkable thing about the intensity guide is the total absence of estimates and graphs of intensities.

This note demonstrates how the first example in the intensity guide on mortality of PAD patients can be analyzed using flexible parametric models for the intensities, leading to better insight in the underlying intensities and more intuitive testing of models, while maintaining the same results for the rate ratios and cumulative risks.

The flexible parametric modeling of the underlying intensities requires time-splitting of the data, which is facilitated by representation of follow-up in a `Lexis` object as provided by the `Epi` package. Also, the `Lexis` machinery allows simpler and more accurate data representation and more informative overview of the data.

2 Data

The intensity guide explains the amendment of data alongside the analysis; here we start by setting up the data as needed.

The following loads the relevant packages and the PAD datasets that were downloaded from the site indicated in the paper:

```
> library(Epi)
> library(popEpi)
> library(tidyverse)
> library(survival)
> clear()
> setwd("/home/bendix/teach/AdvCoh/art/STRATOS")
> load("pad.rda", v = T )
Loading objects:
  pad1
  pad2
  pad3
> str(pad1)
'data.frame':        1455 obs. of  23 variables:
 $ id             : int  1 2 3 4 5 6 7 8 9 10 ...
 $ site           : int  2 2 2 2 2 2 2 2 2 2 ...
 $ pad            : Factor w/ 2 levels "Control","PAD": 1 2 2 2 2 2 2 2 2 2 ...
 $ d.first        : Date, format: "2005-06-09" "2005-06-10" ...
 $ sex            : Factor w/ 2 levels "female","male": 2 1 2 2 1 2 2 1 2 2 ...
 $ d.birth        : Date, format: "1935-11-29" "1941-07-14" ...
 $ claud.yr       : int  NA 1990 1998 2001 2004 2004 1996 2001 2004 1997 ...
 $ claud.m        : int  NA 200 2000 300 250 500 200 100 300 500 ...
 $ limb.isch      : int  NA NA NA NA NA NA 2000 2003 NA NA ...
 $ diabetes       : num  0 0 0 0 1 0 0 0 1 0 ...
 $ yr.diab        : int  NA NA NA NA 2005 NA NA NA 1991 NA ...
 $ hyperlipidemia: num  1 1 1 1 1 1 1 1 1 1 ...
 $ yr.hyperl      : int  1997 1990 1993 NA 1999 2005 1996 1993 1991 1990 ...
 $ hypertension   : num  0 1 1 0 1 0 0 1 1 1 ...
 $ yr.hyperten    : int  NA 2000 1985 NA 2003 NA NA 1976 1986 1986 ...
 $ smoking        : Factor w/ 3 levels "never","current",..: 3 3 3 1 2 2 3 1 2 3 ...
 $ start.smoke    : int  1956 1960 1955 NA 1969 1988 1971 NA 1960 1954 ...
 $ end.smoke      : int  1966 2003 1984 NA NA NA 1998 NA NA 1998 ...
 $ cigarettes     : int  18 20 15 NA 20 20 25 NA 14 20 ...
 $ lastfu         : Date, format: "2011-12-08" "2011-11-07" ...
 $ futime         : num  2373 2341 2037 2412 2482 ...
```

```
 $ status        : Factor w/ 3 levels "censor","CV death",..: 1 3 1 1 1 1 1 2 3 1 ...
 $ age           : num  69.5 63.9 73.7 60.5 66.1 43 57.2 73.2 65.6 69.4 ...
```

```
> str(pad2)
```

```
'data.frame':        7797 obs. of  55 variables:
 $ id          : int  1 1 1 1 1 1 2 2 2 2 ...
 $ visit       : num  0 1 2 3 4 5 0 1 2 3 ...
 $ date        : Date, format: "2005-06-09" "2006-10-09" ...
 $ day         : num  0 487 876 1243 1597 ...
 $ pad         : Factor w/ 2 levels "Control","PAD": 1 1 1 1 1 1 2 2 2 2 ...
 $ abp.r       : num  1.17 1.22 1.25 1.07 1.17 ...
 $ abp.l       : num  1.17 1.13 1.25 1.04 1.15 ...
 $ anxiety     : int  3 2 3 2 2 2 3 4 3 2 ...
 $ sleep       : int  2 2 1 1 1 1 3 3 1 2 ...
 $ alcohol     : int  2 1 2 3 3 3 1 1 1 1 ...
 $ fruit       : int  4 4 4 3 4 4 3 3 4 4 ...
 $ vegetable   : int  4 4 4 4 4 4 3 3 3 4 ...
 $ active1     : int  7 7 7 6 7 7 4 0 7 7 ...
 $ active2     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ hlc1        : int  1 1 1 1 1 1 1 1 1 1 ...
 $ hlc2        : int  2 2 2 2 2 2 2 2 2 2 ...
 $ weight      : int  75 74 73 72 73 73 75 80 80 80 ...
 $ height      : int  173 173 172 171 171 170 152 152 152 151 ...
 $ bmi         : num  25.1 24.7 24.7 24.6 25 ...
 $ circum      : int  91 93 89 91 86 80 93 102 99 97 ...
 $ systolic    : int  130 120 130 122 132 120 148 128 170 170 ...
 $ diastolic   : int  70 70 72 70 68 63 74 70 70 95 ...
 $ dpulse.r    : num  1 1 1 0 1 1 0 0 0 1 ...
 $ dpulse.l    : num  1 1 1 1 1 1 1 0 0 0 1 ...
 $ tpulse.r    : num  1 1 1 1 1 1 1 0 0 0 0 ...
 $ tpulse.l    : num  1 1 1 1 1 1 1 0 0 0 0 ...
 $ erythrocytes: num  3.97 3.87 3.93 4.33 4.08 4.25 3.99 4.22 4.32 4.04 ...
 $ hemoglobin  : int  131 125 133 139 129 142 126 133 137 122 ...
 $ hematocrit  : num  39.1 37.6 39.5 40 38 42 36.8 38.4 38 35 ...
 $ wbc         : num  5.9 4.8 6.7 6 5.2 6.9 5.8 4.6 6.2 5.2 ...
 $ platelets   : int  202 146 184 197 167 173 145 121 191 180 ...
 $ glucose     : num  6.1 5.4 6.2 6.6 5.6 6.9 5.6 5.9 6.1 5.4 ...
 $ potassium   : num  4.2 4.9 4.2 3.9 4 4.6 3.7 4 4.3 3.8 ...
 $ urea        : num  6.8 6.8 5.2 4.2 6.5 5.3 7.2 9.1 9.6 8 ...
 $ creatine    : int  72 73 69 71 74 77 82 84 94 101 ...
 $ crp         : num  3.1 3.1 3 17 3 3 8.7 3.2 6 4 ...
 $ chol        : num  5.03 5.4 4.7 5.29 4.73 5.05 5.57 5.42 5.84 4.78 ...
 $ hdl         : num  1.65 1.78 2.27 2 2.26 2.51 0.97 1.22 1.26 1.07 ...
 $ ldl         : num  2.39 2.82 2.59 2.8 2.33 2.48 3 2.93 3.46 2.63 ...
 $ trig        : num  1.65 10.6 0.91 0.71 0.43 0.58 2 2.36 2.55 1.97 ...
 $ uprotein    : num  0 0 0 0 0 0 0 0 0 0 ...
 $ uglucose    : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cvscore     : num  0.09 0.087 0.1 0.102 0.119 ...
 $ coop.1      : int  1 2 2 3 2 3 3 4 5 3 ...
 $ coop.2      : int  2 2 1 3 2 2 3 4 3 3 ...
 $ coop.3      : int  1 2 1 2 2 2 2 3 2 3 ...
 $ coop.4      : int  1 1 1 3 1 1 1 3 2 3 ...
 $ coop.5      : int  2 3 3 2 3 2 2 3 2 3 ...
 $ coop.6      : int  3 3 3 4 3 3 2 3 3 3 ...
 $ aspirin     : num  1 1 1 1 1 1 1 1 1 1 ...
 $ clopridrogel: num  1 0 0 0 0 0 0 0 0 0 ...
 $ anticlot    : num  0 0 0 0 0 0 0 0 0 0 ...
 $ ezetimibe   : num  0 0 0 0 0 0 0 0 0 0 ...
 $ statins     : num  1 1 1 1 1 1 1 1 1 1 ...
 $ ace         : num  1 1 1 1 1 1 1 1 1 1 ...
```

```
> str(pad3)
```

```
'data.frame':        933 obs. of  6 variables:
 $ id   : int  2 2 5 6 6 6 7 7 8 8 ...
 $ date : Date, format: "2010-05-07" "2011-11-07" ...
 $ event: chr  "malignancy" "other death" "revascularization" "revascularization" ...
```

```
 $ type : chr  "other" "death" "minor" "minor" ...
 $ pad  : Factor w/ 2 levels "Control","PAD": 2 2 2 2 2 2 2 2 2 2 ...
 $ day  : num  1792 2341 1047 448 511 ...
```

As illustration along the way we will be printing select records from data frames with both calendar year and age as columns. It makes little sense to print both with the same number of significant digits, so here is a definition of a print method for `Lexis` objects that print all time scales of `Lexis` objects with 2 digits after the decimal separator:

```
> print.Lexis <-
+ function(Lx, d = 2)
+ {
+ tsx <- c(timeScales(Lx), "lex.dur")
+ Lx[,tsx] <- round(Lx[,tsx], d)
+ first.cols <- c("lex.id", tsx, "lex.Cst", "lex.Xst")
+  rest.cols <- setdiff(names(Lx), first.cols)
+ print.data.frame(Lx[,c(first.cols,rest.cols)], row.names = FALSE)
+ }
```

2.1 Follow-up: a `Lexis` object

With the datasets read we can set up the relevant parts of the datasets as a `Lexis` object [2, 3]. But first we convert all date variables to `cal.yr` format; this is basically just scaling them to units of 365.25 days (which we shall call "years"), formatted so that 1970.00 corresponds to 1970-1-1:

```
> pad1 <- cal.yr(pad1)
> pad2 <- cal.yr(pad2)
> pad3 <- cal.yr(pad3)
```

We extract the `hdl` and `ldl` at first visit from `pad2` and attach them as variables to `dat1` which is just a version of the `pad1` with fewer variables and a different name for the first level of `status`:

```
> dat2 <- ( filter(pad2, visit == 0)
+       %>% select("id", "hdl", "ldl")
+       %>% dplyr::rename(hdl0 = hdl, ldl0 = ldl)
+         )
> #
> dat1 <- ( select(pad1,
+                  c("id", "d.first", "d.birth", "lastfu",
+                    "status", "pad", "sex"))
+       %>% mutate(status = Relevel(status, list("Alive" = 1)))
+       %>% left_join(dat2)
+         )
> str(dat1)
'data.frame':        1455 obs. of  9 variables:
 $ id     : int  1 2 3 4 5 6 7 8 9 10 ...
 $ d.first: 'cal.yr' num  2005 2005 2005 2005 2005 ...
 $ d.birth: 'cal.yr' num  1936 1942 1932 1945 1939 ...
 $ lastfu : 'cal.yr' num  2012 2012 2011 2012 2012 ...
 $ status : Factor w/ 3 levels "Alive","CV death",..: 1 3 1 1 1 1 1 2 3 1 ...
 $ pad    : Factor w/ 2 levels "Control","PAD": 1 2 2 2 2 2 2 2 2 2 ...
 $ sex    : Factor w/ 2 levels "female","male": 2 1 2 2 1 2 2 1 2 2 ...
 $ hdl0   : num  1.65 0.97 2.33 1.32 1.12 1.02 1.21 1.67 1.08 1.1 ...
 $ ldl0   : num  2.39 3 2.36 2.38 3.59 3.87 2.76 2.89 2.66 1.8 ...

> head(dat1)
```

```
  id  d.first  d.birth   lastfu      status     pad    sex hdl0 ldl0
1  1 2005.436 1935.908 2011.933       Alive Control   male 1.65 2.39
2  2 2005.439 1941.532 2011.848 nonCV death     PAD female 0.97 3.00
3  3 2005.431 1931.749 2011.008       Alive     PAD   male 2.33 2.36
4  4 2005.417 1944.891 2012.021       Alive     PAD   male 1.32 2.38
5  5 2005.379 1939.279 2012.174       Alive     PAD female 1.12 3.59
6  6 2005.316 1962.293 2012.155       Alive     PAD   male 1.02 3.87
```

Then we can set up the follow-up in the cohort from the date `d.first` to the date `lastfu` as a `Lexis` object, with the exit status as indicated in the variable `status`. Also we show the summary table of transitions represented:

```
> table(dat1$status, exclude = NULL)
      Alive    CV death nonCV death
       1268          74         113
> Lx <- Lexis(entry = list(Per = d.first,
+                          Age = d.first - d.birth,
+                          tfE = 0),
+             exit = list(Per = lastfu),
+       exit.status = status,
+               id = id,
+             data = dat1) %>% select(-id)
NOTE: entry.status has been set to "Alive" for all.
> summary(Lx, t = T)
Transitions:
     To
From    Alive CV death nonCV death  Records:  Events: Risk time:  Persons:
  Alive  1268       74         113      1455      187    7817.45      1455

Timescales:
Per Age tfE
 ""  ""  ""
```

There is a total of 1455 persons in the study, and the amount of risk time is 7817 years, roughly 5 years of follow up on average.

We can also make an informative plot of the dataset—which is really a competing risks dataset:

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

2.2 Clinical variables

We will also need the clinical variables measured during follow-up attached to the `Lexis` object. To this end we put them in a data frame of the relevant structure and naming. The visit is not really a quantitative variable, but just a name so we designate it as a factor:

```
> clin <- ( select(pad2, c("id", "visit", "date", "hdl", "ldl"))
+       %>% rename("lex.id" = "id",
+                  "Per" = "date")
+       %>% mutate(visit = factor(paste0("v", visit)))
+         )
> str(clin)
'data.frame':        7797 obs. of  5 variables:
 $ lex.id: int  1 1 1 1 1 1 2 2 2 2 ...
 $ visit : Factor w/ 6 levels "v0","v1","v2",..: 1 2 3 4 5 6 1 2 3 4 ...
 $ Per   : 'cal.yr' num  2005 2007 2008 2009 2010 ...
 $ hdl   : num  1.65 1.78 2.27 2 2.26 2.51 0.97 1.22 1.26 1.07 ...
 $ ldl   : num  2.39 2.82 2.59 2.8 2.33 2.48 3 2.93 3.46 2.63 ...
```

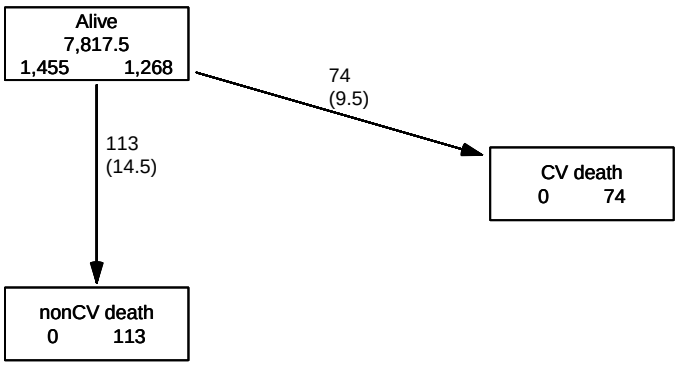

Figure 1: *Follow-up, events and (rates per 1000) in the* `pad1` *dataset. Person-years are indicated inside each box, number of events and transition rates per 1000 PY are shown on the arrows.* `./Sc-box1`

```
> head(clin, 8)
```

```
  lex.id visit      Per  hdl  ldl
1      1    v0 2005.436 1.65 2.39
2      1    v1 2006.769 1.78 2.82
3      1    v2 2007.834 2.27 2.59
4      1    v3 2008.839 2.00 2.80
5      1    v4 2009.808 2.26 2.33
6      1    v5 2010.745 2.51 2.48
7      2    v0 2005.439 0.97 3.00
8      2    v1 2006.695 1.22 2.93
```

The `rename` is needed because we need the id in the data set with clinical variables to have name `lex.id`, and the date to have the name of the corresponding time scale in `Lx`, in this case `Per`.

The values from this data frame can now be added to the follow-up in the `Lexis` object (and propagated by LOCF). We use a workaround via `subset` to purge the `cal.yr` class of variables in the datasets, in order to avoid clashes of variable types:

```
> La <- addCov.Lexis(subset(Lx),
+                    subset(clin),
+                           exnam = "visit",
+                             tfc = "tfV")
> summary(La, t = T)
```

```
Transitions:
     To
From    Alive CV death nonCV death  Records:  Events: Risk time:  Persons:
  Alive  6951       74         113      7138      187    7817.45      1455

Timescales:
Per Age tfE tfV
 ""  ""  ""  ""
```

```
> head(La)
```

```
 lex.id     Per   Age  tfE tfV lex.dur lex.Cst lex.Xst  d.first  d.birth   lastfu status
      1 2005.44 69.53 0.00   0    1.33   Alive   Alive 2005.436 1935.908 2011.933  Alive
      1 2006.77 70.86 1.33   0    1.07   Alive   Alive 2005.436 1935.908 2011.933  Alive
      1 2007.83 71.93 2.40   0    1.00   Alive   Alive 2005.436 1935.908 2011.933  Alive
```

```
     1 2008.84 72.93 3.40   0    0.97   Alive   Alive 2005.436 1935.908 2011.933  Alive
     1 2009.81 73.90 4.37   0    0.94   Alive   Alive 2005.436 1935.908 2011.933  Alive
     1 2010.74 74.84 5.31   0    1.19   Alive   Alive 2005.436 1935.908 2011.933  Alive
    pad  sex hdl0 ldl0 visit  hdl  ldl
Control male 1.65 2.39    v0 1.65 2.39
Control male 1.65 2.39    v1 1.78 2.82
Control male 1.65 2.39    v2 2.27 2.59
Control male 1.65 2.39    v3 2.00 2.80
Control male 1.65 2.39    v4 2.26 2.33
Control male 1.65 2.39    v5 2.51 2.48
```

We locate a couple of persons of interest to be used for illustration of the amended dataset and also later the time-split dataset.

```
> subset(La, lex.Xst=="CV death")[1:5,"lex.id"]
[1]  8 33 38 96 98
> subset(La, lex.Xst=="nonCV death")[1:5,"lex.id"]
[1]  2  9 11 14 40
> whv <- c("lex.id","Per","Age","tfE","tfV","lex.dur",
+          "lex.Cst","lex.Xst","visit","hdl0","ldl0","hdl","ldl")
> subset(La, lex.id %in% c(1,2,96))[,whv]
 lex.id     Per   Age  tfE tfV lex.dur lex.Cst     lex.Xst visit hdl0 ldl0  hdl  ldl
      1 2005.44 69.53 0.00   0    1.33   Alive       Alive    v0 1.65 2.39 1.65 2.39
      1 2006.77 70.86 1.33   0    1.07   Alive       Alive    v1 1.65 2.39 1.78 2.82
      1 2007.83 71.93 2.40   0    1.00   Alive       Alive    v2 1.65 2.39 2.27 2.59
      1 2008.84 72.93 3.40   0    0.97   Alive       Alive    v3 1.65 2.39 2.00 2.80
      1 2009.81 73.90 4.37   0    0.94   Alive       Alive    v4 1.65 2.39 2.26 2.33
      1 2010.74 74.84 5.31   0    1.19   Alive       Alive    v5 1.65 2.39 2.51 2.48
      2 2005.44 63.91 0.00   0    1.26   Alive       Alive    v0 0.97 3.00 0.97 3.00
      2 2006.70 65.16 1.26   0    1.09   Alive       Alive    v1 0.97 3.00 1.22 2.93
      2 2007.79 66.25 2.35   0    0.94   Alive       Alive    v2 0.97 3.00 1.26 3.46
      2 2008.72 67.19 3.29   0    1.00   Alive       Alive    v3 0.97 3.00 1.07 2.63
      2 2009.72 68.19 4.28   0    0.89   Alive       Alive    v4 0.97 3.00 1.13 1.86
      2 2010.61 69.08 5.17   0    1.24   Alive nonCV death    v5 0.97 3.00 1.21 2.29
     96 2005.32 74.17 0.00   0    1.20   Alive       Alive    v0 1.31 2.99 1.31 2.99
     96 2006.53 75.37 1.20   0    1.17   Alive       Alive    v1 1.31 2.99 1.19 3.30
     96 2007.69 76.54 2.37   0    4.46   Alive    CV death    v2 1.31 2.99 1.20 2.92
> summary(La, t = T)
Transitions:
     To
From    Alive CV death nonCV death  Records:  Events: Risk time:  Persons:
  Alive  6951       74         113      7138      187    7817.45      1455

Timescales:
Per Age tfE tfV
 ""  ""  ""  ""
```

We see that `tfV` (`t`ime `f`rom last `V`isit) has been added as new time scale. But it is not a time like the others; it is `NA` before the first visit and it is reset at every new clinical visit, so should be treated with care.

The `tfV` is reset to 0 at each clinical visit, so at the beginning of all of the intervals shown here it is necessarily 0.

2.3 Time splitting and restriction of follow-up

Analysis of intensities with parametric functions requires follow-up to be subdivided in small time intervals. In the analysis of rates we will use the values of the timescales

(current age, date, time since entry, time since last examination) at the beginning of each interval as covariates.

The splitting of time can be done either by `Epi::splitLexis` or `popEpi::splitMulti`. The latter returns a `data.table` unless otherwise instructed, as below. Here we split in intervals of 3 months after date of entry:

```
> summary(La)
Transitions:
     To
From    Alive CV death nonCV death  Records:  Events: Risk time:  Persons:
  Alive  6951       74         113      7138      187    7817.45      1455

> options("popEpi.datatable" = FALSE)
> Sa <- splitMulti(La, tfE = seq(0, 20, 1/4))
> summary(Sa)

Transitions:
     To
From    Alive CV death nonCV death  Records:  Events: Risk time:  Persons:
  Alive 37466       74         113     37653      187    7817.45      1455

> subset(Sa, lex.id == 1)[,whv]

 lex.id     Per   Age  tfE  tfV lex.dur lex.Cst lex.Xst visit hdl0 ldl0  hdl  ldl
      1 2005.44 69.53 0.00 0.00    0.25   Alive   Alive    v0 1.65 2.39 1.65 2.39
      1 2005.69 69.78 0.25 0.25    0.25   Alive   Alive    v0 1.65 2.39 1.65 2.39
      1 2005.94 70.03 0.50 0.50    0.25   Alive   Alive    v0 1.65 2.39 1.65 2.39
      1 2006.19 70.28 0.75 0.75    0.25   Alive   Alive    v0 1.65 2.39 1.65 2.39
      1 2006.44 70.53 1.00 1.00    0.25   Alive   Alive    v0 1.65 2.39 1.65 2.39
      1 2006.69 70.78 1.25 1.25    0.08   Alive   Alive    v0 1.65 2.39 1.65 2.39
      1 2006.77 70.86 1.33 0.00    0.17   Alive   Alive    v1 1.65 2.39 1.78 2.82
      1 2006.94 71.03 1.50 0.17    0.25   Alive   Alive    v1 1.65 2.39 1.78 2.82
      1 2007.19 71.28 1.75 0.42    0.25   Alive   Alive    v1 1.65 2.39 1.78 2.82
      1 2007.44 71.53 2.00 0.67    0.25   Alive   Alive    v1 1.65 2.39 1.78 2.82
      1 2007.69 71.78 2.25 0.92    0.15   Alive   Alive    v1 1.65 2.39 1.78 2.82
      1 2007.83 71.93 2.40 0.00    0.10   Alive   Alive    v2 1.65 2.39 2.27 2.59
      1 2007.94 72.03 2.50 0.10    0.25   Alive   Alive    v2 1.65 2.39 2.27 2.59
      1 2008.19 72.28 2.75 0.35    0.25   Alive   Alive    v2 1.65 2.39 2.27 2.59
      1 2008.44 72.53 3.00 0.60    0.25   Alive   Alive    v2 1.65 2.39 2.27 2.59
      1 2008.69 72.78 3.25 0.85    0.15   Alive   Alive    v2 1.65 2.39 2.27 2.59
      1 2008.84 72.93 3.40 0.00    0.10   Alive   Alive    v3 1.65 2.39 2.00 2.80
      1 2008.94 73.03 3.50 0.10    0.25   Alive   Alive    v3 1.65 2.39 2.00 2.80
      1 2009.19 73.28 3.75 0.35    0.25   Alive   Alive    v3 1.65 2.39 2.00 2.80
      1 2009.44 73.53 4.00 0.60    0.25   Alive   Alive    v3 1.65 2.39 2.00 2.80
      1 2009.69 73.78 4.25 0.85    0.12   Alive   Alive    v3 1.65 2.39 2.00 2.80
      1 2009.81 73.90 4.37 0.00    0.13   Alive   Alive    v4 1.65 2.39 2.26 2.33
      1 2009.94 74.03 4.50 0.13    0.25   Alive   Alive    v4 1.65 2.39 2.26 2.33
      1 2010.19 74.28 4.75 0.38    0.25   Alive   Alive    v4 1.65 2.39 2.26 2.33
      1 2010.44 74.53 5.00 0.63    0.25   Alive   Alive    v4 1.65 2.39 2.26 2.33
      1 2010.69 74.78 5.25 0.88    0.06   Alive   Alive    v4 1.65 2.39 2.26 2.33
      1 2010.74 74.84 5.31 0.00    0.19   Alive   Alive    v5 1.65 2.39 2.51 2.48
      1 2010.94 75.03 5.50 0.19    0.25   Alive   Alive    v5 1.65 2.39 2.51 2.48
      1 2011.19 75.28 5.75 0.44    0.25   Alive   Alive    v5 1.65 2.39 2.51 2.48
      1 2011.44 75.53 6.00 0.69    0.25   Alive   Alive    v5 1.65 2.39 2.51 2.48
      1 2011.69 75.78 6.25 0.94    0.25   Alive   Alive    v5 1.65 2.39 2.51 2.48

> subset(Sa, lex.id == 2)[,whv]

 lex.id     Per   Age  tfE  tfV lex.dur lex.Cst     lex.Xst visit hdl0 ldl0  hdl  ldl
      2 2005.44 63.91 0.00 0.00    0.25   Alive       Alive    v0 0.97    3 0.97 3.00
      2 2005.69 64.16 0.25 0.25    0.25   Alive       Alive    v0 0.97    3 0.97 3.00
      2 2005.94 64.41 0.50 0.50    0.25   Alive       Alive    v0 0.97    3 0.97 3.00
      2 2006.19 64.66 0.75 0.75    0.25   Alive       Alive    v0 0.97    3 0.97 3.00
      2 2006.44 64.91 1.00 1.00    0.25   Alive       Alive    v0 0.97    3 0.97 3.00
      2 2006.69 65.16 1.25 1.25    0.01   Alive       Alive    v0 0.97    3 0.97 3.00
      2 2006.70 65.16 1.26 0.00    0.24   Alive       Alive    v1 0.97    3 1.22 2.93
      2 2006.94 65.41 1.50 0.24    0.25   Alive       Alive    v1 0.97    3 1.22 2.93
```

```
      2 2007.19 65.66 1.75 0.49    0.25   Alive       Alive    v1 0.97    3 1.22 2.93
      2 2007.44 65.91 2.00 0.74    0.25   Alive       Alive    v1 0.97    3 1.22 2.93
      2 2007.69 66.16 2.25 0.99    0.10   Alive       Alive    v1 0.97    3 1.22 2.93
      2 2007.79 66.25 2.35 0.00    0.15   Alive       Alive    v2 0.97    3 1.26 3.46
      2 2007.94 66.41 2.50 0.15    0.25   Alive       Alive    v2 0.97    3 1.26 3.46
      2 2008.19 66.66 2.75 0.40    0.25   Alive       Alive    v2 0.97    3 1.26 3.46
      2 2008.44 66.91 3.00 0.65    0.25   Alive       Alive    v2 0.97    3 1.26 3.46
      2 2008.69 67.16 3.25 0.90    0.04   Alive       Alive    v2 0.97    3 1.26 3.46
      2 2008.72 67.19 3.29 0.00    0.21   Alive       Alive    v3 0.97    3 1.07 2.63
      2 2008.94 67.41 3.50 0.21    0.25   Alive       Alive    v3 0.97    3 1.07 2.63
      2 2009.19 67.66 3.75 0.46    0.25   Alive       Alive    v3 0.97    3 1.07 2.63
      2 2009.44 67.91 4.00 0.71    0.25   Alive       Alive    v3 0.97    3 1.07 2.63
      2 2009.69 68.16 4.25 0.96    0.03   Alive       Alive    v3 0.97    3 1.07 2.63
      2 2009.72 68.19 4.28 0.00    0.22   Alive       Alive    v4 0.97    3 1.13 1.86
      2 2009.94 68.41 4.50 0.22    0.25   Alive       Alive    v4 0.97    3 1.13 1.86
      2 2010.19 68.66 4.75 0.47    0.25   Alive       Alive    v4 0.97    3 1.13 1.86
      2 2010.44 68.91 5.00 0.72    0.17   Alive       Alive    v4 0.97    3 1.13 1.86
      2 2010.61 69.08 5.17 0.00    0.08   Alive       Alive    v5 0.97    3 1.21 2.29
      2 2010.69 69.16 5.25 0.08    0.25   Alive       Alive    v5 0.97    3 1.21 2.29
      2 2010.94 69.41 5.50 0.33    0.25   Alive       Alive    v5 0.97    3 1.21 2.29
      2 2011.19 69.66 5.75 0.58    0.25   Alive       Alive    v5 0.97    3 1.21 2.29
      2 2011.44 69.91 6.00 0.83    0.25   Alive       Alive    v5 0.97    3 1.21 2.29
      2 2011.69 70.16 6.25 1.08    0.16   Alive nonCV death    v5 0.97    3 1.21 2.29
> subset(Sa, lex.id == 96)[,whv]
 lex.id     Per   Age  tfE  tfV lex.dur lex.Cst  lex.Xst visit hdl0 ldl0  hdl  ldl
     96 2005.32 74.17 0.00 0.00    0.25   Alive    Alive    v0 1.31 2.99 1.31 2.99
     96 2005.57 74.42 0.25 0.25    0.25   Alive    Alive    v0 1.31 2.99 1.31 2.99
     96 2005.82 74.67 0.50 0.50    0.25   Alive    Alive    v0 1.31 2.99 1.31 2.99
     96 2006.07 74.92 0.75 0.75    0.25   Alive    Alive    v0 1.31 2.99 1.31 2.99
     96 2006.32 75.17 1.00 1.00    0.20   Alive    Alive    v0 1.31 2.99 1.31 2.99
     96 2006.53 75.37 1.20 0.00    0.05   Alive    Alive    v1 1.31 2.99 1.19 3.30
     96 2006.57 75.42 1.25 0.05    0.25   Alive    Alive    v1 1.31 2.99 1.19 3.30
     96 2006.82 75.67 1.50 0.30    0.25   Alive    Alive    v1 1.31 2.99 1.19 3.30
     96 2007.07 75.92 1.75 0.55    0.25   Alive    Alive    v1 1.31 2.99 1.19 3.30
     96 2007.32 76.17 2.00 0.80    0.25   Alive    Alive    v1 1.31 2.99 1.19 3.30
     96 2007.57 76.42 2.25 1.05    0.12   Alive    Alive    v1 1.31 2.99 1.19 3.30
     96 2007.69 76.54 2.37 0.00    0.13   Alive    Alive    v2 1.31 2.99 1.20 2.92
     96 2007.82 76.67 2.50 0.13    0.25   Alive    Alive    v2 1.31 2.99 1.20 2.92
     96 2008.07 76.92 2.75 0.38    0.25   Alive    Alive    v2 1.31 2.99 1.20 2.92
     96 2008.32 77.17 3.00 0.63    0.25   Alive    Alive    v2 1.31 2.99 1.20 2.92
     96 2008.57 77.42 3.25 0.88    0.25   Alive    Alive    v2 1.31 2.99 1.20 2.92
     96 2008.82 77.67 3.50 1.13    0.25   Alive    Alive    v2 1.31 2.99 1.20 2.92
     96 2009.07 77.92 3.75 1.38    0.25   Alive    Alive    v2 1.31 2.99 1.20 2.92
     96 2009.32 78.17 4.00 1.63    0.25   Alive    Alive    v2 1.31 2.99 1.20 2.92
     96 2009.57 78.42 4.25 1.88    0.25   Alive    Alive    v2 1.31 2.99 1.20 2.92
     96 2009.82 78.67 4.50 2.13    0.25   Alive    Alive    v2 1.31 2.99 1.20 2.92
     96 2010.07 78.92 4.75 2.38    0.25   Alive    Alive    v2 1.31 2.99 1.20 2.92
     96 2010.32 79.17 5.00 2.63    0.25   Alive    Alive    v2 1.31 2.99 1.20 2.92
     96 2010.57 79.42 5.25 2.88    0.25   Alive    Alive    v2 1.31 2.99 1.20 2.92
     96 2010.82 79.67 5.50 3.13    0.25   Alive    Alive    v2 1.31 2.99 1.20 2.92
     96 2011.07 79.92 5.75 3.38    0.25   Alive    Alive    v2 1.31 2.99 1.20 2.92
     96 2011.32 80.17 6.00 3.63    0.25   Alive    Alive    v2 1.31 2.99 1.20 2.92
     96 2011.57 80.42 6.25 3.88    0.25   Alive    Alive    v2 1.31 2.99 1.20 2.92
     96 2011.82 80.67 6.50 4.13    0.25   Alive    Alive    v2 1.31 2.99 1.20 2.92
     96 2012.07 80.92 6.75 4.38    0.08   Alive CV death    v2 1.31 2.99 1.20 2.92
```

We see that the values of `hdl0`, `ldl0`, `hdl` and `ldl` have been carried correctly forward across the splits, and that `tfV` now shows how long the beginning of each interval is from the date of the most recent clinical measurement.

In the intensity guide, the follow-up after 5 years is censored, so for comparability we do this here too. This is actually one of the reasons that we split the time along the `tfE` axis; it makes it easy to restrict to the first 5 years of follow-up; the clause `tfE < 5` has the desired effect because there is a split at precisely 5 years of follow-up:

```
> Sx <- subset(Sa, tfE < 5)
> summary(Sx$tfE + Sx$lex.dur)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.1478  1.2500  2.5000  2.5734  3.7500  5.0000
```

We see that we do have follow-up till precisely 5 years.

2.4 Intermediate event: CV

We shall also be concerned with CV events, defined as occurrence of stroke or myocardial infarction. To this end we need a data frame with dates of CV occurrences. In `Lexis` terms it is a data frame with variables `lex.id`, `cut` (time of event) and `new.state` (name of event); note that in the dataset `pad3` a person may have more than one event, but we only use the first of these:

```
> addmargins(with(pad3, table(event, type)))
                    type
event                death major minor other Sum
  amputation             0    26     0     0  26
  cvd death             74     0     0     0  74
  infarction             0    77     0     0  77
  malignancy             0     0     0   184 184
  other death          113     0     0     0 113
  revascularization      0     0   358    57 415
  stroke                 0    44     0     0  44
  Sum                  187   147   358   241 933
```

```
> with(pad3, addmargins(table(table(id))))
```

```
  1   2   3   4   5   6   7   8  10 Sum
275 145  56  19  13   1   5   1   1 516
```

```
> cv <- as.data.frame( filter(pad3,
+                             event %in% c("stroke","infarction"))
+                  %>% select(id, date)
+                  %>% arrange(id, date)
+                  %>% group_by(id)
+                  %>% summarize(first(date))
+                  %>% rename(lex.id = id,
+                             cut = "first(date)")
+                  %>% mutate(new.state = "CV")
+                    )
> str(cv)
```

```
'data.frame':        97 obs. of  3 variables:
 $ lex.id   : int  8 9 19 45 48 50 67 72 81 82 ...
 $ cut      : num  2007 2008 2011 2007 2011 ...
 $ new.state: chr  "CV" "CV" "CV" "CV" ...
```

```
> head(cv)
```

```
  lex.id      cut new.state
1      8 2006.589        CV
2      9 2008.111        CV
3     19 2010.635        CV
4     45 2007.095        CV
5     48 2010.797        CV
6     50 2006.953        CV
```

```
> with(cv, addmargins(table(table(lex.id))))
```

```
  1 Sum
 97  97
```

We can then cut the follow-up at the intermediate event `CV` to form a proper multistate model, and show the events. If we have more than a few states, a plot of transitions helps to get an overview of the data:

```
> par(mfrow = c(1,2))
> summary(Sx)
Transitions:
     To
From    Alive CV death nonCV death  Records:  Events: Risk time:  Persons:
  Alive 32750       68          91     32909      159    6870.69      1455

> boxes(Sx, boxpos = list(x = c(20,80,80),
+                         y = c(80,20,80)),
+           scale.R = 1000,
+           show.BE = "nz",
+           pos.arr = 0.3, cex=1.1)
> Sc <- cutLexis(Sx, cut = cv)
> summary(Sc)

Transitions:
     To
From    Alive   CV CV death nonCV death  Records:  Events: Risk time:  Persons:
  Alive 31834   84       59          85     32062      228    6689.38      1455
  CV        0  916        9           6       931       15     181.31        84
  Sum   31834 1000       68          91     32993      243    6870.69      1455

> boxes(Sc, boxpos = list(x = c(20,20,80,80),
+                         y = c(80,20,20,80)),
+           scale.R = 1000,
+           show.BE = "nz",
+           pos.arr = 0.3, cex=1.1)
```

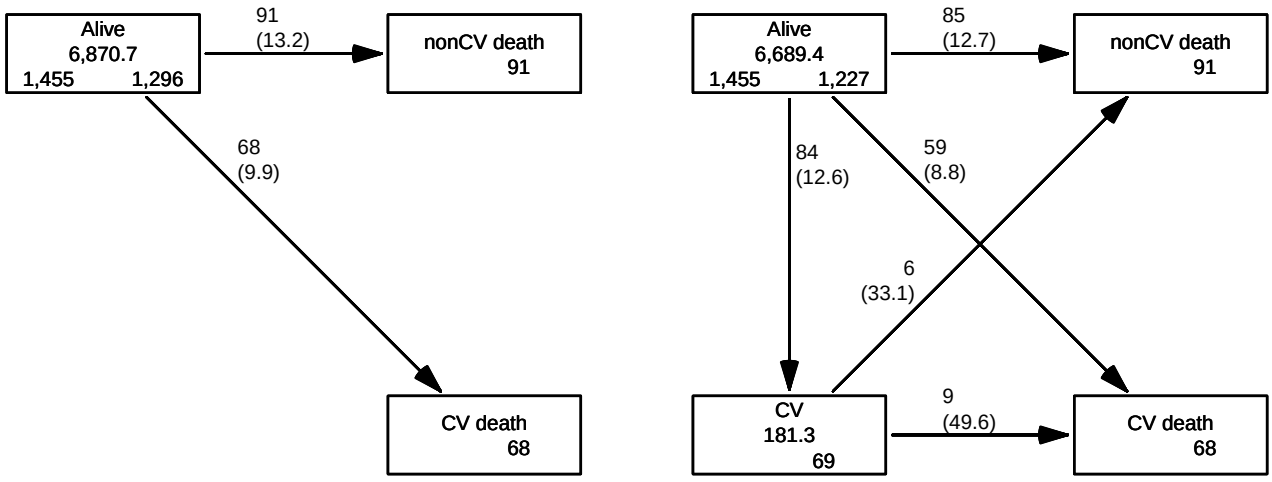

Figure 2: *Multistate models, without (dataset* `Sx`*, left) and with CV event (dataset* `Sc`*, right), defined by date of first occurrence of either stroke or infarction. Follow-up censored at 5 years, hence the difference to figure 1. There are fewer CV events than in the* `cv` *dataset, because some occur more than 5 years after entry, where we censored follow-up.* `./Sc-boxes3`

We now have the possibility of modeling CV events either separately or as "either `CV` or `CV death`".

The `Lexis` object `Sx` represents follow up in the simple setting with two competing risks and censored at 5 years after entry; the object `Sc` represents the same follow-up, but with follow-up sub-divided by CV occurrence.

So now we have a time-split `Lexis` object that we can use as basis for the modeling.

3 Modeling mortality

The `Lexis` object has information on beginning and end of follow up, state transitions etc. This is exploited in the function `coxph.Lexis`, allowing for a simple specification of the model; note that we are using `Sx`, the three-state data in figure 2; first the model with `tfE` as baseline time scale:

```
> ct <- coxph.Lexis(Sx, tfE ~ pad + sex + I(Age / 10))
survival::coxph analysis of Lexis object Sx:
Rates for transitions Alive->CV death, Alive->nonCV death
Baseline timescale: tfE
```

```
> summary(ct)
```

```
Call:
coxph(formula = as.formula(paste("Sobj", as.character(formula[3]),
    sep = "~")), data = Lx)

  n= 32909, number of events= 159

             coef exp(coef) se(coef)     z Pr(>|z|)
padPAD     0.8756    2.4002   0.1739 5.036 4.75e-07
sexmale    0.6928    1.9992   0.1822 3.802 0.000143
I(Age/10) 0.6600    1.9348   0.1054 6.261 3.84e-10

          exp(coef) exp(-coef) lower .95 upper .95
padPAD        2.400     0.4166     1.707     3.375
sexmale       1.999     0.5002     1.399     2.857
I(Age/10)     1.935     0.5168     1.574     2.379

Concordance= 0.701  (se = 0.02 )
Likelihood ratio test= 82.81  on 3 df,   p=<2e-16
Wald test            = 73.39  on 3 df,   p=8e-16
Score (logrank) test = 75.94  on 3 df,   p=2e-16
```

```
> round(ci.exp(ct), 2)
```

```
          exp(Est.) 2.5% 97.5%
padPAD         2.40 1.71  3.37
sexmale        2.00 1.40  2.86
I(Age/10)      1.93 1.57  2.38
```

. . . then the model with `Age` as baseline:

```
> ca <- coxph.Lexis(Sx, Age ~ pad + sex + tfE)
survival::coxph analysis of Lexis object Sx:
Rates for transitions Alive->CV death, Alive->nonCV death
Baseline timescale: Age
```

```
> round(ci.exp(ca), 2)
```

```
        exp(Est.) 2.5% 97.5%
padPAD       2.40 1.70  3.37
sexmale      2.02 1.42  2.90
tfE          1.18 1.05  1.33
```

The small discrepancies from the results in the intensity guide presumably comes from the 5-year censoring which may not be quite the same, in the Lexis object we censored at 5 years in the `cal.yr` sense, that is $365.25 \times 5 = 1826.25$ days after entry, in the analysis in the paper this was at 1825 days, using 365 days as the length of a year.

The `coxph.Lexis` is just a wrapper for `coxph`; the following two statements are equivalent:

```
> ca <- coxph.Lexis(Sx, Age ~ pad + sex + tfE)
> ca <- coxph(Surv(Age, Age + lex.dur, lex.Cst == "Alive" &
+                                      lex.Xst %in% c("CV death",
+                                                     "nonCV death") &
+                                      lex.Xst != lex.Cst)
+                           ~ pad + sex + tfE,
+                   data = Sx)
```

The first one is more handy once the `Lexis` object has been defined; further details is at the help page for `coxph.Lexis`.

3.1 Parametric models

The same models as the Cox-models can be fitted using an explicit smoothing for the baseline effect; we will get the same covariate effect estimates in all models.

To this end we use natural splines to model the non-linear effects of the timescales. Note that this is where the time-split data is required; the underlying model really assumes that mortality rates are constant in each small interval.

We fit models with the explanatory variables `pad` and `sex` and `tfE` as baseline (`mt`) and linear effect of `Age`, with `Age` as baseline (`ma`) and a linear effect of `tfE`, both as (non-linear) time scales (`m2`), including a non-linear effect of the difference (`m3`) and including a general interaction between `tfE` and `Age` (`mi`).

As for the `coxph` we also have a wrapper for fitting Poisson `glm` models to `Lexis` objects; we just need some reasonably spaced knots to define the spline effects:

```
> (knE <- seq(0, 4.5, 1.5))
[1] 0.0 1.5 3.0 4.5
> (knA <- c(4:8) * 10)
[1] 40 50 60 70 80
> mt <- glm.Lexis(Sx, ~ Ns(tfE, kn=knE) + pad + sex + I(Age / 10))
stats::glm Poisson analysis of Lexis object Sx with log link:
Rates for transitions: Alive->CV death, Alive->nonCV death
> ma <- glm.Lexis(Sx, ~ Ns(Age, kn=knA) + pad + sex + tfE)
stats::glm Poisson analysis of Lexis object Sx with log link:
Rates for transitions: Alive->CV death, Alive->nonCV death
> m2 <- glm.Lexis(Sx, ~ Ns(Age, kn=knA) +
+                       Ns(tfE, kn=knE) + pad + sex)
stats::glm Poisson analysis of Lexis object Sx with log link:
Rates for transitions: Alive->CV death, Alive->nonCV death
> m3 <- glm.Lexis(transform(Sx, Ain = Age - tfE),
+                     ~ Ns(Age, kn=knA) +
+                       Ns(tfE, kn=knE) +
+                       Ns(Ain, kn=knA) + pad + sex)
stats::glm Poisson analysis of Lexis object transform(Sx, Ain = Age - tfE) with log link:
Rates for transitions: Alive->CV death, Alive->nonCV death
```

```
> mi <- glm.Lexis(Sx, ~ Ns(Age, kn=knA) + Ns(tfE, kn=knE) +
+                       Ns(Age, kn=knA) : Ns(tfE, kn=knE) + pad + sex)

stats::glm Poisson analysis of Lexis object Sx with log link:
Rates for transitions: Alive->CV death, Alive->nonCV death

> round(ci.exp(mt, subset=c("pad","sex","Age")), 2)

          exp(Est.) 2.5% 97.5%
padPAD         2.40 1.71  3.38
sexmale        2.00 1.40  2.86
I(Age/10)      1.93 1.57  2.38

> round(ci.exp(ma, subset=c("pad","sex","tfE")), 2)

        exp(Est.) 2.5% 97.5%
padPAD       2.39 1.70  3.36
sexmale      2.02 1.41  2.89
tfE          1.18 1.05  1.32

> round(ci.exp(m2, subset=c("pad","sex")), 2)

        exp(Est.) 2.5% 97.5%
padPAD       2.39 1.70  3.36
sexmale      2.02 1.41  2.89

> round(ci.exp(m3, subset=c("pad","sex")), 2)

        exp(Est.) 2.5% 97.5%
padPAD       2.38 1.70  3.35
sexmale      2.02 1.41  2.89

> round(ci.exp(mi, subset=c("pad","sex")), 2)

        exp(Est.) 2.5% 97.5%
padPAD       2.38 1.70  3.35
sexmale      2.02 1.41  2.89
```

For all model fits a note is issued that the model fitted comprises more than one transition from the same state (`Alive`), and therefore the analysis is equivalent to analysis of all-cause mortality. This is precisely what we want; we are modeling the all-cause mortality.

A more flexible class of models, also a bit simpler to specify are `gam` (generalized additive) `m`odels that fits the non-linear effects using `penalized` splines. For there models there is a smoother called "`s`", that will give thin

```
> Mt <- gam.Lexis(Sx, ~ s(tfE, k=10) + pad + sex + I(Age / 10))
mgcv::gam Poisson analysis of Lexis object Sx with log link:
Rates for transitions: Alive->CV death, Alive->nonCV death

> Ma <- gam.Lexis(Sx, ~ s(Age, k=10) + pad + sex + tfE)

mgcv::gam Poisson analysis of Lexis object Sx with log link:
Rates for transitions: Alive->CV death, Alive->nonCV death

> M2 <- gam.Lexis(Sx, ~ s(Age, k=10) +
+                       s(tfE, k=10) + pad + sex)

mgcv::gam Poisson analysis of Lexis object Sx with log link:
Rates for transitions: Alive->CV death, Alive->nonCV death

> M3 <- gam.Lexis(transform(Sx, Ain = Age - tfE),
+                     ~ s(Age, k=10) +
+                       s(tfE, k=10) +
+                       s(Ain, k=10) + pad + sex)

mgcv::gam Poisson analysis of Lexis object transform(Sx, Ain = Age - tfE) with log link:
Rates for transitions: Alive->CV death, Alive->nonCV death

> Mi <- gam.Lexis(Sx, ~ s(Age, tfE, k=20) + pad + sex)

mgcv::gam Poisson analysis of Lexis object Sx with log link:
Rates for transitions: Alive->CV death, Alive->nonCV death
```

```
> round(ci.exp(Mt, subset=c("pad","sex","Age")), 2)

           exp(Est.) 2.5% 97.5%
padPAD          2.40 1.71  3.38
sexmale         2.00 1.40  2.86
I(Age/10)       1.93 1.57  2.38

> round(ci.exp(Ma, subset=c("pad","sex","tfE")), 2)

         exp(Est.) 2.5% 97.5%
padPAD        2.40 1.71  3.38
sexmale       2.02 1.41  2.89
tfE           1.18 1.06  1.33

> round(ci.exp(M2, subset=c("pad","sex")), 2)

         exp(Est.) 2.5% 97.5%
padPAD        2.40 1.71  3.38
sexmale       2.02 1.41  2.89

> round(ci.exp(M3, subset=c("pad","sex")), 2)

         exp(Est.) 2.5% 97.5%
padPAD        2.40 1.71  3.37
sexmale       2.02 1.41  2.89

> round(ci.exp(Mi, subset=c("pad","sex")), 2)

         exp(Est.) 2.5% 97.5%
padPAD        2.40 1.71  3.38
sexmale       2.02 1.41  2.88
```

The spline models involve no penalization, so quite a bit quicker to fit than the `gam` models, but the price to pay is that we need to specify knots for the splines.

We can see that the estimates of the covariate effects are practically the same as in the Cox models regardless of how we choose the shape of the baseline hazard.

3.2 The Poisson model

In the intensity guide [1], a model with non-linear effects of two timescales is fitted, using a Poisson model for the counts and person-years in bins of 1 year of FU and 5 years of current age. The model assumes the baseline to be constant in each bin, without any restrictions on the intensities in the bins, so it violates the general advice of not grouping continuous variables.

Even if the model conceptually purports to be the union of the two models with different underlying time scales, there is no way to compare the two models to the Poisson model; they are not nested.

We can fit the Poisson model from the intensity guide using our `Sx` data frame. However, the follow-up data was only split by `tfE`, so in order to accommodate the grouping of age in 5-year bins of age we need to split by age in 5-year intervals too:

```
> Sp <- splitMulti(Sx, Age = seq(0,120,5))
> summary(Sx) ; summary(Sp)
Transitions:
     To
From    Alive CV death nonCV death  Records:  Events: Risk time:  Persons:
  Alive 32750       68          91     32909      159    6870.69      1455

Transitions:
     To
From    Alive CV death nonCV death  Records:  Events: Risk time:  Persons:
  Alive 34130       68          91     34289      159    6870.69      1455
```

```
> Sp$A <- factor(floor(Sp$Age / 5) * 5)
> Sp$E <- factor(floor(Sp$tfE))
> mp <- glm.Lexis(Sp, ~ A * E + pad + sex)
stats::glm Poisson analysis of Lexis object Sp with log link:
Rates for transitions: Alive->CV death, Alive->nonCV death
> round(ci.exp(mp, subset = c("pad","sex")), 2)
        exp(Est.) 2.5% 97.5%
padPAD       2.38 1.70  3.35
sexmale      2.01 1.41  2.88
```

We see that we again get the same regression parameters as in the paper but again at the expense of a *lot* of baseline parameters; half of which have a standard error exceeding 1000 (meaning that the parameter numerically is infinity):

```
> length(coef(mp))
[1] 57
> table(ci.lin(mp)[,2] > 1000)
FALSE  TRUE
   29    28
```

So the grouped Poisson model is just as useless as the Cox-model in providing information on the shape of the baseline intensities.

If we look at the amount of information in the bins it is quite scarce:

```
> print(      xtabs((lex.Cst != lex.Xst) ~ E + A, data = Sp)    ,
+       zero = ".")
   A
E   35 40 45 50 55 60 65 70 75 80 85
  0  .  .  .  1  1  2  4  2  4  1  .
  1  .  .  .  .  2  2  5  7 12  2  .
  2  .  .  .  5  3  6  6  4  6  6  .
  3  .  .  .  .  3  2  7  7  8 10  .
  4  .  .  .  .  2  5  5  9 10 10  .
> print(round(xtabs( lex.dur             ~ E + A, data = Sp), 1),
+       zero = ".")
   A
E      35    40    45    50    55    60    65    70    75    80    85
  0   2.5  20.7  46.2 149.2 201.3 224.8 300.2 280.3 207.3  15.3     .
  1   0.2  18.2  36.2 133.6 188.9 207.7 291.8 268.4 235.5  40.5     .
  2     .  11.4  32.7 108.3 186.2 185.0 274.2 270.4 242.0  77.6     .
  3     .   7.1  29.0  90.1 169.4 173.1 244.6 270.2 248.0 108.9     .
  4     .   5.2  24.8  62.2 149.9 177.5 206.2 281.6 235.9 129.5   0.5
```

The dots in the last table are bins with less than 0.05 years of total risk time (rounded to 0). We see that the amount of information on each of the rates is scarce; only about half of the bins have any events; this is why half of the parameter estimates will be $-\infty$.

3.3 Assessing the model with two timescales

With parametric models based on finely split follow-up intervals, the union of the two models `mt` and `ma` is easily fitted (`m2`), and therefore it is also straight-forward to assess the relative importance of a non-linear effect of the two time-scales (of course under the assumption of the additivity of the two time-scale effects), because the model with two non-linear time scales contains both of the models as sub-models, so the two tests are just ordinary likelihood-ratio tests in `glms`:

```
> anova(mt, m2, ma, test = "Chisq")
Analysis of Deviance Table

Model 1: cbind(trt(Lx$lex.Cst, Lx$lex.Xst) %in% trnam, Lx$lex.dur) ~ Ns(tfE,
    kn = knE) + pad + sex + I(Age/10)
Model 2: cbind(trt(Lx$lex.Cst, Lx$lex.Xst) %in% trnam, Lx$lex.dur) ~ Ns(Age,
    kn = knA) + Ns(tfE, kn = knE) + pad + sex
Model 3: cbind(trt(Lx$lex.Cst, Lx$lex.Xst) %in% trnam, Lx$lex.dur) ~ Ns(Age,
    kn = knA) + pad + sex + tfE
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1     32902     1887.9
2     32899     1881.5  3   6.3657  0.09511
3     32901     1885.3 -2  -3.7949  0.14995
```

```
> anova(Mt, M2, Ma, test = "Chisq")
Analysis of Deviance Table

Model 1: cbind(trt(Lx$lex.Cst, Lx$lex.Xst) %in% trnam, Lx$lex.dur) ~ s(tfE,
    k = 10) + pad + sex + I(Age/10)
Model 2: cbind(trt(Lx$lex.Cst, Lx$lex.Xst) %in% trnam, Lx$lex.dur) ~ s(Age,
    k = 10) + s(tfE, k = 10) + pad + sex
Model 3: cbind(trt(Lx$lex.Cst, Lx$lex.Xst) %in% trnam, Lx$lex.dur) ~ s(Age,
    k = 10) + pad + sex + tfE
  Resid. Df Resid. Dev      Df Deviance Pr(>Chi)
1     32903     1888.7
2     32899     1883.0  3.0778   5.7165   0.1328
3     32901     1885.9 -1.3775  -2.8806   0.1399
```

There is not much difference in the testing outcome between the `glm` and `gam` approaches, so for the remainder we stick to the `glm`.

The first test is comparing with the model with linear effect of `Age`, the second with the model with linear effect of `tfE`. None of them are significantly worse than the model with two time scales, so we might explore the model with linear effects of both:

```
> m0 <- glm.Lexis(Sx, ~ tfE + pad + sex + I(Age / 10))
stats::glm Poisson analysis of Lexis object Sx with log link:
Rates for transitions: Alive->CV death, Alive->nonCV death
```

```
> anova(mt, m0, ma, test = "Chisq")
Analysis of Deviance Table

Model 1: cbind(trt(Lx$lex.Cst, Lx$lex.Xst) %in% trnam, Lx$lex.dur) ~ Ns(tfE,
    kn = knE) + pad + sex + I(Age/10)
Model 2: cbind(trt(Lx$lex.Cst, Lx$lex.Xst) %in% trnam, Lx$lex.dur) ~ tfE +
    pad + sex + I(Age/10)
Model 3: cbind(trt(Lx$lex.Cst, Lx$lex.Xst) %in% trnam, Lx$lex.dur) ~ Ns(Age,
    kn = knA) + pad + sex + tfE
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1     32902     1887.9
2     32904     1891.6 -2  -3.6978  0.15741
3     32901     1885.3  3   6.2685  0.09925
```

These are also tests of non-linearity of the time effect and age-effects respectively, but versus a model where both are linear. The tests produce almost the same p-values as when comparing to the additive model.

It can be debated whether formal testing of linearity of time scale effects is of any relevance, but it has been included here because the fitting of the different models in table 1 in the intensity guide seem to indicate that linearity of the effect of one or the other of the time scales may be of interest, albeit not explicitly formulated as such. Modeling the intensities reduces this to a trivial testing problem in generalized linear models.

3.4 Testing proportionality

Above, we also expanded the model with an interaction term between the to variables in the guise of a non-linear term in the difference, age at entry (`m3`) as well as with a more flexible interaction between the two variables (`mi`), so testing for non-proportionality can be split in two:

```
> anova(m2, m3, mi, test = "Chisq")
Analysis of Deviance Table

Model 1: cbind(trt(Lx$lex.Cst, Lx$lex.Xst) %in% trnam, Lx$lex.dur) ~ Ns(Age,
    kn = knA) + Ns(tfE, kn = knE) + pad + sex
Model 2: cbind(trt(Lx$lex.Cst, Lx$lex.Xst) %in% trnam, Lx$lex.dur) ~ Ns(Age,
    kn = knA) + Ns(tfE, kn = knE) + Ns(Ain, kn = knA) + pad +
    sex
Model 3: cbind(trt(Lx$lex.Cst, Lx$lex.Xst) %in% trnam, Lx$lex.dur) ~ Ns(Age,
    kn = knA) + Ns(tfE, kn = knE) + Ns(Age, kn = knA):Ns(tfE,
    kn = knE) + pad + sex
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1     32899     1881.5
2     32896     1879.7  3   1.8578   0.6024
3     32887     1869.3  9  10.4245   0.3172
```

We see that there is little evidence for an interaction of any kind between the time scales. So we found no "violation of the proportionality assumption".

3.5 Predicted mortality rates

Both the Cox-models and the fitted grouped Poisson model use parametrizations of the time scales with one parameter per interval — a so-called exchangeable model that does not exploit the quantitative nature of the timescales. This is strongly at variance with normal biostatistical recommendations against grouping of variables.

We should not entirely rely on p-values in the choice of models, it is of relevance to *quantify* the effects of the two time scales. Unlike the Cox-model, the Poisson models with smooth parametric effects allows us to inspect the shape of the baseline hazard.

We can plot the hazard rates for men entering at age, say, 65 and followed till age 70. We just set up a prediction data frame (`nd`), indicating the points where we want the predicted values of the rates; that will allow us to inspect the predicted rates under the models with one non-linear time scale and the model with two. Note that since we specify the follow-up in `lex.dur` in years, the rates returned by `ci.pred` will be in units of events per 1 year, so we multiply by 1000 to get rates per 1000 PY.

```
> nd <- data.frame(tfE = seq(0, 5, 0.1),
+                  sex = "male",
+                  pad = "PAD",
+                  Age = seq(0, 5, 0.1) + 65)
> par(mfrow=c(1,2))
> matshade(nd$tfE, cbind(ci.pred(mt, nd),
+                        ci.pred(m2, nd)) * 1000,
+          lty = 1:2, lwd = 2,
+          xlab = "Time from entry (years)",
+          ylab = "Mortality rate per 1000 PY",
+          ylim = c(5, 100), log = "y", alpha = c(0.15,0.10),
+          plot = TRUE)
> matshade(nd$Age, cbind(ci.pred(ma, nd),
+                        ci.pred(m2, nd)) * 1000,
+          lty = 1:2, lwd = 2, xlab = "Age (years)", ylab = "",
+          ylim = c(5, 100), log = "y", alpha = c(0.15,0.10),
+          plot = TRUE)
```

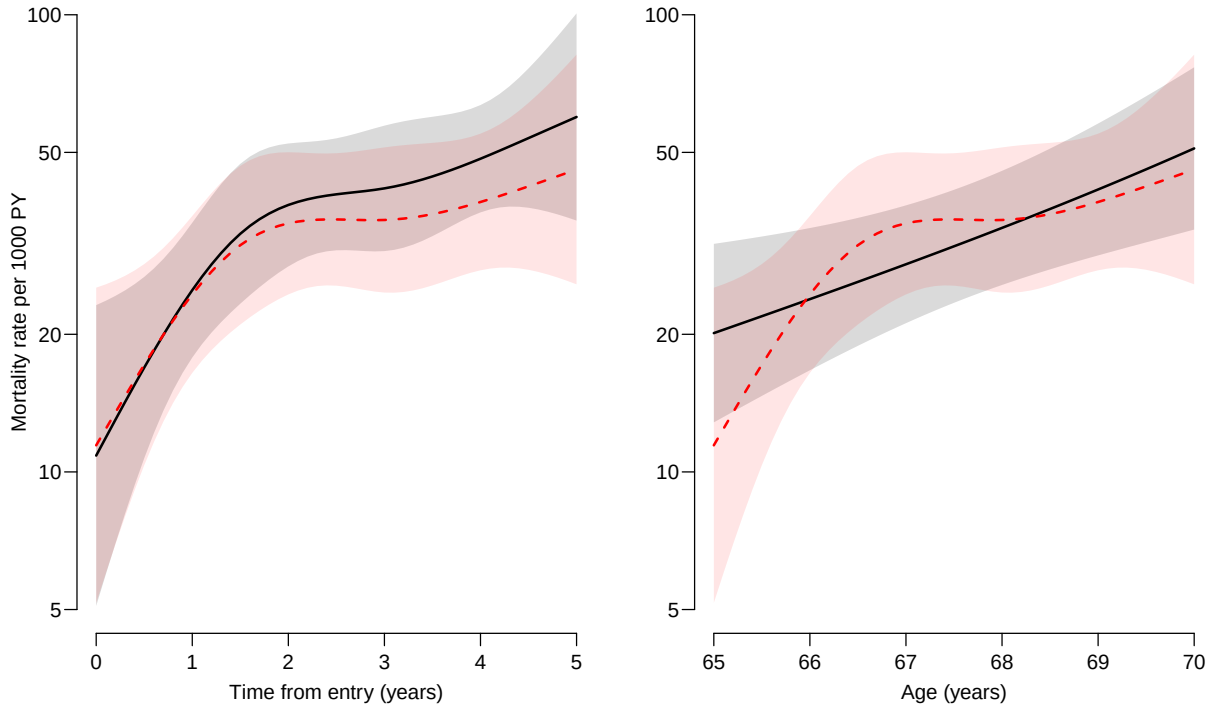

Figure 3: *Estimated mortality rates for PAD persons entering at age 65. The black curve in the left panel is the model with time since entry as time scale, in the right panel the black curve is the model with age as time scale. The red broken curves are the same in the two plots, they represent the joint effect of the two time scales from the model with both timescales modeled with non-linear effects.* `./Sc-rates`

The plots in figure 3 are really the main reason to use a parametric model instead of the Cox-model; they are trivial to produce from the parametric model and really cumbersome to get from the Cox model (at least so cumbersome that they were not included in the intensity guide).

The red dotted curves in figure 3 are the mortality rates from the model `m2` with two non-linear time scales for a person aged 65 at entry followed for 5 years, and whether we label the curve by `tfE` or by `Age` it will be the same, so essentially we are just comparing with two different curves; we could have put them all in the same plot, and have the x-axis labeled both 65 through 70 and 0 through 5.

3.6 Joint effect of timescales

What is of real interest is to see how mortality looks for persons that enter at different ages. We have a model with two time scales, and they advance at the same pace, so it is of course of relevance to see how the joint effect of the two time scales look for different ages at entry.

We can illustrate this by plotting the mortality of persons the first 5 years after entry for persons entering at different ages, say 45, 50,. . . ,75:

```
> nd <- (data.frame(expand.grid(tfE = c(NA, seq(0, 5, 0.1)),
+                               Ain = seq(45, 75, 5)),
+                   sex = "male",
+                   pad = "PAD")
+       %>% mutate(Age = Ain + tfE))
> head(nd)
```

```
  tfE Ain  sex pad  Age
1  NA  45 male PAD   NA
2 0.0  45 male PAD 45.0
3 0.1  45 male PAD 45.1
4 0.2  45 male PAD 45.2
5 0.3  45 male PAD 45.3
6 0.4  45 male PAD 45.4
```

```
> matshade(nd$Age, cbind(ci.pred(m2, nd),
+                        ci.pred(ma, nd),
+                        ci.pred(mt, nd)) * 1000,
+          lty = c("solid","52","11"), lwd = 2, lend = "butt",
+          col = "black", alpha = 0.1,
+          xlab = "Age (years)", ylab = "Mortality per 1000 py",
+          ylim = c(2, 200), log = "y", plot = TRUE)
> nx <- data.frame(Age = seq(47, 77, 5),
+                  tfE = rep(2, 7),
+                  sex = "male",
+                  pad = "PAD")
> lines(nx$Age, ci.pred(mt, nx)[,1] * 1000,
+       col = "blue", type = "b", pch = 16, cex = 0.5)
> abline(v = seq(45, 75, 5), col = gray(0.7), lty=3)
```

From figure 4 we see that mortality increases sharply during the first 2 years after entry and then levels off. This is presumably a clinical artifact because persons who are very sick —too sick to attend the initial clinical visit— are not in the study, so only after a few years the study population achieves the "normal" population mortality rates. But we can also see that the curves after the first two years seem to form a pretty consistent age-shape independent of age at diagnosis and duration.

This can be illustrated by drawing the estimated mortality rates for duration 0 through 5 years for persons diagnosed at ages 45, 47, . . . :

```
> n2 <- (data.frame(expand.grid(tfE = c(NA, seq(0, 5, 0.1)),
+                               Ain = seq(45, 75, 2)),
+                   sex = "male",
+                   pad = "PAD")
+        %>% mutate(Age = Ain + tfE))
> head(n2)
```

```
  tfE Ain  sex pad  Age
1  NA  45 male PAD   NA
2 0.0  45 male PAD 45.0
3 0.1  45 male PAD 45.1
4 0.2  45 male PAD 45.2
5 0.3  45 male PAD 45.3
6 0.4  45 male PAD 45.4
```

```
> matshade(n2$Age, ci.pred(M2, n2) * 1000,
+          lty = c("solid","44","22"), lwd = 2, col = "black",
+          xlab = "Age (years)",
+          ylim = c(2, 200), log = "y", plot = TRUE)
```

From figure **??** we see that even disregarding the first 2 years after entry where mortality increases steeply, the mortality is only well described with functions of both age and time from entry, and that the younger at diagnosis, the higher is the mortality at any age. This type of insight is only achieved from intensity models where the intensity itself is studied.

From a testing-formal point of view we may reduce the model with two time scales to a model with linear effects of both age and time from entry, but that would be to go to the other extreme from the Cox-model. It seems more relevant to accommodate both time scales in the model, after all it is only 7 parameters for both time scales, way less than used in the Cox model behind the scenes.

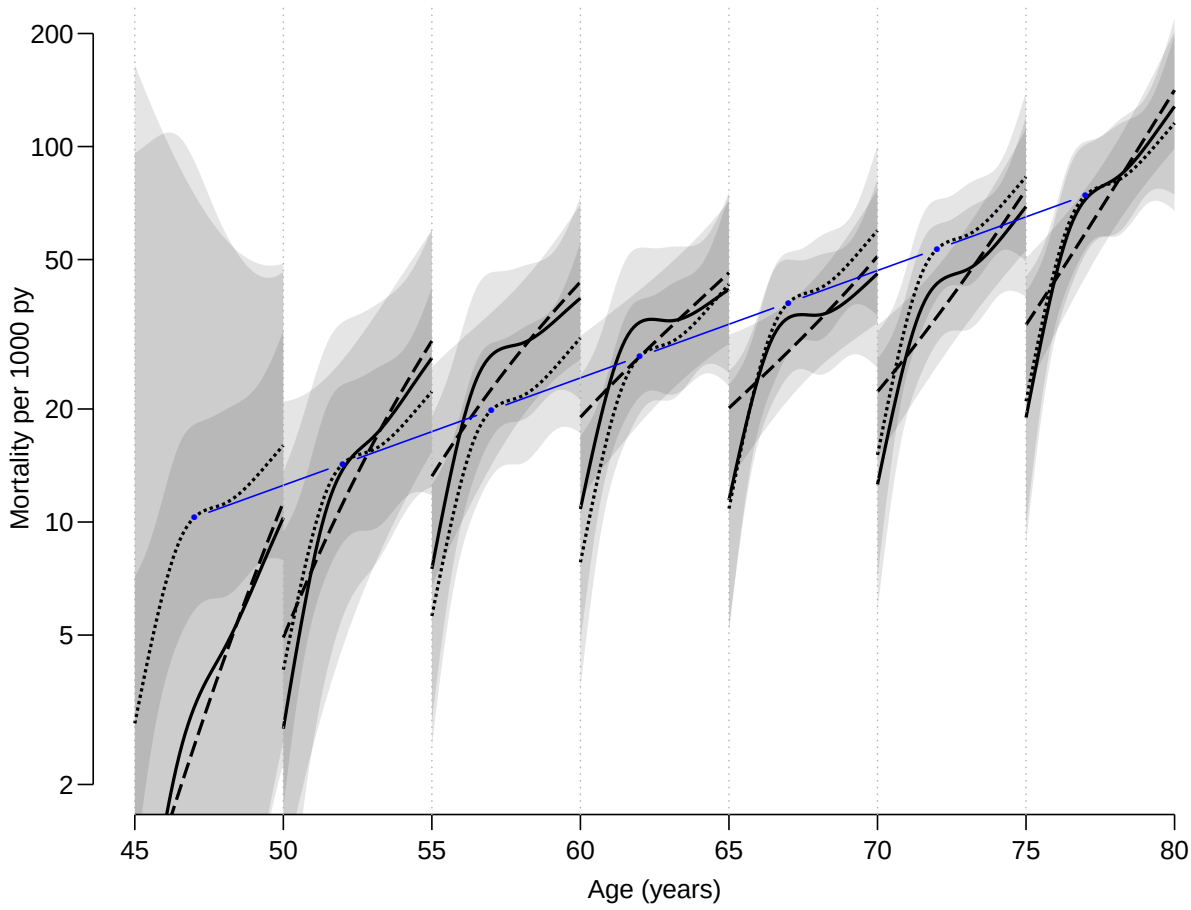

Figure 4: *Mortality rates from three models: The broken (almost straight) lines are from the model with a non-linear effect of current age but a linear effect of time since entry. The dotted non-linear curves are from the model with non-linear effect of time since entry but linear effect of age (as illustrated by the blue line touching the curves at* `tfE`*=2 years). The full lines are mortality rates from the model with non-linear effects of both current age and time from entry (but no interaction); this is the relevant model to report.* `./Sc-2scales`

3.7 Exploring proportionality

In the model with the two timescales the duration effects may be modified by the age-effect. Testing for "proportionality" i.e. interaction between `tfE` and `Age`, we found it negligible by p-value standards. But still it would be worth exploring the actual effect of this by plotting the predictions together for the main effects and the interaction model, interactions may be relevant even if not significant:

```
> matshade(nd$Age, cbind(ci.pred(m2, nd),
+                        ci.pred(m3, nd),
+                        ci.pred(mi, nd)) * 1000,
+          lty = c("solid","12","43"), lwd = 2, lene = "butt",
+          col = "black", alpha = 0.1,
+          xlab = "Age (years)", ylab = "Mortality per 1000 py",
+          ylim = c(2, 200), log = "y", plot = TRUE)
```

From figure 6 it is pretty clear that the model with additional effect of age at entry does not provide any substantial change in predicted mortalities over the main-effects model,

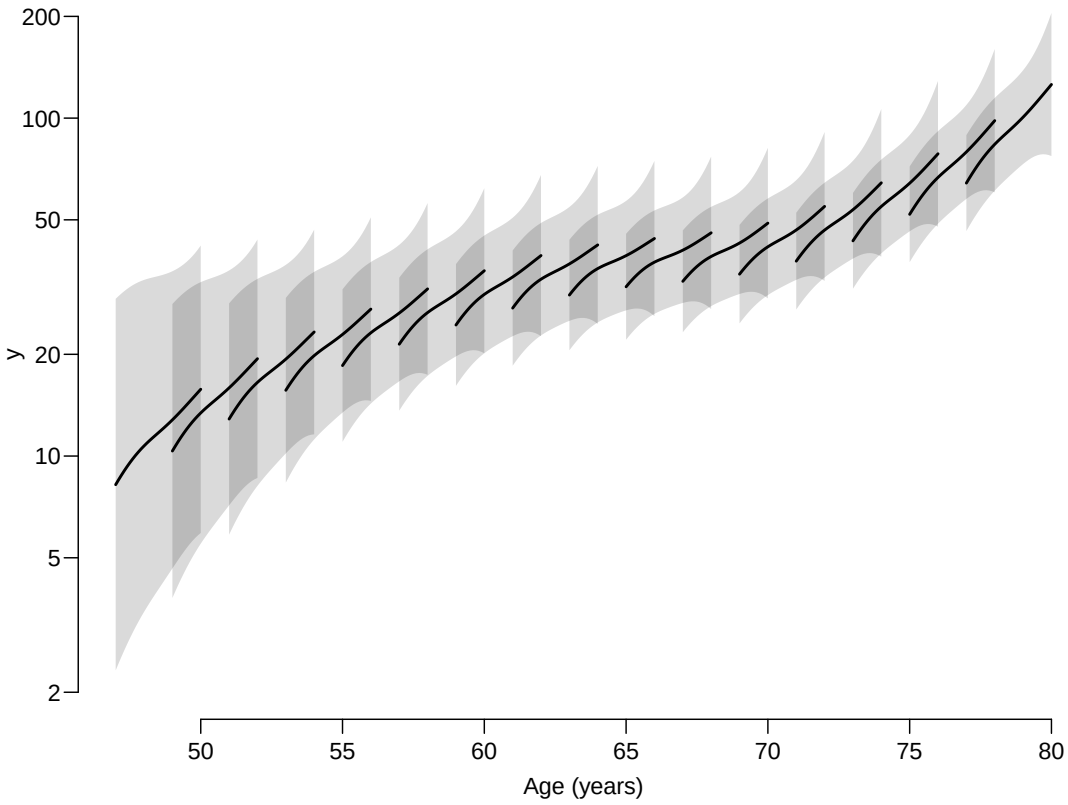

Figure 5: *Mortality rates from the model with non-linear effects of both current age and time from entry for persons diagnosed at ages 45, 47, . . . , for periods 0–5 years after entry (the extent of data). The darker areas are just places where two different predictions overlap.* `./Sc-2scales2-5`

and the general interaction model's predictions are simply not credible. So we conclude that the main effects model is a tenable description of data, and that the discussion of the mortality rates should be based on this:

```
> matshade(nd$Age, cbind(ci.pred(m2, mutate(nd, sex = "female",
+                                                pad = "Control")),
+                        ci.pred(m2, mutate(nd, sex = "female" )),
+                        ci.pred(m2, mutate(nd, pad = "Control")),
+                        ci.pred(m2, nd)) * 1000,
+          lty = c(3,3,1,1), lwd = 2, col = clr <- c("forestgreen","red"),
+          alpha = c(0,0,1,1)/8,
+          xlab = "Age (years)",
+          ylab = "Mortality per 1000 py",
+          ylim = c(2, 200), log = "y", plot = TRUE)
> text(45, c(150,200), c("Contol","PAD"), col = clr, adj = 0)
```

Showing the shape of the mortality rates together with the predictions for different levels of the covariates shows that the sex effect is a bit smaller than the PAD effect (this is also apparent from the HR estimates). But it also shows that these effects are quite large compared the effect of age (after 2 years of FU). We also see a very sharp increase in the mortality during the first two years after entry, possibly owing to a healthy recruitee effect. The latter two points are only available from a parametric model for the intensities, and therefore completely missed in the intensity guide.

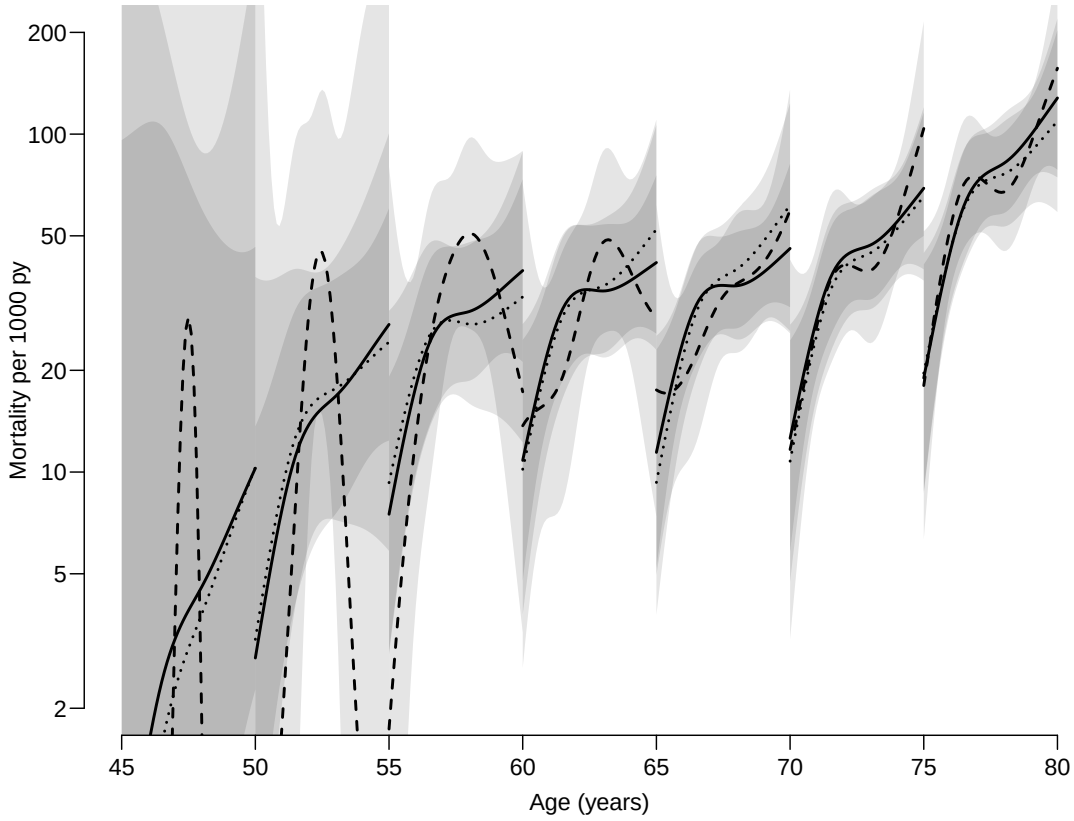

Figure 6: *Comparison of the model with main effects of time from inclusion and current age (full lines), the model with additional non-linear effect of age at entry (dotted lines) and the general interaction model (broken lines).* `./Sc-2int`

4 Multiple states and clinical covariates

We now turn to the effect of the clinical measurements, based on the dataset amended with clinical measurements and a CV complications state (see fig. 2). Inclusion of the state CV and addition of the clinical measurements are really two different topics that can be treated separately.

Anyway, the models fitted in the intensity guide are:

```
> cf <- coxph.Lexis(Sc, tfE ~ pad + sex + I(Age/10) + hdl0 + ldl0,
+                       to = "CV death")
survival::coxph analysis of Lexis object Sc:
Rates for transitions Alive->CV death, CV->CV death
Baseline timescale: tfE

> ct <- coxph.Lexis(Sc, tfE ~ pad + sex + I(Age/10) + hdl + ldl,
+                       to = "CV death")

survival::coxph analysis of Lexis object Sc:
Rates for transitions Alive->CV death, CV->CV death
Baseline timescale: tfE

> cv <- coxph.Lexis(Sc, tfE ~ pad + sex + I(Age/10) + hdl + ldl,
+                       to = c("CV","CV death"))

survival::coxph analysis of Lexis object Sc:
Rates for transitions Alive->CV, Alive->CV death
Baseline timescale: tfE
```

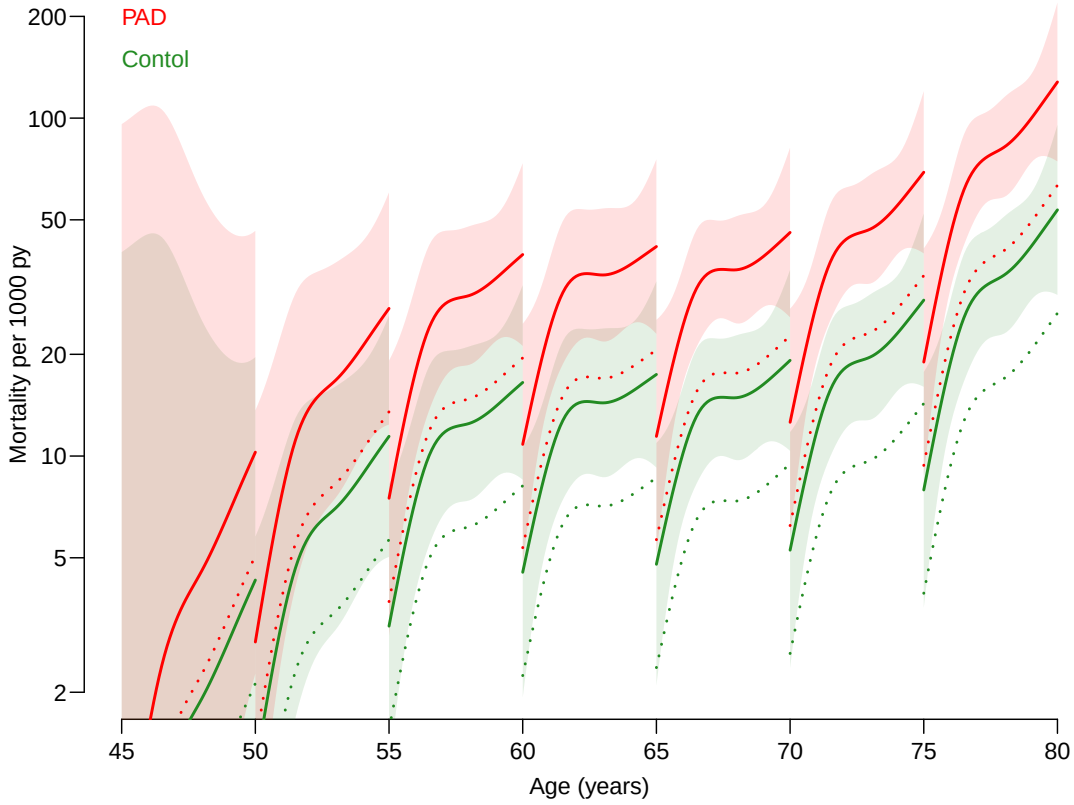

Figure 7: *Mortality rates from the final model for persons entering at ages 45, 50, . . . , 75, separately for persons with (red) and without (green) PAD at entry. Dotted lines are rates for women, full lines are men; the shaded areas are 95% confidence intervals—only shown for men.* `./Sc-finalrates`

```
> round(cbind(ci.exp(cf),
+             ci.exp(ct),
+             ci.exp(cv)), 2)

          exp(Est.) 2.5% 97.5% exp(Est.) 2.5% 97.5% exp(Est.) 2.5% 97.5%
padPAD         2.87 1.65  5.00      2.21 1.26  3.90      2.22 1.53  3.22
sexmale        1.67 0.97  2.88      1.37 0.78  2.38      1.94 1.30  2.89
I(Age/10)      1.93 1.40  2.67      2.17 1.55  3.05      1.58 1.28  1.95
hdl0           0.74 0.39  1.41      0.18 0.08  0.41      0.46 0.28  0.76
ldl0           0.92 0.72  1.18      0.74 0.55  0.99      0.89 0.73  1.07
```

The models with time-updated variables are not quite replicas of the estimates in the intensity guide; the data used in the intensity guide updates the clinical measurements at an annual basis, whereas the `Lexis` objects `Sa`, and by that token also `Sx` and `Sc`, correctly updates the clinical measurements at the actual time of measurement and not at integer years of follow-up.

As an aside, I suspect that this shortcoming in the paper is a product of the complicated programming needed to get it done correctly. This merely emphasizes the need for versatile tools to represent and handle multistate data. Since it is quite a painful exercise to add clinical information to follow-up data, it has been wrapped in a special function `addCov.Lexis` that was used to construct the `Sa` object, and hence its descendants `Sx` and `Sc`.

4.1 Parametric modeling of multistate model

We can directly replicate the results with a parametric models, first using `glm`:

```
> pf <- glm.Lexis(Sc, ~ Ns(Age, kn = knA) + Ns(tfE, kn = knE) +
+                       pad + sex + hdl0 + ldl0,
+                       to = "CV death")
stats::glm Poisson analysis of Lexis object Sc with log link:
Rates for transitions: Alive->CV death, CV->CV death

> pt <- glm.Lexis(Sc, ~ Ns(Age, kn = knA) + Ns(tfE, kn = knE) +
+                       pad + sex + hdl + ldl,
+                       to = "CV death")

stats::glm Poisson analysis of Lexis object Sc with log link:
Rates for transitions: Alive->CV death, CV->CV death

> pv <- glm.Lexis(Sc, ~ Ns(Age, kn = knA) + Ns(tfE, kn = knE) +
+                       pad + sex + hdl + ldl,
+                     from = c("Alive"),
+                       to = c("CV","CV death"))

stats::glm Poisson analysis of Lexis object Sc with log link:
Rates for transitions: Alive->CV, Alive->CV death
```

. . . and subsequently using `gam` models

```
> Pf <- gam.Lexis(Sc, ~ s(Age) + s(tfE) +
+                       pad + sex + hdl0 + ldl0,
+                       to = "CV death")
mgcv::gam Poisson analysis of Lexis object Sc with log link:
Rates for transitions: Alive->CV death, CV->CV death

> Pt <- gam.Lexis(Sc, ~ s(Age) + s(tfE) +
+                       pad + sex + hdl + ldl,
+                       to = "CV death")

mgcv::gam Poisson analysis of Lexis object Sc with log link:
Rates for transitions: Alive->CV death, CV->CV death

> Pv <- gam.Lexis(Sc, ~ s(Age) + s(tfE) +
+                       pad + sex + hdl + ldl,
+                       to = c("CV","CV death"))

mgcv::gam Poisson analysis of Lexis object Sc with log link:
Rates for transitions: Alive->CV, Alive->CV death

> round(cbind(ci.exp(pf, subset=c("pad","sex","dl")),
+             ci.exp(Pf, subset=c("pad","sex","dl")),
+             ci.exp(cf, subset=c("pad","sex","dl"))), 3)

        exp(Est.)  2.5% 97.5% exp(Est.)  2.5% 97.5% exp(Est.)  2.5% 97.5%
padPAD      2.856 1.640 4.973     2.871 1.649 4.998     2.871 1.649 4.998
sexmale     1.706 0.989 2.945     1.686 0.978 2.909     1.671 0.969 2.882
hdl0        0.747 0.394 1.417     0.742 0.389 1.414     0.740 0.388 1.413
ldl0        0.920 0.719 1.179     0.922 0.719 1.182     0.922 0.719 1.182

> round(cbind(ci.exp(pt, subset=c("pad","sex","dl")),
+             ci.exp(Pt, subset=c("pad","sex","dl")),
+             ci.exp(ct, subset=c("pad","sex","dl"))), 3)

        exp(Est.)  2.5% 97.5% exp(Est.)  2.5% 97.5% exp(Est.)  2.5% 97.5%
padPAD      2.206 1.254 3.882     2.216 1.259 3.899     2.215 1.259 3.897
sexmale     1.392 0.795 2.436     1.377 0.788 2.407     1.365 0.782 2.384
hdl         0.176 0.076 0.409     0.176 0.076 0.407     0.177 0.077 0.408
ldl         0.737 0.547 0.993     0.735 0.545 0.991     0.737 0.546 0.995

> round(cbind(ci.exp(pv, subset=c("pad","sex","dl")),
+             ci.exp(Pv, subset=c("pad","sex","dl")),
+             ci.exp(cv, subset=c("pad","sex","dl"))), 3)
```

```
        exp(Est.)  2.5% 97.5% exp(Est.)  2.5% 97.5% exp(Est.)  2.5% 97.5%
padPAD      2.209 1.524 3.201     2.219 1.531 3.216     2.222 1.533 3.220
sexmale     1.955 1.310 2.917     1.953 1.309 2.914     1.938 1.300 2.890
hdl         0.454 0.273 0.756     0.455 0.273 0.756     0.458 0.275 0.763
ldl         0.884 0.732 1.069     0.882 0.729 1.067     0.888 0.734 1.074
```

Again we get the same results as from the Cox model. The Cox model also gives estimates of the linear effect of the time scale `Age`, but that is really of little use if the shape of the effect of the underlying time scale `tfE` is not known.

We can of course check the proportionality assumption, that is any interactions with the time-scales, by adding a suitable parametric interaction term as show the shape and the p-value.

5 Cumulative risks

We can estimate the absolute risk of CV event (diagnosis or CV death), but it requires a model for non-CVD mortality as well, so we model this as well:

```
> mf <- glm.Lexis(Sc, ~ Ns(Age, kn = knA) + Ns(tfE, kn = knE) +
+                       pad + sex + hdl0 + ldl0,
+                     from = "Alive",
+                       to = "nonCV death")
stats::glm Poisson analysis of Lexis object Sc with log link:
Rates for the transition: Alive->nonCV death
```

```
> Mf <- gam.Lexis(Sc, ~ s(Age) + s(tfE) +
+                       pad + sex + hdl0 + ldl0,
+                     from = "Alive",
+                       to = "nonCV death")
mgcv::gam Poisson analysis of Lexis object Sc with log link:
Rates for the transition: Alive->nonCV death
> round(cbind(ci.exp(mf, subset=c("pad","sex","dl")),
+             ci.exp(Mf, subset=c("pad","sex","dl"))), 3)
        exp(Est.)  2.5% 97.5% exp(Est.)  2.5% 97.5%
padPAD      1.857 1.186 2.909     1.870 1.194 2.929
sexmale     2.137 1.291 3.539     2.114 1.278 3.499
hdl0        0.783 0.443 1.384     0.786 0.443 1.394
ldl0        1.026 0.826 1.273     1.028 0.828 1.278
```

Together with the model `pf` we can now compute the cumulative risks with confidence intervals for two different men aged 58 and 72 years, both with a baseline HDL of 1.3 and LDL of 3:

```
> np58 <- data.frame(tfE = seq(0, 5, 0.1),
+                    Age = seq(0, 5, 0.1) + 58,
+                    pad = "PAD",
+                    sex = "male",
+                   hdl0 = 1.3,
+                   ldl0 = 3)
```

We can use the `ci.Crisk` function to compute the cumulative risks (and simulation based confidence intervals) for persons with the given set of covariates for the 4 combinations of PAD and ages 58 and 72:

```
> ci58p <- ci.Crisk(list(CVD = pf,
+                        nonCVD = mf),
+                   nd = np58)
NOTE: Times are assumed to be in the column tfE at equal distances of 0.1
> ci58c <- ci.Crisk(list(CVD = pf,
+                        nonCVD = mf),
+                   nd = mutate(np58, pad = "Control"))
NOTE: Times are assumed to be in the column tfE at equal distances of 0.1
> ci72p <- ci.Crisk(list(CVD = pf,
+                        nonCVD = mf),
+                   nd = mutate(np58, Age = Age + 14))
NOTE: Times are assumed to be in the column tfE at equal distances of 0.1
> ci72c <- ci.Crisk(list(CVD = pf,
+                        nonCVD = mf),
+                   nd = mutate(np58, Age = Age + 14,
+                                     pad = "Control"))
NOTE: Times are assumed to be in the column tfE at equal distances of 0.1
> str(ci58p)
List of 4
 $ Crisk: num [1:51, 1:3, 1:3] 1 0.999 0.997 0.996 0.995 ...
  ..- attr(*, "dimnames")=List of 3
  .. ..$ tfE  : chr [1:51] "0" "0.1" "0.2" "0.3" ...
  .. ..$ cause: chr [1:3] "Surv" "CVD" "nonCVD"
  .. ..$      : chr [1:3] "50%" "2.5%" "97.5%"
 $ Srisk: num [1:51, 1:2, 1:3] 0 0.000322 0.000694 0.001132 0.001636 ...
  ..- attr(*, "dimnames")=List of 3
  .. ..$ tfE  : chr [1:51] "0" "0.1" "0.2" "0.3" ...
  .. ..$ cause: chr [1:2] "nonCVD" "nonCVD+CVD"
  .. ..$      : chr [1:3] "50%" "2.5%" "97.5%"
 $ Stime: num [1:51, 1:3, 1:3] 0 0.0999 0.1997 0.2994 0.399 ...
  ..- attr(*, "dimnames")=List of 3
  .. ..$ tfE  : chr [1:51] "0" "0.1" "0.2" "0.3" ...
  .. ..$ cause: chr [1:3] "Surv" "CVD" "nonCVD"
  .. ..$      : chr [1:3] "50%" "2.5%" "97.5%"
 $ time : num [1:51] 0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 ...
 - attr(*, "int")= num 0.1
```

We can then produce the two plots from figure 6 in the with proper simulation based confidence intervals, by taking the values from the `Crisk` component from the results of `ci.Crisk`.

```
> clr <- c("orange","black")
> par(mfrow=c(1,2), las = 1, bty = "n")
> matshade(ci58p$time,
+          cbind(ci58p$Crisk[,   "CVD",],
+                ci58p$Crisk[,"nonCVD",],
+                ci58c$Crisk[,   "CVD",],
+                ci58c$Crisk[,"nonCVD",])*100, plot = TRUE,
+          lty = c("solid", "23"), lwd = 2, alpha = 0.1*c(0,1,0,1),
+          col = rep(clr, each = 2),
+          ylim = c(0,15), yaxs = "i",
+          xlab = "Time since entry (years)",
+          ylab = "Cumulative risk (%)")
> matshade(ci72p$time,
+          cbind(ci72p$Crisk[,   "CVD",],
+                ci72p$Crisk[,"nonCVD",],
+                ci72c$Crisk[,   "CVD",],
+                ci72c$Crisk[,"nonCVD",])*100, plot = TRUE,
+          lty = c("solid", "23"), lwd = 2, alpha = 0.1*c(0,1,0,1),
+          col = rep(clr, each = 2),
+          ylim = c(0,15), yaxs = "i",
+          xlab = "Time since entry (years)",
+          ylab = "Cumulative risk (%)")
```

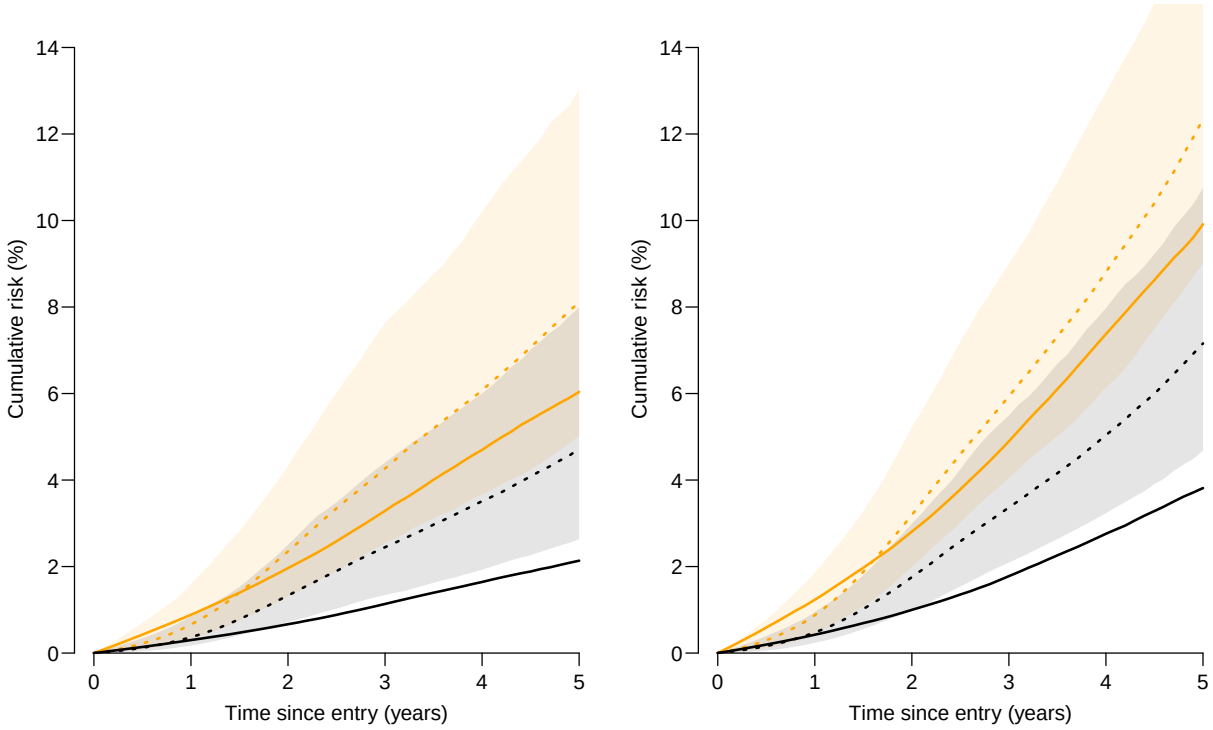

Figure 8: *Cumulative risk of CV death and other-cause death from two separate models, evaluated for men aged 58 and 72. Solid lines are CV death, broken lines are non-CV death. Orange lines are persons with PAD at entry, black are persons without. 95% confidence intervals are shown as shades, but only for the non-CV death (broken lines). This is the counterpart of figure 6 in the intensity guide, except for the added confidence intervals.* `./Sc-crisk`

6 From rates to risks

Hernan [4] advocated the use of survival curves in reporting effects, essentially recovering the baseline hazard from the Cox-model. The point being that large hazard ratios for tiny rates may be of less relevance than small hazard ratios for large rates.

Moreover, Hernan notes that interactions with time can be better shown on the cumulative risk or survival scale. However, the concepts of interaction and survival are not related; we can always compare cumulative risks (or survival, the cumulative risk of being in the alive state), regardless of how rates depend on time and other covariates. Likewise, we can also graph an interaction on the rate scales, which seems more logical, because that is the scale on which it is estimated.

Resorting to cumulative risks requires a definition of the *origin* from which to compute the cumulative risks. This is often conveniently overlooked because reference is to a study entry in a clinical trial, whereas no such thing as a meaningful entry date necessarily exists in an epidemiological study. Many epidemiological cohort studies include persons by invitation to a clinical examination of some sort, so the date of inclusion is basically just a random date in these persons' lives. Register based studies will typically have a fixed calendar time as entry (for some persons at least).

The cumulative risk(s) is a function of the rates, so once we have a model for the rates

we can compute the cumulative risks. These come in different guises:

- conditional on *one* particular set of values of the explanatory variables; this is interpretable as the cumulative fraction seeing an event in a group of *identical* persons.
- using a *set* of values of the explanatory variables. The *average* of the cumulative risks over the set will then represent the cumulative fraction seeing an event in a group of persons with a distribution of explanatory variables as in the chosen set.

The latter was used by Hernan [4] to suggest comparison of population difference associated with a treatment allocation by looking at the cumulative risks in a group of persons identical to the study population but assuming treatment allocation identical to either A or B for the entire population. This gives the average treatment effect in the *study* population, but the exercise can of course be done both for models with and without interactions.

However, the calculation of cumulative risks in the example shown above would not have revealed the distinct healthy-participant effect, it would merely have overestimated the survival in the background population by including the healthy participant effect during the first 1–2 years. A proper analysis and inspection of the rates would allow calculation of survival conditional on being alive at a given age, instaed of conditional on entering a study. The latter is hardly relevant.

7 Additional guidance to the use of intensity models

It seems that some additional advice on the use, or rater choice of intensity models is needed. So here are some suggestions:

- When modeling intensities, show the time scale effects on the intensity scale.
- Use parametric models for the intensity with smooth effects of time scales.
- Use multiple time scales if necessary.
- Show the joint effects of the time scales used.
- Parametric modeling puts the "proportional hazards" testing in the familiar setting of testing for and in particular *estimating* interaction effects. Mere testing is not advisable.
- Do not group quantitative variables, even if they are time scales.

These issues are treated in chapters 5 and 6 of "Epidemiology with R"[5].

References

[1] P. Kragh Andersen, M. Pohar Perme, H. C. van Houwelingen, R. J. Cook, P. Joly, T. Martinussen, J. M. G. Taylor, M. Abrahamowicz, and T. M. Therneau. Analysis of time-to-event for observational studies: Guidance to the use of intensity models. *Stat Med*, 40(1):185–211, Jan 2021.

[2] Martyn Plummer and Bendix Carstensen. Lexis: An R class for epidemiological studies with long-term follow-up. *Journal of Statistical Software*, 38(5):1–12, 1 2011.

[3] Bendix Carstensen and Martyn Plummer. Using Lexis objects for multi-state models in R. *Journal of Statistical Software*, 38(6):1–18, 1 2011.

[4] M. A. Hernán. The hazards of hazard ratios. *Epidemiology*, 21(1):13–15, Jan 2010.

[5] Bendix Carstensen. *Epidemiology with R.* Number ISBN: 978-0-19-884133-3. Oxford University Press, 2020.