

Practice in analysis of multistate models using Epi::Lexis in

Freiburg, Germany

September 2016

<http://bendixcarstensen.com/AdvCoh/courses/Frias-2016>

Version 1.1

Compiled Thursday 15th September, 2016, 11:26

from: /home/bendix/teach/AdvCoh/courses/Freiburg.2016/pracs/pracs.tex

Bendix Carstensen Steno Diabetes Center, Gentofte, Denmark
& Dept. of Biostatistics, University of Copenhagen, Denmark
bxc@steno.dk
<http://BendixCarstensen.com>

Contents

1	Introduction	1
1.1	Computing prerequisites	1
1.2	Statistical prerequisites	1
2	Exercises	2
2.1	Renal complications: Time-dependent variables and multiple states	2
2.1.1	The renal failure dataset	2
2.1.2	Splitting the follow-up time	5
2.1.3	Prediction in a multistate model	7
2.2	Time-splitting, time-scales and SMR: Diabetes in Denmark	11
2.2.1	SMR	14
3	Solutions	17
3.1	Renal complications: Time-dependent variables and multiple states	17
3.1.1	The renal failure dataset	17
3.1.2	Splitting the follow-up time	23
3.1.3	Prediction in a multistate model	28
3.2	Time-splitting, time-scales and SMR: Diabetes in Denmark	37
3.2.1	SMR	46

Chapter 1

Introduction

There are two practicals in this document. This first one, “Renal complications: Time-dependent variables and multiple states” is the one we shall be using, the other is included for self-study if you wish so. Both exercises also has a solution-version in following chapter.

1.1 Computing prerequisites

The practicals assume that you have an up to date version of R (3.3.1) as well as the last version of the `Epi` package (2.5), the following should work and give you the relevant information:

```
> install.packages( "Epi" )  
> library( Epi )  
> sessionInfo()
```

Also, you will need to access a dataset from the website so you might want to download the file it is <http://bendixcarstensen.com/AdvCoh/courses/Frias-2016/renal.Rda>

In the tutorial I shall assume that you are familiar with the following commands in R:

- `glm`, including the `offset=` argument
- update for models
- `predict`, and the wrapper `ci.pred` from the `Epi` package.

1.2 Statistical prerequisites

I will assume that you are familiar with the usual likelihood machinery and the theory of generalized linear models.

And of course the basic probability theory underlying calculation of demographic rates and probabilities derived from these.

Chapter 2

Exercises

2.1 Renal complications: Time-dependent variables and multiple states

The following practical exercise is based on the data from paper:

P Hovind, L Tarnow, P Rossing, B Carstensen, and HH Parving: Improved survival in patients obtaining remission of nephrotic range albuminuria in diabetic nephropathy. *Kidney Int*, **66**(3):1180–1186, Sept 2004.

You can find a .pdf-version of the paper here:

<http://BendixCarstensen.com/~bxc/AdvCoh/papers/Hovind.2004.pdf>

2.1.1 The renal failure dataset

The dataset `renal.dta` contains data on follow up of 125 patients from Steno Diabetes Center. They enter the study when they are diagnosed with nephrotic range albuminuria (NRA). This is a condition where the levels of albumin in the urine is exceeds a certain level as a sign of kidney disease. The levels may however drop as a consequence of treatment, this is called remission. Patients exit the study at death or kidney failure (dialysis or transplant).

Table 2.1: *Variables in renal.dta.*

<code>id</code>	Patient id
<code>sex</code>	M / F
<code>dob</code>	Date of birth
<code>doe</code>	Date of entry into the study (2.5 years after NRA)
<code>dor</code>	Date of remission. Missing if no remission has occurred
<code>dox</code>	Date of exit from study
<code>event</code>	Exit status: 0: censored, 1: death, 2: end stage renal disease, ESRD (kidney failure) and 3: Kidney transplant.

1. The dataset is available at the course website as `renal.Rda`:

```
library( Epi ) ; clear()
load( url("http://BendixCarstensen.com/AdvCoh/courses/Frias-2016/data/renal.Rda") )
# load( "renal.Rda" )
str( renal )
head( renal )
```

Here we shall only be interested in the combined event 1, 2 or 3.

2. Use the `Lexis` function to declare the data as survival data with age, calendar time and time since entry into the study as timescales. Note that any coding of event > 0 will be labeled “ESRD”, i.e. renal death (death of kidney (transplant or dialysis), or person).

Note that you must make sure that the “alive” state (here `NRA`) is the first, as `Lexis` assumes that everyone starts in this state (unless of course `entry.status` is specified).

```
Lr <- Lexis( entry = list( per=doe,
                          age=doe-dob,
                          tfi=0 ),
            exit = list( per=dox ),
            exit.status = factor( event>0, labels=c("NRA","ESRD") ),
            data = renal )
str( Lr )
summary( Lr )
```

3. Visualize the data in a Lexis-diagram, using the `plot` method for `Lexis` objects. What do you see?

```
plot( Lr, col="black", lwd=3 )
```

4. (*Optional, not crucial to the rest of the exercise.* Now try to produce a slightly more fancy Lexis diagram. Note that we have a x -axis of 40 years, and a y -axis of 80 years, so when specifying the output file adjust the *total* width of the plot so that the use `mai` to specify the margins of the plot leaves a plotting area twice as high as wide. You will want to consult the maning of the argument `mai` to the function `par`.

```
# pdf( "lexis-fancy.pdf", height=80/5+1, width=40/5+1 )
# x11( height=80/5+1, width=40/5+1 )
par( mai=c(3,3,1,1)/4, mgp=c(3,1,0)/1.6 )
plot( Lr, 1:2, col=c("blue","red")[Lr$sex], lwd=3, grid=0:20*5,
      xlab="Calendar time", ylab="Age",
      xlim=c(1970,2010), ylim=c(0,80), xaxs="i", yaxs="i", las=1 )
# dev.off()
```

5. Make a Cox-regression analysis with the variables `sex` and `age` at entry into the study, using time since entry to the study as time scale.

Give the hazard ratio between males and females and between two persons who differ 10 years in age at entry. Give the 95% confidence intervals for this as well.

```
library( survival )
mc <- coxph( Surv( lex.dur, lex.Xst=="ESRD" ) ~
             I(age/10) + sex, data=Lr )
summary( mc )
```

6. The main focus of the paper was to assess whether occurrence of remission (return to a lower level of albumin excretion, an indication of kidney recovery) influences mortality.

“Remission” is a time-dependent variable which is initially 0, but takes the value 1 when remission occurs. In order to handle this, each person who see a remission must have two records:

- One record for the time before remission, where entry is `doe`, exit is `dor`, remission is 0, and event is 0.
- One record for the time after remission, where entry is `dor`, exit is `dox`, remission is 1, and event is 0 or 1 according to whether the person had an event at `dox`.

This is accomplished using the `cutLexis` function on the `Lexis` object. You must declare the “NRA” state as a precursor state, i.e. a state that is *less* severe than “Rem” in the sense that a person who see a remission will stay in the “Rem” state unless he goes to the “ESRD” state.

```
Lc <- cutLexis( Lr, cut = Lr$dor, # where to cut follow up
               timescale = "per", # the timescale that "dor" refers to
               new.state = "Rem", # name of the new state
               precursor.states = "NRA" ) # which states are less severe
summary( Lc )
```

List records for a few select persons from `Lr` and from `Lc` to see how the cut has worked.

7. Show how the states are connected and the number of transitions between them by using `boxes`. This is an interactive command that requires you to click in the graph window:

```
boxes( Lc )
```

Alternatively you can let R try to place the boxes for you, and even compute rates (in this case in units of events per 100 PY):

```
boxes( Lc, boxpos=TRUE, scale.R=100, show.BE=TRUE )
```

How many transitions are there from remission to death?

8. (*Optional: Not relevant for the remainder of the exercise.*) Now make a Lexis diagram where different colouring is used for different segments of the follow-up — you should be able to count the 8 transitions from “Rem” to “ESRD”.

```

par( mai=c(3,3,1,1)/4, mgp=c(3,1,0)/1.6 )
plot( Lc, col=c("red","limegreen")[(Lc$lex.Cst=="Rem")+1],
      xlab="Calendar time", ylab="Age",
      lwd=3, grid=0:20*5, xlim=c(1970,2010), ylim=c(0,80), xaxs="i", yaxs="i", las=1
points( Lc, pch=c(NA,16)[(Lc$lex.Xst=="ESRD")+1],
      col=c("red","limegreen")[(Lc$lex.Cst=="Rem")+1])
points( Lc, pch=c(NA,1)[(Lc$lex.Xst=="ESRD")+1],
      col="black", lwd=2 )

```

9. Make a Cox-regression of mortality (i.e. endpoint “ESRD”) with sex, age at entry and remission as explanatory variables, and using time since entry as timescale.

Remember to include `lex.Cst` as time-dependent variable, and to indicate that each recort represbts follow-up from `tfi` to `tfi+lex.dur`. Note the use of the `Lexis` variables `lex.dur` (risk time), `lex.Xst` (exit status) and `lex.Cst` (current status).

```

m1 <- coxph( Surv( tfi, tfi+lex.dur, lex.Xst=="ESRD" ) ~
              sex + I((doe-dob-50)/10) + (lex.Cst=="Rem"), data=Lc )
summary( m1 )

```

10. What is the relation between the rate of ESRD between persons in remission and persons not?
11. What is the assumption about the two rates of remission? Refer to the figure with the three boxes you just made. (??).

2.1.2 Splitting the follow-up time

In order to explore the effect of remission on the rate of ESRD, we will split the data further into small pieces of follow-up. To this end we use the function `splitLexis`. The rates can then be modeled using a Poisson-model, and the shape of the underlying *rates* be explored. Furthermore, we can allow effects of both time since NRA and current age. To this end we will use splines, so we need the splines package, too.

12. First, split the follow-up time every month after entry, and make sure that the number of events and risk time is the same as before (use `summary`):

```

sLc <- splitLexis( Lc, "tfi", breaks=seq(0,30,1/12) )
summary( Lc )
summary(sLc )

```

13. Now try to fit the Poisson-model corresponding to the Cox-model we fitted previously. The function `ns()` produces a model matrix corresponding to a piecewise cubic function, modeling the baseline hazard explicitly (think of the `ns` terms as the baseline hazard that is not visible in the Cox-model).

The outcome is 1 or 0 according to whether an event occurred or not, but sine a Poisson variate by definition is numerical, R will automatically coerce (change) a logical value to numeric; `FALSE` as 0 and `TRUE` as 1, so we can conveniently write:

```
library( splines )
mp <- glm( lex.Xst=="ESRD" ~ ns( tfi, df=4 ) +
          sex + I((doe-dob-40)/10) + (lex.Cst=="Rem"),
          offset = log(lex.dur),
          family = poisson,
          data = sLc )
summary( mp )
```

The `ns` function places knots at the quantiles of the variable, which may not be the most logical as the information is contained in the events, so the natural placement of knots would be at the quantiles of the event times. The `Ns` function in the *Epi* package automatically takes the smallest and the largest of the knots as boundary knots — the number of parameters is one less than the number of knots, so we use 5 knots:

```
t.kn <- with( subset( sLc, lex.Xst=="ESRD"),
             quantile( tfi+lex.dur, 0:4/5 ) )
mp <- glm( lex.Xst=="ESRD" ~ Ns( tfi, knots=t.kn ) +
          sex + I((doe-dob-40)/10) + (lex.Cst=="Rem"),
          offset = log(lex.dur),
          family = poisson,
          data = sLc )
summary( mp )
```

14. You can extract the parameters from the models using `ci.lin` or `ci.exp` try:

```
ci.lin( mp )
ci.exp( mp )
ci.exp( mp, subset=c("sex","dob","Cst"), pval=TRUE )
```

Compare with the estimates from the Cox-model. Use:

```
ci.exp( m1 )
ci.exp( mp, subset=c("sex","dob","Cst") )
ci.exp( mp, subset=c("sex","dob","Cst") ) / ci.exp( m1 )
```

What do you conclude about the models?

15. You can visualize the spline term using `termplot`, try:

```
termplot( mp, terms=1 )
```

... which is not a terribly informative plot

16. `termplot` does not give you the absolute level of the underlying rates because it bypasses the intercept. If you explicitly include the intercept in the baseline split you can use `Termplot` from the *Epi* package to get estimates on the rate scale for a reference person (in units of events per 100 years):

```
mP <- glm( lex.Xst=="ESRD" ~ -1 + Ns( tfi, knots=t.kn, intercept=TRUE ) +
          sex + I((doe-dob-40)/10) + (lex.Cst=="Rem"),
          offset = log(lex.dur/100),
          family = poisson,
          data = sLc )
Termplot( mP, terms=1 )
```


How would you describe this rate function in plain words? And what is the scale of the y -axis.

Annotate the axes of the plot accordingly — consult the help page of `Termplot`.

17. Apart from the baseline timescale, time since NRA, time since remission might be of interest in describing the mortality rate. However this is only relevant for persons who actually have a remission, so start by checking how many events there are in this group:

```
summary( sLc )
```

How many go in remission, and how many deaths are in this group?

18. With this rather limited number of events we can certainly not expect to be able to model anything more complicated than a linear trend with time since remission. Two parameters on 8 events is actually pretty far-fetched.

The variable we want to have in the model is current date (`per`) minus date of remission (`dor`): `per-dor`), but *only* positive values of it. This can be fixed by using `pmax()`, but we must also deal with all those who have missing values, so we use the construct:

```
pmax( per-dor, 0, na.rm=TRUE )
```

Make sure that you understand what goes on here.

19. We can now expand the model with this variable:

```
sLc <- transform( sLc, tfr = pmax( (per-dor)/10, 0, na.rm=TRUE ) )
mPx <- glm( lex.Xst=="ESRD" ~ -1 + Ns( tfr, knots=t.kn, intercept=TRUE ) +
           sex + I((age-tfi-40)/10) + (lex.Cst=="Rem") + tfr,
           offset = log(lex.dur/100),
           family = poisson,
           data = sLc )
round( ci.exp( mPx ), 3 )
Termplot( mPx, terms=1 )
```

20. Is the effect significant? Can a substantial effect of time since remission be ruled out?
21. What is the test of this parameter traditionally called? What is the null and what is the alternative of this test?

2.1.3 Prediction in a multistate model

This part of the practical is about making proper statements about the survival and the disease probabilities. But in order to do this we must know not only how the occurrence of remission influences the rate of death/ESRD, but we must also model the occurrence rate of remission itself.

The following exercise will be quite similar to the example in the help file for `simLexis` (which you should read now!).

22. The rates of ESRD were modelled by a Poisson model with effects of age and time since NRA — in the model `mp`. But in the modelling of the remission rates transition from “NRA” to “Rem”, the number of events is rather small, so we restrict the variables in this model to only time since NRA and sex. Also remember, only the records that relate to the “NRA” state can be used:

```
mr <- glm( lex.Xst=="Rem" ~ ns( tfi, knots=t.kn ) + sex,
          offset = log(lex.dur),
          family = poisson,
          data = subset( sLc, lex.Cst=="NRA" ) )
ci.exp( mr, pval=TRUE )
```

23. If we want to predict the probability of being in each of the three states using these estimated rates, we can either do analytical calculations of the probabilities from the estimated rates, or we can *simulate* the life course through a model using the estimated rates. That will give a simulated cohort (in the form of a Lexis object), and we can then just count the number of persons in each state at each of a set of time points.

This is accomplished using the function `simLexis`. The input to this is the initial status of the persons whose life-course we shall simulate, and the transition rates in suitable form:

- Suppose we want predictions for men aged 50 at NRA. The input is in the form of a Lexis object (where `lex.dur` and `lex.Xst` will be ignored). Note that in order to carry over the `time.scales` and the `time.since` attributes, we construct the input object using `subset` to select columns, and `NULL` to select rows (see the example in the help file for `simLexis`):

```
inL <- subset( sLc, select=1:11 )[NULL,]
str( inL )
timeScales(inL)
inL[1,"lex.id"] <- 1
inL[1,"per"] <- 2000
inL[1,"age"] <- 50
inL[1,"tfi"] <- 0
inL[1,"lex.Cst"] <- "NRA"
inL[1,"lex.Xst"] <- NA
inL[1,"lex.dur"] <- NA
inL[1,"sex"] <- "M"
inL[1,"doe"] <- 2000
inL[1,"dob"] <- 1950
inL
```

- The other input for the simulation is the transitions, which is a list with an element for each transient state (that is “NRA” and “Rem”), each of which is again a list with names equal to the states that can be reached from the transient state. The content of the list will be `glm` objects, in this case the models we just fitted, describing the transition rates:

```
Tr <- list( "NRA" = list( "Rem" = mr,
                        "ESRD" = mp ),
           "Rem" = list( "ESRD" = mp ) )
```

With this as input we can now generate a cohort, using N=10 to simulate life course of 10 persons (with identical starting values):

```
( iL <- simLexis( Tr, inL, N=10 ) )
summary( iL )
```

24. Now generate the life course of 10,000 persons, and look at the summary. The `system.time` command is just to tell you how long it took, you may want to start with 1000 just to see how long that takes.

```
system.time(
  sM <- simLexis( Tr, inL, N=10000 ) )
summary( sM )
```

Why are there so many ESRD-events in the resulting data set?

25. Now we want to count how many persons are present in each state at each time for the first 10 years after entry (which is at age 50). This can be done by using `nState`:

```
nSt <- nState( sM, at=seq(0,10,0.1), from=50, time.scale="age" )
head( nSt )
```

26. Once we have the counts of persons in each state at the designated time points, we compute the cumulative fraction over the states, arranged in order given by `perm`:

```
pp <- pState( nSt, perm=1:3 )
head( pp )
tail( pp )
```

27. Try to plot the cumulative probabilities using the `plot` method for `pState` objects:

```
plot( pp )
```

28. A quantity of particular interest would be how many patients actually get a remission. This is not deductible from the plot just shown, because those who get ESRD are not subdivided according to whether they have a remission prior to ESRD.

The simplest way to do that is to modify the simulated object (`sM` in the above notation), so that those exiting to “ESRD” from “Rem” are counted in a separate state. We must also change the formal set of levels of `lex.Cst`:

```
xM <- transform( sM, lex.Xst = factor( ifelse( lex.Xst=="ESRD" & lex.Cst=="Rem",
                                             "ESRD(Rem)",
                                             as.character(lex.Xst) ),
                                             levels=c("NRA", "Rem", "ESRD(Rem)", "ESRD" ) ),
                 lex.Cst = factor( as.character(lex.Cst),
                                   levels=c("NRA", "Rem", "ESRD(Rem)", "ESRD" ) ) )
summary( sM )
summary( xM )
boxes( xM, boxpos=TRUE, show.BE=TRUE, scale.R=100 )
```

29. Having done this, try to compute the number of persons in each of the 4 states, and the cumulative proportions to be plotted:

```
xSt <- nState( xM, at=seq(0,10,0.1), from=50, time.scale="age" )
xp <- pState( xSt, perm=1:4 )
head( xp )
plot( xp, col=rev(c("pink","limegreen","forestgreen","red")), xlab="Age" )
lines( as.numeric(rownames(xp)), xp[, "Rem"], lwd=4 )
```

What is the probability that a 50-year old man with NRA sees a remission from NRA during the next 10 yezs?

30. Make the same calculations for a 60-year old woman.
31. Normally you would know that a split of the absorbing “ESRD” state according to the preceding state and so define this in the `cutLexis` function, using `split.states`. At the same time it is also possible to define a new timescale using `new.scale`, defined as time since entry to the new state:

```
Lc <- cutLexis( Lr, cut = Lr$dor, # where to cut follow up
               timescale = "per", # the timescale that "dor" refers to
               new.state = "Rem", # name of the new state
               precursor.states = "NRA", # which states are less severe
               new.scale = "tfr", # define a new timescale as time since Rem
               split.states = TRUE ) # subdivide non-precursor states
str( Lc )
# source("/home/bendix/stat/R/lib.src/Epi/pkg/R/summary.Lexis.r")
# summary( Lc, S=T, scale=100 )
summary( Lc )
boxes( Lc, boxpos=list(x=c(20,80,20,80),y=c(80,80,20,20)),
       scale.R=100, show.BE=TRUE )
sLc <- splitLexis( Lc, "tfr", breaks=seq(0,30,1/12) )
summary( sLc )
summary( sLc )
head( subset( sLc, lex.id==2 )[,1:8], 8 )
tail( subset( sLc, lex.id==2 )[,1:8], 3 )
( fl <- levels(Lc)[3:4] )
mp <- glm( lex.Xst %in% fl ~ ns( tfr, df=4 ) +
          sex + I((age-tfr-40)/10) + (lex.Cst=="Rem"),
          offset = log(lex.dur/100),
          family = poisson,
          data = sLc )
# the timescale tfr must be given some value for time before Rem
sLc$tfr <- pmax( 0, sLc$tfr, na.rm=TRUE )
head( subset( sLc, lex.id==2 )[,1:8], 8 )
mr <- glm( lex.Xst=="Rem" ~ ns( tfr, df=4 ) + sex,
          offset = log(lex.dur),
          family = poisson,
          data = subset( sLc, lex.Cst=="NRA" ) )
ci.exp( mr, pval=TRUE )
inL <- subset( sLc, select=1:10 )[NULL,]
str( inL )
timeScales(inL)
inL[1,"lex.id"] <- 1
inL[1,"per"] <- 2000
inL[1,"age"] <- 50
inL[1,"tfr"] <- 0
```

```

inL[1,"lex.Cst"] <- "NRA"
inL[1,"lex.Xst"] <- NA
inL[1,"lex.dur"] <- NA
inL[1,"sex"] <- "M"
inL
Tr <- list( "NRA" = list( "Rem" = mr,
                        "ESRD" = mp ),
           "Rem" = list( "ESRD(Rem)" = mp ) )
( iL <- simLexis( Tr, inL, N=10 ) )
summary( iL )
system.time(
sM <- simLexis( Tr, inL, N=10000, t.range=25, n.int=251 ) )
summary( sM )
nSt <- nState( sM, at=seq(0,24,0.1), from=50, time.scale="age" )
head( nSt )
pp <- pState( nSt, perm=c(1,2,4,3) )
head( pp )
tail( pp )
plot( pp )
# Two colors and the corresponding pale ones for the dead states
clr <- c("limegreen","orange")
col2rgb(clr)
cl4 <- cbind(col2rgb(clr),col2rgb(clr)/2+255/2)[,c(1,2,4,3)]
cl4 <- rgb( t(cl4), max=255 )
# Nicer plot
plot( pp, col=cl4, xlab="Age" )
lines( as.numeric(rownames(pp)), pp[,2], lwd=2 )

```

2.2 Time-splitting, time-scales and SMR: Diabetes in Denmark

This exercise is using data from the National Danish Diabetes register. There is a random sample of 10,000 records from this in the `Epi` package. Actually there are two data sets, we shall use the one with only cases of diabetes diagnosed after 1995, see the help page for `DMlate`.

This is of interest because it is only for these where the data of diagnosis is certain, and hence for whom we can compute the duration of diabetes during follow-up.

The exercise is about assessing how mortality depends age, calendar time and duration of diabetes. And how to understand and compute SMR, and assess how it depends on these factors as well.

1. First load the data and take a look at the data:

```

> library( Epi )
> data( DMlate )
> str( DMlate )

```

You can get a more detailed explanation of the data by referring to the help page:

```

> ?DMlate

```

- Set up the dataset as a `Lexis` object with age, calendar time and duration of diabetes as timescales, and date of death as event. Make sure that you know what each of the arguments to `Lexis` mean:

```
> LL <- Lexis( entry = list( A = dodm-dobth,
+                           P = dodm,
+                           dur = 0 ),
+             exit = list( P = dox ),
+             exit.status = factor( !is.na(dodth),
+                                   labels=c("Alive", "Dead") ),
+             data = DMLate )
```

Take a look at the first few lines of the resulting dataset using `head()`.

- Get an overall overview of the mortality by using `stat.table` to tabulate no. deaths, person-years and the crude mortality rate by sex.
- If we want to assess how mortality depends on age, calendar time and duration, we should split the follow-up along all three time scales. In practice it is sufficient to split it along one of the time-scales and then just use the value of each of the time-scales at the left endpoint of the intervals.

Use `splitLexis` to split the follow-up along the age-axis:

```
> SL <- splitLexis( LL, breaks=seq(0,125,1/2), time.scale="A" )
> summary( SL )
```

How many records are now in the dataset? How many person-years? Compare to the original `Lexis`-dataset.

- Now estimate an age-specific mortality curve for men and women separately, using natural splines:

```
> library( splines )
> r.m <- glm( (lex.Xst=="Dead") ~ ns( A, df=10 ),
+           offset = log( lex.dur ),
+           family = poisson,
+           data = subset( SL, sex=="M" ) )
> r.f <- update( r.m,
+              data = subset( SL, sex=="F" ) )
```

Make sure you understand all the components on this modeling statement.

- Now try to get the estimated rates by using the wrapper function `ci.pred` that computes predicted rates and confidence limits for these.

Note that `lex.dur` is a covariate in the context of prediction; by putting this to 1000 in the prediction dataset we get the rates in units of deaths per 1000 PY:

```
> nd <- data.frame( A = seq(10,90,0.5),
+                 lex.dur = 1000)
> p.m <- ci.pred( r.m, newdata = nd )
> str( p.m )
```

- Plot the predicted rates for men and women together - using for example `matplot`.

Period and duration effects

8. We now want to model the mortality rates among diabetes patients also including current date and duration of diabetes. However, we shall not just use the positioning of knots for the splines as provided by `ns`, because this is based on the allocating knots so that the number of observations in the dataset is the same between knots. The information in a follow-up study is in the number of events, so it would be better to allocate knots so that number of events were the same between knots.

We take the 5th and 95th percentile of deaths as the boundary knots for age (`A`) and calendar time (`P`), but for duration (`dur`) where we actually have follow-up from time 0 on the timescale, we use 0 as the first knot.

Therefore, find points (knots) so that the number of events is the same between each pair. Try this:

```
> kn.A <- with( subset( SL, lex.Xst=="Dead" ),
+              quantile( A+lex.dur, probs=seq(5,95,20)/100 ) )
```

Take a look at where these points are and make a similar construction for calendar time (`P`) and diabetes duration (`dur`) — remember to add 0 as a knot for the latter.

9. With knots for age, period and duration we can now model mortality rates (separately for men and women), as functions of age, calendar time and duration of diabetes. To this end you will need the function `Ns` from the `Epi` package (look that up) to specify a model very simply

```
> mx <- glm( (lex.Xst=="Dead") ~ Ns( A, kn=kn.A ) +
+              Ns( P, kn=kn.P ) +
+              Ns( dur, kn=kn.dur ),
+              offset = log( lex.dur ),
+              family = poisson,
+              data = subset( SL, sex=="M" ) )
```

10. How do these models fit relative to the models with only age as a descriptor of the rates?

(Hint: Use the `anova`-function with the argument `test="Chisq"` to compare the models.

11. If we want to see the shape of the three effects we can use the `type="terms"` facility in the `predict.glm` that makes predictions separately for each term in the model. But this does not include the intercept, so if we want prediction of terms that add up to the total predicted value we must explicitly include the intercept in one of the terms; age, say, thereby making age the term with a rate-dimension and interpretable as age-specific rates.

This requires that we select reference points for the other terms, period and duration.

This is done by using the `intercept` and `ref` arguments to `Ns`:

```

> mm <- glm( (lex.Xst=="Dead") ~ Ns( A, kn=kn.A , intercept=TRUE ) - 1 +
+                               Ns( P, kn=kn.P , ref=2000 ) +
+                               Ns( dur, kn=kn.dur, ref=5 ),
+                               offset = log( lex.dur/100 ),
+                               family = poisson,
+                               data = subset( SL, sex=="M" ) )

```

Check that it actually is the same model, for example by using the deviances from the two models fitted.

12. Once this is done we can use `Termplot`, which is a wrapper for `termplot`. `Termplot` gives plots on the rate / resp RR scale, so that we can actually make sense of the plots. Now make a plot of the three effects in the model:

```

> Termplot( mx )

```

What is the interpretation of the three terms in the model?

13. The model we fitted has three time-scales: current age, current date and current duration of diabetes, so the effects that we report are not immediately interpretable, as they are (as in any kind of multiple regressions) to be interpreted as “all else equal” which they are not, as the three time scales advance simultaneously at the same pace.

The reporting would therefore more naturally be *only* on the mortality scale, but showing the mortality for persons diagnosed in different ages, using separate displays for separate years of diagnosis.

This is most easily done using the `ci.pred` function with the `newdata=` argument. So a person diagnosed in age 50 in 1995 will have a mortality measured in cases per 1000 PY as:

```

> pts <- seq(0,20,1)
> nd <- data.frame( A= 50+pts,
+                  P=1995+pts,
+                  dur= pts,
+                  lex.dur=1000 )
> ci.pred( mm, newdata=nd )

```

Now take a look at the result from the `ci.pred` statement and construct prediction of mortality for men and women diagnosed in a range of ages, say 50, 60, 70, and plot these together in the same graph.

2.2.1 SMR

The SMR is the **Standardized Mortality Ratio**, which is the mortality rate-ratio between the diabetes patients and the general population. In real studies we would subtract the deaths and the person-years among the diabetes patients from those of the general population, but since we do not have access to these, we make the comparison to the general population at large, *i.e.* also including the diabetes patients.

There are two ways to make the comparison to the population mortality; one is to amend the diabetes patient dataset with the population mortality dataset, the other (classical) one is to include the population mortality rates as a fixed variable in the calculations.

The latter requires that each analytical unit in the diabetes patient dataset is amended with a variable with the population mortality rate for the corresponding sex, age and calendar time.

This can be achieved in two ways: Either we just use the current split of follow-up time and allocate the population mortality rates for some suitably chosen (mid-)point of the follow-up in each, or we make a second split by date, so that follow-up in the diabetes patients is in the same classification of age and data as the population mortality table.

14. We will use the former approach, that is in the diabetes dataset to include as an extra variable the population mortality as available from the data set `M.dk`.

First create the variables in the diabetes dataset that we need for matching with the population mortality data, that is age, date and sex at the midpoint of each of the intervals (or rather at a point 3 months after the left endpoint of the interval — recall we split the follow-up in 6 month intervals).

We need to have variables of the same type when we merge, so we must transform the sex variable in `M.dk` to a factor, and must for each follow-up interval in the `SL` data have an age and a period variable that can be used in merging with the population data.

```
> str( SL )
> SL$Am <- floor( SL$A+0.25 )
> SL$Pm <- floor( SL$P+0.25 )
> data( M.dk )
> str( M.dk )
> M.dk <- transform( M.dk, Am = A,
+                    Pm = P,
+                    sex = factor( sex, labels=c("M","F") ) )
> str( M.dk )
```

Then match the rates from `M.dk` into `SL` — `sex`, `Am` and `Pm` are the common variables, and therefore the match is on these variables:

```
> SLr <- merge( SL, M.dk[,c("sex","Am","Pm","rate")] )
> dim( SL )
> dim( SLr )
```

This merge only takes rows that have information from both datasets, hence the slightly fewer rows in `SLr` than in `SL`.

15. Compute the expected number of deaths as the person-time multiplied by the corresponding population rate, and put it in a new variable. Use `stat.table` to make a table of observed, expected and the ratio (SMR) by age (suitably grouped) and sex.
16. Then model the SMR using age and date of diagnosis and diabetes duration as explanatory variables, including the log-expected-number instead of the log-person-years as offset, using separate models for men and women. Remember to exclude those units where no deaths in the population occur (that is where the rate is 0).

Plot the estimates as you did before for the rates, using `Termplot`. What do the extracted effects represent now?

17. Is there any difference between SMR for males and females?
18. Plot the predicted SMR as you did the predicted rates for persons aged 50, 60 and 70 at diagnosis.
19. Try to simplify the model to one with a simple linear effect of date of diagnosis, and using only knots at 0,1,and 2 years for duration, giving an estimate of the change in SMR as duration increases beyond 2 years.
20. What are the estimated annual change in SMR by date of diagnosis and by duration after 2 years?

Chapter 3

Solutions

3.1 Renal complications: Time-dependent variables and multiple states

The following practical exercise is based on the data from paper:

P Hovind, L Tarnow, P Rossing, B Carstensen, and HH Parving: Improved survival in patients obtaining remission of nephrotic range albuminuria in diabetic nephropathy. *Kidney Int*, **66**(3):1180–1186, Sept 2004.

You can find a .pdf-version of the paper here:

<http://BendixCarstensen.com/~bxc/AdvCoh/papers/Hovind.2004.pdf>

3.1.1 The renal failure dataset

The dataset `renal.dta` contains data on follow up of 125 patients from Steno Diabetes Center. They enter the study when they are diagnosed with nephrotic range albuminuria (NRA). This is a condition where the levels of albumin in the urine is exceeds a certain level as a sign of kidney disease. The levels may however drop as a consequence of treatment, this is called remission. Patients exit the study at death or kidney failure (dialysis or transplant).

Table 3.1: *Variables in renal.dta.*

<code>id</code>	Patient id
<code>sex</code>	M / F
<code>dob</code>	Date of birth
<code>doe</code>	Date of entry into the study (2.5 years after NRA)
<code>dor</code>	Date of remission. Missing if no remission has occurred
<code>dox</code>	Date of exit from study
<code>event</code>	Exit status: 0: censored, 1: death, 2: end stage renal disease, ESRD (kidney failure) and 3: Kidney transplant

1. The dataset is available at the course website as `renal.Rda`:

```

library( Epi ) ; clear()
load( url("http://BendixCarstensen.com/AdvCoh/courses/Frias-2016/renal.Rda") )
# load( "renal.Rda" )
str( renal )

'data.frame':      125 obs. of  7 variables:
 $ id   : num  17 26 27 33 42 46 47 55 62 64 ...
 $ sex  : Factor w/ 2 levels "M","F": 1 2 2 1 2 2 1 1 2 1 ...
 $ dob  : num  1968 1959 1962 1951 1961 ...
 $ doe  : num  1996 1990 1988 1995 1988 ...
 $ dor  : num  NA 1990 NA 1996 1997 ...
 $ dox  : num  1997 1996 1993 2004 2004 ...
 $ event: num  2 1 3 0 0 2 1 0 2 0 ...

head( renal )

   id sex   dob   doe   dor   dox event
1  17  M 1967.944 1996.013    NA 1997.094    2
2  26  F 1959.306 1989.535 1989.814 1996.136    1
3  27  F 1962.014 1987.846    NA 1993.239    3
4  33  M 1950.747 1995.243 1995.717 2003.993    0
5  42  F 1961.296 1987.884 1996.650 2003.955    0
6  46  F 1952.374 1983.419    NA 1991.484    2

```

Here we shall only be interested in the combined event 1, 2 or 3.

- Use the `Lexis` function to declare the data as survival data with age, calendar time and time since entry into the study as timescales. Note that any coding of event > 0 will be labeled “ESRD”, i.e. renal death (death of kidney (transplant or dialysis), or person).

Note that you must make sure that the “alive” state (here `NRA`) is the first, as `Lexis` assumes that everyone starts in this state (unless of course `entry.status` is specified).

```

Lr <- Lexis( entry = list( per=doe,
                          age=doe-dob,
                          tfi=0 ),
            exit = list( per=dox ),
            exit.status = factor( event>0, labels=c("NRA","ESRD") ),
            data = renal )

```

NOTE: `entry.status` has been set to "NRA" for all.

```
str( Lr )
```

```

Classes 'Lexis' and 'data.frame':      125 obs. of  14 variables:
 $ per   : num  1996 1990 1988 1995 1988 ...
 $ age   : num  28.1 30.2 25.8 44.5 26.6 ...
 $ tfi   : num  0 0 0 0 0 0 0 0 0 0 ...
 $ lex.dur: num  1.08 6.6 5.39 8.75 16.07 ...
 $ lex.Cst: Factor w/ 2 levels "NRA","ESRD": 1 1 1 1 1 1 1 1 1 1 ...
 $ lex.Xst: Factor w/ 2 levels "NRA","ESRD": 2 2 2 1 1 2 2 1 2 1 ...
 $ lex.id : int  1 2 3 4 5 6 7 8 9 10 ...
 $ id    : num  17 26 27 33 42 46 47 55 62 64 ...
 $ sex   : Factor w/ 2 levels "M","F": 1 2 2 1 2 2 1 1 2 1 ...
 $ dob   : num  1968 1959 1962 1951 1961 ...
 $ doe   : num  1996 1990 1988 1995 1988 ...

```

```
$ dor      : num  NA 1990 NA 1996 1997 ...
$ dox      : num  1997 1996 1993 2004 2004 ...
$ event    : num  2 1 3 0 0 2 1 0 2 0 ...
- attr(*, "time.scales")= chr  "per" "age" "tfi"
- attr(*, "time.since")= chr  "" "" ""
- attr(*, "breaks")=List of 3
..$ per: NULL
..$ age: NULL
..$ tfi: NULL
```

```
summary( Lr )
```

Transitions:

	To					
From	NRA	ESRD	Records:	Events:	Risk time:	Persons:
	NRA	48	77	125	77	1084.67
						125

- Visualize the data in a Lexis-diagram, using the `plot` method for Lexis objects. What do you see?

```
plot( Lr, col="black", lwd=3 )
```

- (Optional, not crucial to the rest of the exercise. Now try to produce a slightly more fancy Lexis diagram. Note that we have a x -axis of 40 years, and a y -axis of 80 years, so when specifying the output file adjust the *total* width of the plot so that the use `mai` to specify the margins of the plot leaves a plotting area twice as high as wide. You will want to consult the maning of the argument `mai` to the function `par`.

```
# pdf( "lexis-fancy.pdf", height=80/5+1, width=40/5+1 )
# x11( height=80/5+1, width=40/5+1 )
par( mai=c(3,3,1,1)/4, mgp=c(3,1,0)/1.6 )
plot( Lr, 1:2, col=c("blue","red")[Lr$sex], lwd=3, grid=0:20*5,
      xlab="Calendar time", ylab="Age",
      xlim=c(1970,2010), ylim=c(0,80), xaxs="i", yaxs="i", las=1 )
# dev.off()
```

- Make a Cox-regression analysis with the variables `sex` and `age` at entry into the study, using `time since entry to the study` as time scale.

Give the hazard ratio between males and females and between two persons who differ 10 years in age at entry. Give the 95% confidence intervals for this as well.

```
library( survival )
mc <- coxph( Surv( lex.dur, lex.Xst=="ESRD" ) ~
            I(age/10) + sex, data=Lr )
summary( mc )
```

Call:

```
coxph(formula = Surv(lex.dur, lex.Xst == "ESRD") ~ I(age/10) +
      sex, data = Lr)
```

```
n= 125, number of events= 77

      coef exp(coef) se(coef)      z Pr(>|z|)
I(age/10) 0.5514    1.7357   0.1402  3.932 8.43e-05
```

```

sexF      -0.1817    0.8338    0.2727 -0.666    0.505

              exp(coef) exp(-coef) lower .95 upper .95
I(age/10)    1.7357    0.5761    1.3186    2.285
sexF         0.8338    1.1993    0.4886    1.423

Concordance= 0.612 (se = 0.036 )
Rsquare= 0.121 (max possible= 0.994 )
Likelihood ratio test= 16.07 on 2 df, p=0.0003237
Wald test = 16.38 on 2 df, p=0.0002774
Score (logrank) test = 16.77 on 2 df, p=0.0002282

```

6. The main focus of the paper was to assess whether occurrence of remission (return to a lower level of albumin excretion, an indication of kidney recovery) influences mortality.

“Remission” is a time-dependent variable which is initially 0, but takes the value 1 when remission occurs. In order to handle this, each person who see a remission must have two records:

- One record for the time before remission, where entry is `doe`, exit is `dor`, remission is 0, and event is 0.
- One record for the time after remission, where entry is `dor`, exit is `dox`, remission is 1, and event is 0 or 1 according to whether the person had an event at `dox`.

This is accomplished using the `cutLexis` function on the `Lexis` object. You must declare the “NRA” state as a precursor state, i.e. a state that is *less* severe than “Rem” in the sense that a person who see a remission will stay in the “Rem” state unless he goes to the “ESRD” state.

```

Lc <- cutLexis( Lr, cut = Lr$dor, # where to cut follow up
               timescale = "per", # the timescale that "dor" refers to
               new.state = "Rem", # name of the new state
               precursor.states = "NRA" ) # which states are less severe
summary( Lc )

```

Transitions:

	To							
From	NRA	Rem	ESRD	Records:	Events:	Risk time:	Persons:	
NRA	24	29	69	122	98	824.77	122	
Rem	0	24	8	32	8	259.90	32	
Sum	24	53	77	154	106	1084.67	125	

List records for a few select persons from `Lr` and from `Lc` to see how the cut has worked.

7. Show how the states are connected and the number of transitions between them by using `boxes`. This is an interactive command that requires you to click in the graph window:

```
boxes( Lc )
```

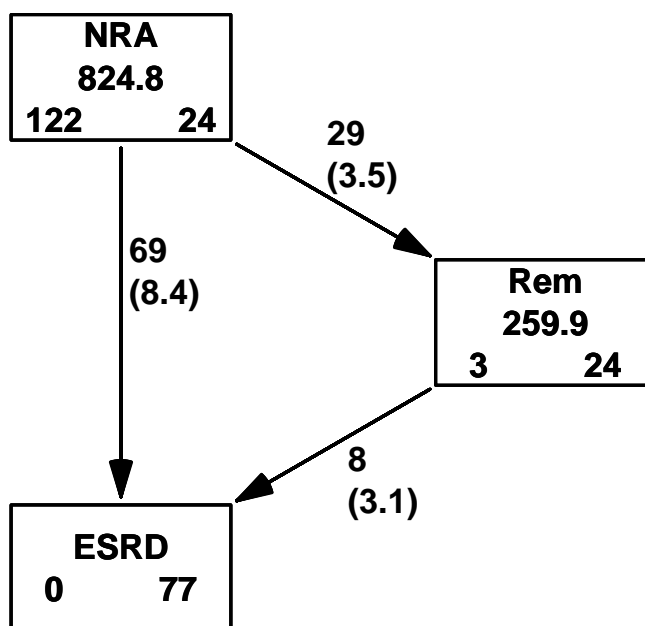


Figure 3.1

Alternatively you can let R try to place the boxes for you, and even compute rates (in this case in units of events per 100 PY):

```
boxes( Lc, boxpos=TRUE, scale.R=100, show.BE=TRUE )
```

How many transitions are there from remission to death?

- (Optional: Not relevant for the remainder of the exercise.) Now make a Lexis diagram where different colouring is used for different segments of the follow-up — you should be able to count the 8 transitions from “Rem” to “ESRD”.

```
par( mai=c(3,3,1,1)/4, mgp=c(3,1,0)/1.6 )
plot( Lc, col=c("red","limegreen")[(Lc$lex.Cst=="Rem")+1],
      xlab="Calendar time", ylab="Age",
      lwd=3, grid=0:20*5, xlim=c(1970,2010), ylim=c(0,80), xaxs="i", yaxs="i", las=1
      points( Lc, pch=c(NA,16)[(Lc$lex.Xst=="ESRD")+1],
             col=c("red","limegreen")[(Lc$lex.Cst=="Rem")+1])
      points( Lc, pch=c(NA,1)[(Lc$lex.Xst=="ESRD")+1],
             col="black", lwd=2 )
```

- Make a Cox-regression of mortality (i.e. endpoint “ESRD”) with sex, age at entry and remission as explanatory variables, and using time since entry as timescale.

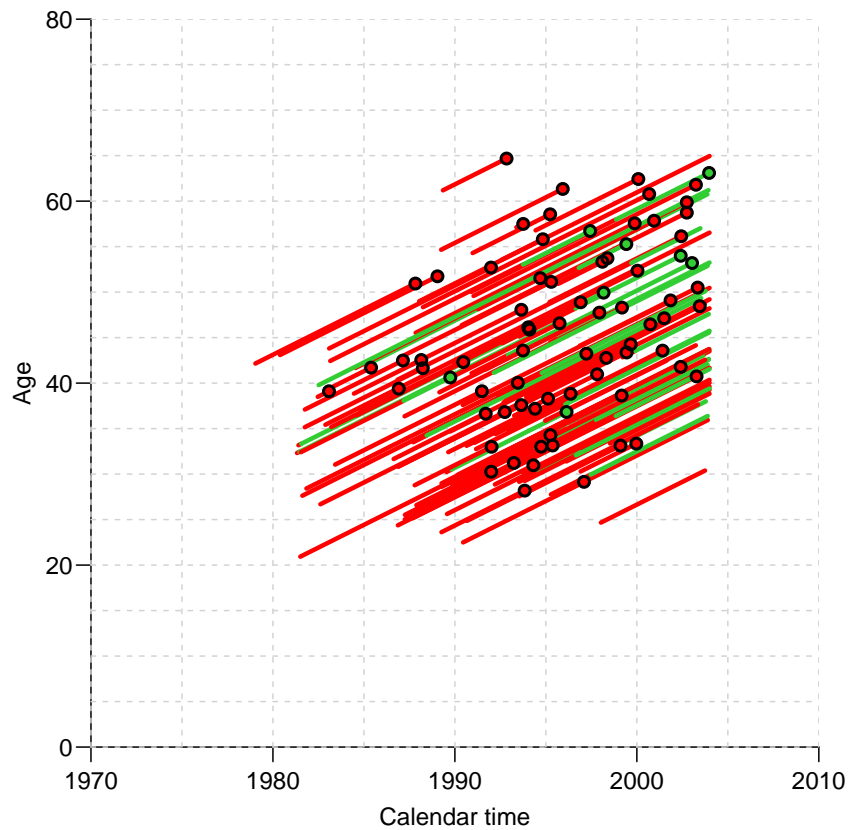


Figure 3.2: *Lexis diagram for the split data, where time after remission is shown in green.*

Remember to include `lex.Cst` as time-dependent variable, and to indicate that each record represents follow-up from `tfi` to `tfi+lex.dur`. Note the use of the Lexis variables `lex.dur` (risk time), `lex.Xst` (exit status) and `lex.Cst` (current status).

```
m1 <- coxph( Surv( tfi, tfi+lex.dur, lex.Xst=="ESRD" ) ~
              sex + I((doe-dob-50)/10) + (lex.Cst=="Rem"), data=Lc )
summary( m1 )
```

Call:

```
coxph(formula = Surv(tfi, tfi + lex.dur, lex.Xst == "ESRD") ~
      sex + I((doe - dob - 50)/10) + (lex.Cst == "Rem"), data = Lc)
```

n= 154, number of events= 77

	coef	exp(coef)	se(coef)	z	Pr(> z)
sexF	-0.05534	0.94616	0.27500	-0.201	0.840517
I((doe - dob - 50)/10)	0.52190	1.68522	0.13655	3.822	0.000132
lex.Cst == "Rem"TRUE	-1.26241	0.28297	0.38483	-3.280	0.001036

	exp(coef)	exp(-coef)	lower .95	upper .95
sexF	0.9462	1.0569	0.5519	1.6220
I((doe - dob - 50)/10)	1.6852	0.5934	1.2895	2.2024
lex.Cst == "Rem"TRUE	0.2830	3.5339	0.1331	0.6016

Concordance= 0.664 (se = 0.036)

Rsquare= 0.179 (max possible= 0.984)


```
Likelihood ratio test= 30.31 on 3 df, p=1.189e-06
Wald test              = 27.07 on 3 df, p=5.683e-06
Score (logrank) test = 29.41 on 3 df, p=1.84e-06
```

10. What is the relation between the rate of ESRD between persons in remission and persons not?
11. What is the assumption about the two rates of remission? Refer to the figure with the three boxes you just made. (??).

3.1.2 Splitting the follow-up time

In order to explore the effect of remission on the rate of ESRD, we will split the data further into small pieces of follow-up. To this end we use the function `splitLexis`. The rates can then be modeled using a Poisson-model, and the shape of the underlying *rates* be explored. Furthermore, we can allow effects of both time since NRA and current age. To this end we will use splines, so we need the splines package, too.

12. First, split the follow-up time every month after entry, and make sure that the number of events and risk time is the same as before (use `summary`):

```
sLc <- splitLexis( Lc, "tfi", breaks=seq(0,30,1/12) )
summary( Lc )
```

```
Transitions:
  To
From  NRA Rem ESRD  Records:  Events: Risk time:  Persons:
  NRA  24  29  69      122      98      824.77      122
  Rem   0  24   8       32       8       259.90       32
  Sum  24  53  77      154     106     1084.67     125
```

```
summary(sLc )
```

```
Transitions:
  To
From  NRA Rem ESRD  Records:  Events: Risk time:  Persons:
  NRA 9854  29  69     9952      98     824.77     122
  Rem   0 3139  8     3147       8     259.90     32
  Sum 9854 3168 77    13099     106     1084.67     125
```

13. Now try to fit the Poisson-model corresponding to the Cox-model we fitted previously. The function `ns()` produces a model matrix corresponding to a piecewise cubic function, modeling the baseline hazard explicitly (think of the `ns` terms as the baseline hazard that is not visible in the Cox-model).

The outcome is 1 or 0 according to whether an event occurred or not, but since a Poisson variate by definition is numerical, R will automatically coerce (change) a logical value to numeric; FALSE as 0 and TRUE as 1, so we can conveniently write:

```

library( splines )
mp <- glm( lex.Xst=="ESRD" ~ ns( tfi, df=4 ) +
          sex + I((doe-dob-40)/10) + (lex.Cst=="Rem"),
          offset = log(lex.dur),
          family = poisson,
          data = sLc )
summary( mp )

Call:
glm(formula = lex.Xst == "ESRD" ~ ns(tfi, df = 4) + sex + I((doe -
  dob - 40)/10) + (lex.Cst == "Rem"), family = poisson, data = sLc,
  offset = log(lex.dur))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.2379  -0.1250  -0.0935  -0.0669   3.7987

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)      -3.93862    0.72879  -5.404 6.51e-08
ns(tfi, df = 4)1    2.10754    0.72379   2.912 0.003593
ns(tfi, df = 4)2    1.42695    0.69738   2.046 0.040741
ns(tfi, df = 4)3    3.49151    1.66427   2.098 0.035912
ns(tfi, df = 4)4    2.47260    1.08261   2.284 0.022376
sexF              -0.08043    0.27427  -0.293 0.769331
I((doe - dob - 40)/10) 0.53187    0.13714   3.878 0.000105
lex.Cst == "Rem"TRUE  -1.27858    0.38530  -3.318 0.000905

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 898.74  on 13098  degrees of freedom
Residual deviance: 853.54  on 13091  degrees of freedom
AIC: 1023.5

Number of Fisher Scoring iterations: 8

```

The `ns` function places knots at the quantiles of the variable, which may not be the most logical as the information is contained in the events, so the natural placement of knots would be at the quantiles of the event times. The `Ns` function in the *Epi* package automatically takes the smallest and the largest of the knots as boundary knots — the number of parameters is one less than the number of knots, so we use 5 knots:

```

t.kn <- with( subset( sLc, lex.Xst=="ESRD"),
             quantile( tfi+lex.dur, 0:4/5 ) )
mp <- glm( lex.Xst=="ESRD" ~ Ns( tfi, knots=t.kn ) +
          sex + I((doe-dob-40)/10) + (lex.Cst=="Rem"),
          offset = log(lex.dur),
          family = poisson,
          data = sLc )
summary( mp )

Call:
glm(formula = lex.Xst == "ESRD" ~ Ns(tfi, knots = t.kn) + sex +
  I((doe - dob - 40)/10) + (lex.Cst == "Rem"), family = poisson,
  data = sLc, offset = log(lex.dur))

Deviance Residuals:

```

```

      Min      1Q   Median      3Q      Max
-0.2573 -0.1250 -0.0923 -0.0662  3.7779

```

Coefficients:

```

              Estimate Std. Error z value Pr(>|z|)
(Intercept)    -3.59600    0.51242  -7.018 2.25e-12
Ns(tfi, knots = t.kn)1  1.95218    0.56113   3.479 0.000503
Ns(tfi, knots = t.kn)2  1.10038    0.41479   2.653 0.007982
Ns(tfi, knots = t.kn)3  2.30320    1.27583   1.805 0.071035
Ns(tfi, knots = t.kn)4  1.31387    0.32577   4.033 5.50e-05
sexF            -0.06981    0.27476  -0.254 0.799427
I((doe - dob - 40)/10)  0.53114    0.13723   3.871 0.000109
lex.Cst == "Rem"TRUE  -1.27896    0.38555  -3.317 0.000909

```

(Dispersion parameter for poisson family taken to be 1)

```

Null deviance: 898.74 on 13098 degrees of freedom
Residual deviance: 852.46 on 13091 degrees of freedom
AIC: 1022.5

```

Number of Fisher Scoring iterations: 8

14. You can extract the parameters from the models using `ci.lin` or `ci.exp` try:

```
ci.lin( mp )
```

```

              Estimate   StdErr      z      P      2.5%
(Intercept) -3.59600346 0.5124153 -7.0177518 2.254666e-12 -4.6003190
Ns(tfi, knots = t.kn)1  1.95218041 0.5611331  3.4789971 5.032941e-04  0.8523797
Ns(tfi, knots = t.kn)2  1.10038347 0.4147933  2.6528474 7.981594e-03  0.2874034
Ns(tfi, knots = t.kn)3  2.30319625 1.2758310  1.8052518 7.103529e-02 -0.1973866
Ns(tfi, knots = t.kn)4  1.31387392 0.3257688  4.0331480 5.503459e-05  0.6753787
sexF          -0.06981245 0.2747561 -0.2540888 7.994269e-01 -0.6083245
I((doe - dob - 40)/10)  0.53114373 0.1372277  3.8705290 1.085995e-04  0.2621824
lex.Cst == "Rem"TRUE  -1.27896398 0.3855503 -3.3172430 9.091051e-04 -2.0346287
              97.5%
(Intercept) -2.5916879
Ns(tfi, knots = t.kn)1  3.0519811
Ns(tfi, knots = t.kn)2  1.9133635
Ns(tfi, knots = t.kn)3  4.8037791
Ns(tfi, knots = t.kn)4  1.9523691
sexF          0.4686996
I((doe - dob - 40)/10)  0.8001050
lex.Cst == "Rem"TRUE  -0.5232993

```

```
ci.exp( mp )
```

```

              exp(Est.)      2.5%      97.5%
(Intercept)  0.02743314 0.01004863  0.07489352
Ns(tfi, knots = t.kn)1  7.04402972 2.34522125 21.15721685
Ns(tfi, knots = t.kn)2  3.00531826 1.33296189  6.77584104
Ns(tfi, knots = t.kn)3 10.00611340 0.82087321 121.97048700
Ns(tfi, knots = t.kn)4  3.72055898 1.96477698  7.04535900
sexF          0.93256870 0.54426203  1.59791487
I((doe - dob - 40)/10)  1.70087654 1.29976361  2.22577473
lex.Cst == "Rem"TRUE  0.27832550 0.13072902  0.59256227

```

```
ci.exp( mp, subset=c("sex","dob","Cst"), pval=TRUE )
```

```

                exp(Est.)    2.5%    97.5%    P
sexF            0.9325687 0.544262 1.5979149 0.7994269292
I((doe - dob - 40)/10) 1.7008765 1.299764 2.2257747 0.0001085995
lex.Cst == "Rem"TRUE 0.2783255 0.130729 0.5925623 0.0009091051

```

Compare with the estimates from the Cox-model. Use:

```

ci.exp( m1 )

                exp(Est.)    2.5%    97.5%
sexF            0.9461646 0.5519334 1.621985
I((doe - dob - 50)/10) 1.6852196 1.2895097 2.202360
lex.Cst == "Rem"TRUE 0.2829710 0.1330996 0.601599

ci.exp( mp, subset=c("sex","dob","Cst") )

                exp(Est.)    2.5%    97.5%
sexF            0.9325687 0.544262 1.5979149
I((doe - dob - 40)/10) 1.7008765 1.299764 2.2257747
lex.Cst == "Rem"TRUE 0.2783255 0.130729 0.5925623

ci.exp( mp, subset=c("sex","dob","Cst") ) / ci.exp( m1 )

                exp(Est.)    2.5%    97.5%
sexF            0.9856305 0.9861009 0.9851603
I((doe - dob - 40)/10) 1.0092907 1.0079518 1.0106315
lex.Cst == "Rem"TRUE 0.9835830 0.9821891 0.9849789

```

What do you conclude about the models?

15. You can visualize the spline term using `termplot`, try:

```
termplot( mp, terms=1 )
```

... which is not a terribly informative plot

16. `termplot` does not give you the absolute level of the underlying rates because it bypasses the intercept. If you explicitly include the intercept in the baseline split you can use `Termplot` from the `Epi` package to get estimates on the rate scale for a reference person (in units of events per 100 years):

```

mP <- glm( lex.Xst=="ESRD" ~ -1 + Ns( tfi, knots=t.kn, intercept=TRUE ) +
           sex + I((doe-dob-40)/10) + (lex.Cst=="Rem"),
           offset = log(lex.dur/100),
           family = poisson,
           data = sLc )
Termplot( mP, terms=1 )

```

How would you describe this rate function in plain words? And what is the scale of the y -axis.

Annotate the axes of the plot accordingly — consult the help page of `Termplot`.

17. Apart from the baseline timescale, time since NRA, time since remission might be of interest in describing the mortality rate. However this is only relevant for persons who actually have a remission, so start by checking how many events there are in this group:

```
summary( sLc )
```

Transitions:

From	To	NRA	Rem	ESRD	Records:	Events:	Risk time:	Persons:
NRA	NRA	9854	29	69	9952	98	824.77	122
Rem	Rem	0	3139	8	3147	8	259.90	32
Sum	Sum	9854	3168	77	13099	106	1084.67	125

How many go in remission, and how many deaths are in this group?

18. With this rather limited number of events we can certainly not expect to be able to model anything more complicated than a linear trend with time since remission. Two parameters on 8 events is actually pretty far-fetched.

The variable we want to have in the model is current date (`per`) minus date of remission (`dor`): `per-dor`), but *only* positive values of it. This can be fixed by using `pmax()`, but we must also deal with all those who have missing values, so we use the construct:

```
pmax( per-dor, 0, na.rm=TRUE )
```

Make sure that you understand what goes on here.

19. We can now expand the model with this variable:

```
sLc <- transform( sLc, tfr = pmax( (per-dor)/10, 0, na.rm=TRUE ) )
mPx <- glm( lex.Xst=="ESRD" ~ -1 + Ns( tfr, knots=t.kn, intercept=TRUE ) +
           sex + I((age-tfi-40)/10) + (lex.Cst=="Rem") + tfr,
           offset = log(lex.dur/100),
           family = poisson,
           data = sLc )
round( ci.exp( mPx ), 3 )
```

	exp(Est.)	2.5%	97.5%
Ns(tfi, knots = t.kn, intercept = TRUE)1	4.789	1.466	15.641
Ns(tfi, knots = t.kn, intercept = TRUE)2	17.935	7.985	40.283
Ns(tfi, knots = t.kn, intercept = TRUE)3	5.581	2.649	11.760
Ns(tfi, knots = t.kn, intercept = TRUE)4	51.347	13.438	196.202
Ns(tfi, knots = t.kn, intercept = TRUE)5	6.427	3.368	12.266
sexM	1.079	0.628	1.853
sexF	1.000	1.000	1.000
I((age - tfi - 40)/10)	1.703	1.302	2.229
lex.Cst == "Rem"TRUE	0.310	0.097	0.989
tfr	0.847	0.210	3.412

```
Termplot( mPx, terms=1 )
```

20. Is the effect significant? Can a substantial effect of time since remission be ruled out?
21. What is the test of this parameter traditionally called? What is the null and what is the alternative of this test?

3.1.3 Prediction in a multistate model

This part of the practical is about making proper statements about the survival and the disease probabilities. But in order to do this we must know not only how the occurrence of remission influences the rate of death/ESRD, but we must also model the occurrence rate of remission itself.

The following exercise will be quite similar to the example in the help file for `simLexis` (which you should read now!).

22. The rates of ESRD were modelled by a Poisson model with effects of age and time since NRA — in the model `mp`. But in the modelling of the remission rates transition from “NRA” to “Rem”, the number of events is rather small, so we restrict the variables in this model to only time since NRA and sex. Also remember, only the records that relate to the “NRA” state can be used:

```
mr <- glm( lex.Xst=="Rem" ~ ns( tfi, knots=t.kn ) + sex,
          offset = log(lex.dur),
          family = poisson,
          data = subset( sLc, lex.Cst=="NRA" ) )
ci.exp( mr, pval=TRUE )
```

	exp(Est.)	2.5%	97.5%	P
(Intercept)	0.05606873	0.0155421035	0.2022701	1.075954e-05
ns(tfi, knots = t.kn)1	1.56250187	0.1758966092	13.8798132	6.888024e-01
ns(tfi, knots = t.kn)2	0.12621768	0.0105935727	1.5038272	1.015853e-01
ns(tfi, knots = t.kn)3	0.61154986	0.0701435838	5.3318238	6.562519e-01
ns(tfi, knots = t.kn)4	0.97532990	0.0280655093	33.8945715	9.889910e-01
ns(tfi, knots = t.kn)5	0.08049791	0.0004655089	13.9200643	3.378924e-01
ns(tfi, knots = t.kn)6	0.65167781	0.0002104090	2018.3737166	9.168447e-01
sexF	2.64124116	1.2658909206	5.5108657	9.645522e-03

23. If we want to predict the probability of being in each of the three states using these estimated rates, we can either do analytical calculations of the probabilities from the estimated rates, or we can *simulate* the life course through a model using the estimated rates. That will give a simulated cohort (in the form of a Lexis object), and we can then just count the number of persons in each state at each of a set of time points.

This is accomplished using the function `simLexis`. The input to this is the initial status of the persons whose life-course we shall simulate, and the transition rates in suitable form:

- Suppose we want predictions for men aged 50 at NRA. The input is in the form of a Lexis object (where `lex.dur` and `lex.Xst` will be ignored). Note that in order to carry over the `time.scales` and the `time.since` attributes, we construct the input object using `subset` to select columns, and `NULL` to select rows (see the example in the help file for `simLexis`):

```
inL <- subset( sLc, select=1:11 )[NULL,]
str( inL )
Classes 'Lexis' and 'data.frame':      0 obs. of  11 variables:
 $ lex.id : int
 $ per    : num
```

```

$ age      : num
$ tfi      : num
$ lex.dur  : num
$ lex.Cst  : Factor w/ 3 levels "NRA","Rem","ESRD":
$ lex.Xst  : Factor w/ 3 levels "NRA","Rem","ESRD":
$ id       : num
$ sex      : Factor w/ 2 levels "M","F":
$ dob      : num
$ doe      : num
- attr(*, "breaks")=List of 3
..$ per: NULL
..$ age: NULL
..$ tfi: num 0 0.0833 0.1667 0.25 0.3333 ...
- attr(*, "time.scales")= chr "per" "age" "tfi"
- attr(*, "time.since")= chr "" "" ""
timeScales(inL)
[1] "per" "age" "tfi"
inL[1,"lex.id"] <- 1
inL[1,"per"] <- 2000
inL[1,"age"] <- 50
inL[1,"tfi"] <- 0
inL[1,"lex.Cst"] <- "NRA"
inL[1,"lex.Xst"] <- NA
inL[1,"lex.dur"] <- NA
inL[1,"sex"] <- "M"
inL[1,"dob"] <- 2000
inL[1,"dob"] <- 1950
inL
lex.id per age tfi lex.dur lex.Cst lex.Xst id sex dob doe
1      1 2000 50 0      NA      NRA      <NA> NA  M 1950 2000

```

- The other input for the simulation is the transitions, which is a list with an element for each transient state (that is “NRA” and “Rem”), each of which is again a list with names equal to the states that can be reached from the transient state. The content of the list will be `glm` objects, in this case the models we just fitted, describing the transition rates:

```

Tr <- list( "NRA" = list( "Rem" = mr,
                        "ESRD" = mp ),
           "Rem" = list( "ESRD" = mp ) )

```

With this as input we can now generate a cohort, using `N=10` to simulate life course of 10 persons (with identical starting values):

```

( iL <- simLexis( Tr, inL, N=10 ) )
lex.id per age tfi lex.dur lex.Cst lex.Xst id sex dob doe cens
1      1 2000.000 50.00000 0.000000 3.9537448      NRA      ESRD NA  M 1950 2000 2020
2      2 2000.000 50.00000 0.000000 3.5109129      NRA      ESRD NA  M 1950 2000 2020
3      3 2000.000 50.00000 0.000000 7.7976187      NRA      Rem  NA  M 1950 2000 2020
4      3 2007.798 57.79762 7.797619 12.2023813      Rem      Rem  NA  M 1950 2000 2020
5      4 2000.000 50.00000 0.000000 5.8765578      NRA      ESRD NA  M 1950 2000 2020
6      5 2000.000 50.00000 0.000000 3.5747715      NRA      ESRD NA  M 1950 2000 2020
7      6 2000.000 50.00000 0.000000 0.4049354      NRA      ESRD NA  M 1950 2000 2020
8      7 2000.000 50.00000 0.000000 5.8315212      NRA      ESRD NA  M 1950 2000 2020
9      8 2000.000 50.00000 0.000000 6.1983270      NRA      ESRD NA  M 1950 2000 2020
10     9 2000.000 50.00000 0.000000 2.3599612      NRA      ESRD NA  M 1950 2000 2020
11    10 2000.000 50.00000 0.000000 3.9291606      NRA      ESRD NA  M 1950 2000 2020

```

```
summary( iL )
```

```
Transitions:
```

```
  To
From  NRA  Rem  ESRD  Records:  Events:  Risk time:  Persons:
  NRA   0   1   9       10       10       43.44       10
  Rem   0   1   0        1        0       12.20        1
  Sum   0   2   9       11       10       55.64       10
```

24. Now generate the life course of 10,000 persons, and look at the summary. The `system.time` command is just to tell you how long it took, you may want to start with 1000 just to see how long that takes.

```
system.time(
  sM <- simLexis( Tr, inL, N=10000 ) )
```

```
  user  system elapsed
16.266   0.192  16.466
```

```
summary( sM )
```

```
Transitions:
```

```
  To
From  NRA  Rem  ESRD  Records:  Events:  Risk time:  Persons:
  NRA  28 1404 8568   10000    9972   56134.77   10000
  Rem   0  391 1013    1404    1013   14345.93    1404
  Sum  28 1795 9581   11404   10985   70480.70   10000
```

Why are there so many ESRD-events in the resulting data set?

25. Now we want to count how many persons are present in each state at each time for the first 10 years after entry (which is at age 50). This can be done by using `nState`:

```
nSt <- nState( sM, at=seq(0,10,0.1), from=50, time.scale="age" )
head( nSt )
```

```
  State
when  NRA  Rem  ESRD
  50   10000  0    0
  50.1  9915  43   42
  50.2  9815  104  81
  50.3  9719  154  127
  50.4  9632  200  168
  50.5  9557  225  218
```

26. Once we have the counts of persons in each state at the designated time points, we compute the cumulative fraction over the states, arranged in order given by `perm`:

```
pp <- pState( nSt, perm=1:3 )
head( pp )
```



```

      State
when   NRA   Rem ESRD
  50    1.0000 1.0000    1
  50.1 0.9915 0.9958    1
  50.2 0.9815 0.9919    1
  50.3 0.9719 0.9873    1
  50.4 0.9632 0.9832    1
  50.5 0.9557 0.9782    1

```

```
tail( pp )
```

```

      State
when   NRA   Rem ESRD
  59.5 0.1470 0.2371    1
  59.6 0.1430 0.2329    1
  59.7 0.1392 0.2287    1
  59.8 0.1362 0.2254    1
  59.9 0.1330 0.2220    1
  60    0.1295 0.2186    1

```

27. Try to plot the cumulative probabilities using the `plot` method for `pState` objects:

```
plot( pp )
```

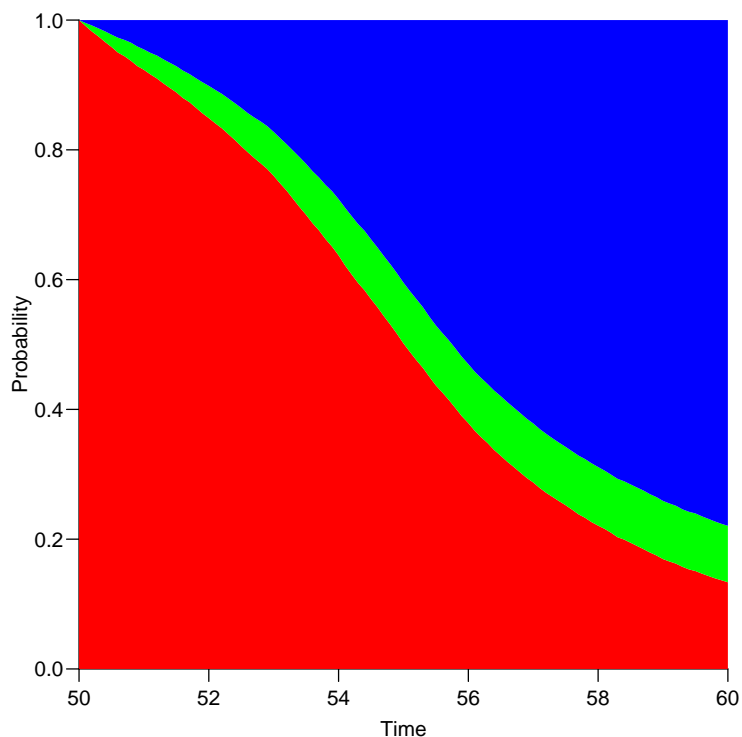


Figure 3.3: *Standard plot of state occupancy probabilities.*

28. A quantity of particular interest would be how many patients actually get a remission. This is not deductible from the plot just shown, because those who get ESRD are not subdivided according to whether they have a remission prior to ESRD.

The simplest way to doctor that is to modify the simulated object (`sM` in the above notation), so that those exiting to “ESRD” from “Rem” are counted in a separate state. We must also change the formal set of levels of `lex.Cst`:

```
xM <- transform( sM, lex.Xst = factor( ifelse( lex.Xst=="ESRD" & lex.Cst=="Rem",
                                             "ESRD(Rem)",
                                             as.character(lex.Xst) ),
                                       levels=c("NRA", "Rem", "ESRD(Rem)", "ESRD" ) ,
lex.Cst = factor( as.character(lex.Cst),
                  levels=c("NRA", "Rem", "ESRD(Rem)", "ESRD" ) ) )

summary( sM )

Transitions:
  To
From NRA  Rem  ESRD  Records:  Events: Risk time:  Persons:
NRA  28 1404 8568    10000    9972   56134.77    10000
Rem   0  391 1013     1404     1013   14345.93     1404
Sum  28 1795 9581    11404    10985   70480.70    10000

summary( xM )

Transitions:
  To
From  NRA  Rem  ESRD(Rem)  ESRD  Records:  Events: Risk time:  Persons:
NRA  28 1404         0 8568    10000    9972   56134.77    10000
Rem   0  391     1013   0    1404    1013   14345.93     1404
Sum  28 1795     1013 8568    11404    10985   70480.70    10000

boxes( xM, boxpos=TRUE, show.BE=TRUE, scale.R=100 )
```

29. Having done this, try to compute the number of persons in each of the 4 states, and the cumulative proportions to be plotted:

```
xSt <- nState( xM, at=seq(0,10,0.1), from=50, time.scale="age" )
xp <- pState( xSt, perm=1:4 )
head( xp )

      State
when   NRA   Rem  ESRD(Rem)  ESRD
50     1.0000 1.0000    1.0000    1
50.1   0.9915 0.9958    0.9958    1
50.2   0.9815 0.9919    0.9919    1
50.3   0.9719 0.9873    0.9873    1
50.4   0.9632 0.9832    0.9832    1
50.5   0.9557 0.9782    0.9782    1

plot( xp, col=rev(c("pink", "limegreen", "forestgreen", "red")), xlab="Age" )
lines( as.numeric(rownames(xp)), xp[, "Rem"], lwd=4 )
```

What is the probability that a 50-year old man with NRA sees a remission from NRA during the next 10 yezs?

30. Make the same calculations for a 60-year old woman.
31. Normally you would know that a split of the absorbing “ESRD” state according to the preceding state and so define this in the `cutLexis` function, using `split.states`. At the same time it is also possible to define a new timescale using `new.scale`, defined as time since entry to the new state:

```

Lc <- cutLexis( Lr, cut = Lr$dor, # where to cut follow up
               timescale = "per", # the timescale that "dor" refers to
               new.state = "Rem", # name of the new state
               precursor.states = "NRA", # which states are less severe
               new.scale = "tfr", # define a new timescale as time since Rem
               split.states = TRUE ) # subdivide non-precursor states

str( Lc )

Classes 'Lexis' and 'data.frame':      154 obs. of  15 variables:
 $ per   : num  1996 1990 1990 1988 1995 ...
 $ age   : num  28.1 30.2 30.5 25.8 44.5 ...
 $ tfi   : num  0 0 0.279 0 0 ...
 $ tfr   : num  NA NA 0 NA NA 0 NA 0 NA NA ...
 $ lex.dur: num  1.081 0.279 6.322 5.393 0.473 ...
 $ lex.Cst: Factor w/ 4 levels "NRA","Rem","ESRD",...: 1 1 2 1 1 2 1 2 1 1 ...
 $ lex.Xst: Factor w/ 4 levels "NRA","Rem","ESRD",...: 3 2 4 3 2 2 2 2 3 3 ...
 $ lex.id : int  1 2 2 3 4 4 5 5 6 7 ...
 $ id     : num  17 26 26 27 33 33 42 42 46 47 ...
 $ sex    : Factor w/ 2 levels "M","F": 1 2 2 2 1 1 2 2 2 1 ...
 $ dob    : num  1968 1959 1959 1962 1951 ...
 $ doe    : num  1996 1990 1990 1988 1995 ...
 $ dor    : num  NA 1990 1990 NA 1996 ...
 $ dox    : num  1997 1996 1996 1993 2004 ...
 $ event  : num  2 1 1 3 0 0 0 0 2 1 ...
 - attr(*, "breaks")=List of 4
 ..$ per: NULL
 ..$ age: NULL
 ..$ tfi: NULL
 ..$ tfr: NULL
 - attr(*, "time.scales")= chr  "per" "age" "tfi" "tfr"
 - attr(*, "time.since")= chr  "" "" "" "Rem"

# source("/home/bendix/stat/R/lib.src/Epi/pkg/R/summary.Lexis.r")
# summary( Lc, S=T, scale=100 )
summary( Lc )

```

Transitions:

	To				Records:	Events:	Risk time:	Persons:
From	NRA	Rem	ESRD	ESRD(Rem)				
NRA	24	29	69	0	122	98	824.77	122
Rem	0	24	0	8	32	8	259.90	32
Sum	24	53	69	8	154	106	1084.67	125

```

boxes( Lc, boxpos=list(x=c(20,80,20,80),y=c(80,80,20,20)),
       scale.R=100, show.BE=TRUE )
sLc <- splitLexis( Lc, "tfi", breaks=seq(0,30,1/12) )
summary( Lc )

```

Transitions:

	To				Records:	Events:	Risk time:	Persons:
From	NRA	Rem	ESRD	ESRD(Rem)				
NRA	24	29	69	0	122	98	824.77	122
Rem	0	24	0	8	32	8	259.90	32
Sum	24	53	69	8	154	106	1084.67	125

```
summary( sLc )
```

Transitions:

	To				Records:	Events:	Risk time:	Persons:
From	NRA	Rem	ESRD	ESRD(Rem)				
NRA	24	29	69	0	122	98	824.77	122
Rem	0	24	0	8	32	8	259.90	32
Sum	24	53	69	8	154	106	1084.67	125

NRA	9854	29	69	0	9952	98	824.77	122
Rem	0	3139	0	8	3147	8	259.90	32
Sum	9854	3168	69	8	13099	106	1084.67	125

```
head( subset( sLc, lex.id==2 ),[1:8], 8 )
```

	lex.id	per	age	tfi	tfr	lex.dur	lex.Cst	lex.Xst
14	2	1989.535	30.22895	0.00000000	NA	0.08333333	NRA	NRA
15	2	1989.618	30.31229	0.08333333	NA	0.08333333	NRA	NRA
16	2	1989.702	30.39562	0.16666667	NA	0.08333333	NRA	NRA
17	2	1989.785	30.47895	0.25000000	NA	0.02891855	NRA	Rem
18	2	1989.814	30.50787	0.27891855	0.00000000	0.05441478	Rem	Rem
19	2	1989.868	30.56229	0.33333333	0.05441478	0.08333333	Rem	Rem
20	2	1989.952	30.64562	0.41666667	0.13774812	0.08333333	Rem	Rem
21	2	1990.035	30.72895	0.50000000	0.22108145	0.08333333	Rem	Rem

```
tail( subset( sLc, lex.id==2 ),[1:8], 3 )
```

	lex.id	per	age	tfi	tfr	lex.dur	lex.Cst	lex.Xst
92	2	1995.952	36.64562	6.416667	6.137748	0.08333333	Rem	Rem
93	2	1996.035	36.72895	6.500000	6.221081	0.08333333	Rem	Rem
94	2	1996.118	36.81229	6.583333	6.304415	0.01728268	Rem	ESRD(Rem)

```
( fl <- levels(Lc)[3:4] )
```

```
[1] "ESRD"      "ESRD(Rem)"
```

```
mp <- glm( lex.Xst %in% fl ~ ns( tfi, df=4 ) +
           sex + I((age-tfi-40)/10) + (lex.Cst=="Rem"),
           offset = log(lex.dur/100),
           family = poisson,
           data = sLc )
# the timescale tfr must be given some value for time before Rem
sLc$tfr <- pmax( 0, sLc$tfr, na.rm=TRUE )
head( subset( sLc, lex.id==2 ),[1:8], 8 )
```

	lex.id	per	age	tfi	tfr	lex.dur	lex.Cst	lex.Xst
14	2	1989.535	30.22895	0.00000000	0.00000000	0.08333333	NRA	NRA
15	2	1989.618	30.31229	0.08333333	0.00000000	0.08333333	NRA	NRA
16	2	1989.702	30.39562	0.16666667	0.00000000	0.08333333	NRA	NRA
17	2	1989.785	30.47895	0.25000000	0.00000000	0.02891855	NRA	Rem
18	2	1989.814	30.50787	0.27891855	0.00000000	0.05441478	Rem	Rem
19	2	1989.868	30.56229	0.33333333	0.05441478	0.08333333	Rem	Rem
20	2	1989.952	30.64562	0.41666667	0.13774812	0.08333333	Rem	Rem
21	2	1990.035	30.72895	0.50000000	0.22108145	0.08333333	Rem	Rem

```
mr <- glm( lex.Xst=="Rem" ~ ns( tfi, df=4 ) + sex,
           offset = log(lex.dur),
           family = poisson,
           data = subset( sLc, lex.Cst=="NRA" ) )
ci.exp( mr, pval=TRUE )
```

	exp(Est.)	2.5%	97.5%	P
(Intercept)	0.03606128	0.011013035	0.1180797	4.016649e-08
ns(tfi, df = 4)1	0.43778959	0.094970457	2.0180984	2.894125e-01
ns(tfi, df = 4)2	1.15591640	0.112100187	11.9191838	9.031269e-01
ns(tfi, df = 4)3	0.57520635	0.017327786	19.0943229	7.569600e-01
ns(tfi, df = 4)4	0.69162506	0.003446815	138.7788899	8.915761e-01
sexF	2.63407462	1.261956986	5.4980868	9.889849e-03

```

inL <- subset( sLc, select=1:10 )[NULL,]
str( inL )

Classes 'Lexis' and 'data.frame':      0 obs. of  10 variables:
 $ lex.id : int
 $ per    : num
 $ age    : num
 $ tfi    : num
 $ tfr    : num
 $ lex.dur: num
 $ lex.Cst: Factor w/ 4 levels "NRA","Rem","ESRD",...
 $ lex.Xst: Factor w/ 4 levels "NRA","Rem","ESRD",...
 $ id     : num
 $ sex    : Factor w/ 2 levels "M","F":
- attr(*, "breaks")=List of 4
..$ per: NULL
..$ age: NULL
..$ tfi: num  0 0.0833 0.1667 0.25 0.3333 ...
..$ tfr: NULL
- attr(*, "time.scales")= chr  "per" "age" "tfi" "tfr"
- attr(*, "time.since")= chr  "" "" "" "Rem"

timeScales(inL)

[1] "per" "age" "tfi" "tfr"

inL[1,"lex.id"] <- 1
inL[1,"per"] <- 2000
inL[1,"age"] <- 50
inL[1,"tfi"] <- 0
inL[1,"lex.Cst"] <- "NRA"
inL[1,"lex.Xst"] <- NA
inL[1,"lex.dur"] <- NA
inL[1,"sex"] <- "M"
inL

lex.id per age tfi tfr lex.dur lex.Cst lex.Xst id sex
1      1 2000 50  0  NA      NA      NRA  <NA> NA  M

Tr <- list( "NRA" = list( "Rem" = mr,
                        "ESRD" = mp ),
           "Rem" = list( "ESRD(Rem)" = mp ) )
( iL <- simLexis( Tr, inL, N=10 ) )

lex.id per age tfi tfr lex.dur lex.Cst lex.Xst id sex cens
1      1 2000.000 50.0000 0.0000000 NA 4.4229390 NRA ESRD NA M 2020
2      2 2000.000 50.0000 0.0000000 NA 0.2273966 NRA Rem NA M 2020
3      2 2000.227 50.2274 0.2273966 0 6.9166222 Rem ESRD(Rem) NA M 2020
4      3 2000.000 50.0000 0.0000000 NA 8.2789058 NRA ESRD NA M 2020
5      4 2000.000 50.0000 0.0000000 NA 8.9320274 NRA ESRD NA M 2020
6      5 2000.000 50.0000 0.0000000 NA 17.8870698 NRA ESRD NA M 2020
7      6 2000.000 50.0000 0.0000000 NA 4.2673109 NRA ESRD NA M 2020
8      7 2000.000 50.0000 0.0000000 NA 14.5124058 NRA ESRD NA M 2020
9      8 2000.000 50.0000 0.0000000 NA 5.4584126 NRA ESRD NA M 2020
10     9 2000.000 50.0000 0.0000000 NA 7.8017705 NRA ESRD NA M 2020
11    10 2000.000 50.0000 0.0000000 NA 17.0273563 NRA ESRD NA M 2020

summary( iL )

```

```

Transitions:
  To
From  NRA Rem ESRD ESRD(Rem) Records: Events: Risk time: Persons:
  NRA  0  1  9         0         10      10      88.82      10
  Rem  0  0  0         1          1       1       6.92       1
  Sum  0  1  9         1         11      11      95.73      10

```

```

system.time(
  sM <- simLexis( Tr, inL, N=10000, t.range=25, n.int=251 ) )

```

```

  user system elapsed
24.655  0.456  25.130

```

```
summary( sM )
```

```

Transitions:
  To
From  NRA Rem ESRD ESRD(Rem) Records: Events: Risk time: Persons:
  NRA  1 1438 8561         0      10000     9999  56556.84  10000
  Rem  0  148  0        1290     1438     1290  15638.24  1438
  Sum  1 1586 8561     1290    11438    11289  72195.08  10000

```

```

nSt <- nState( sM, at=seq(0,24,0.1), from=50, time.scale="age" )
head( nSt )

```

```

      State
when   NRA  Rem ESRD ESRD(Rem)
  50  10000   0   0         0
  50.1 9928  41  31         0
  50.2 9858  81  61         0
  50.3 9787 116  97         0
  50.4 9727 144 129         0
  50.5 9659 182 159         0

```

```

pp <- pState( nSt, perm=c(1,2,4,3) )
head( pp )

```

```

      State
when   NRA  Rem ESRD(Rem) ESRD
  50  1.0000 1.0000  1.0000  1
  50.1 0.9928 0.9969  0.9969  1
  50.2 0.9858 0.9939  0.9939  1
  50.3 0.9787 0.9903  0.9903  1
  50.4 0.9727 0.9871  0.9871  1
  50.5 0.9659 0.9841  0.9841  1

```

```
tail( pp )
```

```

      State
when   NRA  Rem ESRD(Rem) ESRD
  73.5 2e-04 0.0207  0.144  1
  73.6 2e-04 0.0204  0.144  1
  73.7 2e-04 0.0202  0.144  1
  73.8 2e-04 0.0199  0.144  1
  73.9 2e-04 0.0196  0.144  1
  74   2e-04 0.0191  0.144  1

```

```

plot( pp )
# Two colors and the corresponding pale ones for the dead states
clr <- c("limegreen","orange")
col2rgb(clr)

```

```

      [,1] [,2]
red      50 255
green    205 165
blue     50   0

c14 <- cbind(col2rgb(c1r),col2rgb(c1r)/2+255/2)[,c(1,2,4,3)]
c14 <- rgb( t(c14), max=255 )
# Nicer plot
plot( pp, col=c14, xlab="Age" )
lines( as.numeric(rownames(pp)), pp[,2], lwd=2 )

```

3.2 Time-splitting, time-scales and SMR: Diabetes in Denmark

This exercise is using data from the National Danish Diabetes register. There is a random sample of 10,000 records from this in the `Epi` package. Actually there are two data sets, we shall use the one with only cases of diabetes diagnosed after 1995, see the help page for `DMlate`.

This is of interest because it is only for these where the data of diagnosis is certain, and hence for whom we can compute the duration of diabetes during follow-up.

The exercise is about assessing how mortality depends age, calendar time and duration of diabetes. And how to understand and compute SMR, and assess how it depends on these factors as well.

1. First, we load the `Epi` package and the dataset, and take a look at it:

```

> options( width=90 )
> library( Epi )
> data( DMlate )
> str( DMlate )

'data.frame':      10000 obs. of  7 variables:
 $ sex   : Factor w/ 2 levels "M","F": 2 1 2 2 1 2 1 1 2 1 ...
 $ dobth: num  1940 1939 1918 1965 1933 ...
 $ dodm  : num  1999 2003 2005 2009 2009 ...
 $ dodth: num  NA NA NA NA NA ...
 $ dooad: num  NA 2007 NA NA NA ...
 $ doins: num  NA NA NA NA NA NA NA NA NA NA ...
 $ dox   : num  2010 2010 2010 2010 2010 ...

> head( DMlate )

      sex  dobth  dodm  dodth  dooad doins  dox
50185  F 1940.256 1998.917    NA     NA    NA 2009.997
307563  M 1939.218 2003.309    NA 2007.446    NA 2009.997
294104  F 1918.301 2004.552    NA     NA    NA 2009.997
336439  F 1965.225 2009.261    NA     NA    NA 2009.997
245651  M 1932.877 2008.653    NA     NA    NA 2009.997
216824  F 1927.870 2007.886 2009.923    NA    NA 2009.923

> summary( DMlate )

```

```

sex          dobth          dodm          dodth          dooad          doins
M:5185      Min.    :1898      Min.    :1995      Min.    :1995      Min.    :1995      Min.    :1995
F:4815      1st Qu.:1930      1st Qu.:2000      1st Qu.:2002      1st Qu.:2001      1st Qu.:2001
              Median :1941      Median :2004      Median :2005      Median :2004      Median :2005
              Mean   :1942      Mean   :2003      Mean   :2005      Mean   :2004      Mean   :2004
              3rd Qu.:1951      3rd Qu.:2007      3rd Qu.:2008      3rd Qu.:2007      3rd Qu.:2007
              Max.   :2008      Max.   :2010      Max.   :2010      Max.   :2010      Max.   :2010
              NA's   :2008      NA's   :2010      NA's   :2010      NA's   :4503      NA's   :8209

dox
Min.    :1995
1st Qu.:2010
Median :2010
Mean   :2009
3rd Qu.:2010
Max.   :2010

```

- We then set up the dataset as a `Lexis` object with age, calendar time and duration of diabetes as timescales, and date of death as event.

In the dataset we have a date of exit `dox` which is either the day of censoring or the date of death:

```

> with( DMLate, table( dead=!is.na(dodth),
+                      same=(dodth==dox), exclude=NULL ) )

      same
dead  TRUE <NA>
FALSE  0 7497
TRUE   2503  0
<NA>   0    0

```

So we can set up the `Lexis` object by specifying the timescales and the exit status via `!is.na(dodth)`:

```

> LL <- Lexis( entry = list( A = dodm-dobth,
+                           P = dodm,
+                           dur = 0 ),
+             exit = list( P = dox ),
+             exit.status = factor( !is.na(dodth),
+                                   labels=c("Alive","Dead") ),
+             data = DMLate )

```

NOTE: `entry.status` has been set to "Alive" for all.

The 4 persons are persons that have identical date of diabetes and date of death.

We can get an overview of the data by using the `summary` function on the object:

```

> summary( LL )

Transitions:
  To
From  Alive Dead  Records:  Events: Risk time:  Persons:
  Alive 7497 2499     9996     2499   54273.27     9996

> head( LL )

```



```

      A      P dur      lex.dur lex.Cst lex.Xst lex.id sex      dobth      dodm
50185  58.66119 1998.917    0 11.0800821   Alive   Alive     1    F 1940.256 1998.917
307563 64.09035 2003.309    0  6.6885695   Alive   Alive     2    M 1939.218 2003.309
294104 86.25051 2004.552    0  5.4455852   Alive   Alive     3    F 1918.301 2004.552
336439 44.03559 2009.261    0  0.7364819   Alive   Alive     4    F 1965.225 2009.261
245651 75.77550 2008.653    0  1.3442847   Alive   Alive     5    M 1932.877 2008.653
216824 80.01643 2007.886    0  2.0369610   Alive   Dead      6    F 1927.870 2007.886
      dodth      dooad doins      dox
50185      NA      NA      NA 2009.997
307563      NA 2007.446      NA 2009.997
294104      NA      NA      NA 2009.997
336439      NA      NA      NA 2009.997
245651      NA      NA      NA 2009.997
216824 2009.923      NA      NA 2009.923

```

3. A very crude picture of the mortality by sex can be obtained by the `stat.table` function:

```

> stat.table( sex,
+             list( D=sum( lex.Xst=="Dead" ),
+                 Y=sum( lex.dur ),
+                 rate=ratio( lex.Xst=="Dead", lex.dur, 1000 ) ),
+             data=LL )

```

```

-----
sex      D      Y      rate
-----
M      1343.00 27614.21  48.63
F      1156.00 26659.05  43.36
-----

```

So not surprising, we see that men have a higher mortality than women.

4. We now want to assess how mortality depends on age, calendar time and duration. In principle we could split the follow-up along all three time scales, but in practice it would be sufficient to split it along one of the time-scales and then just use the value of each of the time-scales at the left endpoint of the intervals.

We note that the total follow-up time was some 54,000 person-years, so if we split the follow-up in 6-month intervals we get a bit more than 110,000 records:

```

> SL <- splitLexis( LL, breaks=seq(0,125,1/2), time.scale="A" )
> summary( SL )

```

Transitions:

```

      To
From   Alive Dead Records: Events: Risk time: Persons:
  Alive 115974 2499   118473    2499   54273.27    9996

```

```

> summary( LL )

```

Transitions:

```

      To
From   Alive Dead Records: Events: Risk time: Persons:
  Alive  7497 2499    9996    2499   54273.27    9996

```

We see that the number of records have increased, but the number of persons, events and person-years is still the same as in LL

- We now use this dataset to estimate models with age-specific mortality curves for men and women separately, using natural splines (the function `ns` from the `splines` package).

```
> library( splines )
> r.m <- glm( (lex.Xst=="Dead") ~ ns( A, df=10 ),
+           offset = log( lex.dur ),
+           family = poisson,
+           data = subset( SL, sex=="M" ) )
> r.f <- update( r.m, data = subset( SL, sex=="F" ) )
```

Here we are modeling the follow-up (events (`(lex.Xst=="Dead")`) and person-years (`(lex.dur)`) as a non-linear function of age — represented by the spline function `ns`.

- From these objects we could get the estimated log-rates by using `predict`, by supplying a data frame of values for the variables corresponding to the predictor variables in the model.

The default `predict.glm` function is a bit clunky as it gives the prediction and the standard errors of these in two different elements of a list, so in `Epi` there is a wrapper function `ci.pred` that uses this and computes predicted rates and confidence limits for these.

Note that `lex.dur` is a covariate too; by putting this to 1000 we get the rates in units of deaths per 1000 PY:

```
> nd <- data.frame( A = seq(10,90,0.5),
+                 lex.dur = 1000)
> p.m <- ci.pred( r.m, newdata = nd )
> p.f <- ci.pred( r.f, newdata = nd )
> str( p.m )

 num [1:161, 1:3] 1.33 1.34 1.34 1.34 1.35 ...
- attr(*, "dimnames")=List of 2
 ..$ : chr [1:161] "1" "2" "3" "4" ...
 ..$ : chr [1:3] "Estimate" "2.5%" "97.5%"
```

- We can then plot the predicted rates for men and women together using `matplot`:

```
> matplot( nd$A, cbind(p.m,p.f),
+         type="l", col=rep(c("blue","red"),each=3), lwd=c(3,1,1), lty=1,
+         log="y", xlab="Age", ylab="Mortality of DM ptt per 1000 PY")
```

Period and duration effects

- We model the mortality rates among diabetes patients also including current date and duration of diabetes. However, we shall not just use the positioning of knots for the splines as provided by `ns`, because this is based on the allocating knots so that the number of observations (lines in the dataset), is the same between knots.

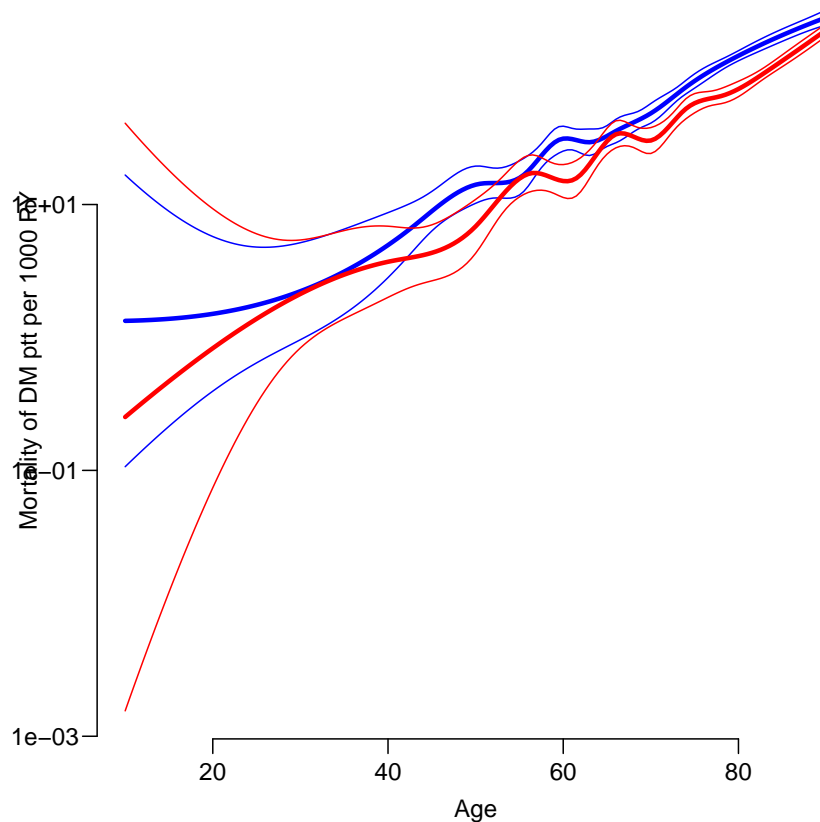


Figure 3.4: Age-specific mortality rates for Danish diabetes patients as estimated from a model with only age. Blue: men, red: women.

However the information in a follow-up study is in the number of events, so it would be better to allocate knots so that number of events were the same between knots.

We use the so-called *natural splines* that are linear beyond the boundary knots, and hence we take the 5th and 95th percentile of deaths as the boundary knots for age (A) and calendar time (P) but for duration where we actually have follow-up from time 0 on the timescale we use 0 as the first knot.

```
> ( kn.A <- with( subset( SL, lex.Xst=="Dead" ),
+               quantile( A+lex.dur, probs=seq(5,95,20)/100 ) ) )
      5%      25%      45%      65%      85%
56.02519 69.06092 76.29021 81.42094 87.66598

> ( kn.P <- with( subset( SL, lex.Xst=="Dead" ),
+               quantile( P+lex.dur, probs=seq(5,95,20)/100 ) ) )
      5%      25%      45%      65%      85%
1998.117 2002.120 2004.694 2006.826 2008.761

> ( kn.dur <- c(0,with( subset( SL, lex.Xst=="Dead" ),
+                    quantile( dur+lex.dur, probs=seq(10,90,20)/100 ) ) ) )
      10%      30%      50%      70%      90%
0.0000000 0.3055441 1.5961670 3.4250513 5.6629706 9.1723477
```

9. With these we can now model mortality rates (separately for men and women), as functions of age, calendar time and duration:

```
> Mm <- glm( (lex.Xst=="Dead") ~ Ns( A, kn=kn.A ) +
+           Ns( P, kn=kn.P ) +
+           Ns( dur, kn=kn.dur ),
+           offset = log( lex.dur ),
+           family = poisson,
+           data = subset( SL, sex=="M" ) )
> summary( Mm )
```

Call:

```
glm(formula = (lex.Xst == "Dead") ~ Ns(A, kn = kn.A) + Ns(P,
kn = kn.P) + Ns(dur, kn = kn.dur), family = poisson, data = subset(SL,
sex == "M"), offset = log(lex.dur))
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-0.8367	-0.2308	-0.1595	-0.1115	4.4965

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-3.16121	0.10390	-30.426	< 2e-16
Ns(A, kn = kn.A)1	1.52180	0.11720	12.985	< 2e-16
Ns(A, kn = kn.A)2	1.89400	0.09175	20.643	< 2e-16
Ns(A, kn = kn.A)3	2.98735	0.12279	24.328	< 2e-16
Ns(A, kn = kn.A)4	2.05374	0.07824	26.250	< 2e-16
Ns(P, kn = kn.P)1	-0.19507	0.13352	-1.461	0.144009
Ns(P, kn = kn.P)2	-0.29731	0.10694	-2.780	0.005435
Ns(P, kn = kn.P)3	-0.43455	0.17152	-2.533	0.011293
Ns(P, kn = kn.P)4	-0.29586	0.08978	-3.295	0.000982
Ns(dur, kn = kn.dur)1	-0.76626	0.15497	-4.945	7.63e-07
Ns(dur, kn = kn.dur)2	-0.63208	0.15325	-4.124	3.72e-05
Ns(dur, kn = kn.dur)3	-0.46099	0.12080	-3.816	0.000136
Ns(dur, kn = kn.dur)4	-1.29240	0.21518	-6.006	1.90e-09
Ns(dur, kn = kn.dur)5	-0.12241	0.09654	-1.268	0.204796

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 12999 on 60346 degrees of freedom
Residual deviance: 11727 on 60333 degrees of freedom
AIC: 14441

Number of Fisher Scoring iterations: 7

```
> Mf <- update( Mm, data = subset( SL, sex=="F" ) )
> round( cbind( ci.exp(Mm), ci.exp(Mf) ), 3 )
```

	exp(Est.)	2.5%	97.5%	exp(Est.)	2.5%	97.5%
(Intercept)	0.042	0.035	0.052	0.025	0.019	0.032
Ns(A, kn = kn.A)1	4.580	3.640	5.763	4.247	3.199	5.639
Ns(A, kn = kn.A)2	6.646	5.552	7.955	5.079	4.180	6.170
Ns(A, kn = kn.A)3	19.833	15.591	25.230	20.611	15.288	27.788
Ns(A, kn = kn.A)4	7.797	6.689	9.089	7.806	6.572	9.272
Ns(P, kn = kn.P)1	0.823	0.633	1.069	0.908	0.686	1.202
Ns(P, kn = kn.P)2	0.743	0.602	0.916	0.730	0.579	0.921
Ns(P, kn = kn.P)3	0.648	0.463	0.906	0.768	0.524	1.125
Ns(P, kn = kn.P)4	0.744	0.624	0.887	0.668	0.551	0.809
Ns(dur, kn = kn.dur)1	0.465	0.343	0.630	0.541	0.387	0.756

Ns(dur, kn = kn.dur)2	0.531	0.394	0.718	0.472	0.338	0.658
Ns(dur, kn = kn.dur)3	0.631	0.498	0.799	0.871	0.678	1.118
Ns(dur, kn = kn.dur)4	0.275	0.180	0.419	0.398	0.248	0.641
Ns(dur, kn = kn.dur)5	0.885	0.732	1.069	0.982	0.800	1.206

It is not possible to attach any meaning to the single parameters from the model, so we shall look at the estimated non-linear effects of each of the variables.

10. These models fit substantially better than the model with only age as we can see from this comparison:

```
> anova( Mm, r.m, test="Chisq" )
```

Analysis of Deviance Table

```
Model 1: (lex.Xst == "Dead") ~ Ns(A, kn = kn.A) + Ns(P, kn = kn.P) + Ns(dur,
kn = kn.dur)
```

```
Model 2: (lex.Xst == "Dead") ~ ns(A, df = 10)
```

	Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
1	60333	11727			
2	60336	11808	-3	-81.122	< 2.2e-16

```
> anova( Mf, r.f, test="Chisq" )
```

Analysis of Deviance Table

```
Model 1: (lex.Xst == "Dead") ~ Ns(A, kn = kn.A) + Ns(P, kn = kn.P) + Ns(dur,
kn = kn.dur)
```

```
Model 2: (lex.Xst == "Dead") ~ ns(A, df = 10)
```

	Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
1	58112	10203			
2	58115	10258	-3	-54.302	9.675e-12

The models are not formally nested since the location of the age-knots is different, so from a formal point of view these test are not valid, but it is clear that the more extensive modeling provides a much better description of the rates.

11. We can inspect the shape of the estimated effects (and their relative size) using the `Termplot` function in the `Epi` package.

However in order for this to work properly we need a model specification where *all* of the prediction is part of a term, essentially including the intercept in one of the terms — notably age. Moreover, the age-specific rates must refer to a specific period and diabetes duration.

This is done by using the `intercept` and `ref` arguments to `Ns`:

```
> mm <- glm( (lex.Xst=="Dead") ~ Ns( A, kn=kn.A , intercept=TRUE ) - 1 +
+                               Ns( P, kn=kn.P , ref=2000 ) +
+                               Ns( dur, kn=kn.dur, ref=5 ),
+       offset = log( lex.dur/100 ),
+       family = poisson,
+       data = subset( SL, sex=="M" ) )
> mf <- update( mm, data = subset( SL, sex=="F" ) )
```

We can check that it actually is the same model, by using the deviances from the two models fitted.

```
> c( deviance(Mm), deviance(mm) )
[1] 11726.61 11726.61
```

12. We use `Termplot`, which is a wrapper for `termplot`. `Termplot` gives plots on the rate / resp RR scale, so that we can actually make sense of the plots.

```
> Termplot( mm )
```

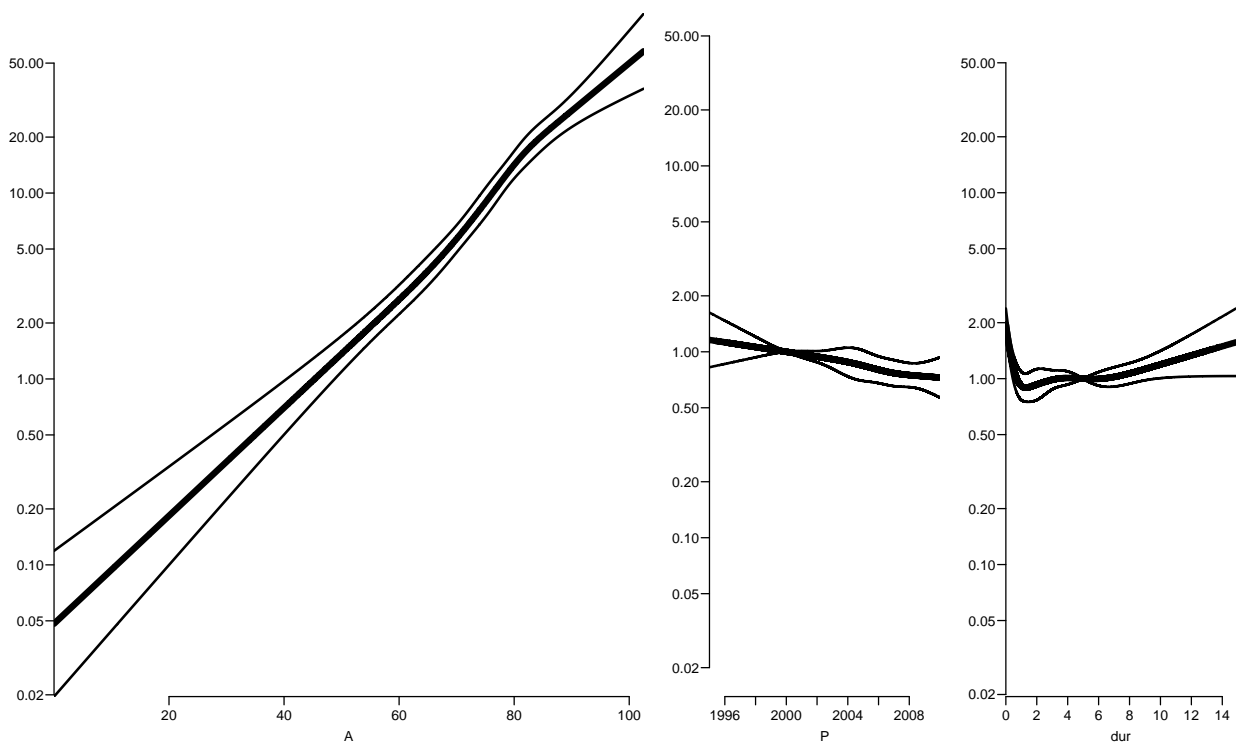


Figure 3.5: Age, period and duration terms for mortality among Danish male diabetes patients. The age effect is age-specific rates for persons with 5 years of diabetes duration in the year 2000.

```
> Termplot( mf )
```

13. Since the fitted model has three time-scales: current age, current date and current duration of diabetes, so the effects that we see in the `Termplot` are not really interpretable; they are (as in any kind of multiple regressions) to be interpreted as “all else equal” which they are not; the three time scales advance simultaneously at the same pace.

The reporting would therefore more naturally be *only* on one time scale, showing the mortality for persons diagnosed in different ages in a given year.

This is most easily done using the `ci.pred` function with the `newdata=` argument. So a person diagnosed in age 50 in 1995 will have a mortality measured in cases per 1000 PY as:

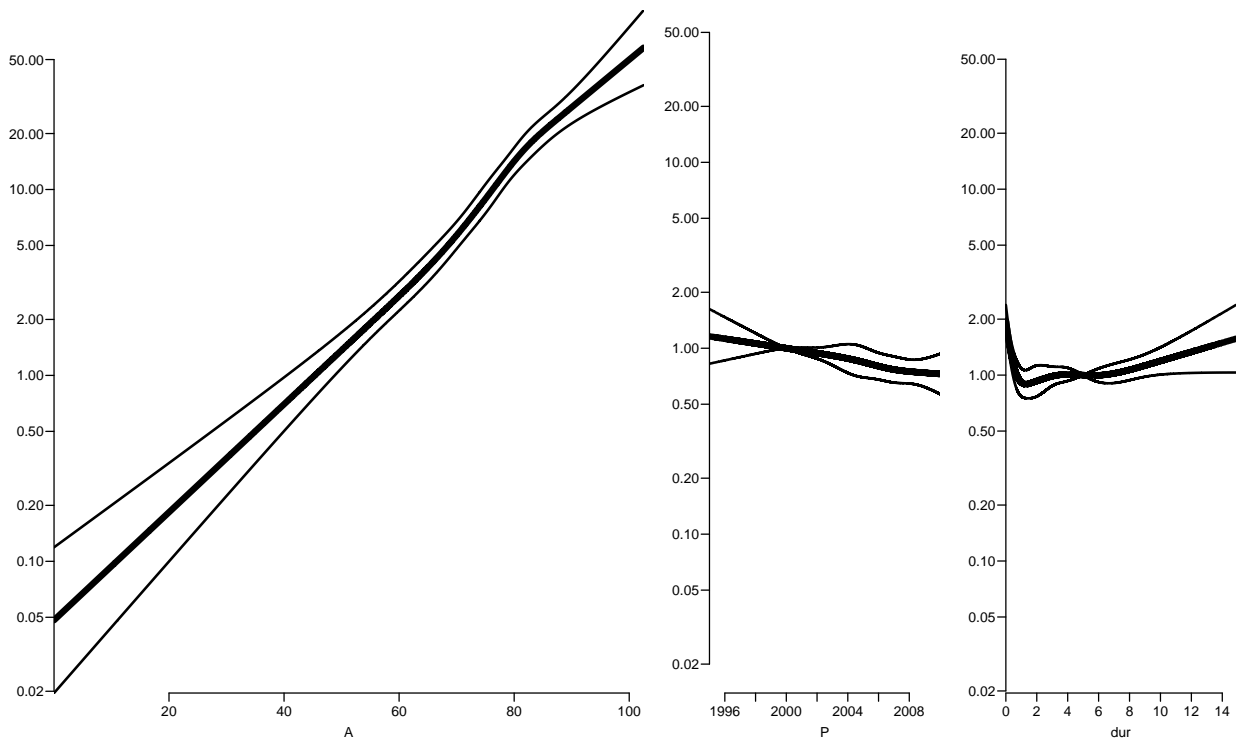


Figure 3.6: Age, period and duration terms for mortality among Danish female diabetes patients. The age effect is age-specific rates for persons with 5 years of diabetes duration in the year 2000.

```
> pts <- seq(0,20,2)
> nd <- data.frame( A= 50+pts,
+                   P=1995+pts,
+                   dur= pts,
+                   lex.dur=1000 )
> cbind( nd$A, ci.pred( mm, newdata=nd ) )
```

	Estimate	2.5%	97.5%	
1	50	31.02982	21.72823	44.31332
2	52	15.85329	12.17126	20.64919
3	54	18.54048	15.12524	22.72687
4	56	19.75660	16.17241	24.13514
5	58	22.71326	18.97210	27.19216
6	60	26.79926	22.01942	32.61667
7	62	31.43090	24.29985	40.65463
8	64	38.99649	27.98085	54.34883
9	66	49.17746	29.96508	80.70802
10	68	62.48983	31.13183	125.43363
11	70	80.12938	32.18765	199.47763

Since there is no duration beyond 18 years in the dataset we only make predictions for 20 years of duration, and do it for persons diagnosed in 1995 and 2005 — the latter is quite dubious too because we are extrapolating calendar time trends way beyond data.

We form matrices of predictions, that we will plot in the same frame:

```

> mpr <- fpr <- NULL
> pts <- seq(0,20,0.1)
> for( ip in c(1995,2005) )
+ for( ia in c(50,60,70) )
+ {
+ nd <- data.frame( A=ia+pts,
+                  P=ip+pts,
+                  dur= pts,
+                  lex.dur=1000 )
+ mpr <- cbind( mpr, ci.pred( mm, nd) )
+ fpr <- cbind( fpr, ci.pred( mf, nd) )
+ }
> str( fpr )

num [1:201, 1:18] 14.5 13.1 12 11 10.3 ...
- attr(*, "dimnames")=List of 2
..$ : chr [1:201] "1" "2" "3" "4" ...
..$ : chr [1:18] "Estimate" "2.5%" "97.5%" "Estimate" ...

```

These 18 columns are 9 columns for 1995, and 9 for 2005, each of these chunks are estimate and lower and upper confidence bound for persons diagnosed in ages 50, 60 and 70.

These can now be plotted:

```

> par( mfrow=c(1,2) )
> matplot( cbind(50+pts,60+pts,70+pts)[,rep(1:3,2,each=3)],
+          cbind( mpr[,1:9], fpr[,1:9] ), ylim=c(5,500),
+          log="y", xlab="Age", ylab="Mortality, diagnosed 1995",
+          type="l", lwd=c(4,1,1), lty=1,
+          col=rep(c("blue","red"),each=9) )
> matplot( cbind(50+pts,60+pts,70+pts)[,rep(1:3,2,each=3)],
+          cbind( mpr[,1:9+9], fpr[,1:9+9] ), ylim=c(5,500),
+          log="y", xlab="Age", ylab="Mortality, diagnosed 2005",
+          type="l", lwd=c(4,1,1), lty=1,
+          col=rep(c("blue","red"),each=9) )

```

3.2.1 SMR

There are two ways to make the comparison of the diabetes mortality to the population mortality; one is to amend the diabetes patient dataset with the population mortality dataset, the other (classical) one is to include the population mortality rates as a fixed variable in the calculations.

The latter requires that each analytic unit in the diabetes patient dataset is amended with a variable with the population mortality rate for the corresponding sex, age and calendar time.

This can be achieved in two ways: Either we just use the current split of follow-up time and allocate the population mortality rates for some suitably chosen (mid-)point of the follow-up in each, or we make a second split by date, so that follow-up in the diabetes patients is in the same classification of age and data as the population mortality table.

- Using the former approach we shall include as an extra variable the population mortality as available from the data set `M.dk`.

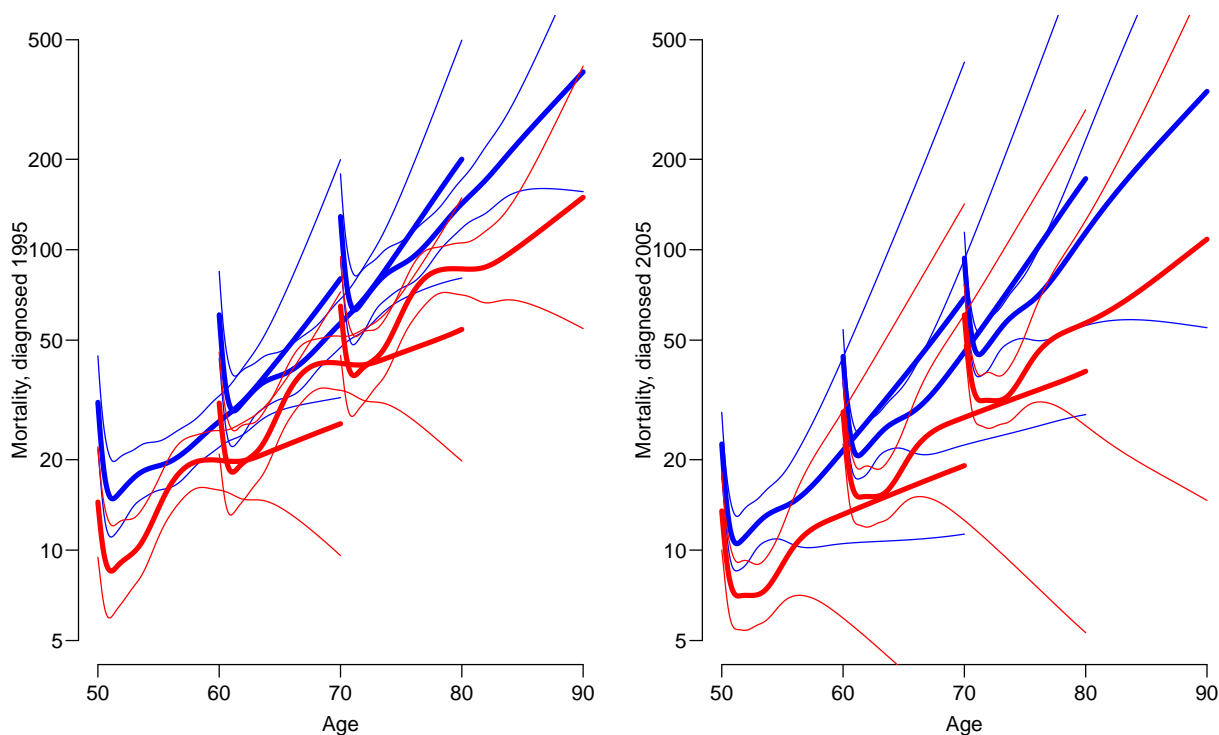


Figure 3.7: *Mortality rates for diabetes patients diagnosed 1995 and 2005 in ages 50, 60 and 70. Men blue, women red.*

First create the variables in the diabetes dataset that we need for matching with the age and period classification of the population mortality data, that is age, date (and sex) at the midpoint of each of the intervals (or rather at a point 3 months after the left endpoint of the interval — recall we split the follow-up in 6 month intervals).

We need to have variables of the same type when we merge, so we must transform the sex variable in `M.dk` to a factor, and must for each follow-up interval in the `SL` data have an age and a period variable that can be used in merging with the population data.

```
> str( SL )
```

```
Classes 'Lexis' and 'data.frame':      118473 obs. of  14 variables:
 $ lex.id : int  1 1 1 1 1 1 1 1 1 1 ...
 $ A      : num  58.7 59 59.5 60 60.5 ...
 $ P      : num  1999 1999 2000 2000 2001 ...
 $ dur    : num  0 0.339 0.839 1.339 1.839 ...
 $ lex.dur: num  0.339 0.5 0.5 0.5 0.5 ...
 $ lex.Cst: Factor w/ 2 levels "Alive","Dead": 1 1 1 1 1 1 1 1 1 1 ...
 $ lex.Xst: Factor w/ 2 levels "Alive","Dead": 1 1 1 1 1 1 1 1 1 1 ...
 $ sex    : Factor w/ 2 levels "M","F": 2 2 2 2 2 2 2 2 2 2 ...
 $ dobth  : num  1940 1940 1940 1940 1940 ...
 $ dodm   : num  1999 1999 1999 1999 1999 ...
 $ dodth  : num  NA NA NA NA NA NA NA NA NA NA ...
 $ dooad  : num  NA NA NA NA NA NA NA NA NA NA ...
 $ doins  : num  NA NA NA NA NA NA NA NA NA NA ...
 $ dox    : num  2010 2010 2010 2010 2010 ...
 - attr(*, "breaks")=List of 3
```

```

..$ A : num 0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 ...
..$ P : NULL
..$ dur: NULL
- attr(*, "time.scales")= chr "A" "P" "dur"
- attr(*, "time.since")= chr "" "" ""

> SL$Am <- floor( SL$A+0.25 )
> SL$Pm <- floor( SL$P+0.25 )
> data( M.dk )
> str( M.dk )

'data.frame':      7800 obs. of  6 variables:
 $ A : num 0 0 0 0 0 0 0 0 0 0 ...
 $ sex : num 1 2 1 2 1 2 1 2 1 2 ...
 $ P : num 1974 1974 1975 1975 1976 ...
 $ D : num 459 303 435 311 405 258 332 205 312 233 ...
 $ Y : num 35963 34383 36099 34652 34965 ...
 $ rate: num 12.76 8.81 12.05 8.97 11.58 ...
- attr(*, "Contents")= chr "Number of deaths and risk time in Denmark"

> M.dk <- transform( M.dk, Am = A,
+                   Pm = P,
+                   sex = factor( sex, labels=c("M","F") ) )
> str( M.dk )

'data.frame':      7800 obs. of  8 variables:
 $ A : num 0 0 0 0 0 0 0 0 0 0 ...
 $ sex : Factor w/ 2 levels "M","F": 1 2 1 2 1 2 1 2 1 2 ...
 $ P : num 1974 1974 1975 1975 1976 ...
 $ D : num 459 303 435 311 405 258 332 205 312 233 ...
 $ Y : num 35963 34383 36099 34652 34965 ...
 $ rate: num 12.76 8.81 12.05 8.97 11.58 ...
 $ Am : num 0 0 0 0 0 0 0 0 0 0 ...
 $ Pm : num 1974 1974 1975 1975 1976 ...

```

We then match the rates from `M.dk` into `SL` — `sex`, `Am` and `Pm` are the common variables, and therefore the match is on these variables:

```

> SLr <- merge( SL, M.dk[,c("sex","Am","Pm","rate")] )
> dim( SL )

[1] 118473    16

> dim( SLr )

[1] 118454    17

```

This merge only takes rows that have information from both data sets, hence the slightly fewer rows in `SLr` than in `SL` — there are a few record in `SL` with age and period values that do not exist in the population mortality data.

15. We compute the expected number of deaths as the person-time multiplied by the corresponding population rate recalling that the rate is given in units of deaths per 1000 PY, whereas `lex.dur` is in units of 1 PY:

```
> SLr$E <- SLr$lex.dur * SLr$rate / 1000
> stat.table( sex,
+           list( D = sum(lex.Xst=="Dead"),
+               Y = sum(lex.dur),
+               E = sum(E),
+               SMR = ratio(lex.Xst=="Dead",E) ),
+           data = SLr )
```

sex	D	Y	E	SMR
M	1342.00	27611.40	796.11	1.69
F	1153.00	26654.52	747.77	1.54

```
> stat.table( list( Age = floor(pmax(A,39)/10)*10 ),
+           list( D = sum(lex.Xst=="Dead"),
+               Y = sum(lex.dur),
+               E = sum(E),
+               SMR = ratio(lex.Xst=="Dead",E) ),
+           data = SLr )
```

Age	D	Y	E	SMR
30	11.00	4706.00	3.18	3.45
40	47.00	5776.18	14.48	3.25
50	181.00	10765.19	70.47	2.57
60	432.00	14052.52	216.39	2.00
70	817.00	12225.99	480.11	1.70
80	771.00	5952.59	573.73	1.34
90	236.00	787.46	185.51	1.27

We see that the SMR is slightly higher for women than for men, but also that there is a much larger variation in SMR by age.

16. We can the SMR exactly as mortality rates by including the log expected numbers instead of the log person-years as offset, again using separate models for men and women.

We exclude those records where no deaths in the population occur (that is where the rate is 0) — you could say that this correspond to parts of the data where no follow-up on the population mortality scale is available.

```
> sm <- glm( (lex.Xst=="Dead") ~ Ns( A, kn=kn.A , intercept=TRUE ) - 1 +
+           Ns( P, kn=kn.P , ref=2000 ) +
+           Ns( dur, kn=kn.dur, ref=5 ),
+           offset = log( E ),
+           family = poisson,
+           data = subset( SLr, E>0 & sex=="M" ) )
> sf <- update( mm, data = subset( SLr, E>0 & sex=="F" ) )
```

We can plot the estimates as before for the rates, using `Termplot`. What do the extracted effects represent now?

```
> Termplot( sm )
```

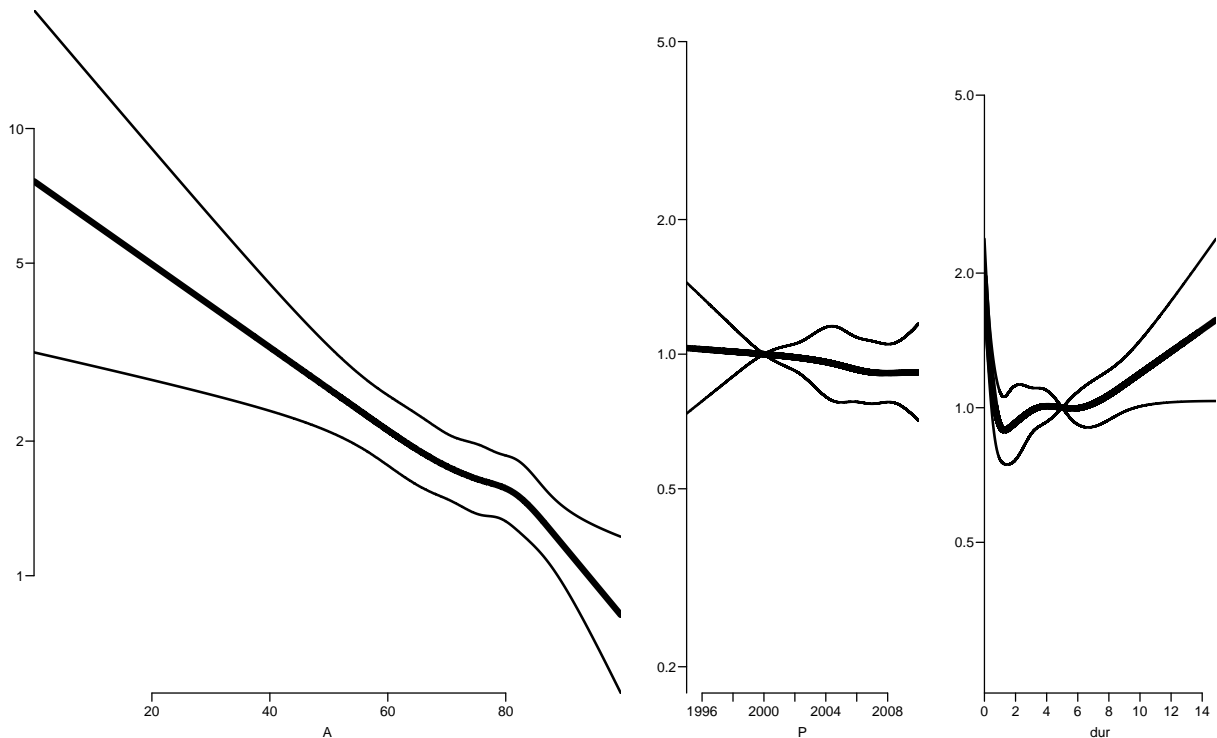


Figure 3.8: Age, period and duration terms for SMR among Danish male diabetes patients. The age effect is age-specific SMR for persons with 5 years of diabetes duration in the year 2000.

```
> Termplot( sf )
```

17. We can check if there are different SMRs between men and women by fitting a joint model and expanding it with (linear) sex-effect(s):

```
> s0 <- glm( (lex.Xst=="Dead") ~ Ns( A, kn=kn.A , intercept=TRUE ) - 1 +
+          Ns( P, kn=kn.P , ref=2000 ) +
+          Ns( dur, kn=kn.dur, ref=5 ),
+          offset = log( E ),
+          family = poisson,
+          data = subset( SLr, E>0 ) )
> s1 <- update( s0, . ~ . + sex )
> sA <- update( s1, . ~ . + sex:A )
> sAP <- update( sA, . ~ . + sex:P )
> sAPd <- update( sAP, . ~ . + sex:dur )
> anova( s0, s1, sA, sAP, sAPd, test="Chisq" )
```

Analysis of Deviance Table

```
Model 1: (lex.Xst == "Dead") ~ Ns(A, kn = kn.A, intercept = TRUE) - 1 +
  Ns(P, kn = kn.P, ref = 2000) + Ns(dur, kn = kn.dur, ref = 5)
Model 2: (lex.Xst == "Dead") ~ Ns(A, kn = kn.A, intercept = TRUE) + Ns(P,
  kn = kn.P, ref = 2000) + Ns(dur, kn = kn.dur, ref = 5) +
  sex - 1
Model 3: (lex.Xst == "Dead") ~ Ns(A, kn = kn.A, intercept = TRUE) + Ns(P,
  kn = kn.P, ref = 2000) + Ns(dur, kn = kn.dur, ref = 5) +
  sex + sex:A - 1
```

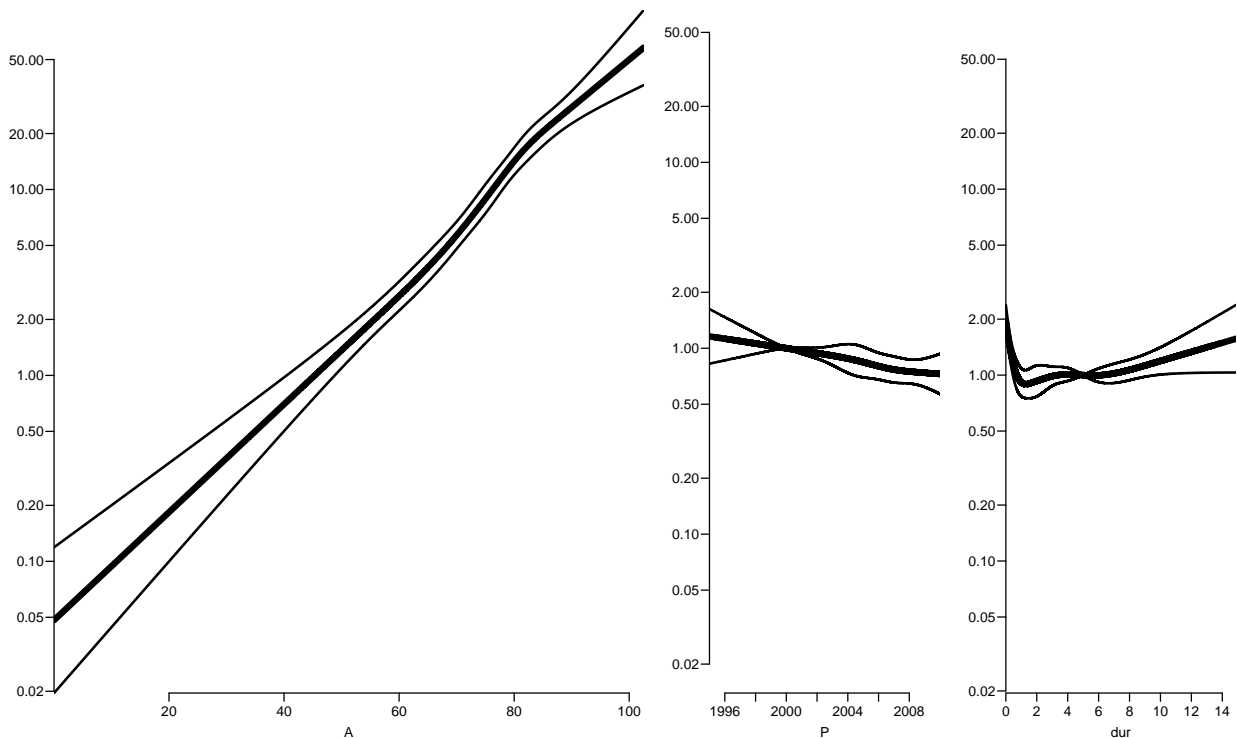


Figure 3.9: Age, period and duration terms for mortality among Danish female diabetes patients. The age effect is age-specific SMR for persons with 5 years of diabetes duration in the year 2000.

```

Model 4: (lex.Xst == "Dead") ~ Ns(A, kn = kn.A, intercept = TRUE) + Ns(P,
  kn = kn.P, ref = 2000) + Ns(dur, kn = kn.dur, ref = 5) +
  sex + sex:A + sex:P - 1
Model 5: (lex.Xst == "Dead") ~ Ns(A, kn = kn.A, intercept = TRUE) + Ns(P,
  kn = kn.P, ref = 2000) + Ns(dur, kn = kn.dur, ref = 5) +
  sex + sex:A + sex:P + sex:dur - 1
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1      118418      21925
2      118417      21925  1  0.00799  0.9288
3      118416      21925  1  0.12948  0.7190
4      118415      21925  1  0.23766  0.6259
5      118414      21924  1  0.45765  0.4987

```

So by this simple check we see there is no really compelling evidence that the SMR differs between men and women.

Of course we might repeat it all by including quadratic effects too:

```

> sA <- update( s1, . ~ . + sex:A + sex:I(A^2) )
> sAP <- update( sA, . ~ . + sex:P + sex:I(P^2) )
> sAPd <- update( sAP, . ~ . + sex:dur + sex:I(dur^2) )
> anova( s0, s1, sA, sAP, sAPd, test="Chisq" )

```

Analysis of Deviance Table

```

Model 1: (lex.Xst == "Dead") ~ Ns(A, kn = kn.A, intercept = TRUE) - 1 +
  Ns(P, kn = kn.P, ref = 2000) + Ns(dur, kn = kn.dur, ref = 5)
Model 2: (lex.Xst == "Dead") ~ Ns(A, kn = kn.A, intercept = TRUE) + Ns(P,

```

```

kn = kn.P, ref = 2000) + Ns(dur, kn = kn.dur, ref = 5) +
sex - 1
Model 3: (lex.Xst == "Dead") ~ Ns(A, kn = kn.A, intercept = TRUE) + Ns(P,
kn = kn.P, ref = 2000) + Ns(dur, kn = kn.dur, ref = 5) +
sex + sex:A + sex:I(A^2) - 1
Model 4: (lex.Xst == "Dead") ~ Ns(A, kn = kn.A, intercept = TRUE) + Ns(P,
kn = kn.P, ref = 2000) + Ns(dur, kn = kn.dur, ref = 5) +
sex + sex:A + sex:I(A^2) + sex:P + sex:I(P^2) - 1
Model 5: (lex.Xst == "Dead") ~ Ns(A, kn = kn.A, intercept = TRUE) + Ns(P,
kn = kn.P, ref = 2000) + Ns(dur, kn = kn.dur, ref = 5) +
sex + sex:A + sex:I(A^2) + sex:P + sex:I(P^2) + sex:dur +
sex:I(dur^2) - 1
Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1 118418 21925
2 118417 21925 1 0.0080 0.9288
3 118414 21923 3 2.4179 0.4903
4 118411 21919 3 3.8787 0.2749
5 118408 21918 3 1.2737 0.7354

```

So there really is no difference, so we can report the SMR between the diabetes patients and the population as sex-independent:

```
> Termplot( s0 )
```

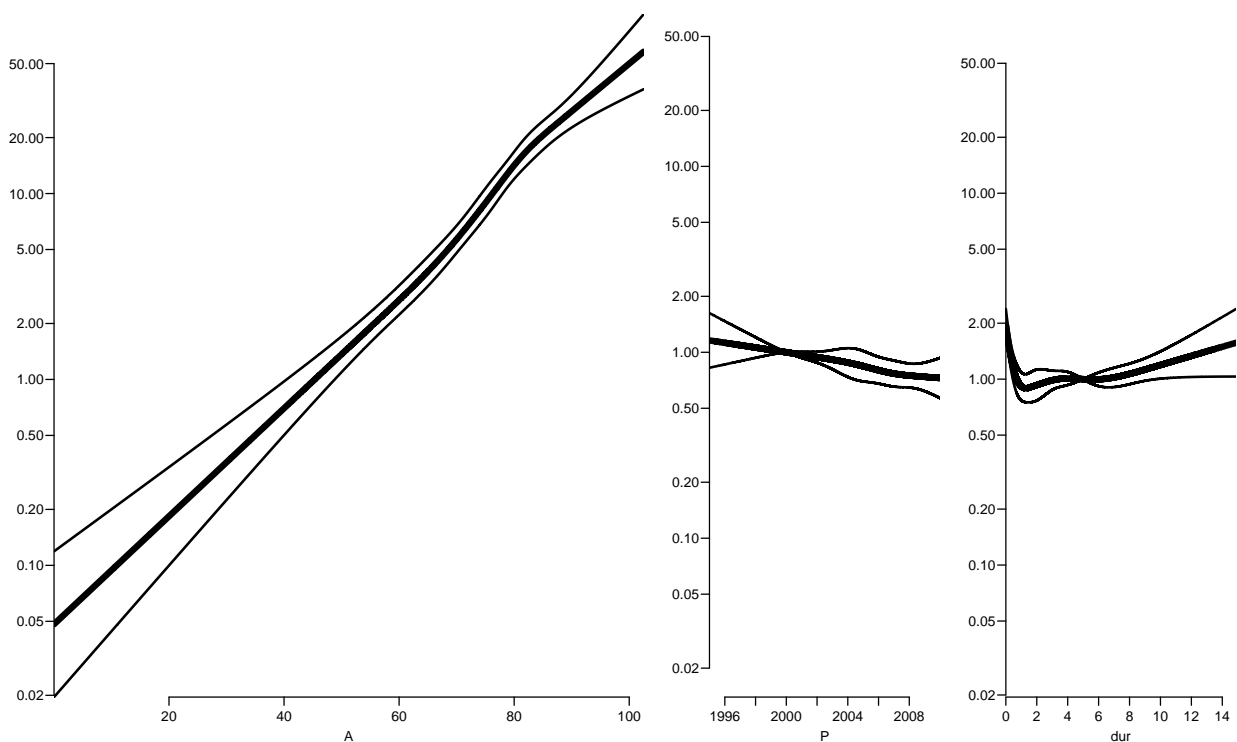


Figure 3.10: Age, period and duration terms for mortality among Danish diabetes patients, male as well as female. The age effect is age-specific SMR for persons with 5 years of diabetes duration in the year 2000.

18. As before, it would be more sensible to show the SMR as a function of age for persons diagnosed with DM at ages 50, 60 and 70. The code is essentially the same as before:

```

> psmr <- NULL
> pts <- seq(0,20,0.1)
> for( ip in c(1995,2005) )
+ for( ia in c(50,60,70) )
+ {
+ nd <- data.frame( A=ia+pts,
+                   P=ip+pts,
+                   dur= pts,
+                   E=1 )
+ psmr <- cbind( psmr, ci.pred( s0, nd) )
+ }
> str( psmr )

num [1:201, 1:18] 4.9 4.34 3.86 3.47 3.17 ...
- attr(*, "dimnames")=List of 2
..$ : chr [1:201] "1" "2" "3" "4" ...
..$ : chr [1:18] "Estimate" "2.5%" "97.5%" "Estimate" ...

```

These 18 columns are 9 columns for 1995, and 9 for 2005, each of these chunks are estimate and lower and upper confidence bound for persons diagnosed in ages 50, 60 and 70.

These can now be plotted:

```

> par( mfrow=c(1,2) )
> matplot( cbind(50+pts,60+pts,70+pts)[,rep(1:3,each=3)],
+          psmr[,1:9], ylim=c(0.7,7),
+          log="y", xlab="Age", ylab="SMR, diagnosed 1995",
+          type="l", lwd=c(4,1,1), lty=1, col="black" )
> abline( h=1 )
> matplot( cbind(50+pts,60+pts,70+pts)[,rep(1:3,each=3)],
+          psmr[,1:9+9], ylim=c(0.7,7),
+          log="y", xlab="Age", ylab="SMR, diagnosed 2005",
+          type="l", lwd=c(4,1,1), lty=1, col="black" )
> abline( h=1 )

```

From the figure it seems that the conclusion is that there is no effect of age or *current* age on SMR, but pretty much that there is an effect of age **at diagnosis** and a very strong initial effect of diabetes duration.

19. Try to simplify the model to one with a simple linear effect of date of diagnosis, and using only knots at 0,1,and 2 years for duration, giving an estimate of the change in SMR as duration increases beyond 2 years.

It would be natural to simplify the model to one with a non-linear effect of duration and linear effects of age at diagnosis and calendar time. We choose knots with successive distances of 1,2,3 and 4 years (a bit out of the blue):

```

> sx <- glm( (lex.Xst=="Dead") ~ I(A-dur) +
+           I(P-2000) +
+           Ns( dur, kn=c(0,1,3,6,10), ref=5 ),
+           offset = log( E ),
+           family = poisson,
+           data = subset( SLr, E>0 ) )
> anova( s0, sx, test="Chisq" )

```

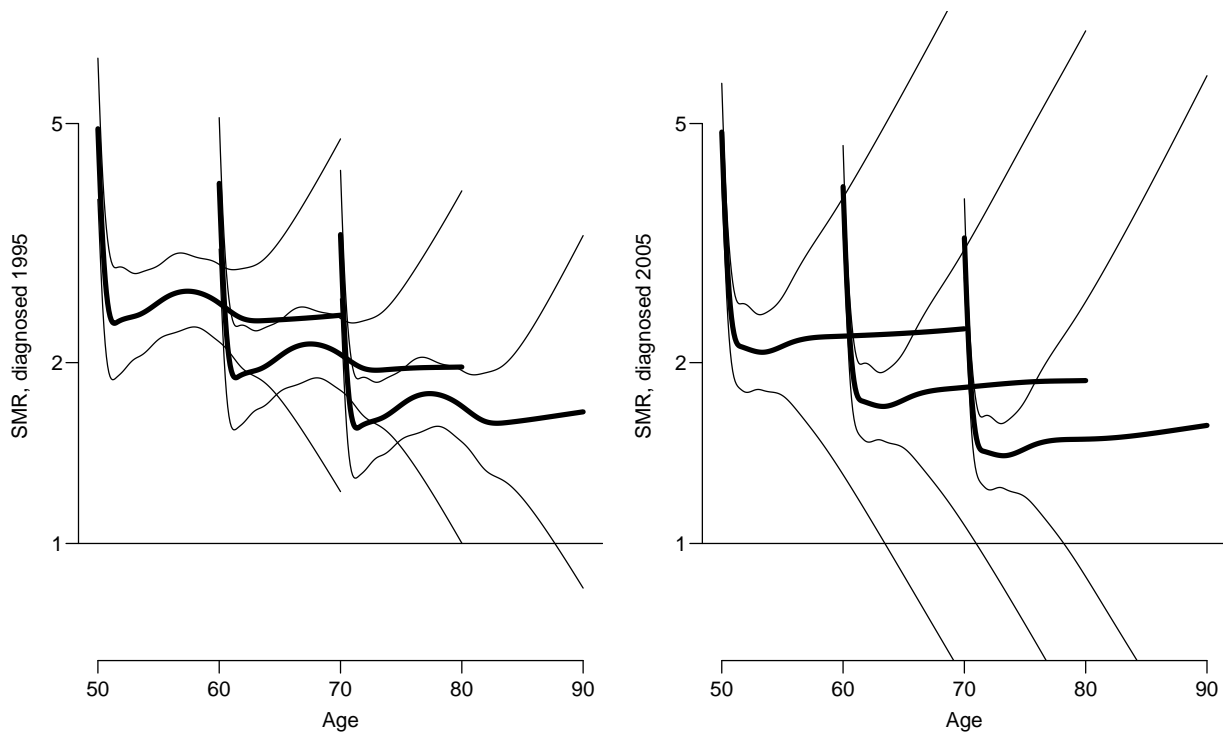


Figure 3.11: *SMR rates for diabetes patients diagnosed 1995 and 2005 in ages 50, 60 and 70.*

Analysis of Deviance Table

```
Model 1: (lex.Xst == "Dead") ~ Ns(A, kn = kn.A, intercept = TRUE) - 1 +
  Ns(P, kn = kn.P, ref = 2000) + Ns(dur, kn = kn.dur, ref = 5)
```

```
Model 2: (lex.Xst == "Dead") ~ I(A - dur) + I(P - 2000) + Ns(dur, kn = c(0,
  1, 3, 6, 10), ref = 5)
```

	Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
1	118418	21925			
2	118425	21935	-7	-9.9485	0.1915

Thus there is no difference between the *very* simple model for SMR and the more complicated ones; and we see that the change in SMR per year of age at diagnosis and calendar year is pretty much the same, namely some 2% per year, or some 15–18% per 10 years:

```
> round( ci.exp(sx), 4 )
```

	exp(Est.)	2.5%	97.5%
(Intercept)	6.7442	5.1886	8.7660
I(A - dur)	0.9809	0.9775	0.9843
I(P - 2000)	0.9843	0.9726	0.9962
Ns(dur, kn = c(0, 1, 3, 6, 10), ref = 5)1	0.5777	0.4830	0.6908
Ns(dur, kn = c(0, 1, 3, 6, 10), ref = 5)2	0.7085	0.6036	0.8316
Ns(dur, kn = c(0, 1, 3, 6, 10), ref = 5)3	0.2638	0.1956	0.3560
Ns(dur, kn = c(0, 1, 3, 6, 10), ref = 5)4	0.9187	0.8040	1.0497

```
> round( ci.exp( sx, subset=c("A","P"), ctr.mat=10*diag(2) ), 4 )
```



```

exp(Est.)  2.5%  97.5%
[1,]      0.8247 0.7966 0.8537
[2,]      0.8536 0.7572 0.9623

```

20. We can also see that the predicted SMRs looks pretty much the same:

```

> xsmr <- NULL
> for( ip in c(1995,2005) )
+ for( ia in c(50,60,70) )
+ {
+ nd <- data.frame( A=ia+pts,
+                   P=ip+pts,
+                   dur= pts,
+                   E=1 )
+ xsmr <- cbind( xsmr, ci.pred( sx, nd) )
+ }
> par( mfrow=c(1,2) )
> matplot( cbind(50+pts,60+pts,70+pts)[,rep(1:3,each=3)],
+          xsmr[,1:9], ylim=c(0.7,7),
+          log="y", xlab="Age", ylab="SMR, diagnosed 1995",
+          type="l", lwd=c(4,1,1), lty=1, col="black" )
> abline( h=1 )
> matplot( cbind(50+pts,60+pts,70+pts)[,rep(1:3,each=3)],
+          xsmr[,1:9+9], ylim=c(0.7,7),
+          log="y", xlab="Age", ylab="SMR, diagnosed 2005",
+          type="l", lwd=c(4,1,1), lty=1, col="black" )
> abline( h=1 )

```

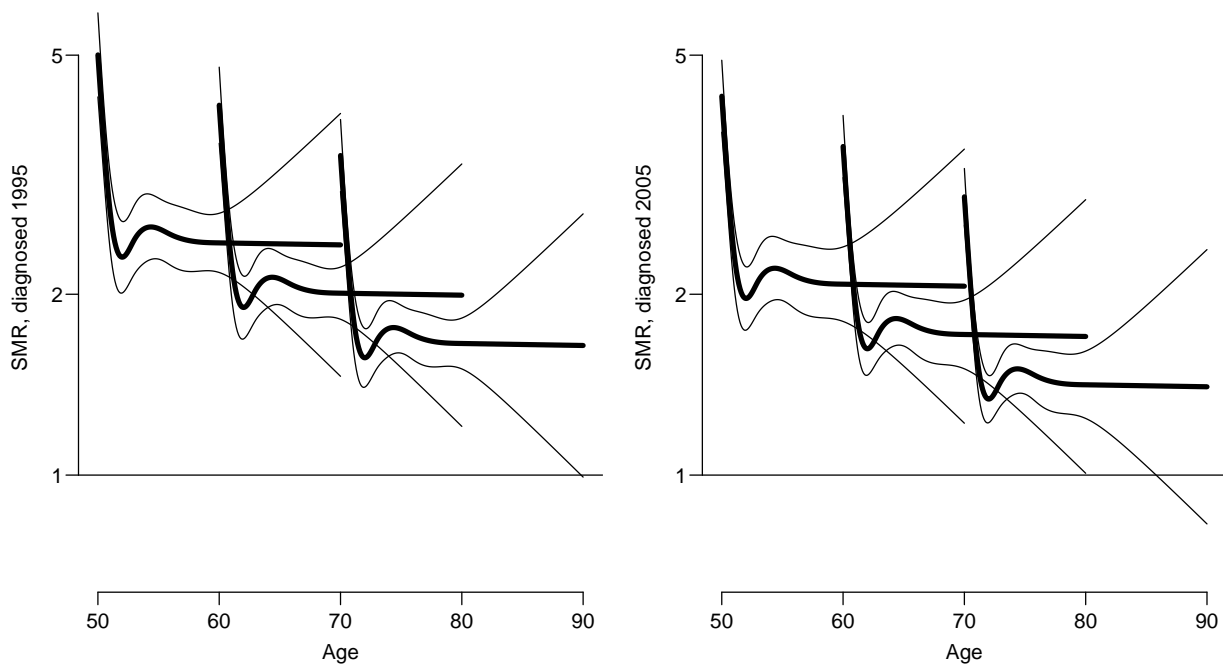


Figure 3.12: SMR rates for diabetes patients diagnosed 1995 and 2005 in ages 50, 60 and 70. Simplified model.

From the figure it seems that the conclusion is that there is no effect of *current* age on SMR, but pretty much that there is an effect of age at **diagnosis** and a very strong initial effect of diabetes duration.