Modern Demographic Methods in Epidemiology with 

University of Edinburgh
August 2014
http://BendixCarstensen.com/AdvCoh/courses/Scot-2014
Version 1.1

Compiled Monday 25th August, 2014, 12:49
from: C:/Bendix/undervis/AdvCoh/courses/Scot.2014/pracs/pracs.tex

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

Contents

Programme of the course

The course is centered around practical calculations in R, illustrating the concepts in analysis of real datasets. All sessions will be alternating between lectures and practicals, most followed by a walk-through of the computing issues. There will be more informal breaks than those listed in the program.

The last part of the course is a combination of a practical and demonstration of a thorough analysis of prevalence, incidence and mortality of diabetes patients in Scotland. In principle, the course participants will at the end (on their own computer) have a complete analysis of the demography of diabetes in Scotland.

Please note the details of the computing requirements on the course web-site, http://bendixcarstensen.com/AdvCoh/Scot-2014/, including download of datasets and programs for the practicals.

Tuesday 26 August 2014

12:00 – 13:00	**Lunch**
13:00 – 15:00	Brief introduction to R.
	Introduction to rates and survival.
	Computing rates, RRs and RDs
15:00 – 15:30	**Tea break**
15:30 – 17:30	Kaplan-Meier, Cox and Lexis.
	Fitting a Cox model and a Poisson model for comparison.
17:30 – 18:00	Summary of the day.

Wednesday 27 August 2014

09:00 – 10:30	Representation of follow-up data.
	Who needs the Cox model anyway?
10:30 – 11:00	**Coffee**
11:00 – 12:30	Estimating — and drawing — a smooth curve.
	Multiple time scales.
12:30 – 13:30	**Lunch**
13:30 – 15:00	Mortality of Danish Diabetes patients.
15:00 – 15:30	**Afternoon Tea**
15:30 – 17:00	SMR and Poisson-modelling.
	SMR of Danish Diabetes patients.
17:00 – 18:00	Summary of the day.

Thursday 28 August 2014

09:00 – 10:30	Interactions, categorical and continuous.
10:30 – 11:00	**Coffee**
11:00 – 12:30	Competing risks and multistate models.
12:30 – 13:30	**Lunch**
13:30 – 15:00	Introducing Scottish diabetes data.
	Binomial regression for (Scottish) prevalence data.
15:00 – 15:30	**Afternoon Tea**
15:30 – 17:00	Scottish diabetes incidence data.
17:00 – 18:00	Summary of the day.

Friday 29 August 2014

09:00 – 10:30	Scottish diabetes mortality rates.
10:30 – 11:00	**Coffee**
11:00 – 12:45	Scottish diabetes SMR.
12:45 – 13:00	Evaluation, feedback, closing remarks and farewell.

Chapter 1

Basic concepts in survival and demography

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

1.1 Probability

Survival function:

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

Conditional survival function:

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

Cumulative distribution function of death times (cumulative risk):

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

Density function of death times:

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

Intensity:

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

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

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

Relationships between terms:

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

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

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

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

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

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

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

1.2 Statistics

Likelihood from one person:

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

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

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

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

$$\ell(\lambda) = d\log(\lambda) - \lambda y$$

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

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

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

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

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

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

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

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

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

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

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

1.3 Competing risks

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

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

The survival function is then:

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

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

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

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

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

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

Subdistribution hazard Fine and Gray defined models for the so-called subdistribution hazard. Recall the relationship between between the hazard (λ) and the cumulative risk (F):

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

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

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

This is what is called the subdistribution hazard, it depends on the survival function S, which depends on *all* the cause-specific hazards:

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

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

1.4 Demography

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

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

where f is the density of the distribution of lifetimes.

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

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

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

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

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

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

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

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

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

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

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

and hence:

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

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

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

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

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

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

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

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

Chapter 2

Exercises

2.1 Calculation of rates, RR and RD

Recall that the standard error of log-rate is $1/\sqrt{D}$, so that a 95% confidence interval for the log of a rate is:

$$\hat{\theta} \pm 1.96/\sqrt{D} = \log(\lambda) \pm 1.96/\sqrt{D}$$

If we take the exponential, we get the confidence interval for the rate:

$$\lambda \overset{\times}{\div} \underbrace{\exp(1.96/\sqrt{D})}_{\text{error factor,erf}}$$

1. Now, suppose you have 15 events during 5532 person-years. Now use R as a simple desk calculator to derive the rate and a confidence interval:

```
> library( Epi )
```

```
> D <- 15
> Y <- 5532
> rate <- D / Y
> erf <- exp( 1.96 / sqrt(D) )
> c( rate, rate/erf, rate*erf )
```

   You can explore the function `ci.mat()`, which lets you use matrix multiplication to produce confidence interval from an estimate and a standard error (or columns of such):

```
> ci.mat()
> exp( c( log(D/Y), 1/sqrt(D) ) %*% ci.mat() )
```

2. Try to achieve this using a Poisson model. Use the number of events as the respoins and the log-person-years as offset:

```
> mm <- glm( D ~ 1, offset=log(Y), family=poisson )
> summary( mm )
```

   What is the interpretation of the parameter in this model?

3. You can extract a confidence interval directly from the model with the `ci.lin()`-function from `Epi`:

```
> ci.lin( mm )
> ci.lin( mm, E=T)[,5:7]
```

4. There is an alternative way to fit a Poisson model, using the rates a the Poisson response, and the person-years as weights instead (albeit it will give you a warning about non-integer response in a Poisson model):

```
> mmx <- glm( D/Y ~ 1, weight=Y, family=poisson )
> ci.lin( mmx, E=T )[,5:7]
```

   Verify that this give the same results as above.

5. The advantage of this approach is that it will also make sense to use an identity link — the response is the same but the paramter estimated is now the rate, not the log-rate:

```
> ma <- glm( D/Y ~ 1, weight=Y, family=poisson(link=identity) )
```

   What is the meaning of the intercept in this model?

   Verify that you actually get the same rate estimate as before.

6. Now use `ci.lin` to produce the estimate and the confidence intervals from this model:

```
> ci.lin( ma )
> ci.lin( ma )[,c(1,5,6)]
```

   Why are the confidence limits not the same as from the multiplicative model?

   Derive the formula for the standard error of this estimated rate.

7. Now, suppose the events and person years are collected over three periods:

```
> Dx <- c(3,7,5)
> Yx <- c(1412,2783,1337)
> Px <- 1:3
```

   Try to fit the same model as before to the data from the separate periods.

```
> m1 <- glm( Dx ~ 1, offset=log(Yx), family=poisson )
```

8. Now test wheter the are rates the same in the three periods: Try to fit a model with the period as a factor in the model:

```
> mp <- glm( Dx ~ factor(Px), offset=log(Yx), family=poisson )
```

   and compare the two models using `anova` with the argument `test="Chisq"`:

```
> anova( m1, mp, test="Chisq" )
```

Compare the test statistic to the deviance of the model `mp`.

9. Suppose instead that we had single observations of each year of follow-up, so that we for each of the 5532 years had an observation of (d, y) where d was either 1 (15 times) or 0 (5517 times), and all the intervals were of length 1:

```
> Dl <- rep( rep(0:1,3), c(1412-3,3,2783-7,7,1337-5,5) )
> Yl <- rep(1,5532)
> Pl <- rep( 1:3, c(1412,2783,1337) )
```

How long are the vectors `Dl`, `Yl` and `Pl`?

10. Now fit the same models as before

```
> mL <- glm( Dl ~ 1, offset=log(Yl), family=poisson )
> mP <- glm( Dl ~ factor(Pl), offset=log(Yl), family=poisson )
```

Compare the two models using `anova` and compare the test statistic to the (residual) deviance of the model `mL`.

What is the deviance good for?

11. If we have observations of two rates λ_1 and λ_0, based on (D_1, Y_1) and (D_0, Y_0) the variance of the difference of the log of the rates, that is the log(RR), is:

$$\begin{aligned} (\log(\mathrm{RR})) &= (\log(\lambda_1/\lambda_0)) \\ &= (\log(\lambda_1)) + (\log(\lambda_0)) \\ &= 1/D_1 + 1/D_0 \end{aligned}$$

As before a 95% c.i. for the RR is then:

$$\mathrm{RR} \overset{\times}{\div} \exp\left(1.96\sqrt{\frac{1}{D_1} + \frac{1}{D_0}}\right)$$

Suppose you have 15 events during 5532 person-years in an unexposed group and 28 events during 4783 person-years in an exposed group:

Compute the the rate-ratio and c.i. by:

```
> D0 <- 15   ; D1 <- 28
> Y0 <- 5532 ; Y1 <- 4783
> RR <- (D1/Y1)/(D0/Y0)
> erf <- exp( 1.96 * sqrt(1/D0+1/D1) )
> c( RR, RR/erf, RR*erf )
> exp( c( log(RR), sqrt(1/D0+1/D1) ) %*% ci.mat() )
```

12. Now achieve this using a Poisson model:

```
> D <- c(D0,D1) ; Y <- c(Y0,Y1); xpos <- 0:1
> mm <- glm( D ~ factor(xpos), offset=log(Y), family=poisson )
```

What does the parameters mean in this model?

Ypu can extract the exponentiated parameters by:

```
> ci.lin( mm, E=T)[,5:7]
```

13. If we instead want the rate-difference, we just subtract the rates, and the variance of the diference is (since the rates are based on independent samples) just the sum of the variances:

$$\begin{aligned} (\log(\text{RD})) &= (\lambda_1) + (\lambda_0) \\ &= D_1/Y_1^2 + D_0/Y_0^2 \end{aligned}$$

Use this formula to compute the rate difference and a 95% confidence interval for it:

```
> rd <- diff( D/Y )
> sd <- sqrt( sum( D/Y^2 ) )
> c( rd, sd ) %*% ci.mat()
```

14. Verify that this is the confidence interval you get when you fit an additive model with exposure as factor:

```
> ma <- glm( D/Y ~ factor(xpos), weight=Y,
+                  family=poisson(link=identity) )
> ci.lin( ma )[,c(1,5,6)]
```

15. Normally one would like to get both the rates and the difference between them. This can be achieved in one go using the `ctr.mat` argument to `ci.lin`. Try:

```
> CM <- rbind( c(1,0), c(1,1), c(0,1) )
> rownames( CM ) <- c("rate 0","rate 1","RR 1 vs. 0")
> CM
> mm <- glm( D ~ factor(xpos),
+                offset=log(Y), family=poisson )
> ci.lin( mm, ctr.mat=CM, E=T)[,5:7]
> round( ci.lin( mm, ctr.mat=CM, E=T)[,5:7], 3 )
```

16. Refit the model with `Y/1000` as the person time, so you get the estimated rates in units of cases per 1000.

17. Use the same machinery to the additive model to get the rates and the rate-difference in one go. Note that the annotation of the resulting estimates are via the column-names of the contrast matrix.

```
> rownames( CM ) <- c("rate 0","rate 1","RD 1 vs. 0")
> ma <- glm( D/Y ~ factor(xpos), weight=Y,
+                  family=poisson(link=identity) )
> ci.lin( ma, ctr.mat=CM )[,c(1,5,6)]
```

2.2 Cox and Poisson modelling

This practical is to show how results from a Cox-model can be reproduced exactly by a Poisson model, and in particular how more sensible and relevant results can be obtained from a Poisson model.

2.2.1 The lung cancer data

The data is the lung cancer data from the `survival` package which comes with R by default. We start by declaring a really large chunk of memory, because we need that to fit a silly model for illustration:

```
> memory.size( 3000 )
> library( Epi )
> library( survival )
> sessionInfo()
```

Note that loading the `survival` package automatically also loads the `splines` package, which is also needed in the exercise.

1. First, load the `lung` data set and have a look at it:

   ```
   > data( lung )
   > str( lung )
   > lung[1:10,]
   ```

2. The deaths are indicated by `status` being equal to 2 — how may deaths are there?

3. How many distinct survival times are there?

2.2.2 Cox-models

4. Fit a traditional Cox-model for the the Mayo clinic lung cancer by `coxph`, where the response is a `Surv` object:

   ```
   > system.time(
   + m0.cox <- coxph( Surv( time, status==2 ) ~ age + factor( sex ),
   +                  method="breslow", eps=10^-8, iter.max=25, data=lung )
   +              )
   > summary( m0.cox )
   ```

5. Create a Lexis object from the dataset

   ```
   > Lung <- Lexis( exit = list( tfe=time ),
   +                exit.status = factor(status,labels=c("Alive","Dead")),
   +                data = lung )
   > summary( Lung )
   ```

   What do you see from the `summary` command?

6. Now try to fit the same Cox-model to data using the formal structures of the `Lexis` object:

   ```
   > mL.cox <- coxph( Surv( tfe, tfe+lex.dur, lex.Xst=="Dead" ) ~
   +                  age + factor( sex ),
   +                  method="breslow", eps=10^-8, iter.max=25, data=Lung )
   > cbind( coef(m0.cox), coef(mL.cox) )
   ```

2.2.3 Poisson models

7. Now split the follow-up data split in small intervals, using all recorded survival times as breakpoints:

```
> Lung.s <- splitLexis( Lung,
+                       breaks=c(0,sort(unique(Lung$time))),
+                       time.scale="tfe" )
> summary( Lung.s )
```

   List all records from one person you choose — use a table of the variable `lex.id` to identify a person with not too many records.

8. Now fit the Cox model to the split dataset

```
> system.time(
+ mLs.cox <- coxph( Surv( tfe, tfe+lex.dur, lex.Xst=="Dead" ) ~
+                    age + factor( sex ),
+                    method="breslow", eps=10^-8, iter.max=25, data=Lung.s )
+             )
```

   Are the results the same?

9. Now fit a Poisson model with a factor accommodating the time-scale defined in the `Lexis` object. You should use the command `factor` to devise a categorical variable:

```
> nlevels( factor( Lung.s$tfe ) )
```

   Note it involves fitting a model with many parameters, so will take some time. Note that the response variable `lex.Xst=="Dead"` is a logical, but by R converts it into a 0/1 numeric:

```
> system.time(
+ mLs.pois.fc <- glm( lex.Xst=="Dead" ~ factor( tfe ) +
+                                age + factor( sex ),
+                                offset = log(lex.dur),
+                     family=poisson, data=Lung.s, eps=10^-8, maxit=25 )
+             )
> length( coef(mLs.pois.fc) )
```

   How does the regression coefficients look compared to the Cox-model?

10. Now replace the factor-model for the time-scale by a smooth spline function. A (cubic) spline is a function made up of 3rd degree polynomials in different intervals defined by knots, in such a way that the polynomials fit nicely together at the knots.

    First defining the knots for the spline, for example:

```
> t.kn <- c(0,25,100,500,1000)
> dim( Ns(Lung.s$tfe,knots=t.kn) )
```

    and then fit the model using `Ns` (look it up!) from the `Epi` package:

```
> system.time(
+ mLs.pois.sp <- glm( lex.Xst=="Dead" ~ Ns( tfe, knots=t.kn ) +
+                                age + factor( sex ),
+                     offset = log(lex.dur),
+                     family=poisson, data=Lung.s, eps=10^-8, maxit=25 )
+             )
> ci.exp( mLs.pois.sp )
> ci.exp( mLs.pois.sp, subset=c("age","sex") )
```

2.2.4 Comparing Cox and Poisson models

11. Compare the estimates of the regression parameters and their confidence intervals between the Cox-model, the factor-Poisson-model and the spline Poisson model.

    What do you conclude?

12. Now use the fitted model to derive the estimated mortality at 0, 10, 20, ..., 1000 days after diagnosis. You must set up a *contrast matrix* with columns corresponding to the parameters of the model, and rows corresponding to the points in time where you want the mortality:

```
> CM <- cbind( 1, Ns( seq(0,1000,10), knots=t.kn ), 60, 1 )
> CM[1:5,]
```

    The mortality rates at these time points, for a 60-year old man are then:

```
> lambda <- ci.exp( mLs.pois.sp, ctr.mat=CM )
```

    What are the units in which `lambda` is measured?

    Also compute the *cumulative* mortality rates (including the s.e.of this), by using the function `ci.cum` (look it up!):

```
> Lambda <- ci.cum( mLs.pois.sp, ctr.mat=CM, intl=10 )
> Lambda <- rbind( 0, Lambda )
```

    Also get the estimate of the survival curve for a male aged 60 from the Cox-model; remember that sex must be specified as a factor with two levels in the data frame in the argument `newdata`:

```
> sf <-  survfit( m0.cox,
+                 newdata=data.frame( sex=factor(2,levels=1:2),
+                                     age=c(60) ) )
```

13. Plot the mortality rates (`lambda`) as a function of time since diagnosis.

    Also plot the estimated survival function from the Cox model on top of the estimated survival function based on the cumulative hazard, using the relationship:

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

    How do the survival curves from the two approaches compare? Which one do you consider the more sensible summary of the survival of 60 year old men with lung cancer?

2.3 Fitting a smooth curve

1. For illustration we fit a very crude model to the mortality follow-up of the men in the Danish Diabetes register:

```
> library( Epi )
> library( splines )
> data( DMlate )
> head( DMlate )
```

2. Now define outcome and age and date of diagnosis for convenience, and restrict data to only men:

```
> DMlate <- transform( DMlate, D = !is.na(dodth),
+                              Y = dox-dodm,
+                              A = dodm-dobth,
+                              P = dodm )
> DMlate <- subset( DMlate, Y>0 & sex=="M" )
> str( DMlate )
```

3. Now fit a model for mortality only depending on age and date at entry:

```
> m0 <- glm( D ~ A + P, family=poisson, offset=log(Y), data=DMlate )
```

— and use `ci.lin` and `ci.exp` to explore the parameters:

```
> ci.lin( m0 )
> round( ci.exp( m0 )[-1,], 3 )
```

How much is mortality changing by age and how much by time?

4. Now try to add a quadratic term in date of diagnosis to the model:

```
> mq <- glm( D ~ A + P + I(P^2), family=poisson, offset=log(Y), data=DMlate )
> round( ci.lin( mq ), 3 )
```

What is the interpretation of the coefficients now (if any)?

5. Now try to make a graph of the RR of death relative to 2005, say: For a given time `P` the log-rate is

$$\mu + \alpha \mathtt{A} + \beta_1 \mathtt{P} + beta_2 \mathtt{P}^2$$

and for 2005 the log-rate is:

$$\mu + \alpha \mathtt{A} + \beta_1 2005 + beta_2 2005^2$$

so the log-RR between these is:

$$\beta_1 \mathtt{P} + beta_2 \mathtt{P}^2 - \beta_1 2005 - beta_2 2005^2 = \beta_1 (\mathtt{P} - 2005) + \beta_2 (\mathtt{P}^2 - 2005^2)$$

Now devise some points between 1995 and 2010 and construct the two columns of numbers to be multiplied by the two parameters:

```
> P.pt <- 1995:2010
> p1 <- P.pt - 2005
> p2 <- P.pt^2 - 2005^2
> coef( mq )
> lRR <- coef(mq)[3] * p1 + coef(mq)[4] * p2
>  RR <- exp( lRR )
> plot( P.pt, RR, type="l", lwd=3, log="y" )
> abline( h=1,v=2005)
```

What is the substantial conclusion from the shape of the RR relative to 2005?

6. Draw the RR relative to year 2000 Is the conclusion substantially different?

7. This curve does not give the confidence limits; to this end we must use the estimated standard errors of β_1 and β_2 *and* the estimated covariance between them. That is hairy.

   There is a facility in the functions `ci.lin` and `ci.exp`, that will both select the relevant parameters (in this case those with names `P` and `I(P^2)`).

   Try (and look at the help page for `ci.lin`):

```
> ci.lin( mq, subset="P" )
> ci.exp( mq, subset="P" )
```

   There is a further argument to these functions, `ctr.mat` — contrast matrix which is the columns of numbers we defined above to multiply by each of the parameters, try:

```
> cbind( p1, p2 )
> ci.exp( mq, subset="P", ctr.mat=cbind(p1,p2) )
```

8. The result is the estimated curve with confidence intervals, so plot it (use the function `matplot`):

```
> matplot( P.pt, ci.exp( mq, subset="P", ctr.mat=cbind(p1,p2) ),
+          type="l", col="Black", lty=1, lwd=c(3,1,1), log="y" )
> abline( h=1,v=2000)
```

9. What really goes on here is that we as contrast matrix use the difference between the matrix of `P` and $\texttt{P}^2$, and the matrix that consists of the 2000-row all way through:

```
> ( MP <- cbind( P.pt, P.pt^2 ) )
> ( Mr <- cbind( rep(2000,length(P.pt)), 2000^2 ) )
> MP-Mr
> ci.exp( mq, subset="P", ctr.mat=MP-Mr )
> matplot( P.pt, ci.exp( mq, subset="P", ctr.mat=MP-Mr ),
+          type="l", col="Black", lty=1, lwd=c(3,1,1), log="y" )
> abline( h=1,v=2000)
```

10. Now suppose we want to model the period effect by a `cubic spline` instead. This is a function that is a cubic between a set of points called *knots*. In the `Epi` package is a function `Ns`, that will generate a set of columns corresponding to this — think of it as the counterpart of the columns `P` and $\texttt{P}^2$:

```
> p.kn <- c(1997,2000,2003,2006,2009)
> Ns( P.pt, knots=p.kn )
```

    Now put this in a model:

```
> ms <- glm( D ~ A + Ns(P,knots=p.kn), family=poisson, offset=log(Y), data=DMlate )
> ci.exp( ms )
> ci.exp( ms, subset="P" )
```

The parameters do not have any meaning *per se*, just as in the case of the coefficients in the quadratic case.

11. Now extract the RR relative to 2005 from this new model as before.

    First construct the matrix for the points 1995—2010 and the reference for 2005:

```
> CP <- Ns( P.pt, knots=p.kn )
> Cr <- Ns( rep(2005,length(P.pt)), knots=p.kn )
> CP-Cr
```

    Then we can extract the RR ad plot it:

```
> RR <- ci.exp( ms, subset="P", ctr.mat=CP-Cr )
> matplot( P.pt, RR,
+          type="l", col="Black", lty=1, lwd=c(3,1,1), log="y" )
> abline( h=1,v=2000)
```

12. Now explore whether there is any non-linearity by age (on the log-mortality scale, that is):

```
> ma <- update( ms, . ~ . + I(A^2) )
> round( ci.lin( ma ), 3 )
```

13. In order to report the model `ms` in full, we will also need to show the estimated mortality rates as a function of age. For that purpose we of course must use the 2005 reference point.

    (a) The first possibility is to devise a prediction data frame:

```
> nd <- data.frame( A = 40:85, P=2005, Y=1000 )
```

    Note that you must provide values for *all* covariates, including the person-years, that enter in the model as offset, that is as a covariate with fixed regression coefficient. The function `ci.pred` is a simple convenience wrapper for `predict.glm`:

```
> rate <- ci.pred( ma, newdata=nd )
> matplot( nd$A, rate,
+          type="l", col="Black", lty=1, lwd=c(3,1,1), log="y" )
```

    Note that since you initially entered `Y` in units of 1 person-year, we get the rates in units of events per 1000 person-years by entering Y with the value of 1000 in the prediction frame.

    (b) The other possibility is to use `ci.exp` directly to extract the predicted rates from the model, but they will be in he units of the `Y` entered into the model. Note that you in this case must be careful to get the order of the columns in `ctr.mat` right:

```
> ci.exp( ma )
> Rate <- ci.exp( ma, ctr.mat=cbind(1,40:85,Ns(rep(2005,46),knots=p.kn),(40:85)^2) )*1000
> matplot( nd$A, Rate,
+          type="l", col="Black", lty=1, lwd=c(3,1,1), log="y" )
```

14. Try to explore if there are further non-linearities in the age-effect by including a spline with several knots

```
> ( a.kn <- 3:9*10 )
> mA <- update( ms, . ~ . - A + Ns(A,knots=a.kn) )
> round( ci.lin( mA ), 3 )
```

How many parameters are in the age-effect?

```
> rate <- ci.pred( mA, newdata=nd )
> matplot( nd$A, rate,
+          type="l", col="Black", lty=1, lwd=c(3,1,1), log="y" )


> ci.exp( ms )
```

2.4 Simple analysis of Estonian stroke data

```
> library(Epi)
```

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

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

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

1. First, read in the data using the `read.table()` or `read.csv()` command. Do not forget to *look into the file before* to see, what the field separator is.

   Calculate an id variable in the dataframe.

```
> stroke <- read.table( url("http://BendixCarstensen.com/AdvCoh/Scot-2014/data/stroke.csv"),
+                       sep=";", header=TRUE, na.strings="." )
> stroke$id <- 1:nrow(stroke)
> str( stroke )
> head( stroke )
```

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

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

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

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

   How many deaths occurred after 1 January 1996?

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

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

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

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

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

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

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

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

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

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

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

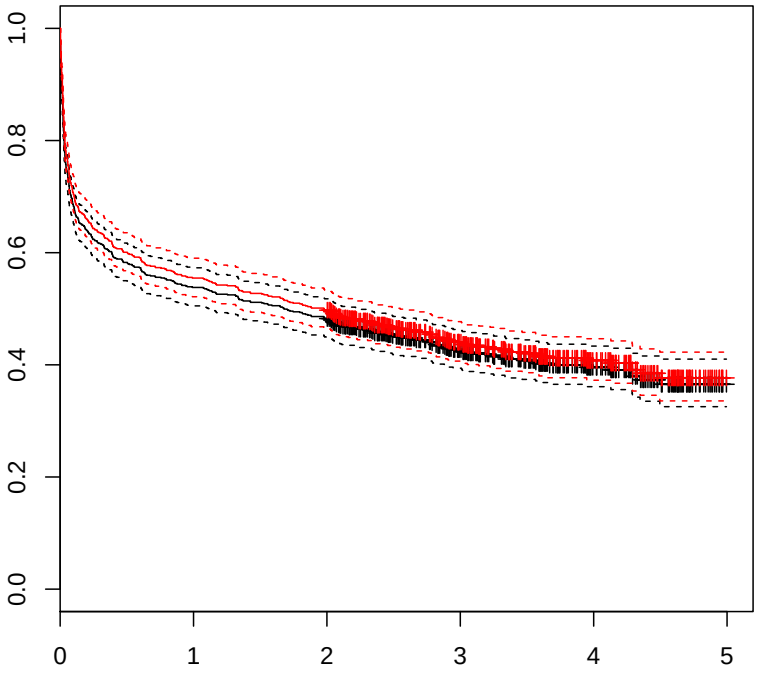

Figure 2.1: *The Kaplan-Meier estimator with (black) and without (red) the 0-survivors, i.e. the persons who die at the same time as their stroke.*

8. Plot the result as 4 curves.

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

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

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

   What do you conclude?

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

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

```
> Lst <- Lexis( data=stroke, entry=list(Per=dstr,Age=age,Tfs=dstr-dstr),
+                            exit=list(Per=dox),
+                    exit.status=as.numeric(stroke$D) )
> head( Lst )
```

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

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

```
> summary( Lst )
```

11. Get an overview of how the number of deaths and person years is distributed by time:

```
> plot( Lst )
```

Try to enhance the Lexis diagram by using the graphical arguments to `plot.Lexis` and `points.Lexis`. By default, `plot.Lexis` makes a plot using the first two timescales of the `Lexis` object. So it matters in which order the timescales are defined.

Below you see the necessary graphical formatting necessary to get squares in the Lexis diagram, i.e. the same physical scale on both axes: `mai=` gives the margins on the four sides of the plot in inches, a total of 1 inch in each direction. Thus, the `height=10+1,width=3+1` gives a plot area of 3 by 10 inches, accommodating a 30 year period (horizoantal) and a 100 year age-span (vertical). You probably want to use another path name for the file, though.

```
> pdf( "../graph/stroke1-LexisX.pdf", height=10+1, width=3+1 )
> par( mai=c(3,3,1,1)/4, mgp=c(3,1,0)/1.6 )
> plot(Lst,xlim=1980+c(0,30),ylim=c(0,100),
+           col=c("red","blue")[Lst$sex+1],grid=0:20*5,xaxs="i",yaxs="i")
> points( subset(Lst,lex.Xst==TRUE),pch=16,cex=0.6,
+          col=c("red","blue")[Lst$sex+1][Lst$lex.Xst==TRUE])
> dev.off()
```

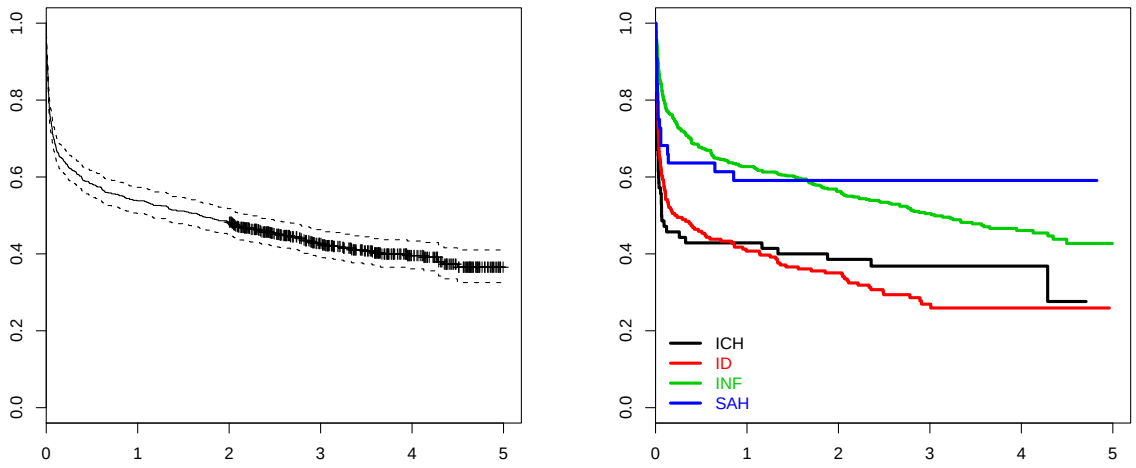

Figure 2.2: *Kaplan-Meier plot for the Estonian stroke data, overall and subdivided by diagnosis.*

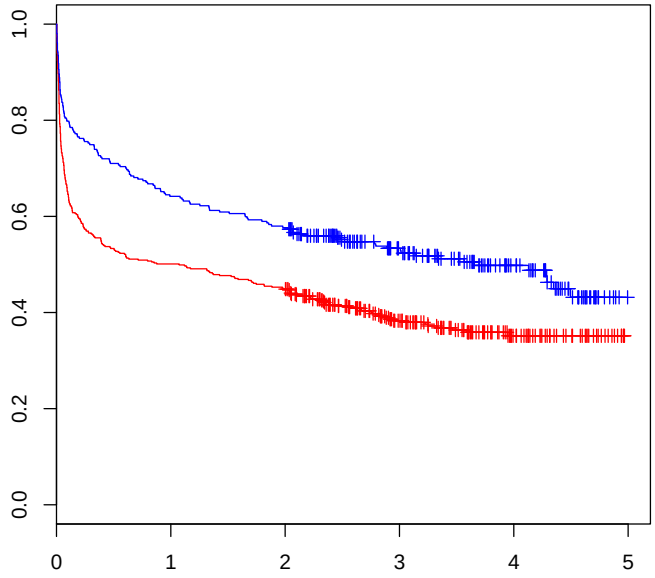

Figure 2.3: *Kaplan-Meier plot for the Estonian stroke data, subdivided by sex.*

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

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

13. What is the time-scale we are using here?

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

```
> save( stroke, Lst, file="../data/from-exc-stroke1.Rdata" )
```

2.5 Cox model and time-splitting using Estonian stroke data

As the previous exercise, this is also very prescriptive, in that almost all the R-code is given to you in the text, but it is important that you make sure you understand the code, for example by trying slightly different arguments to functions, and inspecting the data frames and other object you create.

Now first attach the packages you will need:

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

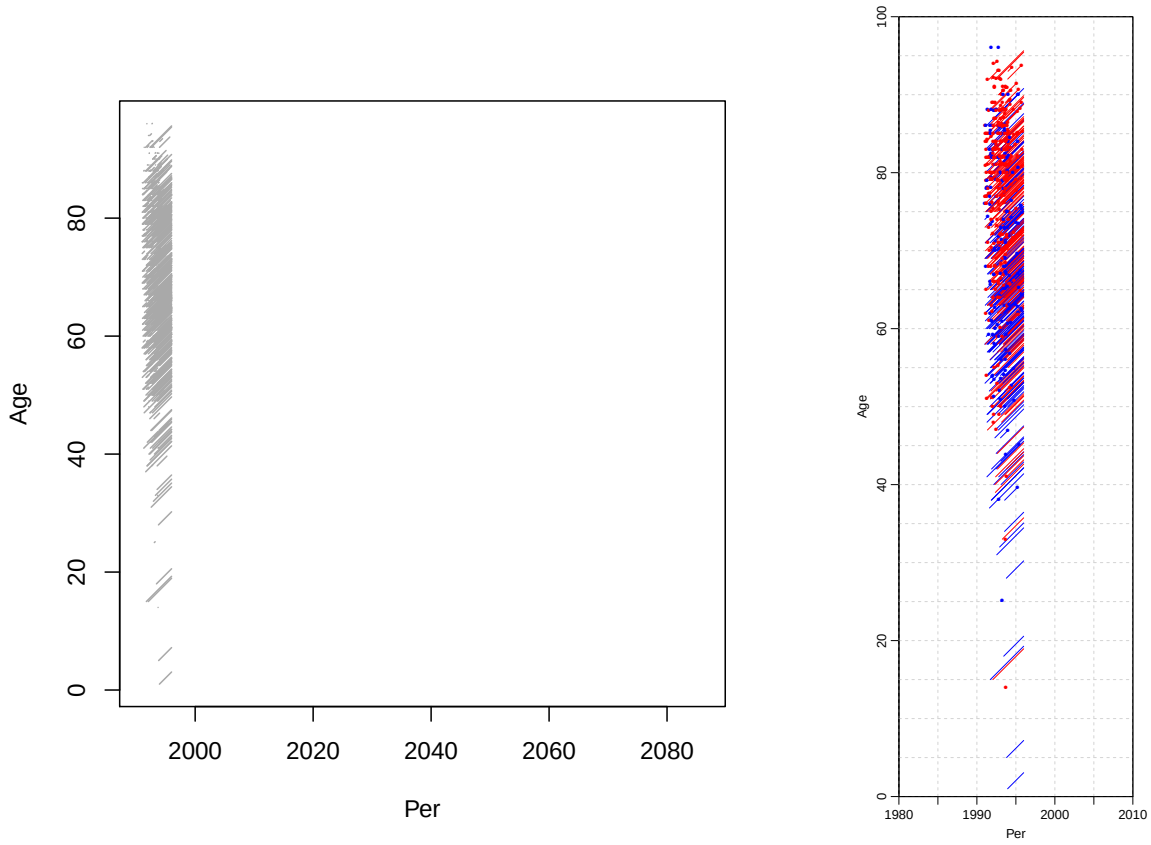

Figure 2.4: *Lexis diagram for the Estonian stroke data. Left panel is the default lay-out, the right-hand panel is the result of adjusting the plotsize etc. as shown above. Red: Women, Blue: Men.*

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

```
> memory.size(3000)
> load( file="../data/from-exc-stroke1.Rdata" )
> str( Lst )
```

Alternatively you must read the data afresh, transform etc.

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

   ```
   >   mc <- coxph( Surv(dox-dstr,D) ~ sex, data=stroke )
   >   summary( mc )
   >   mL <- coxph( Surv(lex.dur,lex.Xst==1) ~ sex, data=Lst )
   >   summary( mL )
   ```

   Are there any differences?

   What is the underlying time scale used here?

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

```
>   mLa <- coxph( Surv(lex.dur,lex.Xst==1) ~ sex + age, data=Lst )
>   summary( mLa )
```

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

You may also use the `ci.exp` function if you only want to see the RR parameters:

```
> ci.exp( mL )
> ci.exp( mLa )
```

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

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

```
> plot( survfit( Surv(lex.dur,lex.Xst==1) ~ interaction(sex,age<75),
+                data=Lst ),
+                col=c("red","blue"), lwd=3 )
```

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

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

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

```
> length( unique(Lst$lex.dur[Lst$lex.Xst==1]) )
> sLst <- splitLexis( Lst, breaks=c(0:9/100,seq(0.1,10,0.05)), "Tfs" )
> summary(  Lst )
> summary( sLst )
```

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

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

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

```
> mCs <- coxph( Surv(lex.dur,lex.Xst==1) ~ sex + age, data=Lst )
> ci.exp( mLa )
> mC <- coxph( Surv(Tfs,Tfs+lex.dur,lex.Xst==1) ~ sex + age, data=sLst )
> ci.exp( mC )
```

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

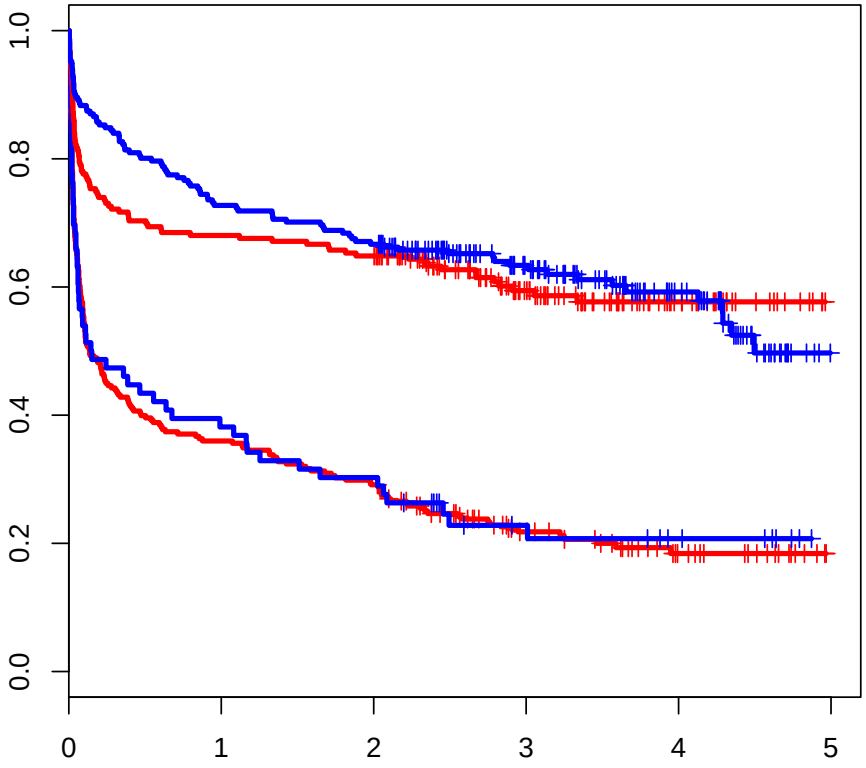

Figure 2.5: *Kaplan-Meier curves for males (blue) and females (red) over and under 75 years at stroke.*

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

8. Now take a look at the estimated coefficients:

```
> coef( mP )
> length( coef(mP) )
```

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

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

Are there any major differences?

9. If time permits (this takes rather long computing time):

   Split time since stroke in intervals of length 0.01 years instead of 0.05 years and repeat the analysis.

10. Now use a parametric function for the baseline hazard. We will use restricted cubic splines (natural splines) with knots at 0.05, 0.2, 0.7, 1.5, 3 and 4.8 years, but we also need a quantitative variable giving the midpoint of the interval, which is achieved by the function `timeBand`:

```
> sLst$Tfs.m <- timeBand( sLst, "Tfs", "middle" )
> kn <- c(0,0.05,0.2,0.7,1.5,3,4.8)
> mS <- glm( lex.Xst ~ Ns( Tfs.m, knots=kn ) + sex + age,
+             offset = log(lex.dur),
+             family = poisson, data=sLst )
```

   Compare the parameter estimates with the previous models.

```
> ci.exp( mC )
> ci.exp( mP, subset=c("sex","age") )
> ci.exp( mS, subset=c("sex","age") )
> ci.exp( mS )
```

11. Obtain an estimate of the baseline hazard function for a female aged 60. You will need to generate a sequence of times where you compute it:

```
> t.pt <- seq(0,5,0.01)
> CM <- cbind( 1, Ns( t.pt, knots=kn ), 0, 60 )
> hz <- ci.exp( mS, ctr.mat=CM ) * 100
> matplot( t.pt, hz,
+          ylim=c(1,500), log="y", ylab="Hazard rate (%/y)",
+          type="l", lty=1, col="black", lwd=c(3,1,1) )
```

12. Alternatively, you can obtain the hazard by `predict` using the `newdata=` argument. Note taht you also need to specify values of `lex.dur` which is in the offset of the model:

```
> nd <- data.frame( Tfs.m=t.pt, sex=0, age=60, lex.dur=100 )
> prhz <- predict( mS, newdata=nd, type="link", se.fit=T )
> str( prhz )
> prhz <- exp( cbind( prhz$fit, prhz$se.fit ) %*% ci.mat() )
```

   Verify that you get the same estimates:

```
> matplot( hz, prhz, pch=16, log="xy" )
```

13. Obtain an estimate of the survival function for a female aged 60. You can reuse the sequence of times from before with the modification that you should not use 0. Consult the help page for `ci.cum` first.

```
> t.pt <- t.pt[-1]
> CM <- cbind( 1, Ns( t.pt-0.005, knots=kn ), 0, 60 )
> Hz <- ci.cum( mS, ctr.mat=CM, intl=0.01 )
> matplot( t.pt, exp(-Hz)[,-4],
+          type="l", lwd=c(3,1,1), ylim=c(0,1), lty=1, col="black" )
```

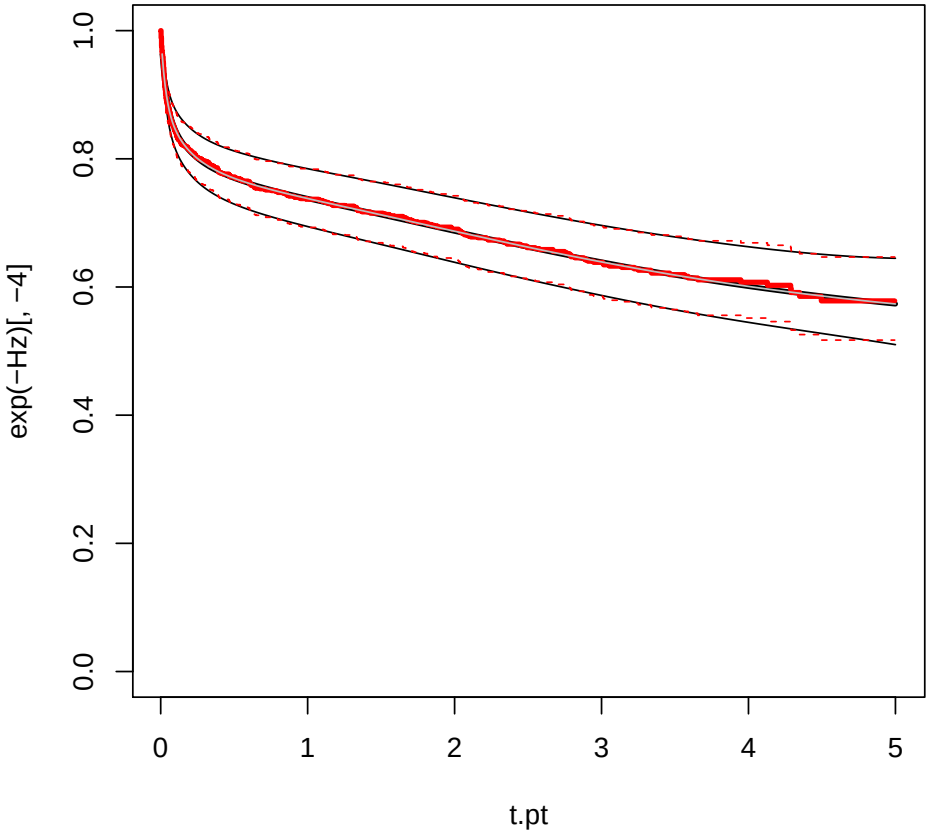

Figure 2.6: *Estimated survival curve for a female 75-year at stroke, computed by the Breslow-setimator from the Cox-model, and by using the approximation from the Poisson model.*

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

```
> matplot( t.pt, exp(-Hz)[,-4],
+          type="l", lwd=c(3,1,1), ylim=c(0,1), lty=1, col="black" )
> lines( survfit(mC,newdata=data.frame(sex=0,age=60)),
+        conf.int=TRUE, col="red", mark.time=FALSE )
> # overplot the estimate with a thicker line:
> lines( survfit(mC,newdata=data.frame(sex=0,age=60)),
+        conf.int=FALSE, col="red", lwd=3, mark.time=FALSE )
> # and a thin line with the smoth estimate:
> lines( t.pt, exp(-Hz)[,1], col="gray" )
```

Which estimator seems the most realistic to you?

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

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

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

- Time should be split finely.
- The effect of time should be modelled detailed.
- Compute the hazards at the mipoint of the intervals, but plot the cumulative hazard (or equivalently, the survival function) at the upper end of the intervals.

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

This exercise is using data from the National Danish Diabetes register. There is a 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, `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
```

2. 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()`.

3. Get an overall overview of the mortality by using `stat.table` to tabulate no. deaths, person-years and the crude mortality rate by sex.

4. 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), time.scale="A" )
> summary( SL )
```

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

5. 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.

   With these objects we can get the estimated log-rates by using `predict`, and supplying a data frame of prediction points, so first make a data frame of prediction points, it must have variables corresponding to the predictor variables in the model, including the off-set variable.

```
> nd <-  data.frame( A = seq(10,90,0.5),
+              lex.dur = 1000)
> p.m <- predict.glm( r.m, type = "link",
+                         newdata = nd,
+                          se.fit = TRUE )
> p.f <- predict.glm( r.f, type = "link",
+                         newdata = nd,
+                          se.fit = TRUE )
> str( p.m )
```

   What is the structure of the prediction objects, when you use the `se.fit=TRUE` argument?

   Construct estimated rates with 95% c.i.s from these, and plot the two sets of estimated rates (men and women). (Hint: use `matplot`)

Graphical comparison with the population rates

6. We can compare the mortality rates of the diabetes patients with the mortality rates from the general population; they are available in the data frame `M.dk`

```
> data( M.dk )
> head( M.dk )
```

Plot the mortality rates from a particular year on top of the estimated rates, for example:

```
> with( subset( M.dk, sex==1 & P==2005 ), lines( A, rate, col="blue", lty="12", lwd=3 ) )
```

Guess how to plot the mortality rates for women...

7. It would more natural to model the population mortality rates in a similar fashion as the diabetes mortality rates, try:

```
> R.m <- glm( D ~ ns( A, df=10 ),
+             offset = log( Y ),
+             family = poisson,
+               data = subset( M.dk, sex==1 & P>1994 ) )
```

Now obtain the same model for women, and construct the predicted rates as before — note that you will need a new dataset for prediction, because in this dataset the persons-years are called `Y`, while in the dataset with the patient follow-up, the person-years were in a variable called `lex.dur`.

Add the curves with predicted rates to the plot of the patient mortality rates.

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 (lines 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 will be using 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.

   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(5,95,10)/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 these we can now model mortality rates (separately for men and women), as functions of age, calendar time and duration. To this end you will need the `splines` package, and you will need the function `ns` (look that up). Specification of natural splines is a bit clumsy, you will need expressions like in your model formula:

```
> ns( A, kn=kn.A[-c(1,length(kn.A))],Bo=kn.A[ c(1,length(kn.A))] )
```

Therefore you can use a convenience wrapper, `Ns` that does the allocation of knots. You do not have to key it in, it is a pert of the `Epi` package. You can now specify a model very simply (remember to check the names of the vectors where you put the positions of the knots; here assumed to be `kn.A`, `kn.P` and `kn.dur`):

```
> 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 )
> mf <- update( mm, data = subset( SL, sex=="F" ) )
```

What is the interpretation of the parameters (if any)?

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.

    What is the problem with this approach?

11. The models that you fitted separately for men and women has three terms: age (`A`), calendar time (`P`) and diabetes duration (`dur`). Since the outcome is a rate with dimension time^{-1} we must put the rate dimension on one of these terms and leave the two others as rate-ratios. In order to do this we must fix reference values for the two rate-ratio terms. The natural variable for the rate-dimension is age, so that we get estimated age-specific rate-ratios for a specific calendar time, 1.1.2008, say, and a specific duration of diabetes, 2 years, say.

    In order to extract these terms from the model we need contrast matrices, that is matrices where each row corresponds to a set of values for age or period or duration, and the columns correspond to the parameters in the model.

    This is one reason for explicitly fixing the knots in the spline definitions; when we extract the effects we must use the same set of knots as in the model specification.

    We will need matrices for specified sets of values for age, calendar time and duration, but also matrices where all rows refer to the chosen reference values for calendar time and duration.

    We begin by specifying the prediction points for the time scales and the reference points. There is formally no reason to require that the matrices all have the same number of rows, but it makes the handling of the reference points much easier.

    ```
    > N <- 100
    > pr.A <- seq(10,90,,N)
    > pr.P <- seq(1995,2010,,N)
    > pr.d <- seq(0,15,,N)
    > rf.P <- 2009
    > rf.d <- 2
    ```

    With these in place we generate the matrices we shall multiply to the parameter estimates:

```
> AC <- Ns( pr.A, knots=kn.A )
> PC <- Ns( pr.P, knots=kn.P )
> dC <- Ns( pr.d, knots=kn.dur )
> PR <- Ns( rep(rf.P,N), knots=kn.P )
> dR <- Ns( rep(rf.d,N), knots=kn.dur )
```

What are the dimensions of these matrices?

Note that the rows of `AC` refer to `N` points on the age-scale, `PC` to `N` points on the calendar time scale, etc.

These matrices are the necessary input for extracting the effects; this is done by the function `ci.exp` — remember to take a look at the help page for this.

Note that we make use of *all* parameters when extracting the age-effect — this is the effect where we have the dimension of the response (rate), and hence the intercept, and where we have fixed the values of date and duration at their reference values.

The rate-ratios for calendar time and duration are estimated exclusively from the parameters for these terms, but note that we subtract the values at the reference point:

```
> m.A <- ci.exp( mm, ctr.mat=cbind(1,AC,PR,dR) ) * 1000
> m.P <- ci.exp( mm, subset="P"  , ctr.mat=PC-PR )
> m.d <- ci.exp( mm, subset="dur", ctr.mat=dC-dR )
> f.A <- ci.exp( mf, ctr.mat=cbind(1,AC,PR,dR) ) * 1000
> f.P <- ci.exp( mf, subset="P"  , ctr.mat=PC-PR )
> f.d <- ci.exp( mf, subset="dur", ctr.mat=dC-dR )
```

12. Plot the three effects in three panels beside each other. Plot the estimates for men and women together.

    (Hint: Use the function `matplot` to plot both estimates and confidence limits for both sexes in one go.

    How are the estimated effects — is the chosen parametrization plausible? Would you want to reconsider the number of knots you used for any of the terms?

13. We have so far fitted models separately for men and women, we might fit a model for the entire dataset with common period and duration effects, but different age-effect for the two sexes. Try to fit this interaction model and take a look at the estimates:

```
> m2 <- glm( (lex.Xst=="Dead") ~ sex +
+                               sex:Ns( A, kn=kn.A ) +
+                                   Ns( P, kn=kn.P ) +
+                                   Ns( dur, kn=kn.dur ),
+            offset = log( lex.dur ),
+            family = poisson,
+              data = SL )
> ci.exp(m2)
```

    What do these estimates refer to?

14. Now test this model against the separate models; the deviance and degrees of freedom from the separate models for men and women add up to that of a joint model with interaction between all terms and sex. Compare the total deviance for these with that of the fitted interaction model

Hint: To find the deviance and degrees of freedom, look at the model object by saying: `names(m2)`.

Is there any evidence of different period and duration effects between thew sexes?

15. Would you test whether there were proportional (or even identical) mortality rates between men and women? Why? And how?

16. Extract the parameters from the model, showing the separate age-effects for men and women, and the common period and duration effects. To get the age-specific rates for men and women at the reference time and reference duration you may need the subset-argument to `ci.exp`, for example:

```
> ci.exp( m2, subset=c("Int","sexF","P","dur") )
```

17. 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 all multiple regression) 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 `predict` function with the `newdata=` argument. So a person diagnosed in age 50 will have a (log-)mortality measure 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 )
> predict( mm, newdata=nd, se.fit=TRUE )
```

    Take a look at the result from the `predict` 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.

18. The model we used for the mortality rates used three time-scales: age, calendar time and duration of diabetes.

    It would be of interest to see whether we would get the same (or better) description by adding age at diagnosis and date of diagnosis to the model.

    Now, age at diagnosis = current age − duration of diabetes, and date of diagnosis = current date − duration of diabetes, so the terms we might add only constitute the *non-linear* effects of these variables.

    When you add the effects of age and date of diagnosis you want to use a set of knots which is aligned to the variables you consider, for example:

```
> kn.Ad <- with( subset( SL, lex.Xst=="Dead" ),
+                quantile( A-dur, probs=seq(5,95,10)/100 ) )
> kn.Pd <- with( subset( SL, lex.Xst=="Dead" ),
+                quantile( P-dur, probs=seq(5,95,20)/100 ) )
```

Now add the effects one at at time and test whether age at diagnosis or current age is the better predictor. If you just want to test whether adding a new term to the model it is convenient to use the `update` function, for example:

```
> anova( mm,
+        update( mm, . ~ . + Ns(A-dur,knots=kn.Ad) ),
+        update( mm, . ~ . + Ns(A-dur,knots=kn.Ad) - Ns(A,knots=kn.A) ),
+        test = "Chisq" )
```

Is there any indication of whether current age or age at diagnosis, or current date or date of diagnosis is the better choice?

19. Fit the models with age at diagnosis and date of diagnosis as explanatory variables instead. To extract the effects you also need new contrast matrices, because the deaths are distributed differently along these "entry"-variables.

20. Show the effects together with the effects from the model with three time scales (that is the model with current age and current date of follow-up.

2.6.1 SMR

The SMR is the standardized mortality ratio, which is 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.

21. We will use the second approach, that is include as an extra variable the population mortality as available from the data set `M.dk`.

    First we 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 rater at a point 6 months after the left endpoint of the interval — recall we split the follow-up in 12 month intervals).

    We need to have variables with the same names in both datasets when we merge them, and moreover, they should be of the same type, so we must transform the sex variable in `M.dk` to a factor:

```
> str( SL )
> SL$Am <- floor( SL$A+0.5 )
> SL$Pm <- floor( SL$P+0.5 )
> 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 we can match up the rates from `M.dk`:

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

22. 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.

23. 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. What do the extracted effects represent now?

24. Is there any difference between SMR for males and females?

25. Fit the model with common SMR for the two sexes. Plot the estimated common effects for SMR.

26. 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.

27. What are the estimated annual change in SMR by date of diagnosis and by duration after 2 years?

Interaction models

This section is quite technical, but nonetheless important from a practical point of view, since it contains exampales of how to construct and report continuous-continuous interactions.

28. We may explore whether there is an interaction between age and duration by including a product of the duration effects and age at diagnosis:

```
> Six <- update( Sx, . ~. + I(A-dur):Ns(dur,knots=kn.dur) )
> anova( Six, Sx, test="Chisq" )
> ci.exp( Six )
```

Even if the effect is not statistically significant, we would still want to explore the shape of it:

```
> Six.A <- ci.exp( Six, ctr.mat=cbind(1,AC,rf.P,dR,dR*pr.A) )
> Six.P <- ci.exp( Six, subset="P"     , ctr.mat=cbind(pr.P-rf.P) )
> Six.d <- ci.exp( Six, subset="kn.dur", ctr.mat=cbind(dC-dR,(dC-dR)*50) )
> for( a in seq(55,90,5) ) Six.d <- cbind( Six.d,
+                 ci.exp( Six, subset="kn.dur", ctr.mat=cbind(dC-dR,(dC-dR)*a) ) )
> dim( Six.d )
> par( mfrow=c(1,3), mar=c(3,3,1,1), mgp=c(3,1,0)/1.6 )
> matplot( pr.A, Six.A,
+          type="l", lty=1, lwd=c(3,1,1), col="black",
+          log="y", ylim=c(1/2,4),
+          xlab="Age at diagnosis", ylab="SMR" )
> abline( h=1 )
> abline( v=4:8*10, col="gray" )
> matplot( pr.P, Six.P,
+          type="l", lty=1, lwd=c(3,1,1), col="black",
+          log="y", ylim=c(1/2,4),
+          xlab="Date of diagnosis", ylab="SMR ratio" )
> abline( h=1 )
> matplot( pr.d, Six.d,
+          type="l", lty=1, lwd=c(3,1,1), col=rep(heat.colors(9),each=3),
+          log="y", ylim=c(1/2,4),
+          xlab="Diabetes duration", ylab="SMR ratio" )
> abline( h=1 )
```

29. This approach is however a bit artificial, because we have fixed the duration effects to be 1 at duration 2 years. It would be appropriate to combine the effects of age at diagnosis and duration to show how the SMR looks as a function of current age, for patients diagnosed with DM at different ages.

    Note the trick with putting an `NA` at the end of `pts` and then stacking all the predictions for persons diagnosed at different ages. This means that the curve plotted will be broken between the different ages at diagnosis. Also note that we use the recycling rule when setting up the data frame.

```
> pts <- c(seq(0,15,0.1),NA)
> np <- length( pts )
> nd <- data.frame( A=rep(seq(50,90,5),each=np)+pts,
+                   P=rf.P+pts,
+                 dur=     pts,
+                   E=1 )
> A.si <- exp(sapply(predict( Six, newdata=nd, se.fit=TRUE )[1:2],cbind) %*% ci.mat())
> A.sm <- exp(sapply(predict( Sx , newdata=nd, se.fit=TRUE )[1:2],cbind) %*% ci.mat())
```

```
> matplot( NA, NA,
+          log="y", ylim=c(1/2,5), xlim=c(50,100),
+          xlab="Age at follow-up", ylab="SMR" )
> abline( h=c(5:19/10,seq(2,5,0.5)), v=seq(50,100,5), col=gray(0.8) )
> matlines( nd$A, cbind(A.si,A.sm),
+           type="l", lty=rep(c(1,3),each=3), lwd=c(3,1,1), col="forestgreen" )
> abline( h=1 )
```

30. This interaction machinery with linear age easily generalizes to more complex age-effects, it is just a question of choosing another age-effect:

```
> SiX <- update( Sx, . ~. + Ns(A-dur,knots=kn.Ad):Ns(dur,knots=kn.dur) )
> anova( SiX, Six, Sx, test="Chisq" )
```

And we can use the exact same code to show the interaction and plot it along the others in a similar plot:

```
> A.sX <- exp(sapply(predict( SiX, newdata=nd, se.fit=TRUE )[1:2],cbind) %*% ci.mat())
> matplot( NA, NA,
+          log="y", ylim=c(1/2,5), xlim=c(50,100),
+          xlab="Age at follow-up", ylab="SMR" )
> abline( h=c(5:19/10,seq(2,5,0.5)), v=seq(50,100,5), col=gray(0.8) )
> matlines( nd$A, cbind(A.sX,A.si,A.sm),
+           type="l", lty=rep(c(1,3),c(6,3)), lwd=c(3,1,1),
+           col=rep(c("magenta","forestgreen"),c(3,6)) )
> abline( h=1 )
```

Chapter 3

Solutions

3.1 Calculation of rates, RR and RD

Recall that the standard error of log-rate is $1/\sqrt{D}$, so that a 95% confidence interval for the log of a rate is:

$$\hat{\theta} \pm 1.96/\sqrt{D} = \log(\lambda) \pm 1.96/\sqrt{D}$$

If we take the exponential, we get the confidence interval for the rate:

$$\lambda \overset{\times}{\div} \underbrace{\exp(1.96/\sqrt{D})}_{\text{error factor,erf}}$$

1. Now, suppose you have 15 events during 5532 person-years. Now use R as a simple desk calculator to derive the rate and a confidence interval:

```
> library( Epi )
```

```
> D <- 15
> Y <- 5532
> rate <- D / Y
> erf <- exp( 1.96 / sqrt(D) )
> c( rate, rate/erf, rate*erf )
```

```
[1] 0.002711497 0.001634654 0.004497720
```

You can explore the function `ci.mat()`, which lets you use matrix multiplication to produce confidence interval from an estimate and a standard error (or columns of such):

```
> ci.mat()
```

```
     Estimate      2.5%    97.5%
[1,]        1  1.000000 1.000000
[2,]        0 -1.959964 1.959964
```

```
> exp( c( log(D/Y), 1/sqrt(D) ) %*% ci.mat() )
```

```
         Estimate        2.5%       97.5%
[1,] 0.002711497 0.001634669 0.004497678
```

2. Try to achieve this using a Poisson model. Use the number of events as the respoins and the log-person-years as offset:

```
> mm <- glm( D ~ 1, offset=log(Y), family=poisson )
> summary( mm )

  Call:
  glm(formula = D ~ 1, family = poisson, offset = log(Y))

  Deviance Residuals:
  [1]  0

  Coefficients:
              Estimate Std. Error z value Pr(>|z|)
  (Intercept)  -5.9103     0.2582  -22.89   <2e-16

  (Dispersion parameter for poisson family taken to be 1)

      Null deviance: -8.8818e-16  on 0  degrees of freedom
  Residual deviance: -8.8818e-16  on 0  degrees of freedom
  AIC: 6.557

  Number of Fisher Scoring iterations: 3
```

   What is the interpretation of the parameter in this model?

3. You can extract a confidence interval directly from the model with the `ci.lin()`-function from `Epi`:

```
> ci.lin( mm )

               Estimate    StdErr         z P      2.5%     97.5%
  (Intercept) -5.910254 0.2581989 -22.89032 0 -6.416315 -5.404194

> ci.lin( mm, E=T)[,5:7]

    exp(Est.)        2.5%       97.5%
  0.002711497 0.001634669 0.004497678
```

4. There is an alternative way to fit a Poisson model, using the rates a the Poisson response, and the person-years as weights instead (albeit it will give you a warning about non-integer response in a Poisson model):

```
> mmx <- glm( D/Y ~ 1, weight=Y, family=poisson )
> ci.lin( mmx, E=T )[,5:7]

    exp(Est.)        2.5%       97.5%
  0.002711497 0.001634669 0.004497678
```

   Verify that this give the same results as above.

5. The advantage of this approach is that it will also make sense to use an identity link — the response is the same but the paramter estimated is now the rate, not the log-rate:

```
> ma <- glm( D/Y ~ 1, weight=Y, family=poisson(link=identity) )
```

What is the meaning of the intercept in this model?

Verify that you actually get the same rate estimate as before.

6. Now use `ci.lin` to produce the estimate and the confidence intervals from this model:

```
> ci.lin( ma )

                Estimate       StdErr        z             P        2.5%
(Intercept) 0.002711497 0.0007001054 3.872983 0.0001075112 0.001339315
                 97.5%
(Intercept) 0.004083678

> ci.lin( ma )[,c(1,5,6)]

    Estimate        2.5%       97.5%
0.002711497 0.001339315 0.004083678
```

Why are the confidence limits not the same as from the multiplicative model?

Derive the formula for the standard error of this estimated rate.

7. Now, suppose the events and person years are collected over three periods:

```
> Dx <- c(3,7,5)
> Yx <- c(1412,2783,1337)
> Px <- 1:3
```

Try to fit the same model as before to the data from the separate periods.

```
> m1 <- glm( Dx ~ 1, offset=log(Yx), family=poisson )
```

8. Now test wheter the are rates the same in the three periods: Try to fit a model with the period as a factor in the model:

```
> mp <- glm( Dx ~ factor(Px), offset=log(Yx), family=poisson )
```

and compare the two models using `anova` with the argument `test="Chisq"`:

```
> anova( m1, mp, test="Chisq" )

Analysis of Deviance Table

Model 1: Dx ~ 1
Model 2: Dx ~ factor(Px)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         2    0.70003
2         0    0.00000  2  0.70003   0.7047
```

Compare the test statistic to the deviance of the model `mp`.

9. Suppose instead that we had single observations of each year of follow-up, so that we for each of the 5532 years had an observation of (d, y) where d was either 1 (15 times) or 0 (5517 times), and all the intervals were of length 1:

```
> Dl <- rep( rep(0:1,3), c(1412-3,3,2783-7,7,1337-5,5) )
> Yl <- rep(1,5532)
> Pl <- rep( 1:3, c(1412,2783,1337) )
```

How long are the vectors `Dl`, `Yl` and `Pl`?

10. Now fit the same models as before

```
> mL <- glm( Dl ~ 1, offset=log(Yl), family=poisson )
> mP <- glm( Dl ~ factor(Pl), offset=log(Yl), family=poisson )
```

Compare the two models using `anova` and compare the test statistic to the (residual) deviance of the model `mL`.

What is the deviance good for?

11. If we have observations of two rates λ_1 and λ_0, based on (D_1, Y_1) and (D_0, Y_0) the variance of the difference of the log of the rates, that is the log(RR), is:

$$\begin{aligned} (\log(\text{RR})) &= (\log(\lambda_1/\lambda_0)) \\ &= (\log(\lambda_1)) + (\log(\lambda_0)) \\ &= 1/D_1 + 1/D_0 \end{aligned}$$

As before a 95% c.i. for the RR is then:

$$\text{RR} \overset{\times}{\div} \exp\left(1.96\sqrt{\frac{1}{D_1}+\frac{1}{D_0}}\right)$$

Suppose you have 15 events during 5532 person-years in an unexposed group and 28 events during 4783 person-years in an exposed group:

Compute the the rate-ratio and c.i. by:

```
> D0 <- 15   ; D1 <- 28
> Y0 <- 5532 ; Y1 <- 4783
> RR <- (D1/Y1)/(D0/Y0)
> erf <- exp( 1.96 * sqrt(1/D0+1/D1) )
> c( RR, RR/erf, RR*erf )

[1] 2.158980 1.153146 4.042153

> exp( c( log(RR), sqrt(1/D0+1/D1) ) %*% ci.mat() )

     Estimate    2.5%    97.5%
[1,]  2.15898 1.15316 4.042106
```

12. Now achieve this using a Poisson model:

```
> D <- c(D0,D1) ; Y <- c(Y0,Y1); xpos <- 0:1
> mm <- glm( D ~ factor(xpos), offset=log(Y), family=poisson )
```

What does the parameters mean in this model?

Ypu can extract the exponentiated parameters by:

```
> ci.lin( mm, E=T)[,5:7]
```

```
                  exp(Est.)        2.5%       97.5%
(Intercept)    0.002711497 0.001634669 0.004497678
factor(xpos)1 2.158979720 1.153159560 4.042106222
```

13. If we instead want the rate-difference, we just subtract the rates, and the variance of the diference is (since the rates are based on independent samples) just the sum of the variances:

$$\begin{aligned} (\log(\text{RD})) &= (\lambda_1) + (\lambda_0) \\ &= D_1/Y_1^2 + D_0/Y_0^2 \end{aligned}$$

Use this formula to compute the rate difference and a 95% confidence interval for it:

```
> rd <- diff( D/Y )
> sd <- sqrt( sum( D/Y^2 ) )
> c( rd, sd ) %*% ci.mat()
```

```
         Estimate         2.5%       97.5%
[1,] 0.00314257 0.0005765288 0.005708611
```

14. Verify that this is the confidence interval you get when you fit an additive model with exposure as factor:

```
> ma <- glm( D/Y ~ factor(xpos), weight=Y,
+                  family=poisson(link=identity) )
> ci.lin( ma )[,c(1,5,6)]
```

```
                  Estimate         2.5%       97.5%
(Intercept)    0.002711497 0.0013393153 0.004083678
factor(xpos)1 0.003142570 0.0005765288 0.005708611
```

15. Normally one would like to get both the rates and the difference between them. This can be achieved in one go using the `ctr.mat` argument to `ci.lin`. Try:

```
> CM <- rbind( c(1,0), c(1,1), c(0,1) )
> rownames( CM ) <- c("rate 0","rate 1","RR 1 vs. 0")
> CM
```

```
            [,1] [,2]
rate 0         1    0
rate 1         1    1
RR 1 vs. 0     0    1
```

```
> mm <- glm( D ~ factor(xpos),
+               offset=log(Y), family=poisson )
> ci.lin( mm, ctr.mat=CM, E=T)[,5:7]
```

```
              exp(Est.)        2.5%       97.5%
rate 0     0.002711497 0.001634669 0.004497678
rate 1     0.005854066 0.004041994 0.008478512
RR 1 vs. 0 2.158979720 1.153159560 4.042106222
```

```
> round( ci.lin( mm, ctr.mat=CM, E=T)[,5:7], 3 )
```

```
            exp(Est.)  2.5% 97.5%
rate 0          0.003 0.002 0.004
rate 1          0.006 0.004 0.008
RR 1 vs. 0      2.159 1.153 4.042
```

16. Refit the model with `Y/1000` as the person time, so you get the estimated rates in units of cases per 1000.

17. Use the same machinery to the additive model to get the rates and the rate-difference in one go. Note that the annotation of the resulting estimates are via the column-names of the contrast matrix.

```
> rownames( CM ) <- c("rate 0","rate 1","RD 1 vs. 0")
> ma <- glm( D/Y ~ factor(xpos), weight=Y,
+                  family=poisson(link=identity) )
> ci.lin( ma, ctr.mat=CM )[,c(1,5,6)]

               Estimate         2.5%       97.5%
rate 0     0.002711497 0.0013393153 0.004083678
rate 1     0.005854066 0.0036857298 0.008022403
RD 1 vs. 0 0.003142570 0.0005765288 0.005708611
```

3.2 Cox and Poisson modelling

This practical is to show how results from a Cox-model can be reproduced exactly by a Poisson model, and in particular how more sensible and relevant results can be obtained from a Poisson model.

3.2.1 The lung cancer data

The data is the lung cancer data from the `survival` package which comes with R by default. We start by declaring a really large chunk of memory, because we need that to fit a silly model for illustration:

```
memory.size( 3000 )
 [1] 3000
library( Epi )
library( survival )
sessionInfo()
 R version 3.1.1 (2014-07-10)
 Platform: i386-w64-mingw32/i386 (32-bit)

 locale:
 [1] LC_COLLATE=Danish_Denmark.1252  LC_CTYPE=Danish_Denmark.1252
 [3] LC_MONETARY=Danish_Denmark.1252 LC_NUMERIC=C
 [5] LC_TIME=Danish_Denmark.1252

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

 other attached packages:
 [1] survival_2.37-7 Epi_1.1.67      foreign_0.8-61
```

Note that loading the `survival` package automatically also loads the `splines` package, which is also needed in the exercise.

1. First we load the `lung` data set and have a look at it:

```
data( lung )
str( lung )
```

```
'data.frame':        228 obs. of  10 variables:
 $ inst     : num  3 3 3 5 1 12 7 11 1 7 ...
 $ time     : num  306 455 1010 210 883 ...
 $ status   : num  2 2 1 2 2 1 2 2 2 2 ...
 $ age      : num  74 68 56 57 60 74 68 71 53 61 ...
 $ sex      : num  1 1 1 1 1 1 2 2 1 1 ...
 $ ph.ecog  : num  1 0 0 1 0 1 2 2 1 2 ...
 $ ph.karno : num  90 90 90 90 100 50 70 60 70 70 ...
 $ pat.karno: num  100 90 90 60 90 80 60 80 80 70 ...
 $ meal.cal : num  1175 1225 NA 1150 NA ...
 $ wt.loss  : num  NA 15 15 11 0 0 10 1 16 34 ...
```

```
lung[1:10,]
```

```
   inst time status age sex ph.ecog ph.karno pat.karno meal.cal wt.loss
1     3  306      2  74   1       1       90       100     1175      NA
2     3  455      2  68   1       0       90        90     1225      15
3     3 1010      1  56   1       0       90        90       NA      15
4     5  210      2  57   1       1       90        60     1150      11
5     1  883      2  60   1       0      100        90       NA       0
6    12 1022      1  74   1       1       50        80      513       0
7     7  310      2  68   2       2       70        60      384      10
8    11  361      2  71   2       2       60        80      538       1
9     1  218      2  53   1       1       70        80      825      16
10    7  166      2  61   1       2       70        70      271      34
```

2. The deaths are indicated by `status` being equal to 2, so we tabulate the number of records with different values of `status`:

```
table( lung$status )
```

```
  1   2
 63 165
```

— so we see there are 165 deaths.

3. Some of the recorded survival times are identical we see:

```
addmargins( table( table( lung$time ) ) )
```

```
  1   2   3 Sum
146  38   2 186
```

In total there are 186 survival times.

3.2.2 Cox-models

4. Fitting a traditional Cox-model for the the Mayo clinic lung cancer data is done by `coxph`, where the response is a `Surv` object:

```
system.time(
m0.cox <- coxph( Surv( time, status==2 ) ~ age + factor( sex ),
                 method="breslow", eps=10^-8, iter.max=25, data=lung )
          )
```

```
   user  system elapsed
   0.02    0.00    0.01
```

```
summary( m0.cox )
```

```
Call:
coxph(formula = Surv(time, status == 2) ~ age + factor(sex),
    data = lung, method = "breslow", eps = 10^-8, iter.max = 25)

  n= 228, number of events= 165

                   coef exp(coef)  se(coef)      z Pr(>|z|)
age           0.017013  1.017158  0.009222  1.845  0.06506
factor(sex)2 -0.512565  0.598957  0.167462 -3.061  0.00221

             exp(coef) exp(-coef) lower .95 upper .95
age              1.017     0.9831    0.9989    1.0357
factor(sex)2     0.599     1.6696    0.4314    0.8316

Concordance= 0.603  (se = 0.026 )
Rsquare= 0.06   (max possible= 0.999 )
Likelihood ratio test= 14.08  on 2 df,   p=0.0008741
Wald test            = 13.44  on 2 df,   p=0.001208
Score (logrank) test = 13.69  on 2 df,   p=0.001067
```

5. Now we create a Lexis object from the dataset

```
Lung <- Lexis( exit = list( tfe=time ),
               exit.status = factor(status,labels=c("Alive","Dead")),
               data = lung )

 NOTE: entry.status has been set to "Alive" for all.
 NOTE: entry is assumed to be 0 on the tfe timescale.

summary( Lung )

 Transitions:
      To
 From    Alive Dead  Records:  Events: Risk time:  Persons:
   Alive    63  165       228      165      69593       228
```

6. We can fit the same Cox-model to data using the formal structures of the `Lexis` object, and we see we get the same estimates:

```
mL.cox <- coxph( Surv( tfe, tfe+lex.dur, lex.Xst=="Dead" ) ~
                 age + factor( sex ),
                 method="breslow", eps=10^-8, iter.max=25, data=Lung )
cbind( coef(m0.cox), coef(mL.cox) )

                     [,1]        [,2]
 age           0.01701289  0.01701289
 factor(sex)2 -0.51256479 -0.51256479
```

3.2.3 Poisson models

7. Now we split data split in small intervals, in fact at all recorded survival times, which mean that all events occur at the end of an interval:

```
Lung.s <- splitLexis( Lung,
                      breaks=c(0,sort(unique(Lung$time))),
                      time.scale="tfe" )
summary( Lung.s )
```

```
 Transitions:
      To
 From    Alive Dead  Records:  Events: Risk time:  Persons:
   Alive 19857  165     20022      165      69593       228
```

```
subset( Lung.s, lex.id==96 )
```

```
      lex.id tfe lex.dur lex.Cst lex.Xst inst time status age sex ph.ecog ph.karno
9235     96   0       5   Alive   Alive   12   30      2  72   1       2       80
9236     96   5       6   Alive   Alive   12   30      2  72   1       2       80
9237     96  11       1   Alive   Alive   12   30      2  72   1       2       80
9238     96  12       1   Alive   Alive   12   30      2  72   1       2       80
9239     96  13       2   Alive   Alive   12   30      2  72   1       2       80
9240     96  15      11   Alive   Alive   12   30      2  72   1       2       80
9241     96  26       4   Alive    Dead   12   30      2  72   1       2       80
      pat.karno meal.cal wt.loss
9235         60      288       7
9236         60      288       7
9237         60      288       7
9238         60      288       7
9239         60      288       7
9240         60      288       7
9241         60      288       7
```

8. We then fit the Cox model to the split dataset

```
system.time(
mLs.cox <- coxph( Surv( tfe, tfe+lex.dur, lex.Xst=="Dead" ) ~
                  age + factor( sex ),
                  method="breslow", eps=10^-8, iter.max=25, data=Lung.s )
           )
```

```
    user  system elapsed
    0.22    0.00    0.22
```

   ... and again we get exactly the same estimates

```
cbind( coef(m0.cox), coef(mL.cox), coef(mLs.cox) )
```

```
                      [,1]        [,2]        [,3]
age             0.01701289  0.01701289  0.01701289
factor(sex)2 -0.51256479 -0.51256479 -0.51256479
```

9. Then we fit a Poisson model with a factor accommodating the time-scale, in this case called `tfe`, which has exactly one level per recorded survival time:

```
nlevels( factor( Lung.s$tfe ) )
```

```
 [1] 186
```

   But it involves fitting a model with 186+2 parameters, so it takes some time, and requires quite some memory, hence the memory allocation at start. Note that the response variable `lex.Xst=="Dead"` is a logical, but by R converted into a 0/1 numeric:

```
system.time(
mLs.pois.fc <- glm( lex.Xst=="Dead" ~ factor( tfe ) +
                                 age + factor( sex ),
                                 offset = log(lex.dur),
                       family=poisson, data=Lung.s, eps=10^-8, maxit=25 )
           )
```

```
   user  system elapsed
  32.80    0.70   33.54
```

```
length( coef(mLs.pois.fc) )
```

```
[1] 188
```

So we have 188, parameters, but is only the last two that are of interest, and they are exactly the same as for the Cox-models:

```
rbind( coef( m0.cox), coef( mLs.pois.fc )[188-1:0] )
```

```
             age factor(sex)2
[1,] 0.01701289   -0.5125648
[2,] 0.01701289   -0.5125648
```

10. Hence the Cox model in reality is a model for the rates that no one in their sane mind would fit, we would of course want to fit a model where the baseline hazard were modelled using the actual values of the time-scale, and devising it as a continuous function of time.

    So we define internal and boundary knots for the spline basis and fit the model with natural splines for the baseline. Using 5 knots gives us a restricted cubic spline (natural spline) basis with 4 parameters, not counting the intercept. Note that we are using the wrapper `Ns` from the Epi package to avoid the hassle of specifying the boundary and internal knots separately.

```
t.kn <- c(0,25,100,500,1000)
dim( Ns(Lung.s$tfe,knots=t.kn) )
```

```
[1] 20022     4
```

As opposed to the model with 188 parameters, this model only has 7, so it is very quickly fitted:

```
system.time(
mLs.pois.sp <- glm( lex.Xst=="Dead" ~ Ns( tfe, knots=t.kn ) +
                                  age + factor( sex ),
                    offset = log(lex.dur),
                    family=poisson, data=Lung.s, eps=10^-8, maxit=25 )
             )
```

```
   user  system elapsed
   0.51    0.00    0.52
```

```
ci.exp( mLs.pois.sp )
```

```
                           exp(Est.)         2.5%        97.5%
(Intercept)           0.0005600982 0.0001311645  0.002391729
Ns(tfe, knots = t.kn)1 2.5751960590 0.9916627245  6.687389350
Ns(tfe, knots = t.kn)2 2.6355488430 0.8560015677  8.114608625
Ns(tfe, knots = t.kn)3 3.2029000448 0.4769285447 21.509655507
Ns(tfe, knots = t.kn)4 3.1689618387 0.6843090390 14.675122733
age                    1.0161894486 0.9980328610  1.034676348
factor(sex)2           0.5998287489 0.4319932401  0.832870736
```

```
ci.exp( mLs.pois.sp, subset=c("age","sex") )
```

```
              exp(Est.)      2.5%     97.5%
age           1.0161894 0.9980329 1.0346763
factor(sex)2  0.5998287 0.4319932 0.8328707
```

3.2.4 Comparing Cox and Poisson models

11. We can now compare the estimates of the regression parameters and their confidence intervals

```
ests <-
rbind( ci.exp(m0.cox),
       ci.exp(mLs.pois.fc,subset=c("age","sex")),
       ci.exp(mLs.pois.sp,subset=c("age","sex")) )
cmp <- cbind( ests[c(1,3,5)  ,],
              ests[c(1,3,5)+1,] )
rownames( cmp ) <- c("Cox","Poisson-factor","Poisson-spline")
colnames( cmp )[c(1,4)] <- c("age","sex")
round( cmp, 5 )
```

```
                     age    2.5%   97.5%     sex    2.5%   97.5%
Cox            1.01716 0.99894 1.03571 0.59896 0.43137 0.83165
Poisson-factor 1.01716 0.99894 1.03571 0.59896 0.43137 0.83165
Poisson-spline 1.01619 0.99803 1.03468 0.59983 0.43199 0.83287
```

We can also take a look at the estimated standard deviations of the log-RR:

```
round(
rbind( ci.lin(m0.cox)[,2],
       ci.lin(mLs.pois.fc,subset=c("age","sex"))[,2],
       ci.lin(mLs.pois.sp,subset=c("age","sex"))[,2] ), 6 )
```

```
          age factor(sex)2
[1,] 0.009222     0.167462
[2,] 0.009222     0.167462
[3,] 0.009199     0.167470
```

For all practical purposes they are the same too, so it is not so that the Cox-model or the factor-Poisson model inflates the s.e. of the regression estimates by estimating all the superfluous parameters.

12. We now use the parametrically estimated baseline intensity from the spline model to compute the estimated cumulative intensities over 100 10-day periods (0–1000 days after diagnosis) for men 60 year old at diagnosis, and then use these to compute the cumulative intensity since diagnosis and subsequently the survival function.

    Now, in order to get the predictions from the spline model we need to devise the right contrast matrix because we need the covariance between the point estimates for log-incidence rates.

    The model matrix, corresponding to times 0,10,20,...,1000:

```
CM <- cbind( 1, Ns( seq(0,1000,10), knots=t.kn ), 60, 1 )
```

    The mortality rates at these time points, for a 60-year old man are then:

```
lambda <- ci.exp( mLs.pois.sp, ctr.mat=CM )
```

    The cumulative mortality mortality rates (including the s.e.of this) are compute using `ci.cum`. Since this is a cumulative measure, we must explicitly supply the length of the intervals that each rate refer to, and for convenience we add

```
Lambda <- ci.cum( mLs.pois.sp, ctr.mat=CM, intl=10 )
Lambda <- rbind( 0, Lambda )
```

The Breslow-estimator of the survival curve for a male aged 60 Note that sex must be specified as a factor with two levels in the data frame in the argument `newdata`:

```
sf <-  survfit( m0.cox,
                newdata=data.frame( sex=factor(2,levels=1:2),
                                    age=c(60) ) )
```

13. We can then plot the mortality rates (`lambda`) and the survival function in two adjacent panels. Note that since we entered the risk time in days, the estimates of lambda we got out were rates *per day*, so we multiply them by 365.25 to the the mortality rates *per year* instead. Also note the we compute the survival function as $\exp(-\Lambda)$ on the fly, using the confidence intervals generated by `ci.cum` on the Λ-scale.

```
par( mfrow=c(1,2), mar=c(3,3,1,1), mgp=c(3,1,0)/1.6, oma=c(0,0,0,0),
     las=1, bty="n" )
matplot( 0:100*10, lambda * 365.25,
         type="l", lwd=c(4,1,1), lty=1, col="black", log="y",
         xlim=c(0,900), xaxs="i", ylim=c(1/10,5),
         xlab="Days since diagnosis",
         ylab="Mortality rate per year")
# Then the survival curves by the two methods
# Here is the Breslow-estimator; note
plot( sf, lwd=c(4,1,1), col="red", conf.int=T, mark.time=F,
          xlab="Days since diagnosis",
          ylab="Survival", xlim=c(0,900), xaxs="i", lty=1)
matlines( 0:101*10, exp(-Lambda[,1:3]), lwd=c(4,1,1), col="black", lty=1 )
```

3.3 Fitting a smooth curve

1. For illustration we fit a very crude model to the mortality follow-up of the men in the Danish Diabetes register:

```
library( Epi )
library( splines )
data( DMlate )
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
```

2. Now define outcome and age and date of diagnosis for convenience, and restrict data to only men:

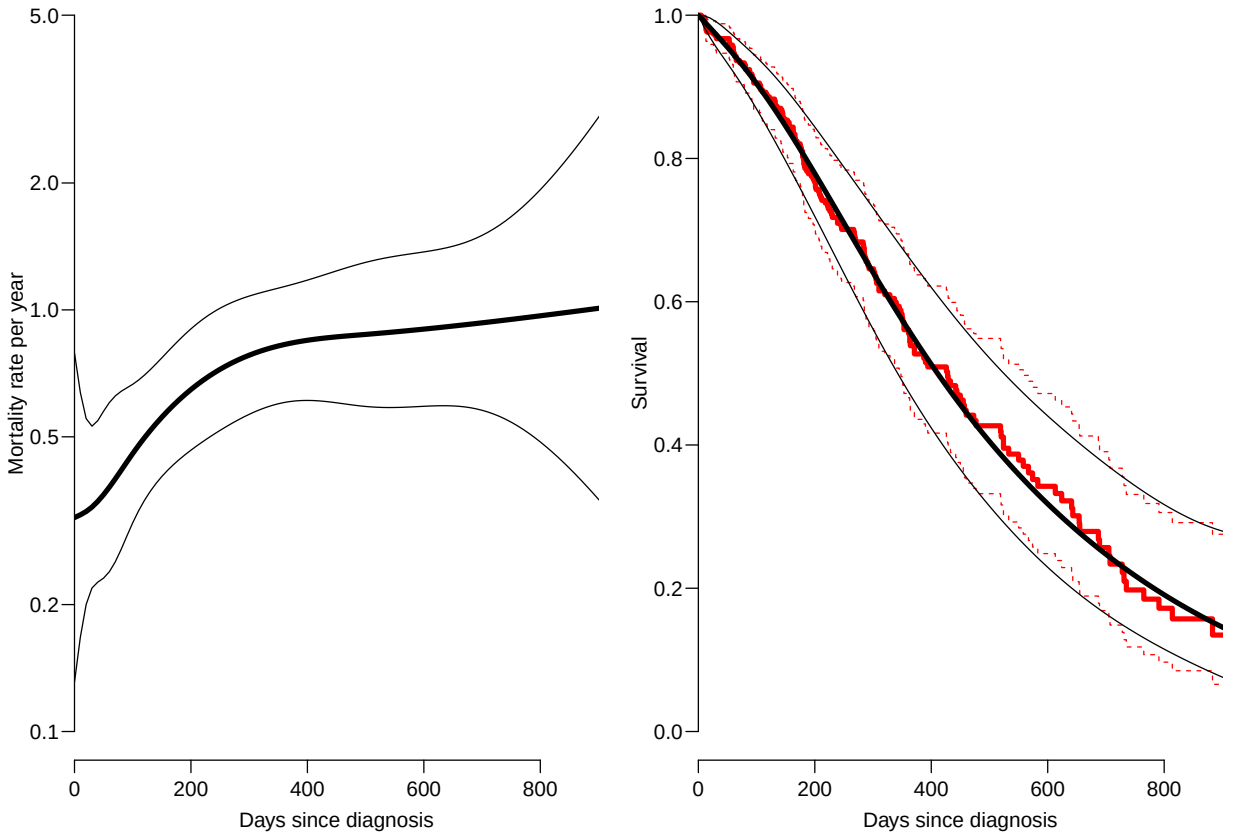

Figure 3.1: *Left: Hazard function for 60-year old men from spline model with 95% c.i. Right: Survival curve for 60 year old men; black from spline model, red from Cox-model.*

```
DMlate <- transform( DMlate, D = !is.na(dodth),
                             Y = dox-dodm,
                             A = dodm-dobth,
                             P = dodm )
DMlate <- subset( DMlate, Y>0 & sex=="M" )
str( DMlate )
```

```
 'data.frame':        5183 obs. of  11 variables:
  $ sex  : Factor w/ 2 levels "M","F": 1 1 1 1 1 1 1 1 1 1 ...
  $ dobth: num  1939 1933 1946 1940 1933 ...
  $ dodm : num  2003 2009 2007 2007 2010 ...
  $ dodth: num  NA NA NA NA NA ...
  $ dooad: num  2007 NA 2007 2007 NA ...
  $ doins: num  NA NA NA NA NA ...
  $ dox  : num  2010 2010 2010 2010 2010 ...
  $ D    : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
  $ Y    : num  6.689 1.344 2.782 3.324 0.214 ...
  $ A    : num  64.1 75.8 60.7 66.2 76.6 ...
  $ P    : num  2003 2009 2007 2007 2010 ...
```

3. Now fit a model for mortality only depending on age and date at entry:

```
m0 <- glm( D ~ A + P, family=poisson, offset=log(Y), data=DMlate )
```

— and use `ci.lin` and `ci.exp` to explore the parameters:

```
ci.lin( m0 )
```

```
                Estimate       StdErr         z             P        2.5%        97.5%
(Intercept) 86.82024444 14.940861088  5.810926  6.212807e-09 57.53669481 116.10379407
A            0.07666534  0.002413784 31.761476 2.204730e-221  0.07193441   0.08139627
P           -0.04736773  0.007475160 -6.336684  2.347625e-10 -0.06201878  -0.03271669
```

```
round( ci.exp( m0 )[-1,], 3 )
```

```
  exp(Est.)  2.5% 97.5%
A     1.080 1.075 1.085
P     0.954 0.940 0.968
```

How much is mortality changing by age and how much by time?

4. Now try to add a quadratic term in date of diagnosis to the model:

```
mq <- glm( D ~ A + P + I(P^2), family=poisson, offset=log(Y), data=DMlate )
round( ci.lin( mq ), 3 )
```

```
              Estimate   StdErr      z     P     2.5%     97.5%
(Intercept) 16425.926 7831.092  2.098 0.036 1077.268 31774.585
A               0.077    0.002 31.733 0.000    0.072     0.081
P             -16.375    7.826 -2.093 0.036  -31.714    -1.037
I(P^2)          0.004    0.002  2.086 0.037    0.000     0.008
```

What is the interpretation of the coefficients now (if any)?

5. Now try to make a graph of the RR of death relative to 2005, say: For a given time `P` the log-rate is

$$\mu + \alpha \mathtt{A} + \beta_1 \mathtt{P} + beta_2 \mathtt{P}^2$$

and for 2005 the log-rate is:

$$\mu + \alpha \mathtt{A} + \beta_1 2005 + beta_2 2005^2$$

so the log-RR between these is:

$$\beta_1 \mathtt{P} + beta_2 \mathtt{P}^2 - \beta_1 2005 - beta_2 2005^2 = \beta_1(\mathtt{P} - 2005) + \beta_2(\mathtt{P}^2 - 2005^2)$$

Now devise some points between 1995 and 2010 and construct the two columns of numbers to be multiplied by the two parameters:

```
P.pt <- 1995:2010
p1 <- P.pt - 2005
p2 <- P.pt^2 - 2005^2
coef( mq )
```

```
  (Intercept)             A             P        I(P^2)
1.642593e+04  7.660107e-02 -1.637542e+01  4.079240e-03
```

```
lRR <- coef(mq)[3] * p1 + coef(mq)[4] * p2
 RR <- exp( lRR )
plot( P.pt, RR, type="l", lwd=3, log="y" )
abline( h=1,v=2005)
```

What is the substantial conclusion from the shape of the RR relative to 2005?

6. Then we draw the RR relative to year 2000

```
P.pt <- 1995:2010
p1 <- P.pt - 2000
p2 <- P.pt^2 - 2000^2
coef( mq )
```

```
   (Intercept)             A             P        I(P^2)
  1.642593e+04  7.660107e-02 -1.637542e+01  4.079240e-03
```

```
lRR <- coef(mq)[3] * p1 + coef(mq)[4] * p2
 RR <- exp( lRR )
plot( P.pt, RR, type="l", lwd=3, log="y" )
abline( h=1,v=2000)
```

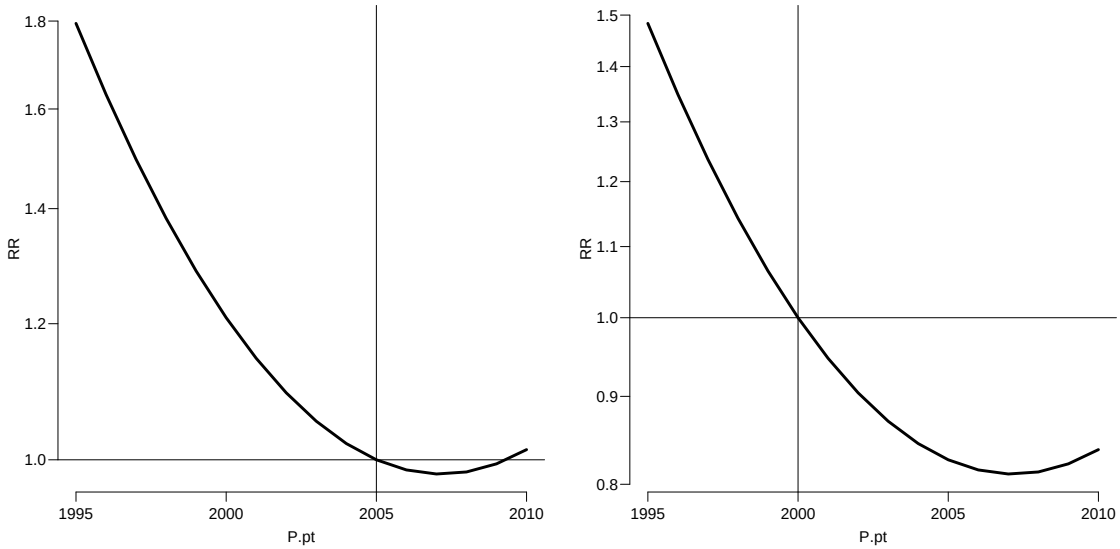

Figure 3.2: *RR relative to 2005 and 2000, respectively. The curves are identical except for the y-axis.*

7. These curves do not have confidence limits; but there is a facility in the functions `ci.lin` and `ci.exp`, that will both select the relevant parameters (in this case those with names `P` and `I(P^2)`):

```
ci.lin( mq, subset="P" )
```

```
             Estimate      StdErr         z          P          2.5%       97.5%
P       -16.37542296 7.825707708 -2.092517 0.03639233 -3.171353e+01 -1.03731770
I(P^2)    0.00407924 0.001955077  2.086486 0.03693467  2.473593e-04  0.00791112
```

```
ci.exp( mq, subset="P" )
```

```
            exp(Est.)         2.5%    97.5%
P       7.731151e-08 1.686513e-14 0.354404
I(P^2) 1.004088e+00 1.000247e+00 1.007942
```

There is a further argument to these functions, `ctr.mat` — a so called contrast matrix which is the columns of numbers we defined above to multiply by each of the parameters:

```
cbind( p1, p2 )
```

```
       p1     p2
 [1,] -5 -19975
 [2,] -4 -15984
 [3,] -3 -11991
 [4,] -2  -7996
 [5,] -1  -3999
 [6,]  0      0
 [7,]  1   4001
 [8,]  2   8004
 [9,]  3  12009
[10,]  4  16016
[11,]  5  20025
[12,]  6  24036
[13,]  7  28049
[14,]  8  32064
[15,]  9  36081
[16,] 10  40100
```

```
ci.exp( mq, subset="P", ctr.mat=cbind(p1,p2) )
```

```
       exp(Est.)      2.5%     97.5%
 [1,] 1.4833518 1.2578707 1.7492519
 [2,] 1.3486802 1.1982418 1.5180060
 [3,] 1.2362803 1.1425703 1.3376762
 [4,] 1.1425313 1.0908379 1.1966744
 [5,] 1.0645412 1.0431817 1.0863380
 [6,] 1.0000000 1.0000000 1.0000000
 [7,] 0.9470670 0.9322492 0.9621203
 [8,] 0.9042835 0.8783501 0.9309826
 [9,] 0.8705058 0.8339919 0.9086184
[10,] 0.8448545 0.7955757 0.8971858
[11,] 0.8266761 0.7606876 0.8983891
[12,] 0.8155151 0.7280788 0.9134517
[13,] 0.8110952 0.6972184 0.9435715
[14,] 0.8133076 0.6678925 0.9903827
[15,] 0.8222066 0.6400008 1.0562858
[16,] 0.8380122 0.6134772 1.1447279
```

8. The result is the estimated curve with confidence intervals, so we can now plot it (use the function `matplot`):

```
matplot( P.pt, ci.exp( mq, subset="P", ctr.mat=cbind(p1,p2) ),
         type="l", col="Black", lty=1, lwd=c(3,1,1), log="y" )
abline( h=1, v=2000 )
```

9. What really goes on here is that we as contrast matrix use the difference between the matrix of `P` and $\texttt{P}^2$, and the matrix that consists of the 2000-row all way through:

```
( MP <- cbind( P.pt, P.pt^2 ) )
```

```
       P.pt
 [1,] 1995 3980025
 [2,] 1996 3984016
 [3,] 1997 3988009
 [4,] 1998 3992004
 [5,] 1999 3996001
 [6,] 2000 4000000
 [7,] 2001 4004001
 [8,] 2002 4008004
 [9,] 2003 4012009
[10,] 2004 4016016
```

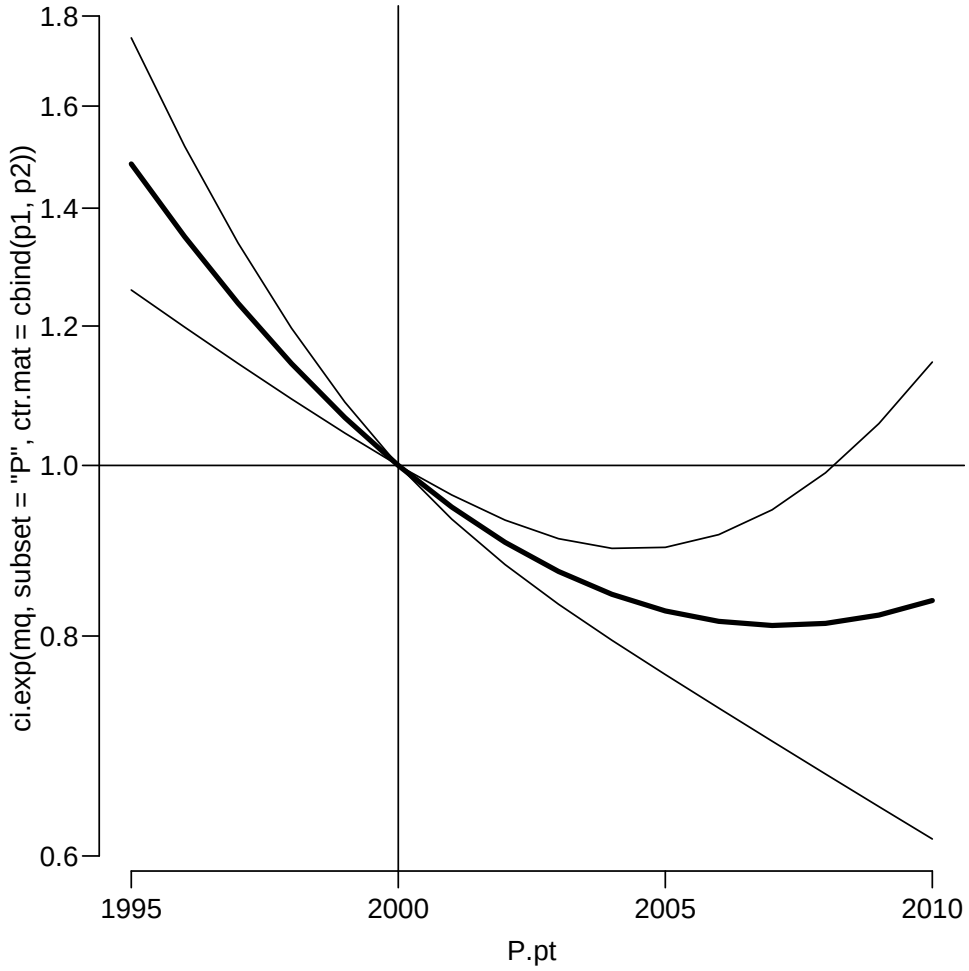

Figure 3.3: *RR by period with confidence intervals*

```
[11,] 2005 4020025
[12,] 2006 4024036
[13,] 2007 4028049
[14,] 2008 4032064
[15,] 2009 4036081
[16,] 2010 4040100
```

```
( Mr <- cbind( rep(2000,length(P.pt)), 2000^2 ) )
```

```
      [,1]  [,2]
 [1,] 2000 4e+06
 [2,] 2000 4e+06
 [3,] 2000 4e+06
 [4,] 2000 4e+06
 [5,] 2000 4e+06
 [6,] 2000 4e+06
 [7,] 2000 4e+06
 [8,] 2000 4e+06
 [9,] 2000 4e+06
[10,] 2000 4e+06
[11,] 2000 4e+06
[12,] 2000 4e+06
[13,] 2000 4e+06
[14,] 2000 4e+06
[15,] 2000 4e+06
[16,] 2000 4e+06
```

```
MP-Mr
```

```
      P.pt
 [1,]   -5 -19975
 [2,]   -4 -15984
 [3,]   -3 -11991
 [4,]   -2  -7996
 [5,]   -1  -3999
 [6,]    0      0
 [7,]    1   4001
 [8,]    2   8004
 [9,]    3  12009
[10,]    4  16016
[11,]    5  20025
[12,]    6  24036
[13,]    7  28049
[14,]    8  32064
[15,]    9  36081
[16,]   10  40100
```

```
ci.exp( mq, subset="P", ctr.mat=MP-Mr )
```

```
      exp(Est.)      2.5%     97.5%
 [1,] 1.4833518 1.2578707 1.7492519
 [2,] 1.3486802 1.1982418 1.5180060
 [3,] 1.2362803 1.1425703 1.3376762
 [4,] 1.1425313 1.0908379 1.1966744
 [5,] 1.0645412 1.0431817 1.0863380
 [6,] 1.0000000 1.0000000 1.0000000
 [7,] 0.9470670 0.9322492 0.9621203
 [8,] 0.9042835 0.8783501 0.9309826
 [9,] 0.8705058 0.8339919 0.9086184
[10,] 0.8448545 0.7955757 0.8971858
[11,] 0.8266761 0.7606876 0.8983891
[12,] 0.8155151 0.7280788 0.9134517
[13,] 0.8110952 0.6972184 0.9435715
[14,] 0.8133076 0.6678925 0.9903827
[15,] 0.8222066 0.6400008 1.0562858
[16,] 0.8380122 0.6134772 1.1447279
```

```
matplot( P.pt, ci.exp( mq, subset="P", ctr.mat=MP-Mr ),
         type="l", col="Black", lty=1, lwd=c(3,1,1), log="y" )
abline( h=1,v=2000)
```

10. If we want to model the period effect by a `cubic spline` instead. This is a function that is a cubic between a set of points called *knots*. In the `Epi` package is a function `Ns`, that will generate a set of columns corresponding to this — think of it as the counterpart of the columns `P` and $\texttt{P}^2$:

```
p.kn <- c(1997,2000,2003,2006,2009)
Ns( P.pt, knots=p.kn )
```

```
                1           2          3           4
 [1,] 0.00000000  0.16903085 -0.5070926  0.33806170
 [2,] 0.00000000  0.08451543 -0.2535463  0.16903085
 [3,] 0.00000000  0.00000000  0.0000000  0.00000000
 [4,] 0.00617284 -0.08138522  0.2441557 -0.16277045
 [5,] 0.04938272 -0.14398924  0.4319677 -0.28797849
 [6,] 0.16666667 -0.16903085  0.5070926 -0.33806170
 [7,] 0.37037037 -0.13982242  0.4379858 -0.29199052
 [8,] 0.57407407 -0.04805065  0.2923001 -0.19486673
 [9,] 0.66666667  0.11250419  0.1624874 -0.10832495
[10,] 0.57407407  0.33051271  0.1195730 -0.07354249
```

```
[11,] 0.37037037  0.52444901  0.1488752 -0.04986741
[12,] 0.16666667  0.59523810  0.2142857  0.02380952
[13,] 0.04938272  0.47266314  0.2857143  0.19223986
[14,] 0.00617284  0.20194004  0.3571429  0.43474427
[15,] 0.00000000 -0.14285714  0.4285714  0.71428571
[16,] 0.00000000 -0.50000000  0.5000000  1.00000000
attr(,"degree")
[1] 3
attr(,"knots")
[1] 2000 2003 2006
attr(,"Boundary.knots")
[1] 1997 2009
attr(,"intercept")
[1] FALSE
attr(,"class")
[1] "ns"     "basis"  "matrix"
```

When we put this in a model, we get parameters that do not have any immediate meaning:

```
ms <- glm( D ~ A + Ns(P,knots=p.kn), family=poisson, offset=log(Y), data=DMlate )
ci.exp( ms )
```

```
                          exp(Est.)         2.5%        97.5%
(Intercept)           0.000430476 0.0003072398 0.0006031433
A                     1.079690213 1.0745850602 1.0848196186
Ns(P, knots = p.kn)1 0.752806894 0.5788409351 0.9790569131
Ns(P, knots = p.kn)2 0.629990707 0.4771921832 0.8317158257
Ns(P, knots = p.kn)3 0.530294086 0.4108359422 0.6844868924
Ns(P, knots = p.kn)4 0.830236504 0.5967627706 1.1550530398
```

```
ci.exp( ms, subset="P" )
```

```
                       exp(Est.)      2.5%     97.5%
Ns(P, knots = p.kn)1 0.7528069 0.5788409 0.9790569
Ns(P, knots = p.kn)2 0.6299907 0.4771922 0.8317158
Ns(P, knots = p.kn)3 0.5302941 0.4108359 0.6844869
Ns(P, knots = p.kn)4 0.8302365 0.5967628 1.1550530
```

11. But we extract the RR relative to 2005 from this new model as before, so we first construct the matrix for the points 1995—2010 and then for the reference for 2005:

```
CP <- Ns( P.pt, knots=p.kn )
Cr <- Ns( rep(2005,length(P.pt)), knots=p.kn )
CP-Cr
```

```
                1           2           3           4
  [1,] -0.3703704 -0.35541816 -0.65596775  0.38792912
  [2,] -0.3703704 -0.43993358 -0.40242147  0.21889827
  [3,] -0.3703704 -0.52444901 -0.14887520  0.04986741
  [4,] -0.3641975 -0.60583423  0.09528048 -0.11290303
  [5,] -0.3209877 -0.66843825  0.28309253 -0.23811107
  [6,] -0.2037037 -0.69347986  0.35821736 -0.28819429
  [7,]  0.0000000 -0.66427143  0.28911059 -0.24212311
  [8,]  0.2037037 -0.57249966  0.14342490 -0.14499932
  [9,]  0.2962963 -0.41194482  0.01361223 -0.05845753
 [10,]  0.2037037 -0.19393630 -0.02930220 -0.02367508
 [11,]  0.0000000  0.00000000  0.00000000  0.00000000
 [12,] -0.2037037  0.07078909  0.06541052  0.07367694
 [13,] -0.3209877 -0.05178587  0.13683909  0.24210727
 [14,] -0.3641975 -0.32250897  0.20826766  0.48461168
 [15,] -0.3703704 -0.66730615  0.27969623  0.76415313
 [16,] -0.3703704 -1.02444901  0.35112480  1.04986741
```

```
attr(,"degree")
[1] 3
attr(,"knots")
[1] 2000 2003 2006
attr(,"Boundary.knots")
[1] 1997 2009
attr(,"intercept")
[1] FALSE
attr(,"class")
[1] "ns"     "basis"  "matrix"
```

Then we can extract the RR ad plot it:

```
RR <- ci.exp( ms, subset="P", ctr.mat=CP-Cr )
matplot( P.pt, RR,
         type="l", col="Black", lty=1, lwd=c(3,1,1), log="y" )
abline( h=1,v=2000)
```

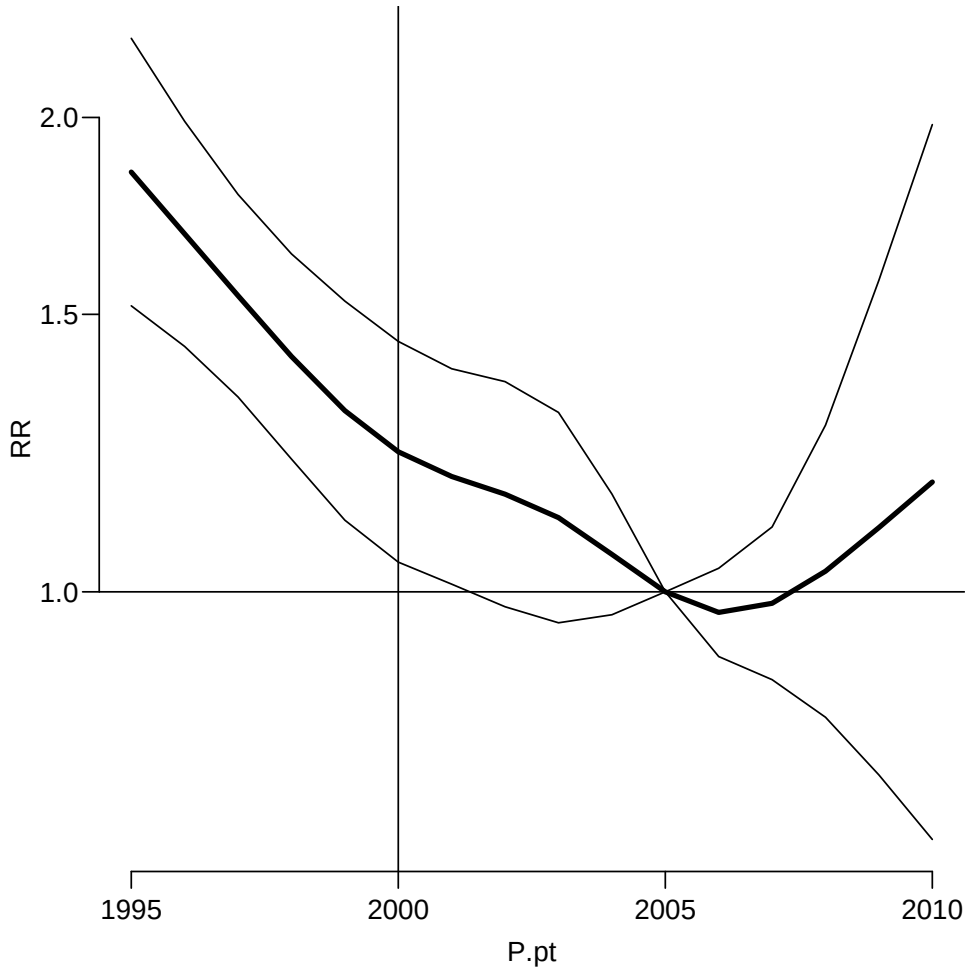

Figure 3.4: *RR by calendar time, modelled by a spline*

12. Now we check if is any non-linearity by age (on the log-mortality scale, that is):

```
ma <- update( ms, . ~ . + I(A^2) )
round( ci.lin( ma ), 3 )
```

```
                      Estimate StdErr       z     P   2.5%  97.5%
(Intercept)             -7.504  0.650 -11.548 0.000 -8.778 -6.231
A                        0.069  0.019   3.574 0.000  0.031  0.107
Ns(P, knots = p.kn)1    -0.284  0.134  -2.117 0.034 -0.547 -0.021
Ns(P, knots = p.kn)2    -0.463  0.142  -3.265 0.001 -0.741 -0.185
Ns(P, knots = p.kn)3    -0.637  0.130  -4.884 0.000 -0.892 -0.381
Ns(P, knots = p.kn)4    -0.188  0.169  -1.115 0.265 -0.518  0.142
I(A^2)                   0.000  0.000   0.392 0.695  0.000  0.000
```

We see from the confidence interval there there is no evidence of non-linearity.

13. In order to report the model `ms` in full, we must show the estimated mortality rates as a function of age. For that purpose we of course must use the 2005 reference point.

    (a) The first possibility is to devise a prediction data frame:

    ```
    nd <- data.frame( A = 40:85, P=2005, Y=1000 )
    ```

    Note that you must provide values for *all* covariates, including the person-years, that enter in the model as offset, that is as a covariate with fixed regression coefficient. The function `ci.pred` is a simple convenience wrapper for `predict.glm`:

    ```
    rate <- ci.pred( ma, newdata=nd )
    matplot( nd$A, rate,
             type="l", col="Black", lty=1, lwd=c(3,1,1), log="y" )
    ```

    Note that since you initially entered `Y` in units of 1 person-year, we get the rates in units of events per 1000 person-years by entering Y with the value of 1000 in the prediction frame.

    (b) The other possibility is to use `ci.exp` directly to extract the predicted rates from the model, but they will be in he units of the `Y` entered into the model. Note that you in this case must be careful to get the order of the columns in `ctr.mat` right:

    ```
    ci.exp( ma )
    ```

    ```
                            exp(Est.)         2.5%       97.5%
     (Intercept)          0.000550656 0.0001540717 0.001968058
     A                    1.071590052 1.0317148908 1.113006364
     Ns(P, knots = p.kn)1 0.752908898 0.5789206188 0.979187458
     Ns(P, knots = p.kn)2 0.629530651 0.4768295353 0.831133164
     Ns(P, knots = p.kn)3 0.528963174 0.4096715676 0.682991112
     Ns(P, knots = p.kn)4 0.828645308 0.5955280996 1.153015360
     I(A^2)               1.000055922 0.9997763008 1.000335621
    ```

    ```
    Rate <- ci.exp( ma, ctr.mat=cbind(1,40:85,Ns(rep(2005,46),knots=p.kn),(40:85)^2) )*1000
    matplot( nd$A, Rate,
             type="l", col="Black", lty=1, lwd=c(3,1,1), log="y" )
    ```

14. Finally we explore if there are further non-linearities in the age-effect by including a spline with many knots

    ```
    ( a.kn <- 3:9*10 )
    ```

    ```
     [1] 30 40 50 60 70 80 90
    ```

    ```
    mA <- update( ms, . ~ . - A + Ns(A,knots=a.kn) )
    round( ci.lin( mA ), 3 )
    ```

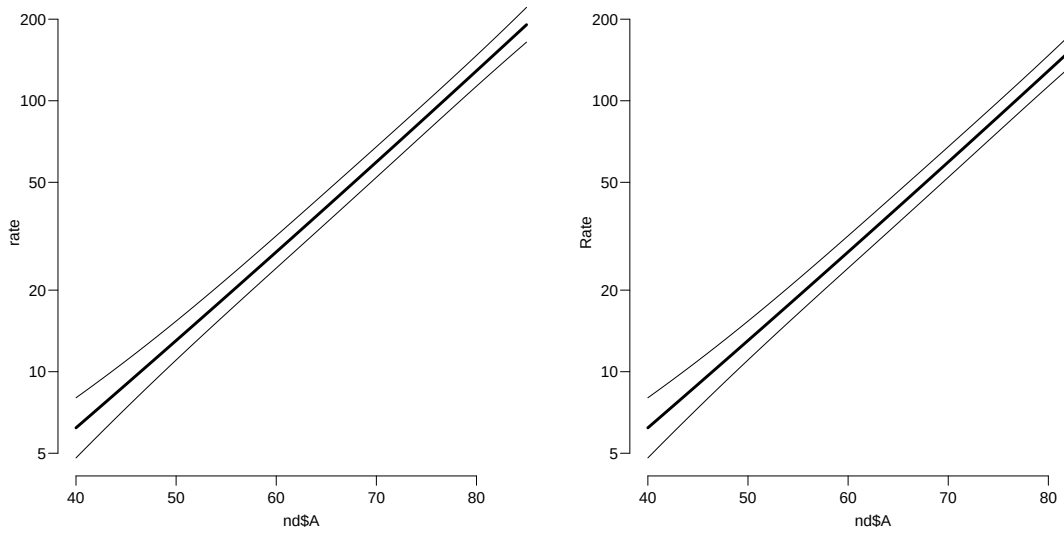

Figure 3.5: *Estimated age-specific mortality on 2005, by the two different approaches.*

```
                     Estimate StdErr       z     P   2.5%  97.5%
(Intercept)            -5.439  0.305 -17.855 0.000 -6.036 -4.842
Ns(P, knots = p.kn)1   -0.275  0.134  -2.051 0.040 -0.538 -0.012
Ns(P, knots = p.kn)2   -0.460  0.142  -3.242 0.001 -0.738 -0.182
Ns(P, knots = p.kn)3   -0.624  0.131  -4.780 0.000 -0.880 -0.368
Ns(P, knots = p.kn)4   -0.178  0.169  -1.053 0.292 -0.508  0.153
Ns(A, knots = a.kn)1    1.758  0.385   4.564 0.000  1.003  2.513
Ns(A, knots = a.kn)2    2.092  0.325   6.432 0.000  1.454  2.729
Ns(A, knots = a.kn)3    3.103  0.328   9.452 0.000  2.460  3.747
Ns(A, knots = a.kn)4    3.667  0.267  13.749 0.000  3.144  4.190
Ns(A, knots = a.kn)5    4.879  0.573   8.517 0.000  3.756  6.002
Ns(A, knots = a.kn)6    4.171  0.265  15.748 0.000  3.652  4.690
```

We see that 7 knots produces a curve with 6 parameters.

```
rate <- ci.pred( mA, newdata=nd )
matplot( nd$A, rate,
         type="l", col="Black", lty=1, lwd=c(3,1,1), log="y" )
```

3.4 Simple analysis of Estonian stroke data

```
> library(Epi)
```

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

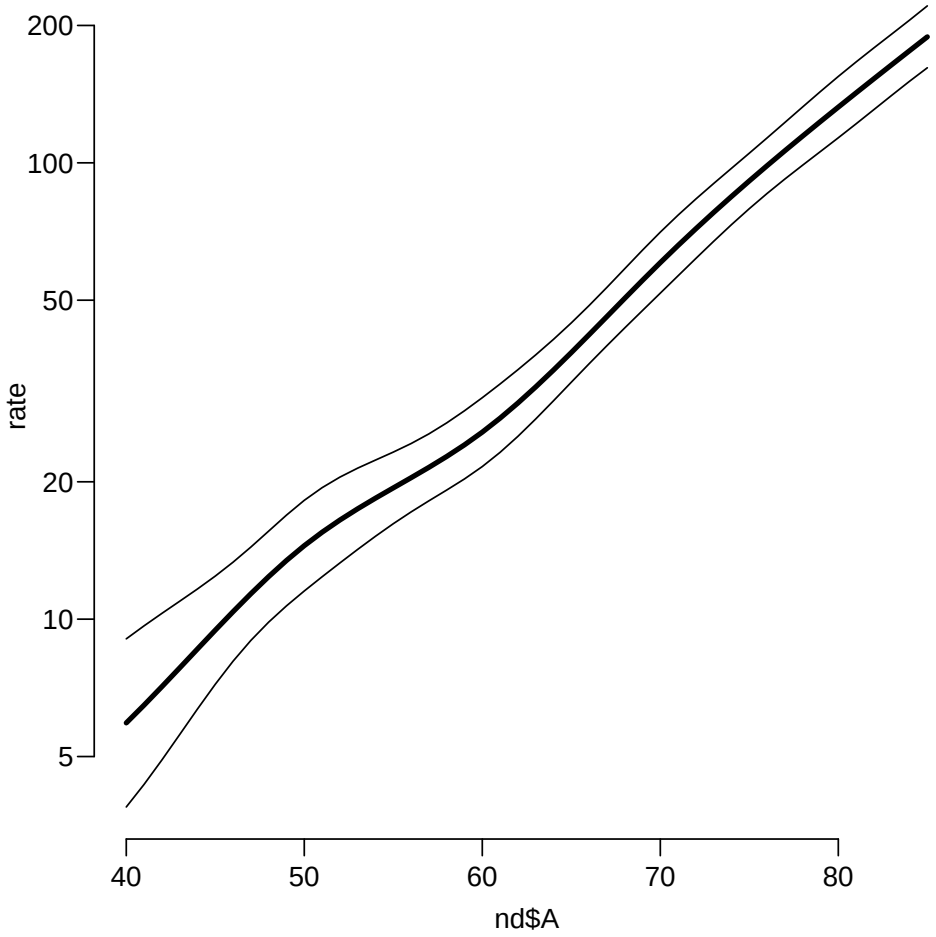

Figure 3.6: *Age-specific mortality among Danish DM men in 2005, using a 6-parameter natural spline.*

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

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

1. First, read in the data using the `read.table()` or `read.csv()` command. Do not forget to *look into the file before* to see, what the field separator is.

   Calculate an id variable in the dataframe.

```
> stroke <- read.table( url("http://BendixCarstensen.com/AdvCoh/Scot-2014/data/stroke.csv"),
+                       sep=";", header=TRUE, na.strings="." )
> stroke$id <- 1:nrow(stroke)
> str( stroke )

  'data.frame':        829 obs. of  10 variables:
   $ sex : int  1 1 1 0 0 1 0 1 0 0 ...
   $ died: Factor w/ 414 levels "1.01.1993","1.02.1992",..: 373 NA 162 60 213 60 16 57 NA 27 .
   $ dstr: Factor w/ 575 levels "1.01.1992","1.01.1993",..: 229 414 542 45 84 84 105 105 124 1
   $ age : int  76 58 74 77 76 48 81 53 73 69 ...
   $ dgn : Factor w/ 4 levels "ICH","ID","INF",..: 3 3 3 1 3 1 3 3 2 3 ...
   $ coma: int  0 0 0 0 0 1 0 0 0 0 ...
   $ diab: int  0 0 0 1 1 0 0 0 0 0 ...
   $ minf: int  1 0 1 0 0 0 0 1 0 0 ...
   $ han : int  0 0 1 1 1 1 1 1 1 1 ...
   $ id  : int  1 2 3 4 5 6 7 8 9 10 ...

> head( stroke )

    sex       died       dstr age dgn coma diab minf han id
1   1  7.01.1991  2.01.1991  76 INF    0    0    1   0  1
2   1       <NA>  3.01.1991  58 INF    0    0    0   0  2
3   1  2.06.1991  8.01.1991  74 INF    0    0    1   1  3
4   0 13.01.1991 11.01.1991  77 ICH    0    1    0   1  4
5   0 23.01.1996 13.01.1991  76 INF    0    1    0   1  5
6   1 13.01.1991 13.01.1991  48 ICH    1    0    0   1  6
```

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

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

  'data.frame':        829 obs. of  10 variables:
   $ sex : int  1 1 1 0 0 1 0 1 0 0 ...
   $ died: Date, format: "1991-01-07" NA ...
   $ dstr: Date, format: "1991-01-02" "1991-01-03" ...
   $ age : int  76 58 74 77 76 48 81 53 73 69 ...
   $ dgn : Factor w/ 4 levels "ICH","ID","INF",..: 3 3 3 1 3 1 3 3 2 3 ...
   $ coma: int  0 0 0 0 0 1 0 0 0 0 ...
   $ diab: int  0 0 0 1 1 0 0 0 0 0 ...
   $ minf: int  1 0 1 0 0 0 0 1 0 0 ...
   $ han : int  0 0 1 1 1 1 1 1 1 1 ...
   $ id  : int  1 2 3 4 5 6 7 8 9 10 ...

> stroke <- cal.yr(stroke)
```

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

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

   How many deaths occurred after 1 January 1996?

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

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

     sex     died     dstr age dgn coma diab minf han  id  dox
5      0 1996.059 1991.032  76 INF    0    1    0   1   5 1996
97     1 1996.037 1991.369  58 INF    0    0    1   1  97 1996
320    1 1996.141 1992.283  59 INF    0    1    0   1 320 1996
360    0 1996.021 1992.396  68 INF    0    0    0   1 360 1996
401    0 1996.026 1992.557  74 INF    0    0    0   1 401 1996
666    0 1996.062 1993.463  70  ID    0    0    0   0 666 1996

> with( stroke, table( died>1996 ) )

  FALSE  TRUE
    485     6

> stroke <- transform( stroke, D = ( dox < 1996 ) )
```

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

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

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

  Call: survfit(formula = sst)

  records   n.max n.start  events  median 0.95LCL 0.95UCL
   829.00  829.00  829.00  485.00    1.68    1.12    2.23

> plot( survfit( sst ) )
```

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

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

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

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

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

```
> with( stroke, table( dgn, D ) )
```

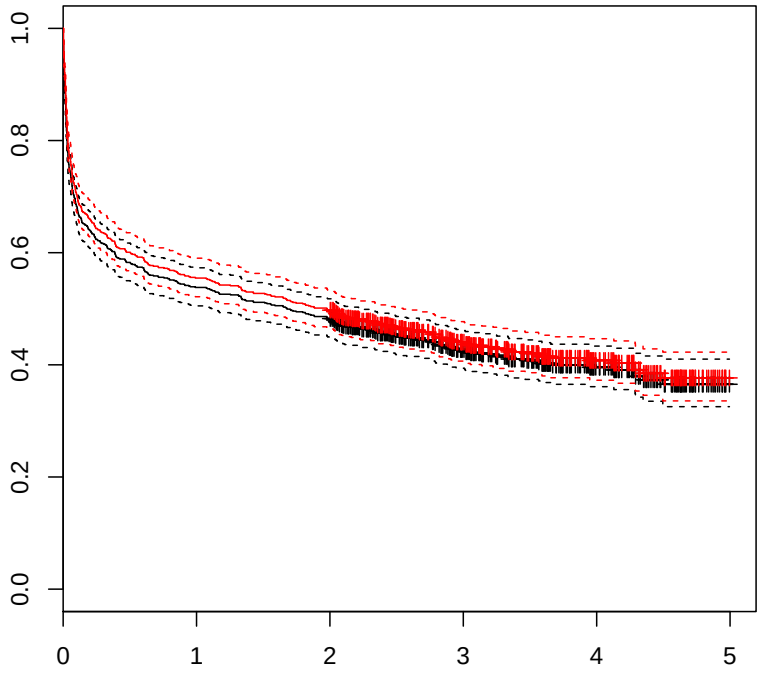

Figure 3.7: *The Kaplan-Meier estimator with (black) and without (red) the 0-survivors, i.e. the persons who die at the same time as their stroke.*

```
       D
dgn    FALSE TRUE
  ICH     25   45
  ID      54  140
  INF    239  257
  SAH     26   18

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

  Call: survfit(formula = Surv(dox - dstr, D) ~ dgn, data = stroke)

          records n.max n.start events median 0.95LCL 0.95UCL
dgn=ICH      70    70      70     45 0.0671  0.0329      NA
dgn=ID      194   194     194    140 0.2231  0.1040    0.86
dgn=INF     496   496     496    257 3.0527  2.3053    4.29
dgn=SAH      44    44      44     18     NA  0.6489      NA
```

8. Plot the result as 4 curves.

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

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

Do the hazards look proportional?

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

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

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

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

Call:
survdiff(formula = Surv(dox - dstr, D) ~ sex, data = stroke)

        N Observed Expected (O-E)^2/E (O-E)^2/V
sex=0 497      308      264      7.16        17
sex=1 307      152      196      9.69        17

 Chisq= 17  on 1 degrees of freedom, p= 3.76e-05
```

What do you conclude?

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

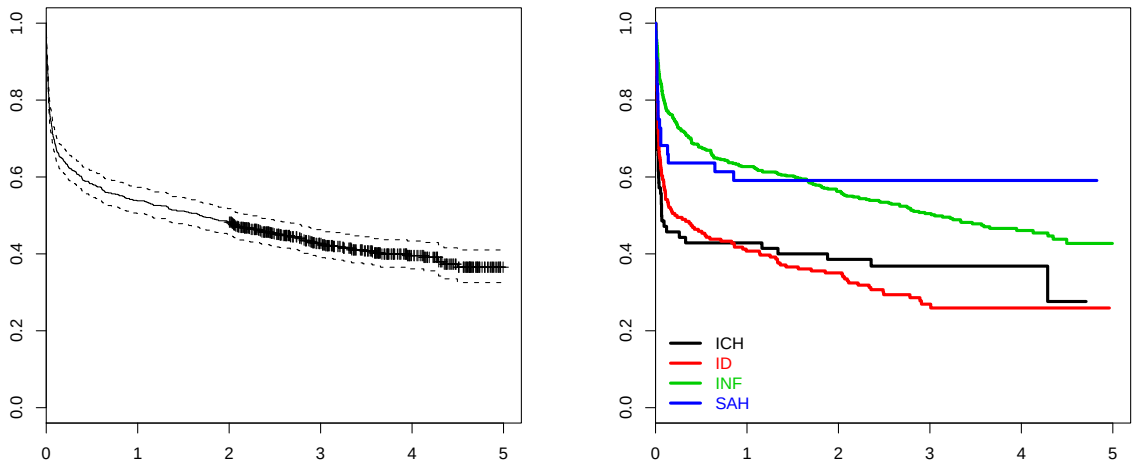

Figure 3.8: *Kaplan-Meier plot for the Estonian stroke data, overall and subdivided by diagnosis.*

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

```
> Lst <- Lexis( data=stroke, entry=list(Per=dstr,Age=age,Tfs=dstr-dstr),
+                            exit=list(Per=dox),
+                    exit.status=as.numeric(stroke$D) )
> head( Lst )
```

```
        Per Age Tfs     lex.dur lex.Cst lex.Xst lex.id sex     died     dstr age
1 1991.002  76   0 0.013689254       0       1      1   1 1991.016 1991.002  76
2 1991.005  58   0 4.995208761       0       0      2   1       NA 1991.005  58
3 1991.018  74   0 0.396988364       0       1      3   1 1991.415 1991.018  74
4 1991.027  77   0 0.005475702       0       1      4   0 1991.032 1991.027  77
5 1991.032  76   0 4.967830253       0       0      5   0 1996.059 1991.032  76
7 1991.035  81   0 2.880219028       0       1      6   0 1993.915 1991.035  81
  dgn coma diab minf han id      dox     D
1 INF    0    0    1   0  1 1991.016  TRUE
2 INF    0    0    0   0  2 1996.000 FALSE
3 INF    0    0    1   1  3 1991.415  TRUE
4 ICH    0    1    0   1  4 1991.032  TRUE
5 INF    0    1    0   1  5 1996.000 FALSE
7 INF    0    0    0   1  7 1993.915  TRUE
```

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

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

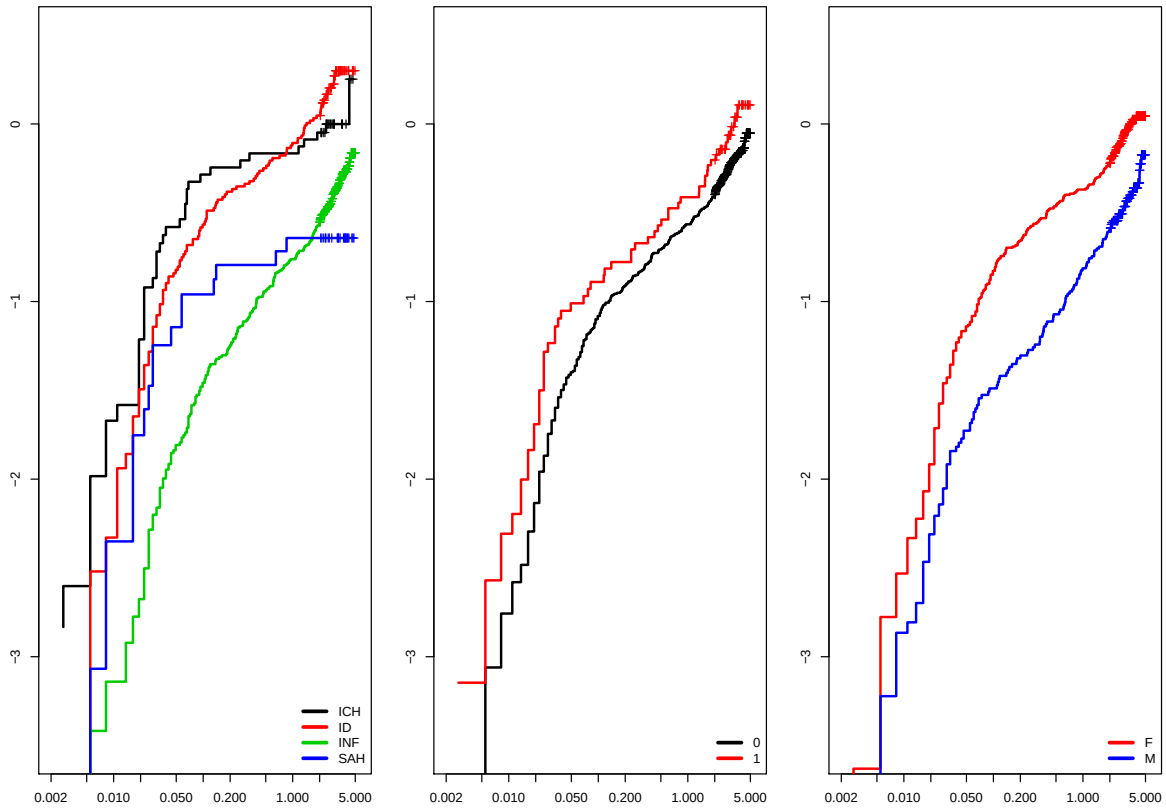

Figure 3.9: *Log-cumulative hazards by diagnosis, diabetes status and sex, respectively, for the Estonian stroke data.*

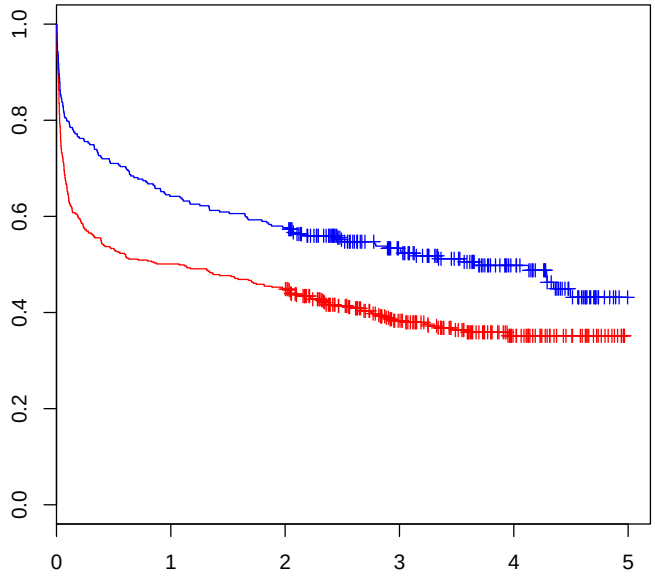

Figure 3.10: *Kaplan-Meier plot for the Estonian stroke data, subdivided by sex.*

```
> summary( Lst )

  Transitions:
       To
  From   0   1  Records:  Events: Risk time:  Persons:
     0 344 460       804      460    1433.08       804
```

12. Get an overview of how the number of deaths and person years is distributed by time:

```
> plot( Lst )
```

Try to enhance the Lexis diagram by using the graphical arguments to `plot.Lexis` and `points.Lexis`. By default, `plot.Lexis` makes a plot using the first two timescales of the `Lexis` object. So it matters in which order the timescales are defined.

Below you see the necessary graphical formatting necessary to get squares in the Lexis diagram, i.e. the same physical scale on both axes: `mai=` gives the margins on the four sides of the plot in inches, a total of 1 inch in each direction. Thus, the `height=10+1,width=3+1` gives a plot area of 3 by 10 inches, accommodating a 30 year period (horizoantal) and a 100 year age-span (vertical). You probably want to use another path name for the file, though.

```
> pdf( "../graph/stroke1-LexisX.pdf", height=10+1, width=3+1 )
> par( mai=c(3,3,1,1)/4, mgp=c(3,1,0)/1.6 )
> plot(Lst,xlim=1980+c(0,30),ylim=c(0,100),
+          col=c("red","blue")[Lst$sex+1],grid=0:20*5,xaxs="i",yaxs="i")
```

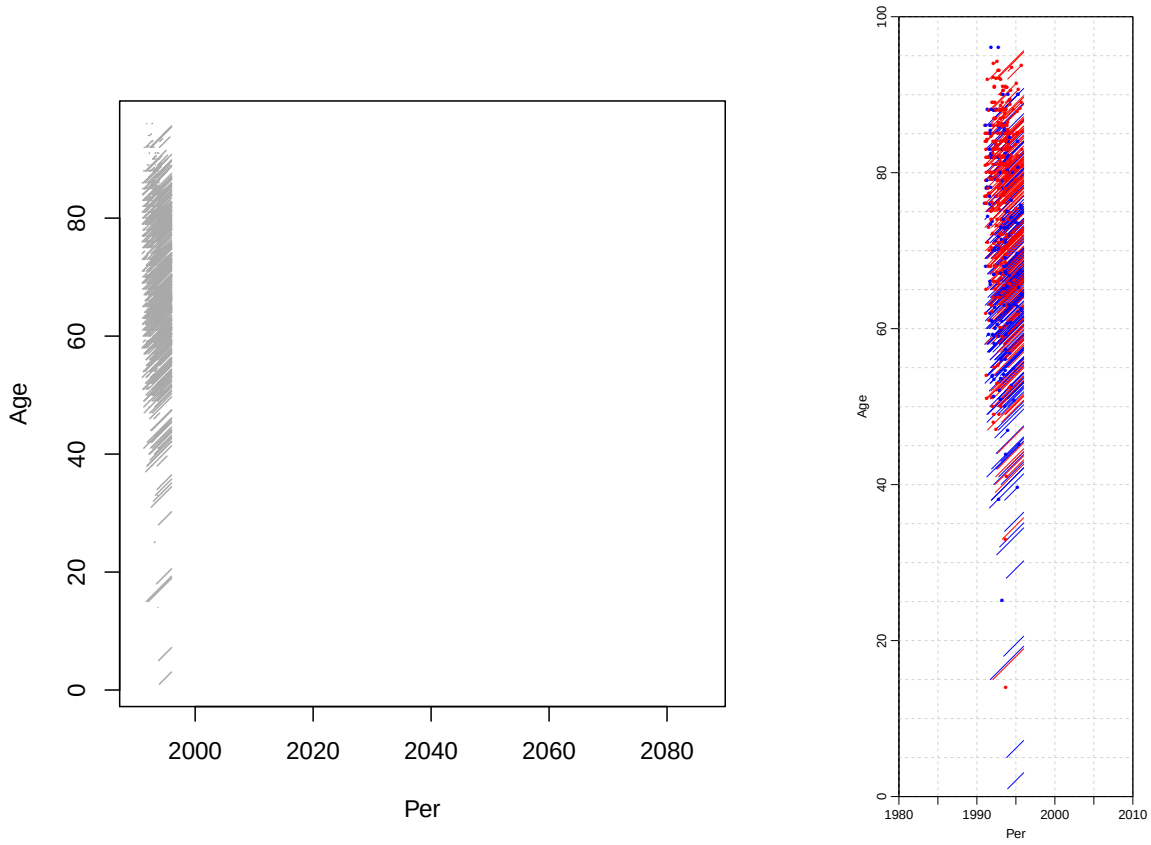

Figure 3.11: *Lexis diagram for the Estonian stroke data. Left panel is the default lay-out, the right-hand panel is the result of adjusting the plotsize etc. as shown above. Red: Women, Blue: Men.*

```
> points( subset(Lst,lex.Xst==TRUE),pch=16,cex=0.6,
+         col=c("red","blue")[Lst$sex+1][Lst$lex.Xst==TRUE])
> dev.off()
```

Clearly, from the enhanced figure with colouring of life-lines by sex, it is immediately apparent that women are much older than men. This may be one explanation of the higher mortality among women as seen in figure 3.10.

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

```
> with( stroke, survfit( Surv( dox-dstr, D ) ~ sex ) )
  Call: survfit(formula = Surv(dox - dstr, D) ~ sex)

        records n.max n.start events median 0.95LCL 0.95UCL
sex=0     497   497     497    308   1.07   0.402    1.88
sex=1     307   307     307    152   3.66   2.489      NA

> with( Lst, survfit( Surv( lex.dur, lex.Xst ) ~ sex ) )

  Call: survfit(formula = Surv(lex.dur, lex.Xst) ~ sex)
```

```
      records n.max n.start events median 0.95LCL 0.95UCL
sex=0     497   497     497    308   1.07   0.402    1.88
sex=1     307   307     307    152   3.66   2.489      NA
```

14. What is the time-scale we are using here?

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

```
> save( stroke, Lst, file="../data/from-exc-stroke1.Rdata" )
```

3.5 Cox model and time-splitting using Estonian stroke data

```
> library(Epi)
```

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

```
> load( file="../data/from-exc-stroke1.Rdata" )
> str( Lst )
  Classes 'Lexis' and 'data.frame':        804 obs. of  19 variables:
   $ Per    : num  1991 1991 1991 1991 1991 ...
   $ Age    : int  76 58 74 77 76 81 53 73 69 86 ...
   $ Tfs    : num  0 0 0 0 0 0 0 0 0 0 ...
   $ lex.dur: num  0.01369 4.99521 0.39699 0.00548 4.96783 ...
   $ lex.Cst: num  0 0 0 0 0 0 0 0 0 0 ...
   $ lex.Xst: num  1 0 1 1 0 1 1 0 1 1 ...
   $ lex.id : int  1 2 3 4 5 6 7 8 9 10 ...
   $ sex    : int  1 1 1 0 0 0 1 0 0 0 ...
   $ died   : num  1991 NA 1991 1991 1996 ...
   $ dstr   : num  1991 1991 1991 1991 1991 ...
   $ age    : int  76 58 74 77 76 81 53 73 69 86 ...
   $ dgn    : Factor w/ 4 levels "ICH","ID","INF",..: 3 3 3 1 3 3 3 2 3 2 ...
   $ coma   : int  0 0 0 0 0 0 0 0 0 0 ...
   $ diab   : int  0 0 0 1 1 0 0 0 0 0 ...
   $ minf   : int  1 0 1 0 0 0 1 0 0 0 ...
   $ han    : int  0 0 1 1 1 1 1 1 1 0 ...
   $ id     : int  1 2 3 4 5 7 8 9 10 11 ...
   $ dox    : num  1991 1996 1991 1991 1996 ...
   $ D      : logi  TRUE FALSE TRUE TRUE FALSE TRUE ...
   - attr(*, "time.scales")= chr  "Per" "Age" "Tfs"
   - attr(*, "breaks")=List of 3
    ..$ Per: NULL
    ..$ Age: NULL
    ..$ Tfs: NULL
```

Alternatively you must read the data afresh, transform etc.

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

```
>   library( survival )
>   mc <- coxph( Surv(dox-dstr,D) ~ sex, data=stroke )
>   summary( mc )
```

```
  Call:
  coxph(formula = Surv(dox - dstr, D) ~ sex, data = stroke)

    n= 804

          coef exp(coef) se(coef)      z Pr(>|z|)
  sex -0.4068    0.6657   0.0993 -4.097 4.18e-05

      exp(coef) exp(-coef) lower .95 upper .95
  sex    0.6658      1.502     0.548    0.8088

  Rsquare= 0.021   (max possible= 0.999 )
  Likelihood ratio test= 17.47  on 1 df,   p=2.923e-05
  Wald test            = 16.79  on 1 df,   p=4.185e-05
  Score (logrank) test = 17.02  on 1 df,   p=3.707e-05
```

```
>   mL <- coxph( Surv(lex.dur,lex.Xst==1) ~ sex, data=Lst )
>   summary( mL )
```

```
  Call:
  coxph(formula = Surv(lex.dur, lex.Xst == 1) ~ sex, data = Lst)

    n= 804

          coef exp(coef) se(coef)      z Pr(>|z|)
  sex -0.4068    0.6657   0.0993 -4.097 4.18e-05

      exp(coef) exp(-coef) lower .95 upper .95
  sex    0.6658      1.502     0.548    0.8088

  Rsquare= 0.021   (max possible= 0.999 )
  Likelihood ratio test= 17.47  on 1 df,   p=2.923e-05
  Wald test            = 16.79  on 1 df,   p=4.185e-05
  Score (logrank) test = 17.02  on 1 df,   p=3.707e-05
```

Are there any differences?

What is the underlying time scale used here?

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

```
>   mLa <- coxph( Surv(lex.dur,lex.Xst==1) ~ sex + age, data=Lst )
>   summary( mLa )
```

```
  Call:
  coxph(formula = Surv(lex.dur, lex.Xst == 1) ~ sex + age, data = Lst)

    n= 804

          coef exp(coef) se(coef)      z Pr(>|z|)
  sex 0.015706  1.015830 0.103890  0.151     0.88
  age 0.054237  1.055735 0.004574 11.857   <2e-16

      exp(coef) exp(-coef) lower .95 upper .95
  sex     1.016     0.9844    0.8287     1.245
  age     1.056     0.9472    1.0463     1.065

  Rsquare= 0.199   (max possible= 0.999 )
  Likelihood ratio test= 178.2  on 2 df,   p=0
  Wald test            = 152.8  on 2 df,   p=0
  Score (logrank) test = 151.1  on 2 df,   p=0
```

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

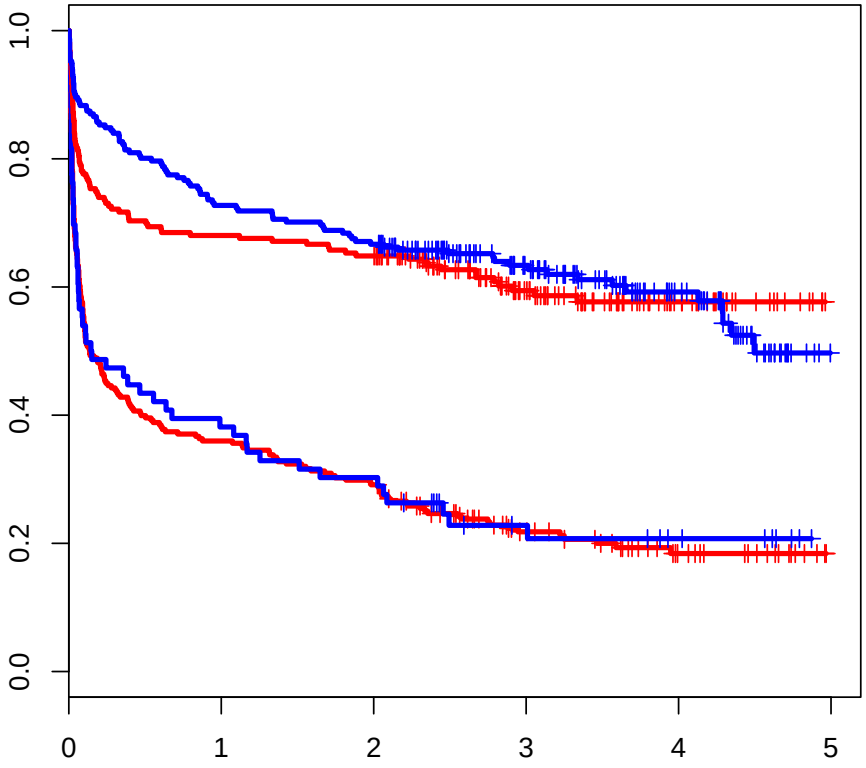

Figure 3.12: *Kaplan-Meier curves for males (blue) and females (red) over and under 75 years at stroke.*

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

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

```
> plot( survfit( Surv(lex.dur,lex.Xst==1) ~ interaction(sex,age<75),
+                data=Lst ),
+                col=c("red","blue"), lwd=3 )
```

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

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

  0.FALSE 1.FALSE  0.TRUE  1.TRUE
      278      76     219     231
```

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

```
> length( unique(Lst$lex.dur[Lst$lex.Xst==1]) )

  [1] 257

> sLst <- splitLexis( Lst, breaks=seq(0,10,0.05), "Tfs" )
> str( sLst )

  Classes 'Lexis' and 'data.frame':        29077 obs. of  19 variables:
   $ lex.id : int  1 2 2 2 2 2 2 2 2 2 ...
   $ Per    : num  1991 1991 1991 1991 1991 ...
   $ Age    : num  76 58 58 58.1 58.1 ...
   $ Tfs    : num  0 0 0.05 0.1 0.15 0.2 0.25 0.3 0.35 0.4 ...
   $ lex.dur: num  0.0137 0.05 0.05 0.05 0.05 ...
   $ lex.Cst: num  0 0 0 0 0 0 0 0 0 0 ...
   $ lex.Xst: num  1 0 0 0 0 0 0 0 0 0 ...
   $ sex    : int  1 1 1 1 1 1 1 1 1 1 ...
   $ died   : num  1991 NA NA NA NA ...
   $ dstr   : num  1991 1991 1991 1991 1991 ...
   $ age    : int  76 58 58 58 58 58 58 58 58 58 ...
   $ dgn    : Factor w/ 4 levels "ICH","ID","INF",..: 3 3 3 3 3 3 3 3 3 3 ...
   $ coma   : int  0 0 0 0 0 0 0 0 0 0 ...
   $ diab   : int  0 0 0 0 0 0 0 0 0 0 ...
   $ minf   : int  1 0 0 0 0 0 0 0 0 0 ...
   $ han    : int  0 0 0 0 0 0 0 0 0 0 ...
   $ id     : int  1 2 2 2 2 2 2 2 2 2 ...
   $ dox    : num  1991 1996 1996 1996 1996 ...
   $ D      : logi  TRUE FALSE FALSE FALSE FALSE FALSE ...
   - attr(*, "breaks")=List of 3
    ..$ Per: NULL
    ..$ Age: NULL
    ..$ Tfs: num  0 0.05 0.1 0.15 0.2 0.25 0.3 0.35 0.4 0.45 ...
   - attr(*, "time.scales")= chr  "Per" "Age" "Tfs"

> summary( Lst )

  Transitions:
       To
  From   0   1  Records:  Events: Risk time:  Persons:
     0 344 460       804      460    1433.08       804

  Rates:
       To
  From 0    1 Total
     0 0 0.32  0.32

> summary( sLst )

  Transitions:
       To
  From     0   1  Records:  Events: Risk time:  Persons:
     0 28617 460     29077      460    1433.08       804

  Rates:
       To
  From 0    1 Total
     0 0 0.32  0.32
```

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

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

```
        Per Age Tfs    lex.dur lex.Cst lex.Xst lex.id sex     died     dstr age
56 1991.194  78   0 0.01916496       0       1     54   1 1991.213 1991.194  78
57 1991.199  59   0 0.33127995       0       1     55   1 1991.530 1991.199  59
   dgn coma diab minf han id      dox    D
56 ICH    0    1    0   1 56 1991.213 TRUE
57 INF    0    0    1   0 57 1991.530 TRUE
```

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

```
     lex.id      Per   Age  Tfs    lex.dur lex.Cst lex.Xst sex     died
2176     54 1991.194 78.00 0.00 0.01916496       0       1   1 1991.213
2177     55 1991.199 59.00 0.00 0.05000000       0       0   1 1991.530
2178     55 1991.249 59.05 0.05 0.05000000       0       0   1 1991.530
2179     55 1991.299 59.10 0.10 0.05000000       0       0   1 1991.530
2180     55 1991.349 59.15 0.15 0.05000000       0       0   1 1991.530
2181     55 1991.399 59.20 0.20 0.05000000       0       0   1 1991.530
2182     55 1991.449 59.25 0.25 0.05000000       0       0   1 1991.530
2183     55 1991.499 59.30 0.30 0.03127995       0       1   1 1991.530
         dstr age dgn coma diab minf han id      dox    D
2176 1991.194  78 ICH    0    1    0   1 56 1991.213 TRUE
2177 1991.199  59 INF    0    0    1   0 57 1991.530 TRUE
2178 1991.199  59 INF    0    0    1   0 57 1991.530 TRUE
2179 1991.199  59 INF    0    0    1   0 57 1991.530 TRUE
2180 1991.199  59 INF    0    0    1   0 57 1991.530 TRUE
2181 1991.199  59 INF    0    0    1   0 57 1991.530 TRUE
2182 1991.199  59 INF    0    0    1   0 57 1991.530 TRUE
2183 1991.199  59 INF    0    0    1   0 57 1991.530 TRUE
```

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

```
> mCs <- coxph( Surv(lex.dur,lex.Xst==1) ~ sex + age, data=Lst )
> ci.lin( mLa )
```

```
       Estimate      StdErr          z        P        2.5%      97.5%
sex 0.01570586 0.103889538  0.1511785 0.879835 -0.18791389 0.21932561
age 0.05423706 0.004574175 11.8572352 0.000000  0.04527185 0.06320228
```

```
> mC <- coxph( Surv(Tfs,Tfs+lex.dur,lex.Xst==1) ~ sex + age, data=sLst )
> ci.lin( mC )
```

```
       Estimate      StdErr          z        P        2.5%      97.5%
sex 0.01570586 0.103889538  0.1511785 0.879835 -0.18791389 0.21932561
age 0.05423706 0.004574175 11.8572352 0.000000  0.04527185 0.06320228
```

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

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

```
    user  system elapsed
   48.39    2.53   51.50
```

8. Now take a look at the estimated coefficients:

```
> coef( mP )
```

```
     (Intercept) factor(Tfs)0.05  factor(Tfs)0.1 factor(Tfs)0.15  factor(Tfs)0.2
      -2.33898954     -1.04190384     -1.65953566     -2.44267852     -2.24653143
 factor(Tfs)0.25  factor(Tfs)0.3 factor(Tfs)0.35  factor(Tfs)0.4 factor(Tfs)0.45
      -2.99051916     -2.68145284     -2.25680685     -3.61670944     -3.09182829
  factor(Tfs)0.5 factor(Tfs)0.55  factor(Tfs)0.6 factor(Tfs)0.65  factor(Tfs)0.7
      -3.07821999     -3.57342153     -2.25177037     -4.63597641     -3.93494442
 factor(Tfs)0.75  factor(Tfs)0.8 factor(Tfs)0.85  factor(Tfs)0.9 factor(Tfs)0.95
      -3.23760466     -3.92530127     -3.22030299     -3.50284809     -3.90421293
    factor(Tfs)1 factor(Tfs)1.05  factor(Tfs)1.1 factor(Tfs)1.15  factor(Tfs)1.2
     -18.74484978     -3.49086905     -3.19075222     -3.46026563    -18.72845570
 factor(Tfs)1.25  factor(Tfs)1.3 factor(Tfs)1.35  factor(Tfs)1.4 factor(Tfs)1.45
      -4.54922037     -2.59333871     -3.82725879     -3.81966931    -18.71103532
  factor(Tfs)1.5 factor(Tfs)1.55  factor(Tfs)1.6 factor(Tfs)1.65  factor(Tfs)1.7
      -3.81129598     -3.39675079     -4.48986574     -3.09318146     -3.08026337
 factor(Tfs)1.75  factor(Tfs)1.8 factor(Tfs)1.85  factor(Tfs)1.9 factor(Tfs)1.95
      -4.45614154     -3.34937707     -3.34287803    -18.69131342     -3.33387116
    factor(Tfs)2 factor(Tfs)2.05  factor(Tfs)2.1 factor(Tfs)2.15  factor(Tfs)2.2
      -2.80060206     -2.75136824     -4.32858427     -4.31093280     -3.18207131
 factor(Tfs)2.25  factor(Tfs)2.3 factor(Tfs)2.35  factor(Tfs)2.4 factor(Tfs)2.45
     -18.65461606     -3.13200176     -3.50319161     -4.16157697     -3.02282066
  factor(Tfs)2.5 factor(Tfs)2.55  factor(Tfs)2.6 factor(Tfs)2.65  factor(Tfs)2.7
      -4.09948386     -4.05861662    -18.61712500     -3.29893651     -3.96422762
 factor(Tfs)2.75  factor(Tfs)2.8 factor(Tfs)2.85  factor(Tfs)2.9 factor(Tfs)2.95
      -2.55270360     -3.91261103     -3.18531360     -3.12961854    -18.56079998
    factor(Tfs)3 factor(Tfs)3.05  factor(Tfs)3.1 factor(Tfs)3.15  factor(Tfs)3.2
      -3.04323576     -3.69222552     -3.67216797    -18.61976106     -2.94103977
 factor(Tfs)3.25  factor(Tfs)3.3 factor(Tfs)3.35  factor(Tfs)3.4 factor(Tfs)3.45
     -18.58778256     -2.89227470    -18.60429587    -18.66085200     -3.47936219
  factor(Tfs)3.5 factor(Tfs)3.55  factor(Tfs)3.6 factor(Tfs)3.65  factor(Tfs)3.7
     -18.62634710     -2.73474221    -18.59394905     -3.28074948    -18.63166864
 factor(Tfs)3.75  factor(Tfs)3.8 factor(Tfs)3.85  factor(Tfs)3.9 factor(Tfs)3.95
     -18.63733648    -18.60361765    -18.62290042     -3.07086878    -18.58610042
    factor(Tfs)4 factor(Tfs)4.05  factor(Tfs)4.1 factor(Tfs)4.15  factor(Tfs)4.2
     -18.59468456    -18.57463979     -2.79915041    -18.55780014    -18.61094521
 factor(Tfs)4.25  factor(Tfs)4.3 factor(Tfs)4.35  factor(Tfs)4.4 factor(Tfs)4.45
      -1.93750315     -2.56191854    -18.60516937    -18.69962444     -2.37552049
  factor(Tfs)4.5 factor(Tfs)4.55  factor(Tfs)4.6 factor(Tfs)4.65  factor(Tfs)4.7
     -18.67912914    -18.67085081    -18.63527714    -18.63930697    -18.64786291
 factor(Tfs)4.75  factor(Tfs)4.8 factor(Tfs)4.85  factor(Tfs)4.9 factor(Tfs)4.95
     -18.79673800    -18.71641558    -18.53409351    -18.44635192    -17.73185419
             sex             age
      0.01404002      0.05471934
```

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

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

       Estimate     StdErr          z         P exp(Est.)      2.5%    97.5%
sex 0.01404002 0.10387588  0.1351615 0.8924842  1.014139 0.8273289 1.243131
age 0.05471934 0.00457603 11.9578186 0.0000000  1.056244 1.0468132 1.065760

> ci.lin( mC, Exp=TRUE )

       Estimate      StdErr          z        P exp(Est.)     2.5%    97.5%
sex 0.01570586 0.103889538  0.1511785 0.879835  1.015830 0.828686 1.245237
age 0.05423706 0.004574175 11.8572352 0.000000  1.055735 1.046312 1.065242
```

Are there any major differences?

9. If time permits (this takes rather long computing time):

   Split time since stroke in intervals of length 0.01 years instead of 0.05 years and repeat the analysis.

10. Now use a parametric function for the baseline hazard. We will use restricted cubic splines (natural splines) with knots at 0.05, 0.2, 0.7, 1.5, 3 and 4.8 years, but we also need a quantitative variable giving the midpoint of the interval, which is achieved by the function `timeBand`:

```
> sLst$Tfs.m <- timeBand( sLst, "Tfs", "middle" )
> library( splines )
> kn <- c(0.05,0.2,0.7,1.5,3)
> Bk <- c(0,4.8)
> mS <- glm( lex.Xst ~
+            ns( Tfs.m, knots=kn, Bo=Bk ) + sex + age + offset(log(lex.dur)),
+            family=poisson, data=sLst )
```

Compare the parameter estimates with the previous models.

```
> ci.lin( mC )

       Estimate      StdErr          z        P        2.5%      97.5%
sex 0.01570586 0.103889538  0.1511785 0.879835 -0.18791389 0.21932561
age 0.05423706 0.004574175 11.8572352 0.000000  0.04527185 0.06320228

> ci.lin( mP, subset=c("sex","age") )

       Estimate     StdErr          z         P        2.5%      97.5%
sex 0.01404002 0.10387588  0.1351615 0.8924842 -0.18955296 0.21763301
age 0.05471934 0.00457603 11.9578186 0.0000000  0.04575048 0.06368819

> ci.lin( mS, subset=c("sex","age") )

       Estimate     StdErr          z         P        2.5%      97.5%
sex 0.01547319 0.10385833  0.1489837 0.8815665 -0.18808539 0.21903178
age 0.05469285 0.00457629 11.9513510 0.0000000  0.04572348 0.06366221
```

11. Obtain an estimate of the baseline hazard function for a female aged 60. You will need to generate a sequence of times where you compute it:

```
> t.pt <- seq(0,5,0.01)
> CM <- cbind( 1, ns( t.pt, knots=kn, Bo=Bk ), 0, 60 )
> hz <- ci.lin( mS, ctr.mat=CM, Exp=TRUE )[,5:7] * 1000
> matplot( t.pt, hz, type="l", lwd=c(3,1,1), ylim=c(0,1), lty=1, col="black" )
```

12. Alternatively, you can obtain the hazard by `predict` using the `newdata=` argument. Note taht you also need to specify values of `lex.dur` which is in the offset of the model:

```
> nd <- data.frame( Tfs.m=t.pt, sex=0, age=60, lex.dur=1000 )
> prhz <- predict( mS, newdata=nd, type="link", se.fit=T )
> str( prhz )

 List of 3
  $ fit           : Named num [1:501] 8.55 8.27 7.99 7.73 7.47 ...
   ..- attr(*, "names")= chr [1:501] "1" "2" "3" "4" ...
  $ se.fit        : Named num [1:501] 0.169 0.143 0.124 0.113 0.111 ...
   ..- attr(*, "names")= chr [1:501] "1" "2" "3" "4" ...
  $ residual.scale: num 1

> prhz <- exp( cbind( prhz$fit, prhz$se.fit ) %*% ci.mat() )
```

Verify that you get the same estimates:

```
> matplot( hz, prhz )
```

13. Obtain an estimate of the survival function for a female aged 60. You can reuse the sequence of times from before with the modification that you should not use 0. Consult the help page for `ci.cum` first.

```
> t.pt <- t.pt[-1]
> CM <- cbind( 1, ns( t.pt-0.005, knots=kn, Bo=Bk ), 0, 60 )
> Hz <- ci.cum( mS, ctr.mat=CM, intl=0.01 )
> matplot( t.pt, exp(-Hz)[,-4],
+          type="l", lwd=c(3,1,1), ylim=c(0,1), lty=1, col="black" )
```

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

```
> matplot( t.pt, exp(-Hz)[,-4],
+          type="l", lwd=c(3,1,1), ylim=c(0,1), lty=1, col="black" )
> lines( survfit(mC,newdata=data.frame(sex=0,age=60)),
+        conf.int=TRUE, col="red" )
> # overplot the estimate with a thicker line:
> lines( survfit(mC,newdata=data.frame(sex=0,age=60)),
+        conf.int=FALSE, col="red", lwd=3 )
```

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

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

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

- Time should be split finely.
- The effect of time should be modelled detailed.
- Compute the hazards at the mipoint of the intervals, but plot the cumulative hazard (or equivalently, the survival function) at the upper end of the intervals.

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

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

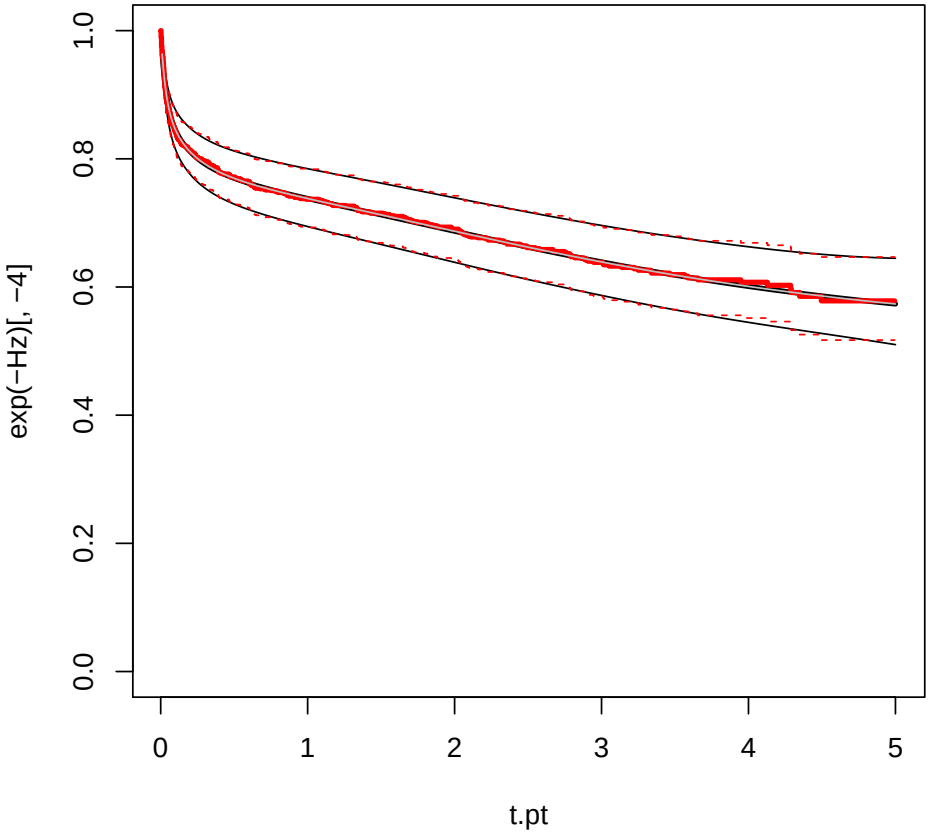

Figure 3.13: *Estimated survival curve for a female 75-year at stroke, computed by the Breslow-setimator from the Cox-model, and by using the approximation from the Poisson model.*

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

```
> options( width=120 )
> 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           do
   M:5185   Min.   :1898   Min.   :1995   Min.   :1995   Min.   :1995   Min.   :1995   Min.
   F:4815   1st Qu.:1930   1st Qu.:2000   1st Qu.:2002   1st Qu.:2001   1st Qu.:2001   1st Qu.
            Median :1941   Median :2004   Median :2005   Median :2004   Median :2005   Median
            Mean   :1942   Mean   :2003   Mean   :2005   Mean   :2004   Mean   :2004   Mean
            3rd Qu.:1951   3rd Qu.:2007   3rd Qu.:2008   3rd Qu.:2007   3rd Qu.:2007   3rd Qu.
            Max.   :2008   Max.   :2010   Max.   :2010   Max.   :2010   Max.   :2010   Max.
                                          NA's   :7497   NA's   :4503   NA's   :8209
```

2. 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:

```
> 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.
```

   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
```

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 12-month intervals we get a bit more than 50,000 records:

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

  Transitions:
       To
  From    Alive Dead  Records:  Events: Risk time:  Persons:
    Alive 61627 2499     64126     2499   54273.27      9996
```

5. With this in place we can start by making a crude age-specific mortality curve for men and women separately, using natural splines:

```
> library( splines )
> r.m <- glm( (lex.Xst=="Dead") ~ ns( A, df=10, intercept=TRUE ) - 1,
+             offset = log( lex.dur ),
+             family = poisson,
+               data = subset( SL, sex=="M" ) )
> r.f <- update( r.m, data = subset( SL, sex=="F" ) )
```

   With these objects we can get the estimated log-rates by using `predict`, and supplying a data frame of prediction points

```
> nd <-  data.frame( A = seq(10,90,0.5),
+               lex.dur = 1000)
> p.m <- predict.glm( r.m, type = "link",
+                       newdata = nd,
+                        se.fit = TRUE )
> p.f <- predict.glm( r.f, type = "link",
+                       newdata = nd,
+                        se.fit = TRUE )
> str( p.m )

  List of 3
   $ fit           : Named num [1:161] -0.149 -0.1233 -0.0975 -0.0716 -0.0455 ...
    ..- attr(*, "names")= chr [1:161] "1" "2" "3" "4" ...
   $ se.fit        : Named num [1:161] 1.45 1.41 1.37 1.33 1.29 ...
    ..- attr(*, "names")= chr [1:161] "1" "2" "3" "4" ...
   $ residual.scale: num 1
```

   From the structure of the predicted rates (`p.m`) we can construct the predicted rates with confidence intervals by using matrix multiplication and the `ci.mat` function:

```
> ci.mat()
     Estimate      2.5%    97.5%
[1,]        1  1.000000 1.000000
[2,]        0 -1.959964 1.959964
> lr.m <- cbind(p.m$fit,p.m$se.fit) %*% ci.mat()
> lr.f <- cbind(p.f$fit,p.f$se.fit) %*% ci.mat()
```

. . . and finally we can plot the two sets of estimated rates:

```
> matplot( seq(10,90,0.5), exp( cbind(lr.m,lr.f) ),
+          type="l", lty=1, lwd=c(3,1,1), las=1,
+          col=rep(c("blue","red"),each=3),
+          log="y", ylim=c(0.1,200),
+          xlab="Age", ylab="Mortality rates per 1000 PY" )
```

Graphical comparison with the population rates

6. We can compare with the mortality rates from the general population; they are available in the data frame `M.dk`

```
> data( M.dk )
> head( M.dk )
  A sex    P   D        Y      rate
1 0   1 1974 459 35963.33 12.762999
2 0   2 1974 303 34382.83  8.812537
3 0   1 1975 435 36099.00 12.050195
4 0   2 1975 311 34652.17  8.974908
5 0   1 1976 405 34965.00 11.583012
6 0   2 1976 258 33278.33  7.752792
```

So we just plot the mortality rates from 2005 on top of this:

```
> with( subset( M.dk, sex==1 & P==2005 ), lines( A, rate, col="blue", lty="12", lwd=3 ) )
> with( subset( M.dk, sex==2 & P==2005 ), lines( A, rate, col="red" , lty="12", lwd=3 ) )
```

7. It would however be more prudent to model these rates in a similar fashion as the diabetes mortality:

```
> R.m <- glm( D ~ ns( A, df=10, intercept=TRUE ) - 1,
+             offset = log( Y ),
+             family = poisson,
+               data = subset( M.dk, sex==1 & P>1994 ) )
> R.f <- update( R.m, data = subset( M.dk, sex==2 & P>1994 ) )
> nd <-  data.frame( A = seq(10,90,0.5),
+                    Y = 1000)
> P.m <- predict.glm( R.m, type = "link", newdata = nd )
> P.f <- predict.glm( R.f, type = "link", newdata = nd )
```

Once we have the predicted rates from a smoothing model we can redo the plot with these overlaid:

```
> matplot( seq(10,90,0.5),
+          exp( cbind(lr.m,lr.f) ),
+          type="l", lty=1, lwd=c(3,1,1),
+          col=rep(c("blue","red"),each=3),
+          log="y", ylim=c(0.1,200),
+          xlab="Age", ylab="Mortality rates per 1000 PY" )
> matlines( seq(10,90,0.5), exp(cbind( P.m,P.f )), lty="12",
+           col=c("blue","red"), lwd=3 )
```

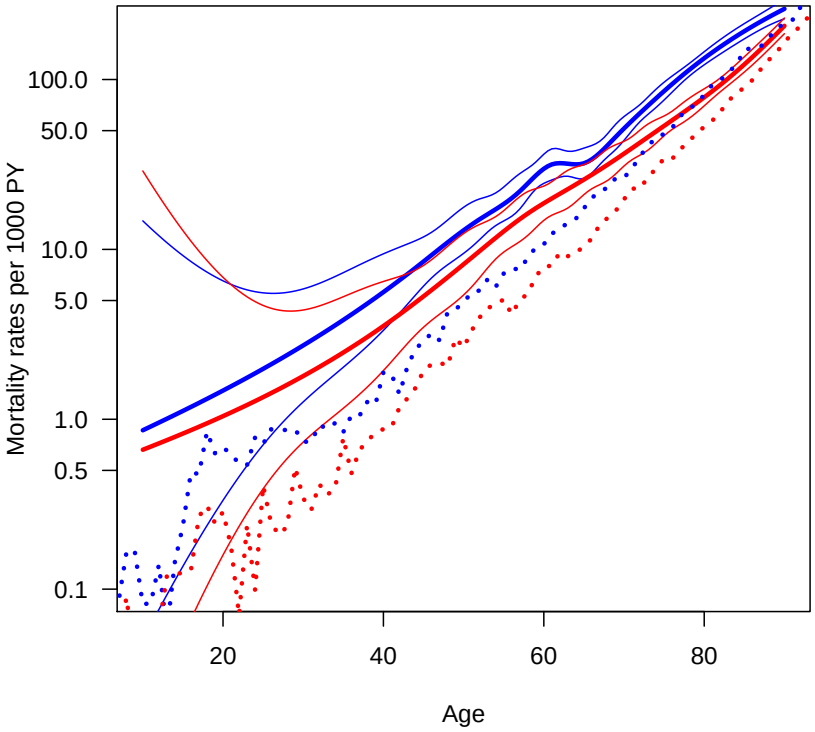

Figure 3.14: *Age-specific mortality rates for Danish diabetes patients as estimated from a model with only age. Broken lines are empirical rates from 2005. Blue: men, red: women.*

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 (lines in the dataset), is the same between knots. 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 will be using 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 tine 0 on the timescale we use 0 as the first knot.

   So we start out by placing knots so that the number of events is the same between each pair of knots (strictly speaking we should do this separately for men and women, but we pass on that one here):

```
> kn.A <- with( subset( SL, lex.Xst=="Dead" ),
+               quantile( A+lex.dur, probs=seq(5,95,10)/100 ) )
> kn.P <- with( subset( SL, lex.Xst=="Dead" ),
+               quantile( P+lex.dur, probs=seq(5,95,30)/100 ) )
```

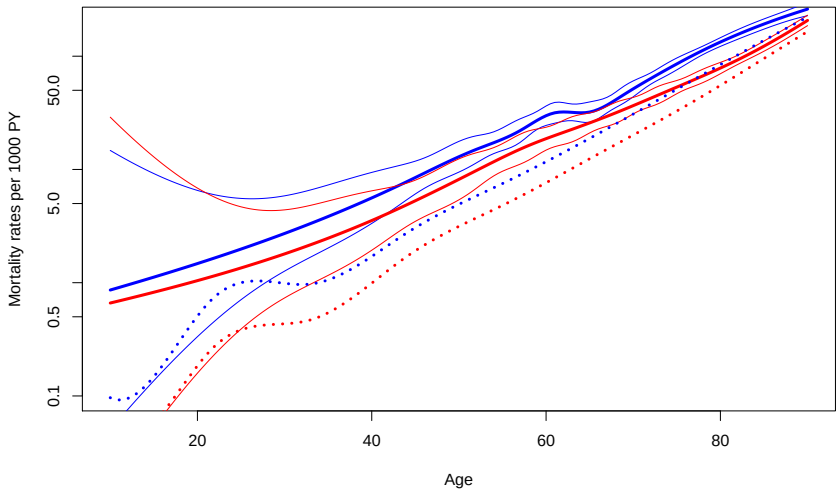

Figure 3.15: *Age-specific mortality rates for Danish diabetes patients as estimated from a model with only age. Broken lines are modeled population rates 1995–2010. Blue: men, red: women.*

```
> kn.dur <- c(0,with( subset( SL, lex.Xst=="Dead" ),
+                quantile( dur+lex.dur, probs=seq(5,95,10)/100 ) ))
```

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[-c(1,length(kn.A))],
+                                       Bo=kn.A[ c(1,length(kn.A))] ) +
+                                ns( P, kn=kn.P[-c(1,length(kn.P))],
+                                       Bo=kn.P[ c(1,length(kn.P))] ) +
+                              ns( dur, kn=kn.dur[-c(1,length(kn.dur))],
+                                       Bo=kn.dur[ c(1,length(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[-c(1, length(kn.A))],
      Bo = kn.A[c(1, length(kn.A))]) + ns(P, kn = kn.P[-c(1, length(kn.P))],
      Bo = kn.P[c(1, length(kn.P))]) + ns(dur, kn = kn.dur[-c(1,
      length(kn.dur))], Bo = kn.dur[c(1, length(kn.dur))]), family = poisson,
      data = subset(SL, sex == "M"), offset = log(lex.dur))

  Deviance Residuals:
      Min       1Q   Median       3Q      Max
  -1.0444  -0.3011  -0.2191  -0.1395   4.0606

  Coefficients:

                                                                                  Estimate St
  (Intercept)                                                                     -3.21711
  ns(A, kn = kn.A[-c(1, length(kn.A))], Bo = kn.A[c(1, length(kn.A))])1           0.68906
  ns(A, kn = kn.A[-c(1, length(kn.A))], Bo = kn.A[c(1, length(kn.A))])2           1.21939
  ns(A, kn = kn.A[-c(1, length(kn.A))], Bo = kn.A[c(1, length(kn.A))])3           1.50702
  ns(A, kn = kn.A[-c(1, length(kn.A))], Bo = kn.A[c(1, length(kn.A))])4           1.92383
  ns(A, kn = kn.A[-c(1, length(kn.A))], Bo = kn.A[c(1, length(kn.A))])5           2.12200
```

```
ns(A, kn = kn.A[-c(1, length(kn.A))], Bo = kn.A[c(1, length(kn.A))])6             1.88170
ns(A, kn = kn.A[-c(1, length(kn.A))], Bo = kn.A[c(1, length(kn.A))])7             2.42353
ns(A, kn = kn.A[-c(1, length(kn.A))], Bo = kn.A[c(1, length(kn.A))])8             3.16568
ns(A, kn = kn.A[-c(1, length(kn.A))], Bo = kn.A[c(1, length(kn.A))])9             2.47621
ns(P, kn = kn.P[-c(1, length(kn.P))], Bo = kn.P[c(1, length(kn.P))])1            -0.27240
ns(P, kn = kn.P[-c(1, length(kn.P))], Bo = kn.P[c(1, length(kn.P))])2            -0.49119
ns(P, kn = kn.P[-c(1, length(kn.P))], Bo = kn.P[c(1, length(kn.P))])3            -0.29024
ns(dur, kn = kn.dur[-c(1, length(kn.dur))], Bo = kn.dur[c(1, length(kn.dur))])1  -0.30091
ns(dur, kn = kn.dur[-c(1, length(kn.dur))], Bo = kn.dur[c(1, length(kn.dur))])2  -0.81243
ns(dur, kn = kn.dur[-c(1, length(kn.dur))], Bo = kn.dur[c(1, length(kn.dur))])3  -0.39040
ns(dur, kn = kn.dur[-c(1, length(kn.dur))], Bo = kn.dur[c(1, length(kn.dur))])4  -0.79923
ns(dur, kn = kn.dur[-c(1, length(kn.dur))], Bo = kn.dur[c(1, length(kn.dur))])5  -0.59604
ns(dur, kn = kn.dur[-c(1, length(kn.dur))], Bo = kn.dur[c(1, length(kn.dur))])6  -0.17280
ns(dur, kn = kn.dur[-c(1, length(kn.dur))], Bo = kn.dur[c(1, length(kn.dur))])7  -0.92102
ns(dur, kn = kn.dur[-c(1, length(kn.dur))], Bo = kn.dur[c(1, length(kn.dur))])8  -0.10567
ns(dur, kn = kn.dur[-c(1, length(kn.dur))], Bo = kn.dur[c(1, length(kn.dur))])9  -0.85446
ns(dur, kn = kn.dur[-c(1, length(kn.dur))], Bo = kn.dur[c(1, length(kn.dur))])10  0.06945

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 11288  on 32697  degrees of freedom
Residual deviance: 10010  on 32675  degrees of freedom
AIC: 12742

Number of Fisher Scoring iterations: 7
```

As a small aside; the specification of the natural splines is cumbersome with specification of both knots and boundary knots and the output from `summary()` is very large, so we use a convenience wrapper that does this on the fly:

```
> source( "http://BendixCarstensen.com/SPE/R/Ns.r" )
> Ns
function (x, df = NULL, knots = NULL, intercept = FALSE, Boundary.knots = NULL)
{
    if (is.null(Boundary.knots)) {
        if (!is.null(knots)) {
            knots <- sort(unique(knots))
            ok <- c(1, length(knots))
            Boundary.knots <- knots[ok]
            knots <- knots[-ok]
        }
    }
    ns(x, df = df, knots = knots, intercept = intercept, Boundary.knots = Boundary.knots)
}
```

Having defined this, the model specification and summary is much simpler:

```
> 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
```

```
-1.0444  -0.3011  -0.2191  -0.1395   4.0606

Coefficients:
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)             -3.21711    0.10190 -31.571  < 2e-16
Ns(A, kn = kn.A)1        0.68906    0.18264   3.773 0.000161
Ns(A, kn = kn.A)2        1.21939    0.16510   7.386 1.52e-13
Ns(A, kn = kn.A)3        1.50702    0.18593   8.105 5.27e-16
Ns(A, kn = kn.A)4        1.92383    0.17609  10.925  < 2e-16
Ns(A, kn = kn.A)5        2.12200    0.18983  11.178  < 2e-16
Ns(A, kn = kn.A)6        1.88170    0.20204   9.314  < 2e-16
Ns(A, kn = kn.A)7        2.42353    0.17150  14.131  < 2e-16
Ns(A, kn = kn.A)8        3.16568    0.13276  23.844  < 2e-16
Ns(A, kn = kn.A)9        2.47621    0.12664  19.554  < 2e-16
Ns(P, kn = kn.P)1       -0.27240    0.11656  -2.337 0.019439
Ns(P, kn = kn.P)2       -0.49119    0.16991  -2.891 0.003842
Ns(P, kn = kn.P)3       -0.29024    0.10322  -2.812 0.004927
Ns(dur, kn = kn.dur)1   -0.30091    0.23451  -1.283 0.199437
Ns(dur, kn = kn.dur)2   -0.81243    0.22716  -3.576 0.000348
Ns(dur, kn = kn.dur)3   -0.39040    0.21414  -1.823 0.068284
Ns(dur, kn = kn.dur)4   -0.79923    0.21901  -3.649 0.000263
Ns(dur, kn = kn.dur)5   -0.59604    0.20574  -2.897 0.003767
Ns(dur, kn = kn.dur)6   -0.17280    0.19660  -0.879 0.379436
Ns(dur, kn = kn.dur)7   -0.92102    0.20048  -4.594 4.35e-06
Ns(dur, kn = kn.dur)8   -0.10567    0.16499  -0.640 0.521896
Ns(dur, kn = kn.dur)9   -0.85446    0.24909  -3.430 0.000603
Ns(dur, kn = kn.dur)10   0.06945    0.14170   0.490 0.624043

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 11288  on 32697  degrees of freedom
Residual deviance: 10010  on 32675  degrees of freedom
AIC: 12742

Number of Fisher Scoring iterations: 7

> mf <- update( mm, data = subset( SL, sex=="F" ) )
```

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, intercept = TRUE) - 1
  Resid. Df Resid. Dev  Df Deviance  Pr(>Chi)
1     32675      10010
2     32688      10097 -13    -86.6 6.222e-13

> 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, intercept = TRUE) - 1
  Resid. Df Resid. Dev  Df Deviance  Pr(>Chi)
1     31405     8744.4
2     31418     8808.1 -13  -63.653 1.157e-08
```

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

11. The model fitted separately for men and women has three terms: age (`A`), calendar time (`P`) and diabetes duration (`dur`). Since the outcome is a rate with dimension time^{-1} we must put the rate dimension on one of these terms and leave the two others as rate-ratios. In order to do this we must fix reference values for the two rate-ratio terms. The natural variable for the rate-dimension is age, so that we get estimated age-specific rate-ratios for a specific calendar time, 1.1.2008, say, and a specific duration of diabetes, 2 years, say.

    In order to extract these terms from the model we need contrast matrices, that is matrices where each row corresponds to a set of values for age or period or duration, and the columns correspond to the spline basis as used in the model.

    This is one reason for explicitly fixing the knots in the spline definitions; when we extract the effects we must use the same set of knots as in the model specification.

    We will need matrices for specified set of values for age, calendar time and duration, but also matrices where all rows refer to the chosen reference values for calendar time and duration.

    We begin by specifying the prediction points for the time scales and the reference points. There is formally no reason to require that the matrices all have the same number of rows, but it makes the handling of the reference points much easier.

```
> N <- 100
> pr.A <- seq(10,90,,N)
> pr.P <- seq(1995,2010,,N)
> pr.d <- seq(0,15,,N)
> rf.P <- 2009
> rf.d <- 2
```

    With these in place we generate the matrices we shall multiply to the parameter estimates:

```
> AC <- Ns( pr.A, knots=kn.A )
> PC <- Ns( pr.P, knots=kn.P )
> dC <- Ns( pr.d, knots=kn.dur )
> PR <- Ns( rep(rf.P,N), knots=kn.P )
> dR <- Ns( rep(rf.d,N), knots=kn.dur )
```

    Note that the rows of `AC` refer to `N` points on the age-scale, `PC` to N points on the calendar time scale, etc.

    These matrices are the necessary input for extracting the effects; this is done by the function `ci.exp`, remember to take a look at the help page for this.

    Note that we make use of *all* parameters when extracting the age-effect — this is the effect where we have the dimension of the response (rate), and hence the intercept, and where we have fixed the values of date and duration at their reference values.

    The rate-ratios for calendar time and duration are estimated exclusively from the parameters for these terms, but note that we subtract the values at the reference point:

```
> m.A <- ci.exp( mm, ctr.mat=cbind(1,AC,PR,dR) ) * 1000
> m.P <- ci.exp( mm, subset="P"  , ctr.mat=PC-PR )
> m.d <- ci.exp( mm, subset="dur", ctr.mat=dC-dR )
> f.A <- ci.exp( mf, ctr.mat=cbind(1,AC,PR,dR) ) * 1000
> f.P <- ci.exp( mf, subset="P"  , ctr.mat=PC-PR )
> f.d <- ci.exp( mf, subset="dur", ctr.mat=dC-dR )
```

12. We now plot the three effects in three panels beside each other:

```
> par( mfrow=c(1,3), mar=c(3,3,1,1), mgp=c(3,1,0)/1.6 )
> matplot( pr.A, cbind(m.A,f.A),
+          type="l", lty=1, lwd=c(3,1,1), col=rep(c("blue","red"),each=3),
+          log="y", xlab="Age", ylab="Mortality rate per 1000 PY" )
> matplot( pr.P,  cbind(m.P,f.P),
+          type="l", lty=1, lwd=c(3,1,1), col=rep(c("blue","red"),each=3),
+          log="y", xlab="Date of follow-up", ylab="Mortality rate ratio" )
> matplot( pr.d,  cbind(m.d,f.d),
+          type="l", lty=1, lwd=c(3,1,1), col=rep(c("blue","red"),each=3),
+          log="y", xlab="Diabetes duration", ylab="Mortality rate ratio" )
```

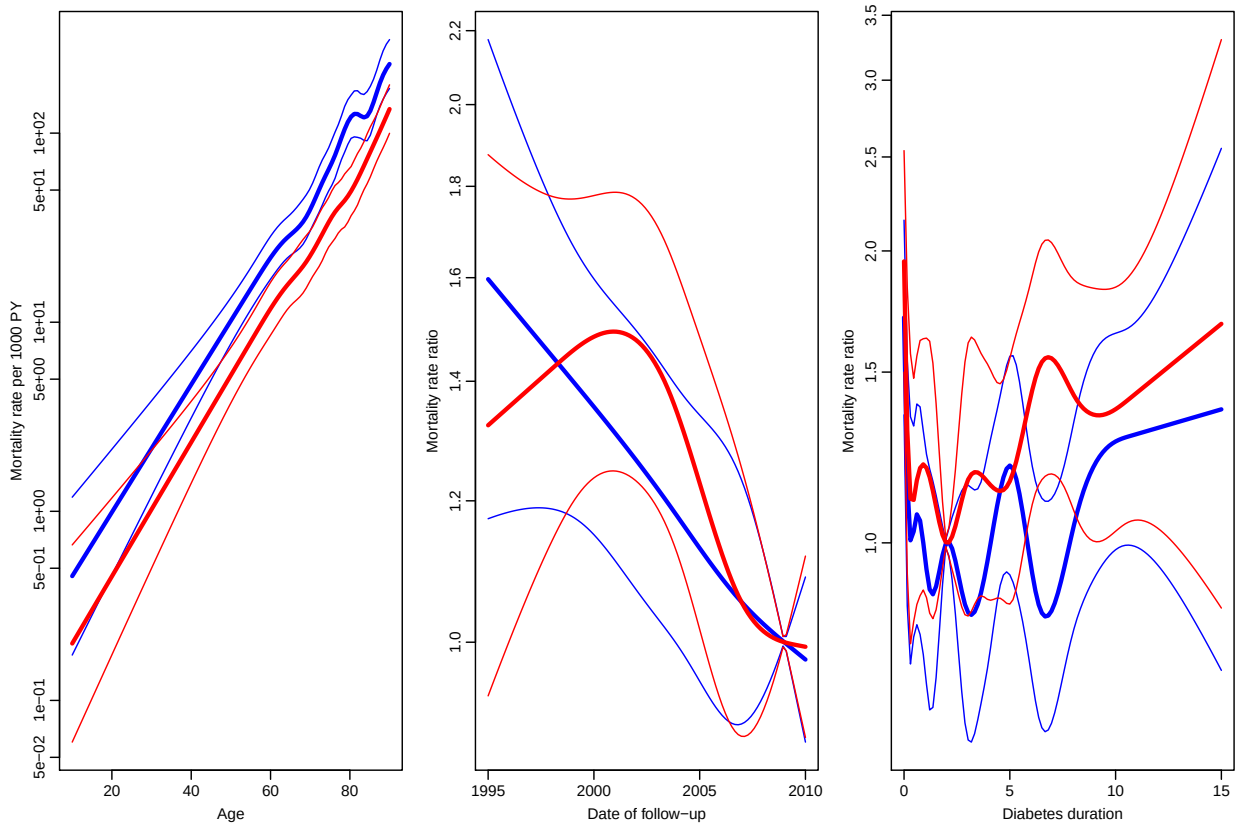

Figure 3.16: *Estimates from model for mortality of Danish diabetes patients. The duration is modeled with 10 parameters, which is clearly way too much.*

Figure **??** clearly shows that the duration effect is grossly over-modeled, and that the rate-ratios have a much smaller variability than the mortality rates.

Moreover the y-axis for mortality rates should be from about 0.1 to 200, and the y-axes for the rate-ratios should be on approximately the same scale. To make the RR-axes symmetric, from 1/30 to 30, that is a factor $30^2 = 900$, and the the rate-axis from 0.2 to 180.

So we redefine the duration knots, refit the models, re-extract parameters and plot using pre-specified axis ranges:

```
> kn.dur <- c(0,with( subset( SL, lex.Xst=="Dead" ),
+               quantile( dur+lex.dur, probs=seq(5,95,30)/100 ) ))
> dC <- Ns( pr.d, knots=kn.dur )
> dR <- Ns( rep(rf.d,N), knots=kn.dur )
> 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" ) )
> mf <- update( mm, data = subset( SL, sex=="F" ) )
> m.A <- ci.exp( mm, ctr.mat=cbind(1,AC,PR,dR) ) * 1000
> m.P <- ci.exp( mm, subset="P"  , ctr.mat=PC-PR )
> m.d <- ci.exp( mm, subset="dur", ctr.mat=dC-dR )
> f.A <- ci.exp( mf, ctr.mat=cbind(1,AC,PR,dR) ) * 1000
> f.P <- ci.exp( mf, subset="P"  , ctr.mat=PC-PR )
> f.d <- ci.exp( mf, subset="dur", ctr.mat=dC-dR )
> par( mfrow=c(1,3), mar=c(3,3,1,1), mgp=c(3,1,0)/1.6 )
> matplot( pr.A, cbind(m.A,f.A),
+          type="l", lty=1, lwd=c(3,1,1), col=rep(c("blue","red"),each=3),
+          log="y", ylim=c(0.2,180),
+          xlab="Age", ylab="Mortality rate per 1000 PY" )
> matplot( pr.P,  cbind(m.P,f.P),
+          type="l", lty=1, lwd=c(3,1,1), col=rep(c("blue","red"),each=3),
+          log="y", ylim=c(1/30,30),
+          xlab="Date of follow-up", ylab="Mortality rate ratio" )
> abline( h=1 )
> matplot( pr.d,  cbind(m.d,f.d),
+          type="l", lty=1, lwd=c(3,1,1), col=rep(c("blue","red"),each=3),
+          log="y", ylim=c(1/30,30),
+          xlab="Diabetes duration", ylab="Mortality rate ratio" )
> abline( h=1 )
```

We might argue that we do not need the same scale for the y-axes for rates and RRs:

```
> par( mfrow=c(1,3), mar=c(3,3,1,1), mgp=c(3,1,0)/1.6 )
> matplot( pr.A, cbind(m.A,f.A),
+          type="l", lty=1, lwd=c(3,1,1), col=rep(c("blue","red"),each=3),
+          log="y", ylim=c(0.2,180),
+          xlab="Age", ylab="Mortality rate per 1000 PY" )
> matplot( pr.P,  cbind(m.P,f.P),
+          type="l", lty=1, lwd=c(3,1,1), col=rep(c("blue","red"),each=3),
+          log="y", ylim=c(1/3,3),
+          xlab="Date of follow-up", ylab="Mortality rate ratio" )
> abline( h=1 )
> matplot( pr.d,  cbind(m.d,f.d),
+          type="l", lty=1, lwd=c(3,1,1), col=rep(c("blue","red"),each=3),
+          log="y", ylim=c(1/3,3),
+          xlab="Diabetes duration", ylab="Mortality rate ratio" )
> abline( h=1 )
```

13. We have so far fitted models separately for men and women, but judging from the display of the parameters in figure **??**, the period and duration effects are the same, so we might fit a model for the entire dataset with common period and duration effects, but different age-effect for the two sexes:

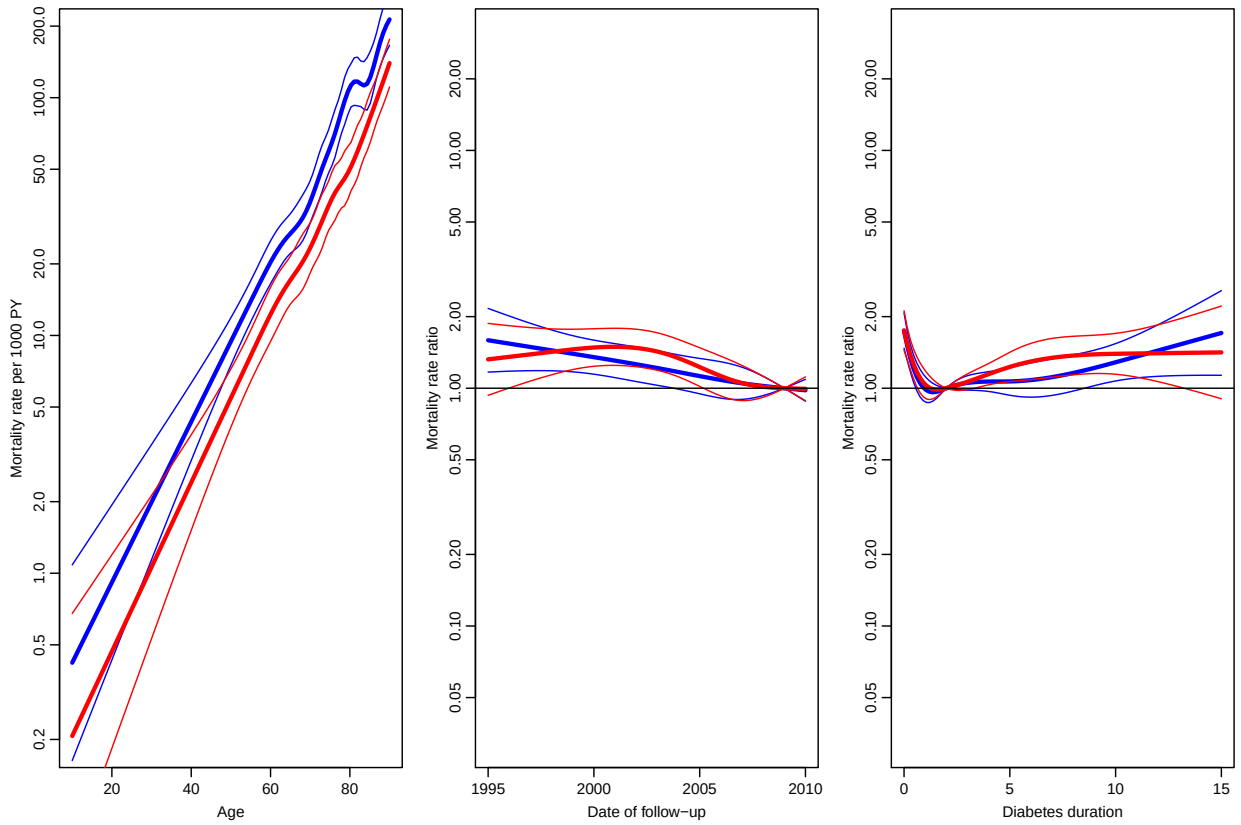

Figure 3.17: *Estimates from the model for mortality of Danish diabetes patients with only 5 knots (corresponding to 4 parameters) for duration.*

```
> m2 <- glm( (lex.Xst=="Dead") ~ sex +
+                               sex:Ns( A, kn=kn.A ) +
+                                   Ns( P, kn=kn.P ) +
+                                   Ns( dur, kn=kn.dur ),
+            offset = log( lex.dur ),
+            family = poisson,
+              data = SL )
> ci.exp(m2)
```

	exp(Est.)	2.5%	97.5%
(Intercept)	0.03579237	0.0302410	0.04236282
sexF	0.66657884	0.5344046	0.83144370
Ns(P, kn = kn.P)1	0.72547140	0.6136877	0.85761656
Ns(P, kn = kn.P)2	0.68310826	0.5324727	0.87635829
Ns(P, kn = kn.P)3	0.70425590	0.6066292	0.81759391
Ns(dur, kn = kn.dur)1	0.59963846	0.4994206	0.71996685
Ns(dur, kn = kn.dur)2	0.83011043	0.7026141	0.98074222
Ns(dur, kn = kn.dur)3	0.43036334	0.3246658	0.57047162
Ns(dur, kn = kn.dur)4	1.07957947	0.9188562	1.26841594
sexM:Ns(A, kn = kn.A)1	1.99255665	1.3930634	2.85003686
sexF:Ns(A, kn = kn.A)1	2.29270232	1.4385172	3.65409872
sexM:Ns(A, kn = kn.A)2	3.37498942	2.4423063	4.66385131
sexF:Ns(A, kn = kn.A)2	3.34775743	2.2559188	4.96803339
sexM:Ns(A, kn = kn.A)3	4.50026933	3.1262878	6.47810609
sexF:Ns(A, kn = kn.A)3	4.57342778	2.9485292	7.09378832
sexM:Ns(A, kn = kn.A)4	6.86263799	4.8597269	9.69103852
sexF:Ns(A, kn = kn.A)4	5.07588761	3.3572466	7.67433495
sexM:Ns(A, kn = kn.A)5	8.26769328	5.6990906	11.99397541
sexF:Ns(A, kn = kn.A)5	6.38110530	4.2594149	9.55964737
sexM:Ns(A, kn = kn.A)6	6.64708490	4.4773994	9.86816990

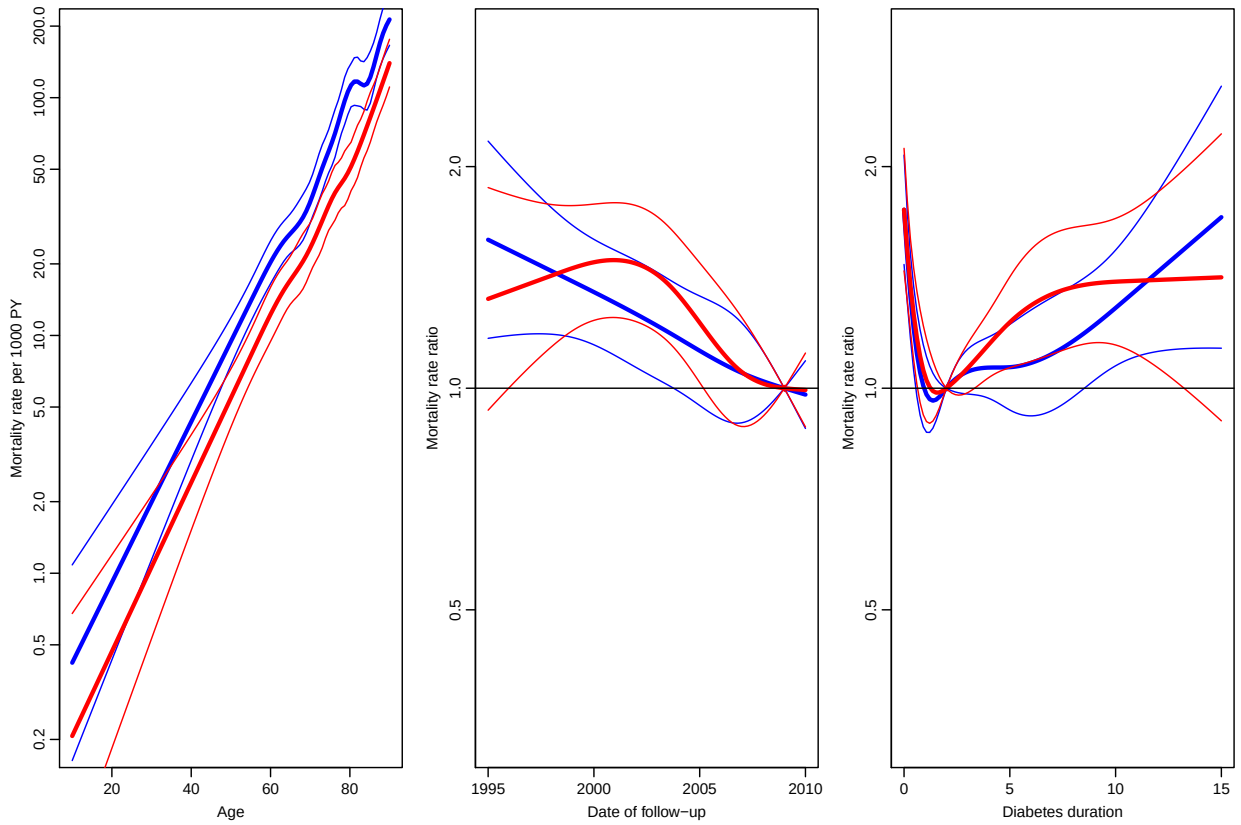

Figure 3.18: *Estimates from model for mortality of Danish diabetes patients.*

```
sexF:Ns(A, kn = kn.A)6  8.63141522  6.0007807 12.41527270
sexM:Ns(A, kn = kn.A)7 11.12936225  7.9570794 15.56635267
sexF:Ns(A, kn = kn.A)7 10.60687235  7.8238629 14.37982012
sexM:Ns(A, kn = kn.A)8 23.40325077 18.0491615 30.34557292
sexF:Ns(A, kn = kn.A)8 27.66533384 21.1941508 36.11235494
sexM:Ns(A, kn = kn.A)9 11.76296454  9.1806331 15.07165505
sexF:Ns(A, kn = kn.A)9 14.83464848 11.5157684 19.11004014
```

14. We can formally test this model against the separate models; the deviance and degrees of freedom from the separate models for men and women add up to that of a joint model with interaction between all terms and sex:

```
> j.dev <- mm$dev + mf$dev
> j.df  <- mm$df.r + mf$df.r
> 1 - pchisq( m2$dev - j.dev, m2$df.r - j.df )

  [1] 0.473991
```

So there is indeed no evidence of different period and duration effects.

15. We might from a purely technical point of view contemplate a model where the difference in age-specific mortality between men and women were either constant or exponentially increasing or decreasing by age. And we might even accept a model of that sort, but given the different biology of men and women over their life span, it would make little sense. And therefore we have not done it here.

16. We can now extract the parameters from the model. Note that the sequence (and hence meaning) naming of the parameters depend on how the model is specified. The age-specific rates for men and women at the reference time and reference duration will need parameters extracted by the following subset-argument to `ci.exp`:

```
> ci.exp( m2, subset=c("Int","sexM","P","dur") )

                          exp(Est.)       2.5%       97.5%
(Intercept)              0.03579237  0.0302410  0.04236282
sexM:Ns(A, kn = kn.A)1   1.99255665  1.3930634  2.85003686
sexM:Ns(A, kn = kn.A)2   3.37498942  2.4423063  4.66385131
sexM:Ns(A, kn = kn.A)3   4.50026933  3.1262878  6.47810609
sexM:Ns(A, kn = kn.A)4   6.86263799  4.8597269  9.69103852
sexM:Ns(A, kn = kn.A)5   8.26769328  5.6990906 11.99397541
sexM:Ns(A, kn = kn.A)6   6.64708490  4.4773994  9.86816990
sexM:Ns(A, kn = kn.A)7  11.12936225  7.9570794 15.56635267
sexM:Ns(A, kn = kn.A)8  23.40325077 18.0491615 30.34557292
sexM:Ns(A, kn = kn.A)9  11.76296454  9.1806331 15.07165505
Ns(P, kn = kn.P)1        0.72547140  0.6136877  0.85761656
Ns(P, kn = kn.P)2        0.68310826  0.5324727  0.87635829
Ns(P, kn = kn.P)3        0.70425590  0.6066292  0.81759391
Ns(dur, kn = kn.dur)1    0.59963846  0.4994206  0.71996685
Ns(dur, kn = kn.dur)2    0.83011043  0.7026141  0.98074222
Ns(dur, kn = kn.dur)3    0.43036334  0.3246658  0.57047162
Ns(dur, kn = kn.dur)4    1.07957947  0.9188562  1.26841594

> ci.exp( m2, subset=c("Int","sexF","P","dur") )

                          exp(Est.)       2.5%       97.5%
(Intercept)              0.03579237  0.0302410  0.04236282
sexF                     0.66657884  0.5344046  0.83144370
sexF:Ns(A, kn = kn.A)1   2.29270232  1.4385172  3.65409872
sexF:Ns(A, kn = kn.A)2   3.34775743  2.2559188  4.96803339
sexF:Ns(A, kn = kn.A)3   4.57342778  2.9485292  7.09378832
sexF:Ns(A, kn = kn.A)4   5.07588761  3.3572466  7.67433495
sexF:Ns(A, kn = kn.A)5   6.38110530  4.2594149  9.55964737
sexF:Ns(A, kn = kn.A)6   8.63141522  6.0007807 12.41527270
sexF:Ns(A, kn = kn.A)7  10.60687235  7.8238629 14.37982012
sexF:Ns(A, kn = kn.A)8  27.66533384 21.1941508 36.11235494
sexF:Ns(A, kn = kn.A)9  14.83464848 11.5157684 19.11004014
Ns(P, kn = kn.P)1        0.72547140  0.6136877  0.85761656
Ns(P, kn = kn.P)2        0.68310826  0.5324727  0.87635829
Ns(P, kn = kn.P)3        0.70425590  0.6066292  0.81759391
Ns(dur, kn = kn.dur)1    0.59963846  0.4994206  0.71996685
Ns(dur, kn = kn.dur)2    0.83011043  0.7026141  0.98074222
Ns(dur, kn = kn.dur)3    0.43036334  0.3246658  0.57047162
Ns(dur, kn = kn.dur)4    1.07957947  0.9188562  1.26841594
```

Note that the two subsets of parameters have different length; the parameters for the women (sex="F") has one more column:

```
> mi.A <- ci.exp( m2, subset=c("Int","sexM","P","dur"), ctr.mat=cbind(1  ,AC,PR,dR) ) * 1000
> fi.A <- ci.exp( m2, subset=c("Int","sexF","P","dur"), ctr.mat=cbind(1,1,AC,PR,dR) ) * 1000
> b.P  <- ci.exp( m2, subset="P"  , ctr.mat=PC-PR )
> b.d  <- ci.exp( m2, subset="dur", ctr.mat=dC-dR )



> par( mfrow=c(1,3), mar=c(3,3,1,1), mgp=c(3,1,0)/1.6 )
> matplot( pr.A, cbind(m.A,f.A,mi.A,fi.A),
+          type="l", lty=rep(c(3,1),each=6), lwd=c(3,1,1), col=rep(c("blue","red"),each=3),
+          log="y", ylim=c(0.2,180),
+          xlab="Age", ylab="Mortality rate per 1000 PY" )
> matplot( pr.P,  cbind(m.P,f.P,b.P),
```

```
+          type="l", lty=rep(c(3,1),c(6,3)), lwd=c(3,1,1), col=rep(c("blue","red","black"),eac
+          log="y", ylim=c(1/3,3),
+          xlab="Date of follow-up", ylab="Mortality rate ratio" )
> abline( h=1 )
> matplot( pr.d,  cbind(m.d,f.d,b.d),
+          type="l", lty=rep(c(3,1),c(6,3)), lwd=c(3,1,1), col=rep(c("blue","red","black"),eac
+          log="y", ylim=c(1/3,3),
+          xlab="Diabetes duration", ylab="Mortality rate ratio" )
> abline( h=1 )
```

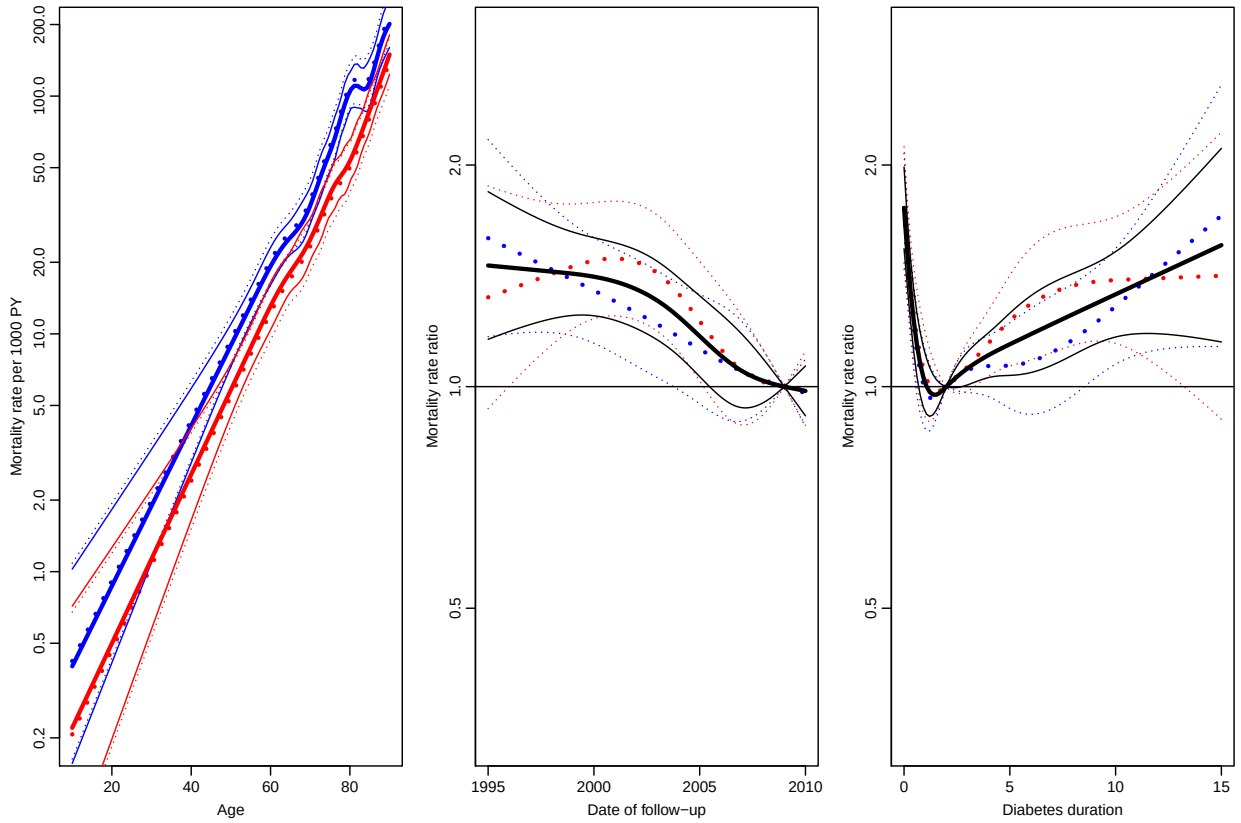

Figure 3.19: *Estimates from models for mortality of Danish diabetes patients. The broken lines are from the full interaction model, full lines with common effects of date and duration. Men:blue, women:red, both (i.e. common): black.*

We shall return to the set-up with separate effects for men and women.

17. 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 all multiple regression) to be interpreted as "all else equal" which they are not, as the three time scales advance by 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.

    Incidentally, this is most easily done using the `predict` function with the `newdata=` argument. So a person diagnosed in age 50 will have a (log-)mortality measure 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 )
> predict( mm, newdata=nd, se.fit=TRUE )

$fit
       1        2        3        4        5        6        7        8        9       10
3.267129 2.743749 2.799280 2.896099 2.953923 2.999912 3.057107 3.128086 3.208354 3.293304 3.
      14       15       16       17       18       19       20       21
3.640437 3.724499 3.804419 3.883603 3.967258 4.060710 4.169289 4.296557

$se.fit
         1          2          3          4          5          6          7          8
0.15795936 0.14233345 0.11950284 0.10897940 0.09461150 0.09037873 0.09600808 0.09967652 0.09
        11         12         13         14         15         16         17         18
0.10552798 0.12224274 0.13906924 0.15579972 0.18262347 0.22314661 0.27243810 0.32571265 0.37
        21
0.48844361

$residual.scale
[1] 1
```

We can wrap this so that we get the predicted rates with confidence intervals:

```
> sapply(predict( mm, newdata=nd, se.fit=TRUE )[1:2],cbind) %*% ci.mat()

      Estimate     2.5%    97.5%
 [1,] 3.267129 2.957534 3.576724
 [2,] 2.743749 2.464781 3.022717
 [3,] 2.799280 2.565059 3.033502
 [4,] 2.896099 2.682503 3.109695
 [5,] 2.953923 2.768488 3.139358
 [6,] 2.999912 2.822773 3.177051
 [7,] 3.057107 2.868934 3.245279
 [8,] 3.128086 2.932723 3.323448
 [9,] 3.208354 3.014682 3.402027
[10,] 3.293304 3.101715 3.484893
[11,] 3.379910 3.173079 3.586741
[12,] 3.466715 3.227123 3.706306
[13,] 3.553489 3.280918 3.826059
[14,] 3.640437 3.335075 3.945799
[15,] 3.724499 3.366563 4.082434
[16,] 3.804419 3.367060 4.241778
[17,] 3.883603 3.349634 4.417572
[18,] 3.967258 3.328872 4.605643
[19,] 4.060710 3.316015 4.805405
[20,] 4.169289 3.318858 5.019719
[21,] 4.296557 3.339225 5.253889
```

This can be nicely wrapped in a function that takes age and date of diagnosis as input and returns the estimated mortality rates for a male and a female diagnosed this age and date:

```
> DMm <-
+ function( A, P, pts=seq(0,15,0.1) )
+ {
+ nd <- data.frame( A=A+pts,
+                   P=P+pts,
+                 dur=  pts,
+             lex.dur=1000 )
+ cbind( nd$A,
```

```
+ exp(sapply(predict( mm, newdata=nd, se.fit=TRUE )[1:2],cbind) %*% ci.mat()),
+ exp(sapply(predict( mf, newdata=nd, se.fit=TRUE )[1:2],cbind) %*% ci.mat())
+       )
+ }
> DMm( 50, 1996, pts=0:10 )

          Estimate     2.5%    97.5%  Estimate      2.5%    97.5%
   [1,] 50 25.39307 19.28622 33.43360 12.923295  9.259298 18.03717
   [2,] 51 15.04576 11.73788 19.28584  8.520286  6.305111 11.51372
   [3,] 52 15.90491 12.94488 19.54178  9.139234  7.081519 11.79487
   [4,] 53 17.52097 14.44036 21.25879 10.808554  8.506719 13.73324
   [5,] 54 18.55717 15.58285 22.09921 12.862220 10.332394 16.01146
   [6,] 55 19.41537 16.30862 23.11395 15.069673 12.140844 18.70505
   [7,] 56 20.53363 17.03991 24.74367 17.052232 13.589632 21.39709
   [8,] 57 22.00858 18.21885 26.58662 18.514317 14.692504 23.33026
   [9,] 58 23.80149 19.82767 28.57175 19.241654 15.303932 24.19256
  [10,] 59 25.89341 21.49848 31.18678 19.346241 15.278189 24.49747
  [11,] 60 28.29472 23.00241 34.80467 19.263952 14.879354 24.94059
```

With this in place we can now plot the mortality rates for persons diagnosed at different ages and different dates:

```
> DMm.1996 <-
+ rbind(
+ DMm( 30, 1996 ), NA,
+ DMm( 40, 1996 ), NA,
+ DMm( 50, 1996 ), NA,
+ DMm( 60, 1996 ), NA,
+ DMm( 70, 1996 ), NA,
+ DMm( 80, 1996 ), NA,
+ DMm( 90, 1996 ) )
> DMm.2005 <-
+ rbind(
+ DMm( 30, 2005 ), NA,
+ DMm( 40, 2005 ), NA,
+ DMm( 50, 2005 ), NA,
+ DMm( 60, 2005 ), NA,
+ DMm( 70, 2005 ), NA,
+ DMm( 80, 2005 ), NA,
+ DMm( 90, 2005 ) )
> par( mfrow=c(1,2), mar=c(3,3,1,1), mgp=c(3,1,0)/1.6 )
> matplot( DMm.1996[,1], DMm.1996[,-1],
+          type="l", lty=1, lwd=c(3,1,1), col=rep(c("blue","red"),each=3),
+          log="y", ylim=c(1,1000), xlim=c(30,95), las=1,
+          xlab="Age", ylab="Mortality rate per 1000 PY" )
> text( 30, 1000, "DM diagnosed 1996", adj=c(0,1) )
> matplot( DMm.2005[,1], DMm.2005[,-1],
+          type="l", lty=1, lwd=c(3,1,1), col=rep(c("blue","red"),each=3),
+          log="y", ylim=c(1,1000), xlim=c(30,95), las=1,
+          xlab="Age", ylab="Mortality rate per 1000 PY" )
> text( 30, 1000, "DM diagnosed 2005", adj=c(0,1) )
```

18. The model we used for the mortality rates used three time-scales: age, calendar time and duration of diabetes.

    It would be of interest to see whether we would get the same (or better) description by adding age at diagnosis and date of diagnosis to the model.

    Now, age at diagnosis = current age − duration of diabetes, and date of diagnosis = current date − duration of diabetes, so the terms we might add only constitute the *non-linear* effects of these variables.

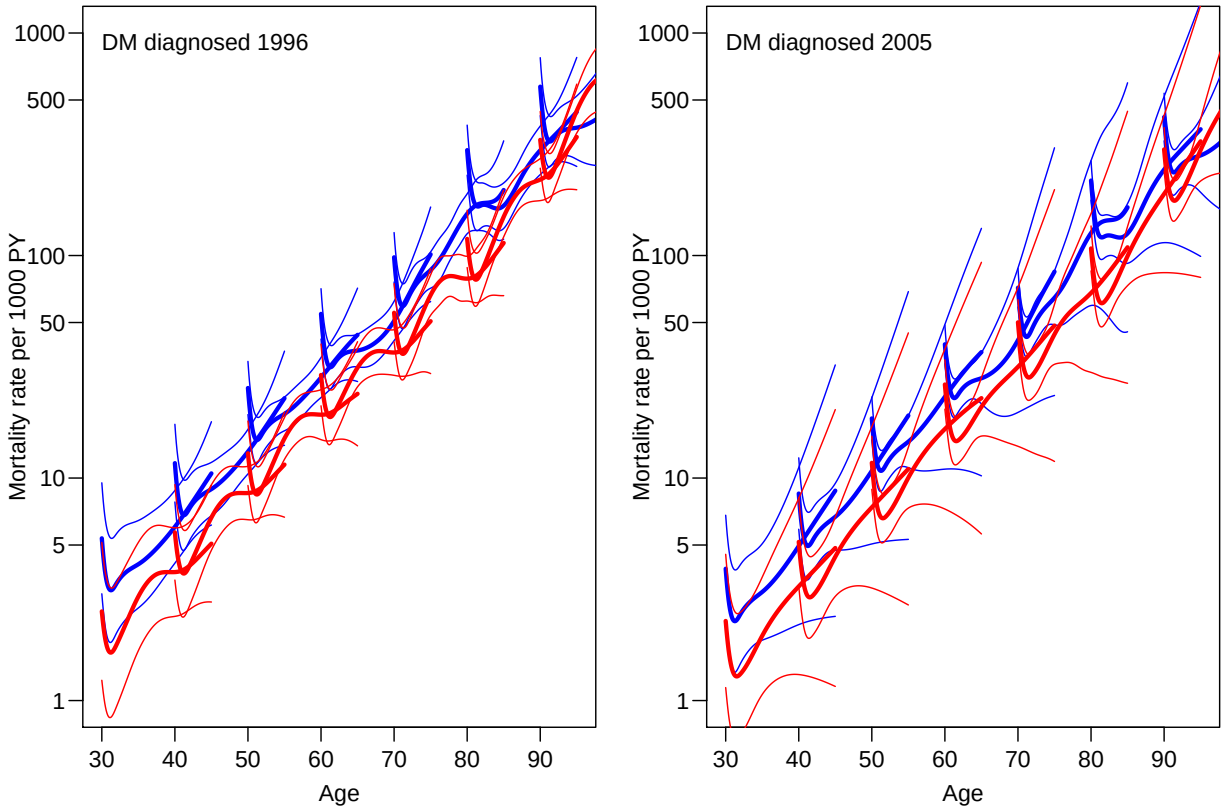

Figure 3.20: *Estimates of mortality of Danish diabetes patients. Note that unlike what is detectable from the plots of the separate effects it seems that for men mortality is higher the younger age at diagnosis, but not for women.*

We add the effects one at at time and test whether age at diagnosis or current age is the better predictor, but we want to use a set of knots which is aligned to the new variables we consider:

```
> kn.Ad <- with( subset( SL, lex.Xst=="Dead" ),
+                quantile( A-dur, probs=seq(5,95,10)/100 ) )
> kn.Pd <- with( subset( SL, lex.Xst=="Dead" ),
+                quantile( P-dur, probs=seq(5,95,20)/100 ) )
> anova( mm,
+        update( mm, . ~ . + Ns(A-dur,knots=kn.Ad) ),
+        update( mm, . ~ . + Ns(A-dur,knots=kn.Ad) - Ns(A,knots=kn.A) ),
+        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, kn = kn.A) + Ns(P, kn = kn.P) + Ns(dur,
      kn = kn.dur) + Ns(A - dur, knots = kn.Ad)
  Model 3: (lex.Xst == "Dead") ~ Ns(P, kn = kn.P) + Ns(dur, kn = kn.dur) +
      Ns(A - dur, knots = kn.Ad)
    Resid. Df Resid. Dev Df Deviance Pr(>Chi)
  1     32681      10024
  2     32673      10014  8   9.6200   0.2927
  3     32681      10024 -8  -9.5799   0.2958

> anova( mm,
+        update( mm, . ~ . + Ns(P-dur,knots=kn.Pd) ),
```

```
+        update( mm, . ~ . + Ns(P-dur,knots=kn.Pd) - Ns(P,knots=kn.P) ),
+        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, kn = kn.A) + Ns(P, kn = kn.P) + Ns(dur,
      kn = kn.dur) + Ns(P - dur, knots = kn.Pd)
  Model 3: (lex.Xst == "Dead") ~ Ns(A, kn = kn.A) + Ns(dur, kn = kn.dur) +
      Ns(P - dur, knots = kn.Pd)
    Resid. Df Resid. Dev Df Deviance Pr(>Chi)
  1     32681      10024
  2     32678      10020  3   4.1180   0.2490
  3     32680      10020 -2  -0.3256   0.8498

> anova( mf,
+        update( mf, . ~ . + Ns(A-dur,knots=kn.Ad) ),
+        update( mf, . ~ . + Ns(A-dur,knots=kn.Ad) - Ns(A,knots=kn.A) ),
+        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, kn = kn.A) + Ns(P, kn = kn.P) + Ns(dur,
      kn = kn.dur) + Ns(A - dur, knots = kn.Ad)
  Model 3: (lex.Xst == "Dead") ~ Ns(P, kn = kn.P) + Ns(dur, kn = kn.dur) +
      Ns(A - dur, knots = kn.Ad)
    Resid. Df Resid. Dev Df Deviance Pr(>Chi)
  1     31411     8752.3
  2     31403     8742.4  8   9.9473   0.2687
  3     31411     8747.4 -8  -5.0449   0.7528

> anova( mf,
+        update( mf, . ~ . + Ns(P-dur,knots=kn.Pd) ),
+        update( mf, . ~ . + Ns(P-dur,knots=kn.Pd) - Ns(P,knots=kn.P) ),
+        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, kn = kn.A) + Ns(P, kn = kn.P) + Ns(dur,
      kn = kn.dur) + Ns(P - dur, knots = kn.Pd)
  Model 3: (lex.Xst == "Dead") ~ Ns(A, kn = kn.A) + Ns(dur, kn = kn.dur) +
      Ns(P - dur, knots = kn.Pd)
    Resid. Df Resid. Dev Df Deviance Pr(>Chi)
  1     31411     8752.3
  2     31408     8750.5  3   1.8693  0.59998
  3     31410     8757.3 -2  -6.8135  0.03315
```

From this it is pretty clear that there is not much difference between using current age or age at diagnosis, and likewise for date of diagnosis, except possibly for period for women, where it seems more appropriate to use current age (since the p-value for removing this from the model is 0.03).

19. So we fit the models with age at diagnosis and date of diagnosis as explanatory variables instead. To this end we also need new contrast matrices, because the deaths are distributed differently along these "entry"-variables, and we therefor placed the knots differently.

```
> AC <- Ns( pr.A, knots=kn.Ad )
> PC <- Ns( pr.P, knots=kn.Pd )
> PR <- Ns( rep(rf.P,N), knots=kn.Pd )
> Mm <- glm( (lex.Xst=="Dead") ~ Ns( A-dur, kn=kn.Ad ) +
+                                Ns( P-dur, kn=kn.Pd ) +
+                                Ns( dur, kn=kn.dur ),
+            offset = log( lex.dur ),
+            family = poisson,
+              data = subset( SL, sex=="M" ) )
> Mf <- update( Mm, data = subset( SL, sex=="F" ) )
> M.A <- ci.exp( Mm, ctr.mat=cbind(1,AC,PR,dR) ) * 1000
> M.P <- ci.exp( Mm, subset="P"  , ctr.mat=PC-PR )
> M.d <- ci.exp( Mm, subset="kn.dur", ctr.mat=dC-dR )
> F.A <- ci.exp( Mf, ctr.mat=cbind(1,AC,PR,dR) ) * 1000
> F.P <- ci.exp( Mf, subset="P"  , ctr.mat=PC-PR )
> F.d <- ci.exp( Mf, subset="kn.dur", ctr.mat=dC-dR )
```

Once the models are fitted, we can plot the estimated effects, as seen in figure **??**

```
> par( mfrow=c(1,3), mar=c(3,3,1,1), mgp=c(3,1,0)/1.6 )
> matplot( pr.A, cbind(M.A,F.A),
+          type="l", lty=1, lwd=c(3,1,1), col=rep(c("blue","red"),each=3),
+          log="y", ylim=c(0.2,180),
+          xlab="Age at diagnosis", ylab="Mortality rate at 2 years duration per 1000 PY" )
> matplot( pr.P,  cbind(M.P,F.P),
+          type="l", lty=1, lwd=c(3,1,1), col=rep(c("blue","red"),each=3),
+          log="y", ylim=c(1/3,3),
+          xlab="Date of diagnosis", ylab="Mortality rate ratio" )
> abline( h=1 )
> matplot( pr.d,  cbind(M.d,F.d),
+          type="l", lty=1, lwd=c(3,1,1), col=rep(c("blue","red"),each=3),
+          log="y", ylim=c(1/3,3),
+          xlab="Diabetes duration", ylab="Mortality rate ratio" )
> abline( h=1 )
```

20. In order to see how the effects from the two approaches using age/date at diagnosis/follow-up relate to each other we can plot them on top of each other:

```
> par( mfrow=c(1,3), mar=c(3,3,1,1), mgp=c(3,1,0)/1.6 )
> matplot( pr.A, cbind(M.A,F.A,m.A,f.A),
+          type="l", lty=rep(c(1,3),each=6), lwd=c(3,1,1), col=rep(c("blue","red"),each=3),
+          log="y", ylim=c(0.2,180),
+          xlab="Age at diagnosis/follow-up", ylab="Mortality rate at 2 years duration per 100
> matplot( pr.P,  cbind(M.P,F.P,m.P,f.P),
+          type="l", lty=rep(c(1,3),each=6), lwd=c(3,1,1), col=rep(c("blue","red"),each=3),
+          log="y", ylim=c(1/3,3),
+          xlab="Date of diagnosis/follow-up", ylab="Mortality rate ratio" )
> abline( h=1 )
> matplot( pr.d,  cbind(M.d,F.d,m.d,f.d),
+          type="l", lty=rep(c(1,3),each=6), lwd=c(3,1,1), col=rep(c("blue","red"),each=3),
+          log="y", ylim=c(1/3,3),
+          xlab="Diabetes duration", ylab="Mortality rate ratio" )
> abline( h=1 )
```

From figure **??** we see that the age and duration curves from the model with two time scales have smaller slopes than those from the model with the age and calendar time as fixed effects. This is because in the latter all the time effect (that is the effect of the clock advancing) is in the duration effect.

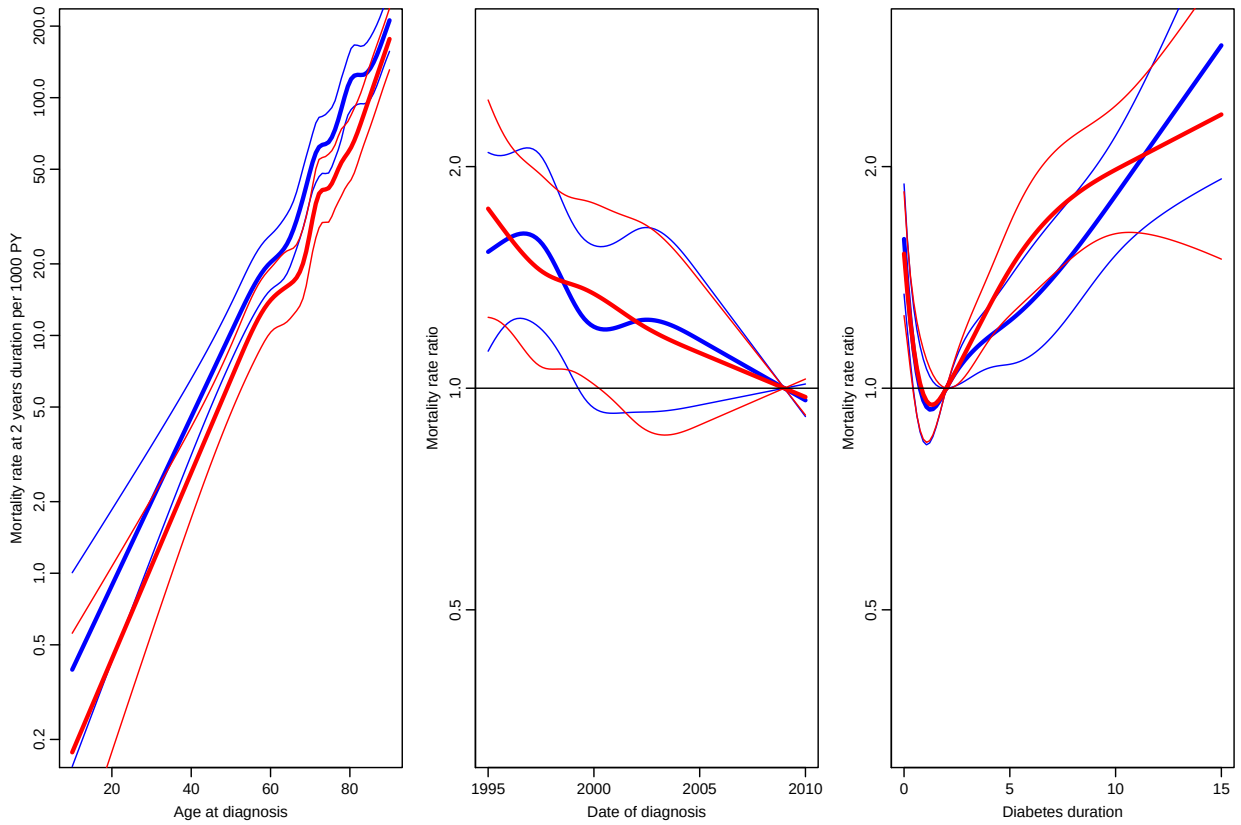

Figure 3.21: *Model for diabetes patient mortality using age and date at diagnosis.*

3.6.1 SMR

The SMR is the standardized mortality ratio, which is 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 toe 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 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.

21. We will use the second approach, that is include as an extra variable the population mortality as available from the data set `M.dk`.

    First we 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

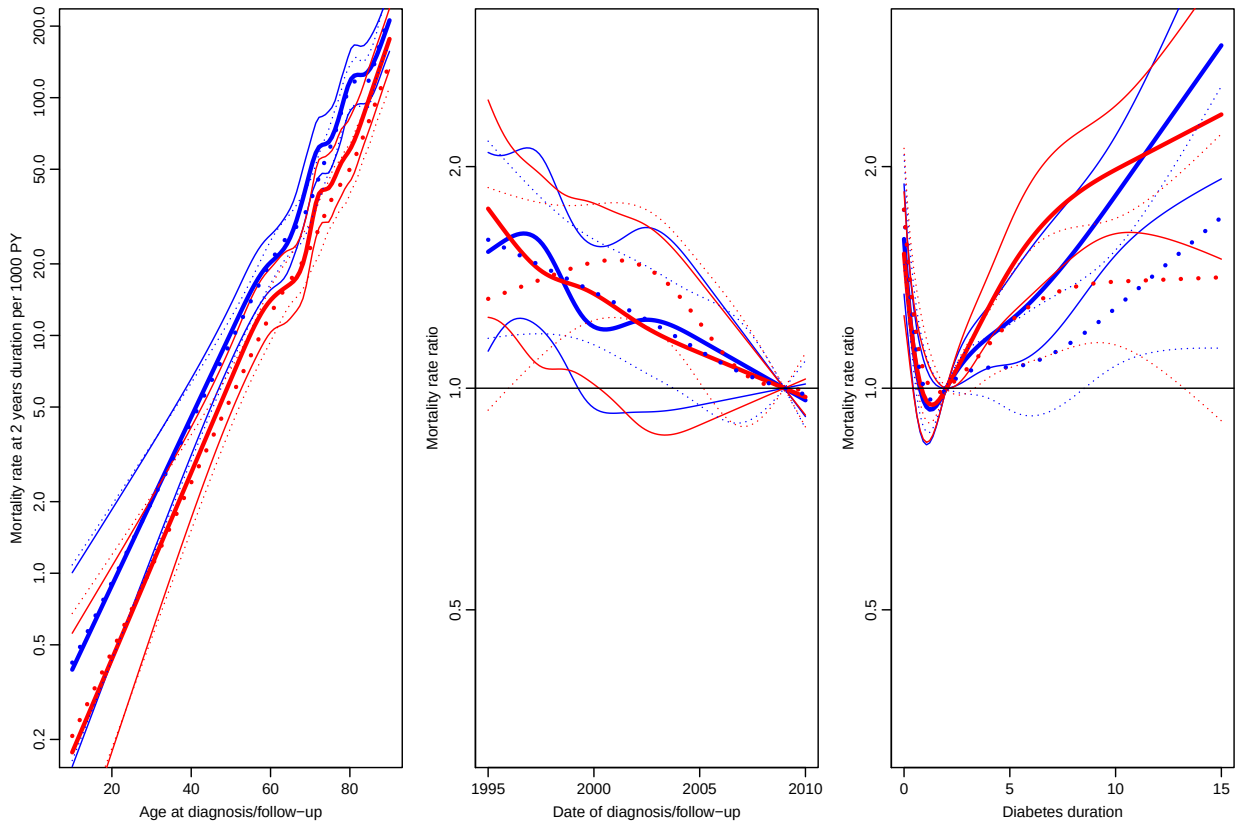

Figure 3.22: *Comparison of estimates from two different models; the full lines give the estimates from the model where age and date are included as fixed variables with the value at diabetes diagnosis, whereas the dotted lines are estimates from the model where age and calendar time are included as time scales.*

the intervals (or rater at a point 3 months after the left end point of the interval — recall we split the follow-up in 6 month intervals).

We need to have variables with the same names in both datasets, moreover, they should be of the same type, so we must transform the sex variable in `M.dk` to a factor:

```
> str( SL )

  Classes 'Lexis' and 'data.frame':        64126 obs. of  14 variables:
   $ lex.id : int  1 1 1 1 1 1 1 1 1 1 ...
   $ A      : num  58.7 59 60 61 62 ...
   $ P      : num  1999 1999 2000 2001 2002 ...
   $ dur    : num  0 0.339 1.339 2.339 3.339 ...
   $ lex.dur: num  0.339 1 1 1 1 ...
   $ 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 1 2 3 4 5 6 7 8 9 ...
```

```
    ..$ P  : NULL
    ..$ dur: NULL
   - attr(*, "time.scales")= chr  "A" "P" "dur"
   - attr(*, "time.since")= chr  "" "" ""
```

```
> SL$Am <- floor( SL$A+0.5 )
> SL$Pm <- floor( SL$P+0.5 )
> 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 ...
```

Then we can match up the rates from `M.dk`:

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

```
  [1] 64126    16
```

```
> dim( SLr )
```

```
  [1] 64114    17
```

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

22. We can now compute the SMR as the observed divided by the expected numbers by say age and sex:

```
> stat.table( list( Age=floor(A/10)*10,
+                     Sex=sex ),
+             list( D=sum(lex.Xst=="Dead"),
+                   E=sum(lex.dur*rate/1000),
+                 SMR=ratio(lex.Xst=="Dead",lex.dur*rate/1000) ),
+             margins = TRUE,
+                data = SLr )
```

```
--------------------------------
          -----------Sex-----------
Age             M       F   Total
--------------------------------
0            0.00    0.00    0.00
             0.02    0.01    0.03
             0.00    0.00    0.00

10           1.00    1.00    2.00
             0.13    0.04    0.17
             7.75   24.84   11.82

20           0.00    0.00    0.00
             0.35    0.18    0.53
             0.00    0.00    0.00

30           5.00    4.00    9.00
             1.43    1.02    2.45
             3.49    3.92    3.67

40          32.00   15.00   47.00
             9.48    5.03   14.51
             3.38    2.98    3.24

50         119.00   62.00  181.00
            48.55   22.03   70.58
             2.45    2.81    2.56

60         275.00  157.00  432.00
           142.55   74.16  216.71
             1.93    2.12    1.99

70         486.00  331.00  817.00
           276.03  204.69  480.71
             1.76    1.62    1.70

80         348.00  423.00  771.00
           255.07  319.26  574.33
             1.36    1.32    1.34

90          76.00  160.00  236.00
            63.41  122.30  185.71
             1.20    1.31    1.27


Total    1342.00 1153.00 2495.00
          797.03  748.71 1545.74
            1.68    1.54    1.61
--------------------------------
```

We see that the SMR is 1.6, but strongly varying with age and to some extent by sex. Moreover, it may seem that the variation with age is not the same for the two sexes.

23. We can now model the SMR by including the log-expected numbers instead of the log-person-years as offset, using separate models for men and women. Also note that we exclude those units where no deaths in the population occur. Also we compute the expected numbers, `E`:

```
> SLr <- subset( SLr, rate>0)
> SLr$E <- SLr$lex.dur * SLr$rate / 1000
> Sm <- glm( (lex.Xst=="Dead") ~ Ns( A-dur, kn=kn.Ad ) +
+                                Ns( P-dur, kn=kn.Pd ) +
+                                Ns( dur, kn=kn.dur ),
```

```
+            offset = log( E ),
+            family = poisson,
+              data = subset( SLr, sex=="M" ) )
> Sf <- update( Sm, data = subset( SLr, sex=="F" ) )
```

The estimates are extracted exactly as for the mortality model; but the results are not mortality rates but rather SMRs (rate-ratios):

```
> sM.A <- ci.exp( Sm, ctr.mat=cbind(1,AC,PR,dR) )
> sM.P <- ci.exp( Sm, subset="P"     , ctr.mat=PC-PR )
> sM.d <- ci.exp( Sm, subset="kn.dur", ctr.mat=dC-dR )
> sF.A <- ci.exp( Sf, ctr.mat=cbind(1,AC,PR,dR) )
> sF.P <- ci.exp( Sf, subset="P"     , ctr.mat=PC-PR )
> sF.d <- ci.exp( Sf, subset="kn.dur", ctr.mat=dC-dR )
```

— plotted using the same code (with obvious adjustments of the axes:

```
> par( mfrow=c(1,3), mar=c(3,3,1,1), mgp=c(3,1,0)/1.6 )
> matplot( pr.A, cbind(sM.A,sF.A),
+          type="l", lty=1, lwd=c(3,1,1), col=rep(c("blue","red"),each=3),
+          log="y", ylim=c(1/3,3),
+          xlab="Age at follow-up", ylab="SMR" )
> abline( h=1 )
> matplot( pr.P,  cbind(sM.P,sF.P),
+          type="l", lty=1, lwd=c(3,1,1), col=rep(c("blue","red"),each=3),
+          log="y", ylim=c(1/3,3),
+          xlab="Date of follow-up", ylab="SMR ratio" )
> abline( h=1 )
> matplot( pr.d,  cbind(sM.d,sF.d),
+          type="l", lty=1, lwd=c(3,1,1), col=rep(c("blue","red"),each=3),
+          log="y", ylim=c(1/3,3),
+          xlab="Diabetes duration", ylab="SMR ratio" )
> abline( h=1 )
```

24. It seems reasonably from figure **??** clear that there is very little difference between SMR for males and females once we controlled for age, date and duration of diabetes. This can be formally tested by fitting models with and without sex-interaction and also a model with no overall effect of sex:

```
> Sb   <- update( Sm, data = SLr )
> Sb.s <- update( Sb, . ~. + sex )
> Sb.i <- update( Sb, . ~. + sex:( Ns( A-dur, kn=kn.Ad  ) +
+                                  Ns( P-dur, kn=kn.Pd  ) +
+                                  Ns(   dur, kn=kn.dur ) ) )
> anova( Sb, Sb.s, Sb.i, test="Chisq" )
Analysis of Deviance Table

Model 1: (lex.Xst == "Dead") ~ Ns(A - dur, kn = kn.Ad) + Ns(P - dur, kn = kn.Pd) +
    Ns(dur, kn = kn.dur)
Model 2: (lex.Xst == "Dead") ~ Ns(A - dur, kn = kn.Ad) + Ns(P - dur, kn = kn.Pd) +
    Ns(dur, kn = kn.dur) + sex
Model 3: (lex.Xst == "Dead") ~ Ns(A - dur, kn = kn.Ad) + Ns(P - dur, kn = kn.Pd) +
    Ns(dur, kn = kn.dur) + Ns(A - dur, kn = kn.Ad):sex + Ns(P -
    dur, kn = kn.Pd):sex + Ns(dur, kn = kn.dur):sex
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1     64083      18764
2     64082      18764  1   0.0004   0.9834
3     64066      18752 16  12.1924   0.7306
```

So we see there is absolutely no difference between the SMR between the sexes.

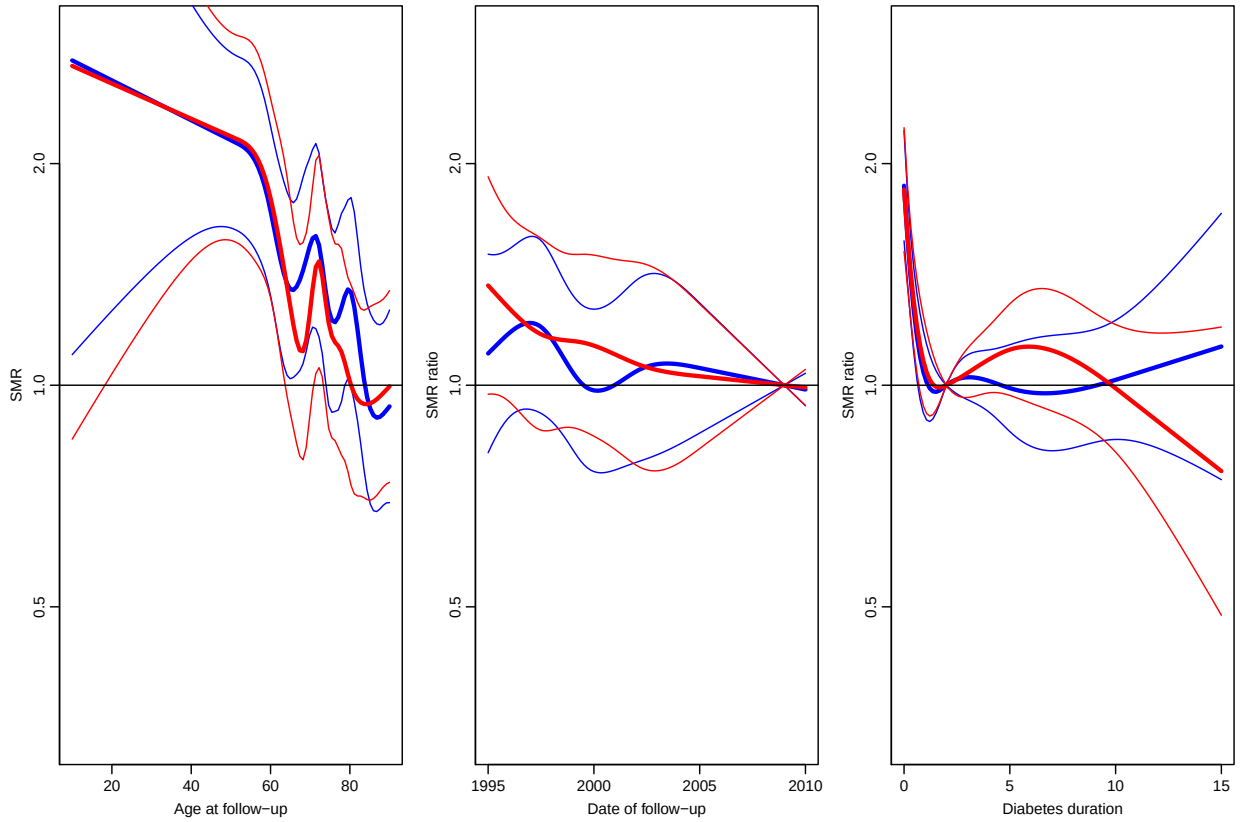

Figure 3.23: *SMR in the diabetic population relative to the (entire) Danish population. Clearly the effect of age is over-modeled.*

25. We therefore extract the parameters from the model with common SMR for the two sexes.

```
> Sb.A <- ci.exp( Sb, ctr.mat=cbind(1,AC,PR,dR) )
> Sb.P <- ci.exp( Sb, subset="P"     , ctr.mat=PC-PR )
> Sb.d <- ci.exp( Sb, subset="kn.dur", ctr.mat=dC-dR )
> par( mfrow=c(1,3), mar=c(3,3,1,1), mgp=c(3,1,0)/1.6 )
> matplot( pr.A, Sb.A,
+          type="l", lty=1, lwd=c(3,1,1), col="black",
+          log="y", ylim=c(1/3,3),
+          xlab="Age at diagnosis", ylab="SMR" )
> abline( h=1 )
> matplot( pr.P, Sb.P,
+          type="l", lty=1, lwd=c(3,1,1), col="black",
+          log="y", ylim=c(1/3,3),
+          xlab="Date of diagnosis", ylab="SMR ratio" )
> abline( h=1 )
> matplot( pr.d, Sb.d,
+          type="l", lty=1, lwd=c(3,1,1), col="black",
+          log="y", ylim=c(1/3,3),
+          xlab="Diabetes duration", ylab="SMR ratio" )
> abline( h=1 )
```

26. We can simplify the model to one the is easier to convey to users by using a 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. At the same time we also limit the number of knots for the age-effect:

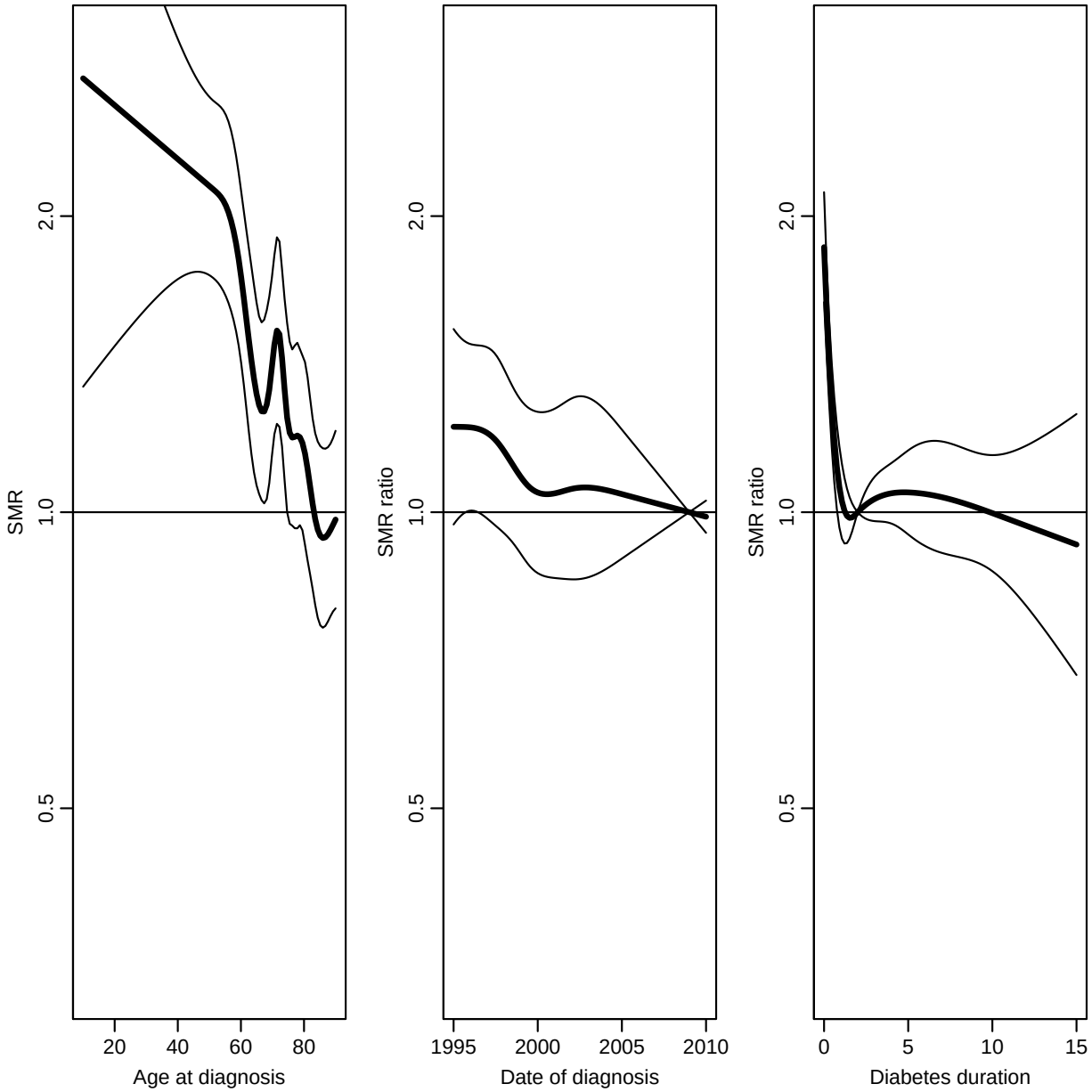

Figure 3.24: *SMR in the diabetic population for both sexes, relative to the (entire) Danish population.*

```
> kn.Ad <- with( subset( SL, lex.Xst=="Dead" ),
+                  quantile( A-dur, probs=seq(5,95,20)/100 ) )
> kn.dur <- 0:2
> AC <- Ns( pr.A, knots=kn.Ad )
> dC <- Ns( pr.d, knots=kn.dur )
> dR <- Ns( rep(rf.d,N), knots=kn.dur )
> Sx <- glm( (lex.Xst=="Dead") ~ Ns( A-dur, kn=kn.Ad ) +
+                                  I( P-dur ) +
+                                 Ns(   dur, kn=kn.dur ),
+            offset = log( E ),
+            family = poisson,
+              data = SLr )
```

Having fitted the model, we can then plot the estimates from it:

```
> Sx.A <- ci.exp( Sx, ctr.mat=cbind(1,AC,rf.P,dR) )
> Sx.P <- ci.exp( Sx, subset="P"     , ctr.mat=cbind(pr.P-rf.P) )
> Sx.d <- ci.exp( Sx, subset="kn.dur", ctr.mat=dC-dR )
> par( mfrow=c(1,3), mar=c(3,3,1,1), mgp=c(3,1,0)/1.6 )
> matplot( pr.A, Sx.A,
+          type="l", lty=1, lwd=c(3,1,1), col="black",
+          log="y", ylim=c(1/2,4),
+          xlab="Age at diagnosis", ylab="SMR" )
> abline( h=1 )
> abline( v=4:8*10, col="gray" )
> matplot( pr.P, Sx.P,
+          type="l", lty=1, lwd=c(3,1,1), col="black",
+          log="y", ylim=c(1/2,4),
+          xlab="Date of diagnosis", ylab="SMR ratio" )
> abline( h=1 )
> matplot( pr.d, Sx.d,
+          type="l", lty=1, lwd=c(3,1,1), col="black",
+          log="y", ylim=c(1/2,4),
+          xlab="Diabetes duration", ylab="SMR ratio" )
> abline( h=1,v=2 )
```

27. We can formulate the period and duration effects by looking at the estimated parameters:

```
> 100*( 1 - ci.exp( Sx, subset="P" ) )

             exp(Est.)     2.5%     97.5%
I(P - dur)  1.539058 2.713708 0.3502251
```

Thus the change in SMR is a 1.5% annuaal decrease (95% c.i.: (0.3-2.7)%).

If we want to assess the annual change in SMR by duration of diabetes we can calculate the duration effects at say 5 and 6 years and subtract them:

```
> d6 <- Ns( 6, knots=kn.dur )
> d5 <- Ns( 5, knots=kn.dur )
> 100*( ci.exp( Sx, subset="kn.dur", ctr.mat=d6-d5 ) - 1 )

      exp(Est.)      2.5%   97.5%
[1,] 0.2676805 -1.369183 1.93171
```

Thus the estimate is an annual increase in SMR or 0.3% (-1.3–1.9)%, thus no evidence of any increasing SMR after 2 years of diabetes duration.

The conclusion is that SMR for diabetes patients diagnosed at age 50 is about 2 after two years of duration and does not change, whereas it for patients aged 70 is about 1.4 after 2 years of diabetes and does not change. The SMR is initially (just after diagnosis) about twice as high, and does not change.

3.6.2 Interaction models

28. We may explore whether there is an interaction between age and duration by including a product of the duration effects and age at diagnosis:

```
> Six <- update( Sx, . ~. + I(A-dur):Ns(dur,knots=kn.dur) )
> anova( Six, Sx, test="Chisq" )
```

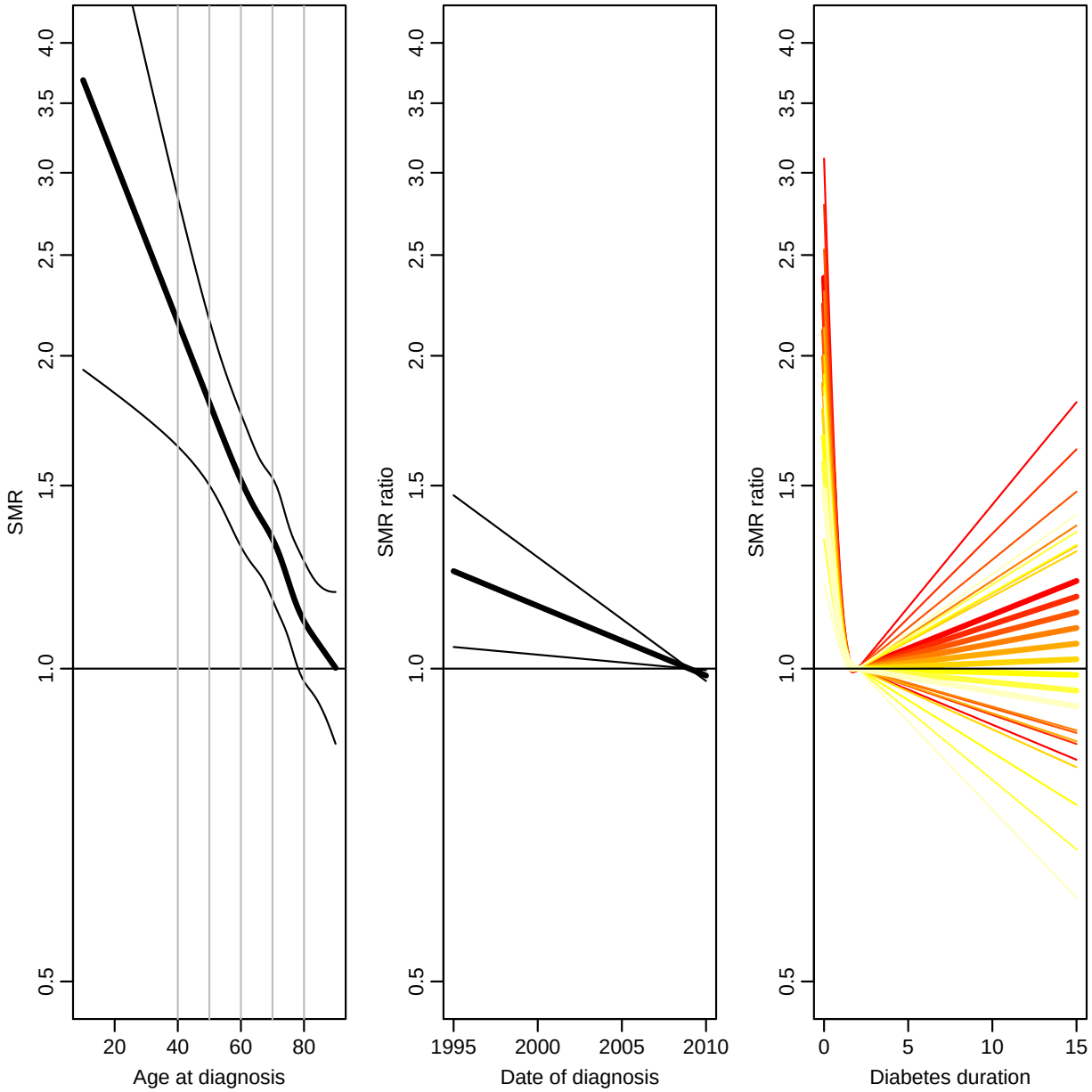

Figure 3.25: *SMR in the diabetic population for both sexes, relative to the (entire) Danish population — simplified model.*

```
Analysis of Deviance Table

Model 1: (lex.Xst == "Dead") ~ Ns(A - dur, kn = kn.Ad) + I(P - dur) +
    Ns(dur, kn = kn.dur) + Ns(dur, kn = kn.dur):I(A - dur)
Model 2: (lex.Xst == "Dead") ~ Ns(A - dur, kn = kn.Ad) + I(P - dur) +
    Ns(dur, kn = kn.dur)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1     64091      18783
2     64093      18789 -2  -5.1995  0.07429

> ci.exp( Six )

                                 exp(Est.)         2.5%        97.5%
(Intercept)                  1.190491e+14 4.349094e+03 3.258768e+24
Ns(A - dur, kn = kn.Ad)1     5.952633e-01 4.728864e-01 7.493099e-01
```

```
Ns(A - dur, kn = kn.Ad)2          5.200933e-01 4.197792e-01 6.443794e-01
Ns(A - dur, kn = kn.Ad)3          3.244669e-01 2.310045e-01 4.557434e-01
Ns(A - dur, kn = kn.Ad)4          5.027709e-01 4.076581e-01 6.200750e-01
I(P - dur)                        9.846898e-01 9.729425e-01 9.965789e-01
Ns(dur, kn = kn.dur)1             8.385022e-02 2.277759e-02 3.086744e-01
Ns(dur, kn = kn.dur)2             4.585490e-01 3.091840e-01 6.800713e-01
Ns(dur, kn = kn.dur)1:I(A - dur) 1.020444e+00 1.002657e+00 1.038546e+00
Ns(dur, kn = kn.dur)2:I(A - dur) 1.006233e+00 1.000913e+00 1.011582e+00
```

Even if the effect is not statistically significant, we would still want to explore the shape of it:

```
> Six.A <- ci.exp( Six, ctr.mat=cbind(1,AC,rf.P,dR,dR*pr.A) )
> Six.P <- ci.exp( Six, subset="P"     , ctr.mat=cbind(pr.P-rf.P) )
> Six.d <- ci.exp( Six, subset="kn.dur", ctr.mat=cbind(dC-dR,(dC-dR)*50) )
> for( a in seq(55,90,5) ) Six.d <- cbind( Six.d,
+                 ci.exp( Six, subset="kn.dur", ctr.mat=cbind(dC-dR,(dC-dR)*a) ) )
> dim( Six.d )

  [1] 100  27

> par( mfrow=c(1,3), mar=c(3,3,1,1), mgp=c(3,1,0)/1.6 )
> matplot( pr.A, Six.A,
+          type="l", lty=1, lwd=c(3,1,1), col="black",
+          log="y", ylim=c(1/2,4),
+          xlab="Age at diagnosis", ylab="SMR" )
> abline( h=1 )
> abline( v=4:8*10, col="gray" )
> matplot( pr.P, Six.P,
+          type="l", lty=1, lwd=c(3,1,1), col="black",
+          log="y", ylim=c(1/2,4),
+          xlab="Date of diagnosis", ylab="SMR ratio" )
> abline( h=1 )
> matplot( pr.d, Six.d,
+          type="l", lty=1, lwd=c(3,1,1), col=rep(heat.colors(9),each=3),
+          log="y", ylim=c(1/2,4),
+          xlab="Diabetes duration", ylab="SMR ratio" )
> abline( h=1 )
```

29. This approach is however a bit artificial, because we have fixed the duration effects to be 1 at duration 2 years. It would be appropriate to combine the effects of age at diagnosis and duration to show how the SMR looks as a function of current age.

```
> pts <- c(seq(0,15,0.1),NA)
> np <- length( pts )
> nd <- data.frame( A=rep(seq(50,90,5),each=np)+pts,
+                   P=rf.P+pts,
+                 dur=     pts,
+                   E=1 )
> A.si <- exp(sapply(predict( Six, newdata=nd, se.fit=TRUE )[1:2],cbind) %*% ci.mat())
> A.sm <- exp(sapply(predict( Sx , newdata=nd, se.fit=TRUE )[1:2],cbind) %*% ci.mat())



> matplot( NA, NA,
+          log="y", ylim=c(1/2,5), xlim=c(50,100),
+          xlab="Age at follow-up", ylab="SMR" )
> abline( h=c(5:19/10,seq(2,5,0.5)), v=seq(50,100,5), col=gray(0.8) )
> matlines( nd$A, cbind(A.si,A.sm),
+           type="l", lty=rep(c(1,3),each=3), lwd=c(3,1,1), col="forestgreen" )
> abline( h=1 )
```

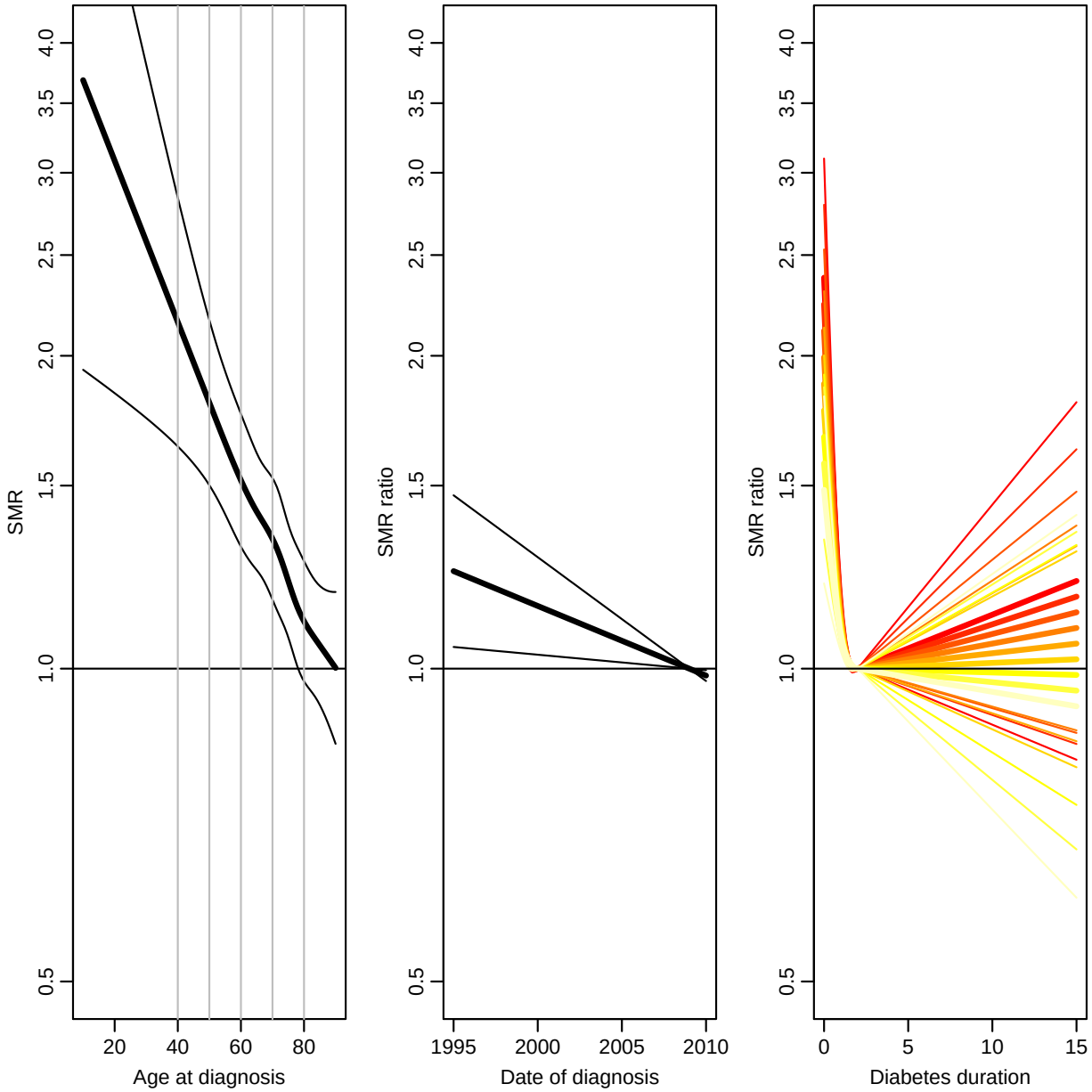

Figure 3.26: *SMR in the diabetic population for both sexes, relative to the (entire) Danish population — interaction model with age-specific duration effects.*

From figure **??** it is clear that the interaction means that the patients diagnosed at young age (50–60, that is) do not experience a declining SMR, on the contrary, they have a relative mortality that is close to what it is a year or so after diagnosis, which is about 2 for 50-year olds , 1.4 for 70 year olds and 1.1 for 80 year olds

30. This interaction machinery with linear age easily generalizes to more complex age-effects, it is just a question of choosing another age-effect:

```
> SiX <- update( Sx, . ~. + Ns(A-dur,knots=kn.Ad):Ns(dur,knots=kn.dur) )
> anova( SiX, Six, Sx, test="Chisq" )

  Analysis of Deviance Table
```

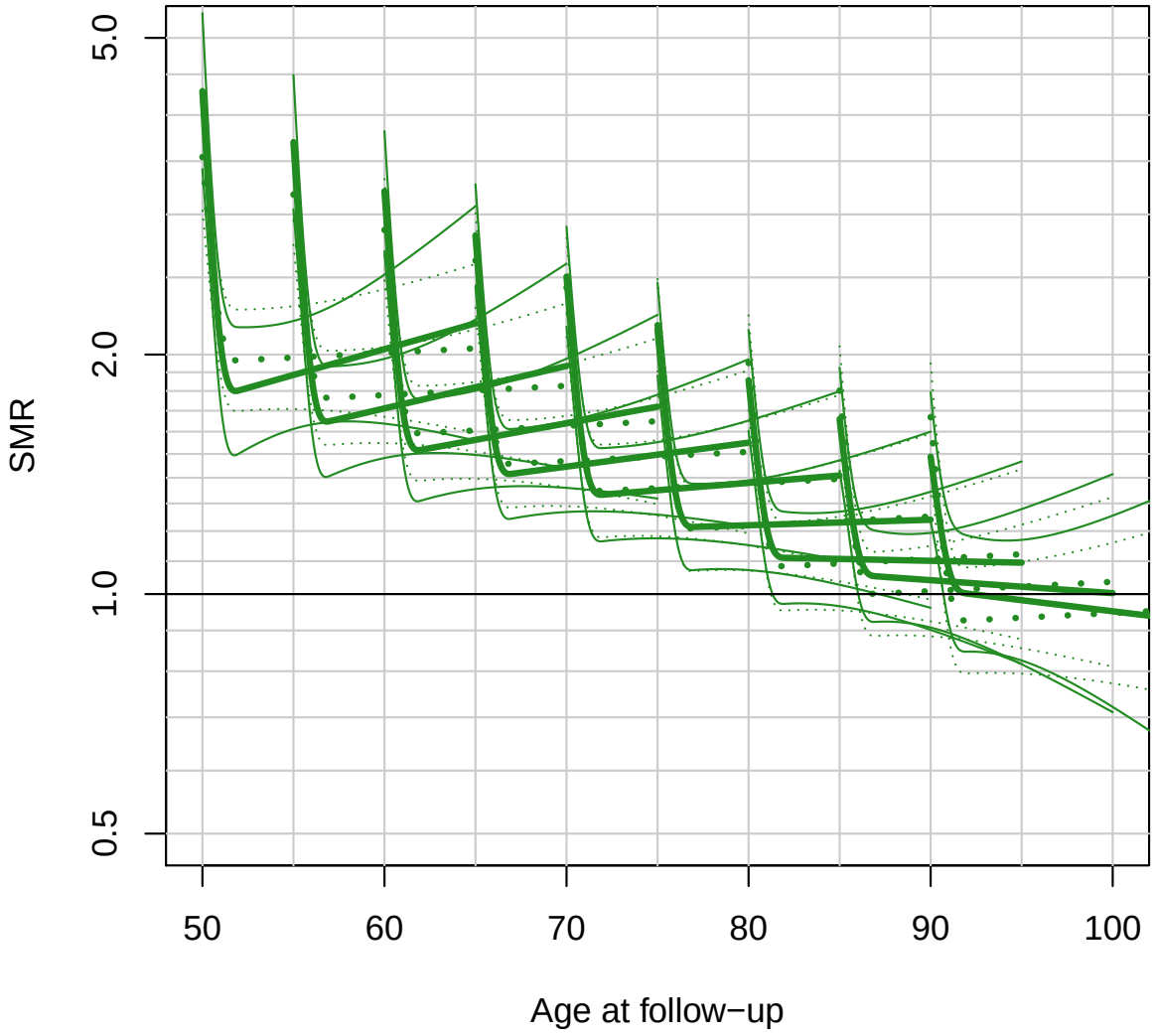

Figure 3.27: *SMR in the diabetic population for both sexes, relative to the (entire) Danish population — interaction model with age-specific duration effects, shown for patients diagnosed at ages 50 to 90.*

```
Model 1: (lex.Xst == "Dead") ~ Ns(A - dur, kn = kn.Ad) + I(P - dur) +
    Ns(dur, kn = kn.dur) + Ns(A - dur, kn = kn.Ad):Ns(dur, kn = kn.dur)
Model 2: (lex.Xst == "Dead") ~ Ns(A - dur, kn = kn.Ad) + I(P - dur) +
    Ns(dur, kn = kn.dur) + Ns(dur, kn = kn.dur):I(A - dur)
Model 3: (lex.Xst == "Dead") ~ Ns(A - dur, kn = kn.Ad) + I(P - dur) +
    Ns(dur, kn = kn.dur)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1     64085      18777
2     64091      18783 -6  -6.1491  0.40670
3     64093      18789 -2  -5.1995  0.07429
```

And we can use the exact same code to show the interaction and plot it along the

others in a similar plot:

```
> A.sX <- exp(sapply(predict( SiX, newdata=nd, se.fit=TRUE )[1:2],cbind) %*% ci.mat())
> matplot( NA, NA,
+          log="y", ylim=c(1/2,5), xlim=c(50,100),
+          xlab="Age at follow-up", ylab="SMR" )
> abline( h=c(5:19/10,seq(2,5,0.5)), v=seq(50,100,5), col=gray(0.8) )
> matlines( nd$A, cbind(A.sX,A.si,A.sm),
+           type="l", lty=rep(c(1,3),c(6,3)), lwd=c(3,1,1),
+           col=rep(c("magenta","forestgreen"),c(3,6)) )
> abline( h=1 )
```

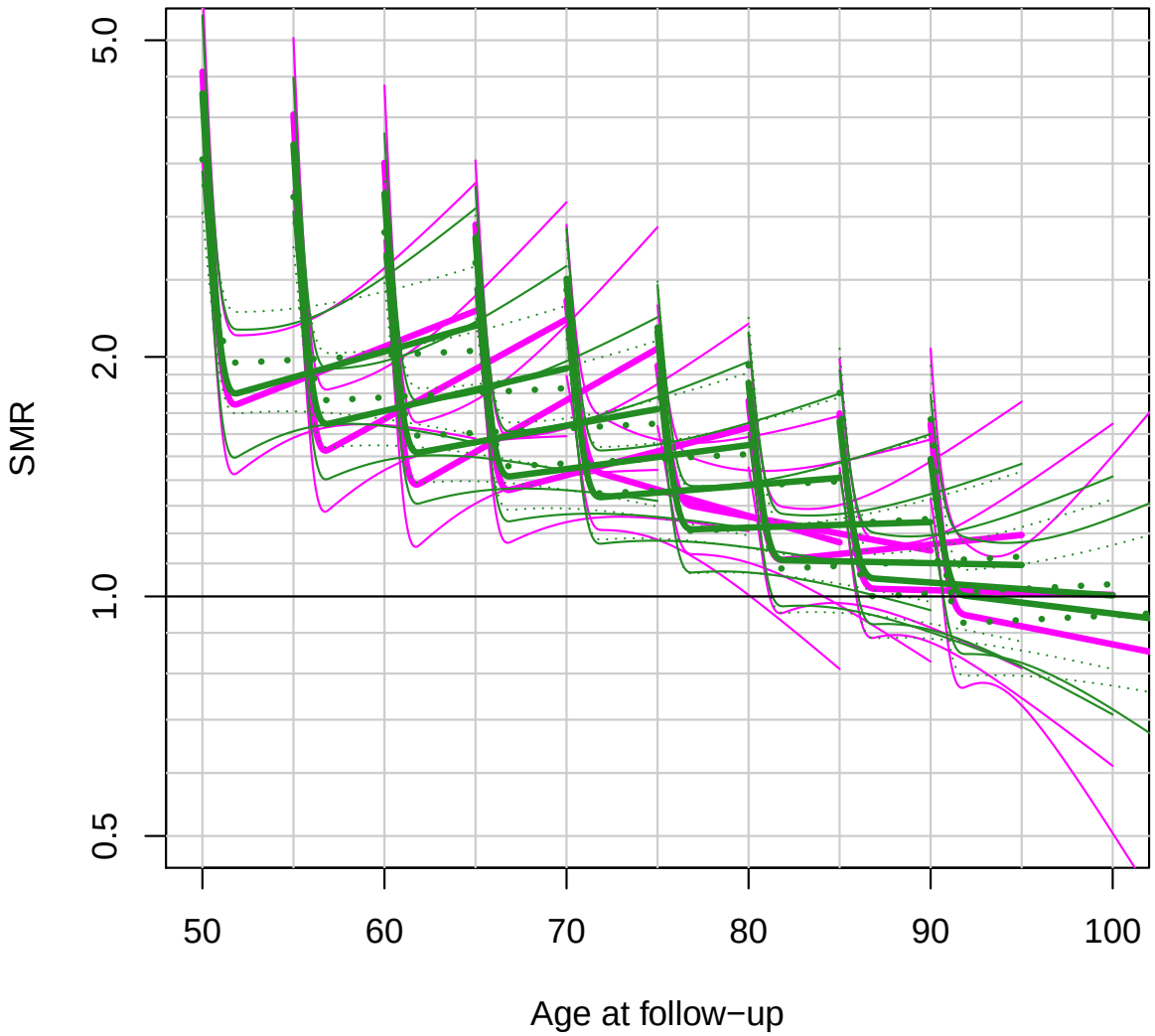

Figure 3.28: *SMR in the diabetic population for both sexes, relative to the (entire) Danish population — interaction model with age-specific duration effects, shown for patients diagnosed at ages 50, 60, 70, 80 and 90. The bright green curves are from the more complex interaction model.*

From figure **??** it is seen that the interaction chosen was way too complex; the long-term variations in the SMR as estimated here do not seem believable. Although the general pattern is pretty much the same; it is the age at diagnosis that determines the SMR.

Chapter 4

Demography of diabetes in Scotland

This exercise is based on population data (risk time and deaths) and diabetes data (individual records of date of birth, DM diagnosis and death) from Scotland.

The purpose of the exercise is to derive measures of prevalence, incidence and mortality from diabetes in Scotland. The data are *actual* data, no shortcuts, so this is a pretty long exercise in 7 sections: Data, Prevalence, Incidence, Mortality, Survival, SMR and Life lost.

4.1 Data

There are two datafiles for this exercise, the mid-year population in 1-year age intervals subdivided by sex and residential area in social deprivation deciles, and individual records.

First, load the `Epi` package:

```
library( Epi )
sessionInfo()
 R version 3.1.0 (2014-04-10)
 Platform: i386-w64-mingw32/i386 (32-bit)

 locale:
 [1] LC_COLLATE=Danish_Denmark.1252  LC_CTYPE=Danish_Denmark.1252
 [3] LC_MONETARY=Danish_Denmark.1252 LC_NUMERIC=C
 [5] LC_TIME=Danish_Denmark.1252

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

 other attached packages:
 [1] Epi_1.1.67     foreign_0.8-61
```

4.1.1 Population data

The population date contains the mid-year population and the number of deaths for Scotland by sex, deprivation index and 1-year classes of age and calendar time.

Then read the population data from the `.csv`-file, and note the funny name of the first variable:

```
pop <- read.csv( "../data/PopulationSIMD2009V2.csv" )
names( pop ) <- tolower( names(pop) )
names( pop )
 [1] "ï..year"  "age"      "sex"      "simd2009" "n_deaths" "pop"
```

```
names( pop )[c(1,4:6)] <- c("per","simd","D","N")
pop$sex <- factor( pop$sex, labels=c("M","F") )
str( pop )
```

```
 'data.frame':        15452 obs. of  6 variables:
  $ per : int  2005 2005 2005 2005 2005 2005 2005 2005 2005 2005 ...
  $ age : int  0 0 0 0 0 0 0 0 0 0 ...
  $ sex : Factor w/ 2 levels "M","F": 1 1 1 1 1 1 1 1 1 ...
  $ simd: int  1 2 3 4 5 6 7 8 9 10 ...
  $ D   : int  21 17 22 20 13 10 20 14 9 8 ...
  $ N   : int  3823 3240 2907 2754 2639 2588 2600 2542 2529 2471 ...
```

```
head( pop )
```

```
    per age sex simd  D    N
1 2005   0   M    1 21 3823
2 2005   0   M    2 17 3240
3 2005   0   M    3 22 2907
4 2005   0   M    4 20 2754
5 2005   0   M    5 13 2639
6 2005   0   M    6 10 2588
```

```
tail( pop, 25 )
```

```
        per age sex simd   D    N
15428 2012  89   F    9  89  720
15429 2012  89   F   10  92  695
15430 2012  89   F   11   1   NA
15431 2012  90   M    1 153  679
15432 2012  90   M    2 187  789
15433 2012  90   M    3 240  971
15434 2012  90   M    4 274 1068
15435 2012  90   M    5 254  967
15436 2012  90   M    6 250 1063
15437 2012  90   M    7 256 1104
15438 2012  90   M    8 264 1082
15439 2012  90   M    9 237  963
15440 2012  90   M   10 247 1153
15441 2012  90   M   11   7   NA
15442 2012  90   F    1 497 2101
15443 2012  90   F    2 550 2413
15444 2012  90   F    3 568 2690
15445 2012  90   F    4 660 2970
15446 2012  90   F    5 670 2799
15447 2012  90   F    6 682 2861
15448 2012  90   F    7 647 2919
15449 2012  90   F    8 714 2947
15450 2012  90   F    9 614 2643
15451 2012  90   F   10 581 2723
15452 2012  90   F   11  14   NA
```

```
summary( pop )
```

```
      per            age        sex           simd             D               N        
 Min.   :2005   Min.   : 0.00   M:7802   Min.   : 1.000   Min.   :  1.0   Min.   : 157  
 1st Qu.:2006   1st Qu.:23.00   F:7650   1st Qu.: 3.000   1st Qu.:  3.0   1st Qu.:2446  
 Median :2008   Median :46.00            Median : 6.000   Median : 13.0   Median :3020  
 Mean   :2008   Mean   :45.75            Mean   : 5.817   Mean   : 33.5   Mean   :2866  
 3rd Qu.:2010   3rd Qu.:68.00            3rd Qu.: 8.000   3rd Qu.: 51.0   3rd Qu.:3569  
 Max.   :2012   Max.   :90.00            Max.   :11.000   Max.   :714.0   Max.   :4720  
                                                          NA's   :2339    NA's   :892   
```

Note that the code 11 for `simd` is used only in conjunction with deaths that it has not been possible to allocate to a particular geographical region. Moreover, it seems that for those combinations of the classifying factors that have 0 deaths are coded 0 for the variable `D`.

```
with( pop, ftable( per, sex, simd ) )
```

```
          simd  1  2  3  4  5  6  7  8  9 10 11
per  sex
2005 M       91 91 91 91 91 91 91 91 91 91 69
     F       91 91 91 91 91 91 91 91 91 91 50
2006 M       91 91 91 91 91 91 91 91 91 91 70
     F       91 91 91 91 91 91 91 91 91 91 51
2007 M       91 91 91 91 91 91 91 91 91 91 64
     F       91 91 91 91 91 91 91 91 91 91 53
2008 M       91 91 91 91 91 91 91 91 91 91 68
     F       91 91 91 91 91 91 91 91 91 91 44
2009 M       91 91 91 91 91 91 91 91 91 91 66
     F       91 91 91 91 91 91 91 91 91 91 49
2010 M       91 91 91 91 91 91 91 91 91 91 63
     F       91 91 91 91 91 91 91 91 91 91 42
2011 M       91 91 91 91 91 91 91 91 91 91 61
     F       91 91 91 91 91 91 91 91 91 91 43
2012 M       91 91 91 91 91 91 91 91 91 91 61
     F       91 91 91 91 91 91 91 91 91 91 38
```

```
round( ftable( xtabs( N/1000 ~
                    sex + per + simd,
                    data=pop ) ), 1 )
```

```
          simd     1     2     3     4     5     6     7     8     9    10
sex per
M   2005      245.3 244.8 245.7 246.9 247.9 246.5 246.4 243.0 247.3 247.5
    2006      247.0 245.7 246.8 248.3 249.9 248.4 248.9 244.7 248.2 247.2
    2007      248.5 246.3 247.9 250.2 252.6 251.5 252.8 248.5 250.3 248.0
    2008      251.1 247.2 249.5 252.5 255.1 254.8 255.8 251.1 251.4 246.7
    2009      252.5 248.4 251.6 254.7 256.8 257.1 257.8 253.9 251.9 247.1
    2010      254.8 250.4 254.0 256.4 258.9 258.3 259.9 256.2 251.6 247.7
    2011      256.0 251.8 256.9 258.6 261.8 260.3 262.9 259.4 253.3 249.4
    2012      256.1 251.8 257.0 259.0 262.5 261.3 265.0 261.8 252.9 249.8
F   2005      275.8 273.0 270.8 268.7 263.2 260.6 260.2 257.6 257.6 261.4
    2006      275.8 273.0 271.2 268.8 264.6 262.4 262.7 259.6 258.9 260.9
    2007      275.9 273.1 271.5 270.4 266.8 265.2 266.5 262.9 260.5 260.6
    2008      277.6 273.2 272.9 271.5 268.7 267.7 268.6 265.7 261.7 260.0
    2009      277.7 273.4 273.5 273.1 270.4 269.2 270.5 268.4 263.1 260.6
    2010      278.1 273.6 275.3 273.5 272.4 271.1 273.4 270.7 264.1 261.8
    2011      277.9 274.2 276.7 275.0 274.3 273.4 275.6 274.6 265.1 262.7
    2012      277.7 273.6 276.8 274.9 275.3 274.2 276.8 277.0 266.7 263.5
```

```
odd <- function( x ) x[length(x)]/sum(x) * 1000
ftable( xtabs( D ~ sex + simd + per, data=pop ) )
```

```
          per 2005 2006 2007 2008 2009 2010 2011 2012
sex simd
M   1         3586 3565 3630 3474 3407 3350 3320 3257
    2         3334 3354 3413 3256 3156 3130 3108 3039
    3         3201 3192 3195 3141 3055 2932 2961 3003
    4         2975 2956 2989 2924 2833 2933 2885 2882
    5         2758 2700 2752 2824 2679 2729 2738 2759
    6         2503 2439 2558 2577 2521 2568 2585 2580
    7         2344 2304 2349 2309 2339 2289 2275 2375
    8         2082 2125 2133 2117 2068 2202 2252 2218
    9         1895 1779 1849 1835 1919 1860 1883 1949
    10        1681 1669 1702 1792 1666 1782 1762 1816
    11         254  229  234  239  220  185  179  172
F   1         3543 3502 3392 3470 3292 3298 3213 3300
    2         3430 3455 3342 3476 3295 3259 3197 3359
    3         3496 3431 3445 3346 3261 3144 3168 3292
    4         3260 3225 3206 3334 3151 3144 3116 3160
    5         2987 2960 2997 3073 2926 2881 2851 3032
    6         2920 2891 2905 2838 2827 2836 2822 2951
    7         2689 2656 2732 2658 2512 2629 2588 2654
    8         2519 2549 2546 2619 2532 2514 2508 2679
    9         2224 2123 2224 2209 2115 2188 2227 2313
    10        2044 2041 2012 2005 2009 2008 2008 2130
    11         203  172  131  122  145  105  118  104
```

```
round( ftable( addmargins( xtabs( D ~ sex + simd + per, data=pop ),
                           margin = 3:2,
                           FUN = list( list("2005-12"=sum),
                                       list(sum,"'11' (per 1000)"=odd) ) ) ) )
```

```
Margins computed over dimensions
in the following order:
1: per
2: simd
                    per   2005   2006   2007   2008   2009   2010   2011   2012 2005-12
sex simd
M   1                     3586   3565   3630   3474   3407   3350   3320   3257   27589
    2                     3334   3354   3413   3256   3156   3130   3108   3039   25790
    3                     3201   3192   3195   3141   3055   2932   2961   3003   24680
    4                     2975   2956   2989   2924   2833   2933   2885   2882   23377
    5                     2758   2700   2752   2824   2679   2729   2738   2759   21939
    6                     2503   2439   2558   2577   2521   2568   2585   2580   20331
    7                     2344   2304   2349   2309   2339   2289   2275   2375   18584
    8                     2082   2125   2133   2117   2068   2202   2252   2218   17197
    9                     1895   1779   1849   1835   1919   1860   1883   1949   14969
    10                    1681   1669   1702   1792   1666   1782   1762   1816   13870
    11                     254    229    234    239    220    185    179    172    1712
    sum                  26613  26312  26804  26488  25863  25960  25948  26050  210038
    '11' (per 1000)         10      9      9      9      9      7      7      7       8
F   1                     3543   3502   3392   3470   3292   3298   3213   3300   27010
    2                     3430   3455   3342   3476   3295   3259   3197   3359   26813
    3                     3496   3431   3445   3346   3261   3144   3168   3292   26583
    4                     3260   3225   3206   3334   3151   3144   3116   3160   25596
    5                     2987   2960   2997   3073   2926   2881   2851   3032   23707
    6                     2920   2891   2905   2838   2827   2836   2822   2951   22990
    7                     2689   2656   2732   2658   2512   2629   2588   2654   21118
    8                     2519   2549   2546   2619   2532   2514   2508   2679   20466
    9                     2224   2123   2224   2209   2115   2188   2227   2313   17623
    10                    2044   2041   2012   2005   2009   2008   2008   2130   16257
    11                     203    172    131    122    145    105    118    104    1100
    sum                  29315  29005  28932  29150  28065  28006  27816  28974  229263
    '11' (per 1000)          7      6      5      4      5      4      4      4       5
```

The last two commands are slightly cryptic (see the help page for `addmargins`); the second number in the margin over `simd` is the fraction of unclassified deaths in 1/1000s, so we see that we have less than 1% unclassified deaths, hence we shall exclude these from the data, since the mortality in the population will be underestimated by less than 1%. Moreover, it seems that for the combinatiosn of sex, age, year and social class where there are no deaths the number of detahs is coded `NA` instaed of 0, so we fix that too:

```
pop <- subset( pop, simd < 11 )
pop$D <- pmax( 0, pop$D, na.rm=TRUE )
```

4.1.2 Diabetes data

The Scottish diabetes data, that contains all diabetes patients in Scotland alive at 1.1.2005 or diagnosed later, and followed for death until 18 May 2012 are in the file `dm_data.csv`.

Read the data using `read.csv` (consult the help page for this) and inspect the data. Note that the character values in the file are converted to factors, but they can be referred to as character variables when converted to dates by `as.Date`:

```
DM  <- read.csv( "../data/dm_data.csv" )
names( DM )
```

```
 [1] "simd_decile" "sex"         "DMtype"      "dod"         "dob"         "doDM"
```

```
str( DM )
```

```
 'data.frame':        300144 obs. of  6 variables:
  $ simd_decile: int  5 4 4 8 2 3 4 2 8 6 ...
  $ sex        : Factor w/ 2 levels "Female","Male": 2 2 2 1 1 1 2 2 1 1 ...
  $ DMtype     : int  2 2 2 2 2 2 2 2 2 2 ...
  $ dod        : Factor w/ 2696 levels "","01/01/2005",..: 2 2 2 2 2 2 2 2 2 2 ...
  $ dob        : Factor w/ 9777 levels "01/01/1906","01/01/1909",..: 1369 7442 1850 3310 4262 1459
  $ doDM       : Factor w/ 15435 levels "","01/01/1945",..: 12528 7827 10746 5559 9146 5026 13639 1
```

```
for( i in 4:6 ) DM[,i] <- cal.yr( as.Date( DM[,i], format="%d/%m/%Y" ) )
names( DM )[1] <- "simd"
levels( DM$sex ) <- c("F","M")
head( DM )
```

```
   simd sex DMtype      dod      dob     doDM
1    5   M      2 2005.001 1929.170 2000.815
2    4   M      2 2005.001 1925.061 1996.538
3    4   M      2 2005.001 1928.259 1997.387
4    8   F      2 2005.001 1936.185 1994.942
5    2   F      2 2005.001 1943.029 2000.133
6    3   F      2 2005.001 1924.256 2003.856
```

```
summary( DM )
```

```
       simd         sex            DMtype           dod              dob
 Min.   : 1.000   F:136324   Min.   :1.000   Min.   :2005     Min.   :1900
 1st Qu.: 3.000   M:163820   1st Qu.:2.000   1st Qu.:2007     1st Qu.:1933
 Median : 5.000              Median :2.000   Median :2009     Median :1943
 Mean   : 5.097              Mean   :1.894   Mean   :2009     Mean   :1945
 3rd Qu.: 7.000              3rd Qu.:2.000   3rd Qu.:2011     3rd Qu.:1954
 Max.   :10.000              Max.   :2.000   Max.   :2012     Max.   :2010
 NA's   :4797                                NA's   :238981   NA's   :1
      doDM
 Min.   :1916
 1st Qu.:1998
 Median :2004
 Mean   :2002
 3rd Qu.:2008
 Max.   :2012
 NA's   :975
```

We see that there are some really old dates of diagnosis:

```
subset( DM, doDM < 1935 )
```

```
        simd sex DMtype      dod      dob     doDM
3257       4   F      2 2005.417 1917.107 1917.324
7966       4   F      2 2006.079 1914.225 1926.868
10026      2   F      2 2006.345 1915.257 1915.936
14839      4   M      2 2007.002 1933.050 1933.691
15770      7   M      2 2007.092 1927.153 1927.418
16712      4   M      2 2007.202 1932.160 1932.212
43403      4   M      2 2010.430 1933.132 1933.674
44465     10   M      1 2010.559 1920.081 1924.771
44959      6   M      2 2010.616 1928.053 1933.630
46195      3   F      2 2010.758 1921.044 1921.277
55021      2   M      2 2011.719 1917.066 1917.187
150998     1   F      2       NA 1917.064 1917.422
152745     7   M      2       NA 1931.227 1931.585
159757     6   F      2       NA 1926.257 1926.088
219438     3   M      2       NA 1930.118 1930.257
246944     2   M      2       NA 1934.140 1934.274
291425     8   M      2       NA 1927.029 1927.276
296309     5   M      2       NA 1931.128 1931.837
300134     5   M      2       NA 1932.037 1932.423
300139     4   M      2       NA 1925.020 1925.201
```

It also appears that most of these are classified as T2DM, ven though most of them are quite young.

4.1.2.1 Depravation index

We can check how many have a missing depravation index:

```
with( DM, ftable( addmargins( table( sex,
                                     doDM = floor( pmax(doDM,1999.999) ),
                                     simd,
                                     useNA="ifany"),
                                     margin = 2:3 ) ) )
```

	simd	1	2	3	4	5	6	7	8	9	10	NA	Sum
sex	doDM												
F	1999	4954	5061	4936	4881	4398	4293	3924	3712	3296	2802	663	42920
	2000	769	795	731	798	647	618	568	565	534	336	69	6430
	2001	824	877	883	808	736	697	618	592	540	404	98	7077
	2002	919	911	904	834	734	673	638	646	553	416	99	7327
	2003	1027	1100	932	1006	875	790	739	686	659	548	133	8495
	2004	1086	1099	967	1048	889	822	822	680	648	520	114	8695
	2005	1062	1000	962	987	887	808	758	657	631	535	120	8407
	2006	993	974	937	962	801	734	797	679	619	484	121	8101
	2007	953	955	977	852	789	739	740	653	626	459	117	7860
	2008	971	969	954	911	883	793	774	682	645	513	126	8221
	2009	1066	995	1021	914	925	800	802	704	634	561	187	8609
	2010	951	1002	1001	905	834	760	756	711	626	477	148	8171
	2011	620	677	659	617	539	546	503	479	429	350	112	5531
	NA	46	50	72	62	63	54	33	45	22	26	7	480
	Sum	16241	16465	15936	15585	14000	13127	12472	11491	10462	8431	2114	136324
M	1999	4866	5318	5411	5394	5250	4990	5097	4694	4490	3826	769	50105
	2000	831	879	826	855	776	776	745	734	658	558	120	7758
	2001	835	883	946	977	887	792	747	729	701	607	123	8227
	2002	985	1020	972	964	943	897	828	815	787	673	136	9020
	2003	1021	1083	1017	1053	953	924	896	801	793	669	134	9344
	2004	1077	1156	1104	1130	969	983	1015	864	825	705	146	9974
	2005	1132	1153	1058	1037	1040	908	956	935	807	711	145	9882
	2006	1068	1066	1119	1083	1081	988	1003	928	875	727	167	10105
	2007	1025	1132	1101	1118	1059	987	998	899	871	644	168	10002
	2008	1187	1224	1120	1091	1051	1009	1038	980	892	773	205	10570
	2009	1157	1154	1197	1177	1171	1052	1091	956	971	787	202	10915
	2010	1126	1161	1156	1117	1150	1038	1001	966	963	744	215	10637
	2011	729	747	743	686	715	652	712	607	569	479	147	6786
	NA	43	35	73	78	68	72	31	38	26	25	6	495
	Sum	17082	18011	17843	17760	17113	16068	16158	14946	14228	11928	2683	163820

We see that the fraction of MD patients with missing depravation index is $2114/136324 = 1.55\%$ for women and $2683/163820 = 1.64\%$ for men, about 2–3 times more frequent than in the population data.

4.1.2.2 Dates

Now check if the three dates are in the correct order for all perons. We make a table where we also cross-tabulate by wheter the dates are actually available (that is non-missing):

```
with( DM, ftable( BB= !is.na(dob),
                  DM= !is.na(doDM),
                  DD= !is.na(dod),
              "b<dm"= dob<doDM,
              "dm<d"= doDM<dod,
                 useNA="ifany",
                 row.vars=1:3 ) )
```

			b<dm	FALSE			TRUE			NA		
			dm<d	FALSE	TRUE	NA	FALSE	TRUE	NA	FALSE	TRUE	NA
BB	DM	DD										
FALSE	FALSE	FALSE		0	0	0	0	0	0	0	0	0
		TRUE		0	0	0	0	0	0	0	0	0

```
      TRUE  FALSE           0      0      0      0      0      0      0      0      0
            TRUE            0      0      0      0      0      0      0      1      0
TRUE  FALSE FALSE           0      0      0      0      0      0      0      0    682
            TRUE            0      0      0      0      0      0      0      0    293
      TRUE  FALSE           0      0     29      0      0 238270      0      0      0
            TRUE            0      0      0     21  60848      0      0      0      0
```

From this table we see that all date variabels have missing values (not so strange for date of death, though) and that even where they are non-mssing some of them are in the wrong order.

Ideally we should only see entries in the last two lines of this table where the date of birth and date of diabetes are known, and then only for the columns whith `b<dm` true and `dm<d` either `TRUE` or `NA`.

We inspect the ones in the wrong order:

```
subset( DM, dob>doDM )
```

```
        simd sex DMtype dod      dob     doDM
63346     NA   M       1  NA 1984.223 1984.070
65596      3   F       2  NA 1939.169 1939.136
71657      2   F       1  NA 1979.120 1978.999
82433     NA   F       1  NA 1971.035 1970.999
102635     5   F       1  NA 2000.240 2000.218
111438     1   M       2  NA 1939.183 1939.054
116100     9   M       1  NA 1992.229 1991.068
119506     9   M       2  NA 1961.231 1961.031
128870     5   F       2  NA 1945.198 1945.001
129172    NA   F       1  NA 1963.090 1962.999
129822     1   M       1  NA 1987.254 1987.169
159757     6   F       2  NA 1926.257 1926.088
161124     8   M       1  NA 1997.028 1997.014
162475    10   M       1  NA 1981.176 1980.927
190732     9   F       2  NA 1950.200 1950.016
204720     9   M       2  NA 1961.211 1961.053
206757    NA   M       2  NA 1981.170 1981.091
217203     6   F       1  NA 1951.150 1950.999
224625     4   F       1  NA 2001.006 2001.001
226098     7   M       1  NA 1970.071 1970.000
241825     3   M       2  NA 1962.011 1961.466
241856     2   F       2  NA 1992.119 1992.089
255131     8   F       2  NA 1948.053 1948.029
259227     1   M       1  NA 1977.012 1977.001
282151     7   F       1  NA 1989.124 1989.001
283971     1   M       1  NA 1988.122 1988.100
294531     5   F       1  NA 1970.118 1970.000
298656     8   F       2  NA 1967.224 1966.999
299627     6   M       1  NA 1983.213 1982.999
```

```
subset( DM, doDM>dod )
```

```
       simd sex DMtype      dod      dob     doDM
319       7   M       2 2005.039 1930.227 2005.072
454       8   F       2 2005.055 1931.101 2010.493
2709      5   M       2 2005.340 1934.235 2007.298
4537      9   M       2 2005.608 1953.094 2005.803
8777      3   M       2 2006.183 1961.228 2010.750
14422    10   M       2 2006.956 1958.022 2008.606
14827     1   F       2 2006.999 1938.088 2011.588
15468     2   F       2 2007.060 1953.118 2007.405
16529     7   M       2 2007.177 1939.216 2010.189
19065     9   F       2 2007.520 1929.247 2008.026
19147     2   M       2 2007.530 1946.063 2008.278
19418     4   M       2 2007.569 1942.036 2008.678
20949     3   F       2 2007.769 1959.087 2009.680
23266     8   F       2 2008.040 1938.216 2009.313
```

```
24518    5   F      2 2008.179 1929.025 2009.822
29124    4   M      2 2008.760 1940.119 2011.281
29618    9   M      2 2008.823 1936.248 2010.402
31041    9   M      2 2008.979 1951.248 2010.871
39036    3   M      2 2009.943 1952.048 2010.550
42285    2   M      2 2010.290 1934.137 2010.668
44908    2   F      2 2010.611 1953.094 2011.114
```

Among those with date of birth after date of diagnosis, the difference is normally only a few days, but none are recorded as dead, and those with date of diagnosis after date of death are quite variable in the difference between the two variables. Thus we exclude persons with missing dates of birth or DM and with wrong order of dates:

```
nrow( DM )
```

```
[1] 300144
```

```
DM <- subset( DM, dob<doDM & pmin(0,dod-doDM,na.rm=T)>=0 )
nrow(DM)
```

```
[1] 299118
```

that is, slightluy more than 1000 persons, about 1 in 300.

In order to check how the dates are distributed, we make a histograms of each of the three date variables (the date of dignosis in two guises):

```
par( mfrow=c(2,2), mar=c(3,3,1,1), mgp=c(3,1,0)/1.6 )
hist( DM$dob, breaks=seq(1900,2013,1),
      col="black", main="", xlab="Date of birth" )
abline(v=seq(1900,2010,10),col="red")
hist( DM$doDM, breaks=seq(1915,2013,1),
      col="black", main="", xlab="Date of DM diagnosis" )
abline(v=seq(1920,2010,10),col="red")
hist( DM$doDM[DM$doDM>2000], breaks=seq(2000,2013,1/12),
      col="black", main="", xlab="Date of DM diagnosis" )
abline(v=seq(2000,2013,1),col="red")
abline(v=2005,col="limegreen")
hist( DM$dod, breaks=seq(2005,2013,1/12),
      col="black", main="", xlab="Date of death" )
abline(v=2004:2013,col="red")
```

The dates of birth shows the well-known post-war baby boom peak.

The dates of diagnosis (given by year) shows a smaller number of diagnoses in 2011 and none in 2012, consistent with a reporting delay. The histogram by month of diagnosis (post 2000) shows a clear seasonal component in diagnoses, particularly for the "old" diagnoses (prior to start of follow-up in 2005). Also it is clear that prevalence of diabetes will not be reliable beyond mid-2011, because of the reporting delay. Similarly we can only have a reliable assessment of incidence until the end of 2010; even for the first half of 2011, the incident cases seem to be under-reported when comparing to previous years.

Inspecting the dates of death by month in the last panel we see that it is consistent with follow-up for only a part of May — note also that mortality drops quite substantially in the summer.

This means that we shall only include persons with a date of diagnosis prior to 1.1.2011, since those reported after this are likely to be a biased sample in some way. However, we shall use follow-up for death till 18 May 2012.

Note three technical features about these histograms that enables us to see these things clearly:

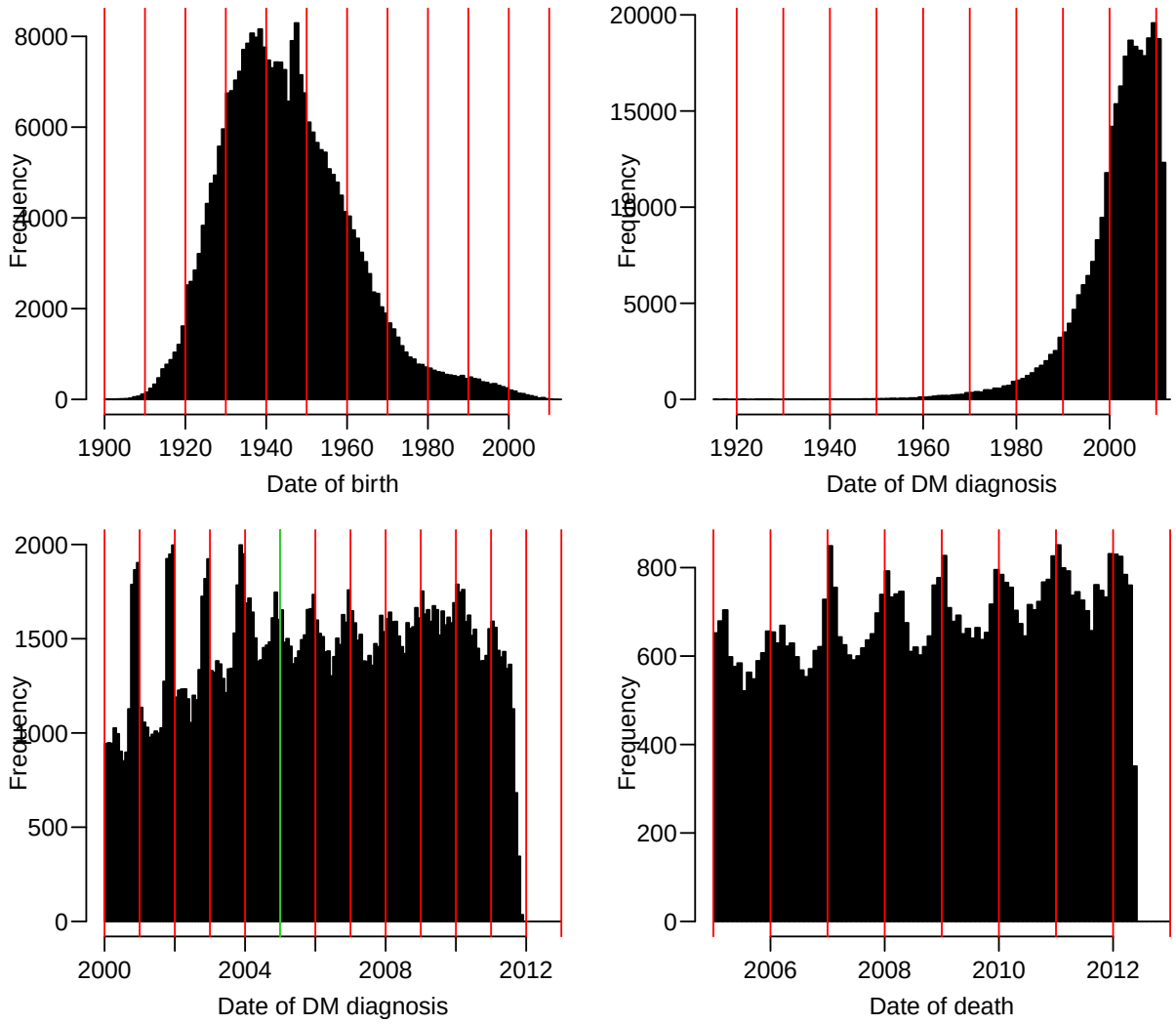

Figure 4.1: *Distribution of dates in the DM data base.*

1. They are given a color ("`black`") that eases inspection — the default is white bars with black outline is not useful. The default color for the border of the bars is black, so if you use another color than black, you shoudl also use that color for the border, for example: `col="red",border="red"`.

2. The breaks are chosen carfully so that each bin corresponds to a month or year so that postulated structures in data can be inspected and verified.

3. Vertical (in this case red or green) reference lines has been put in so that pertinent dates are clearly visible.

Conclusion: Default histograms are invariably useless; never do a histogram before you have drawn up with a pencil on paper what you want.

4.2 Prevalence of diabetes

Since the given population figures are per 1 July each year, we should compute the prevalences at each 1st July from 2005–2011. That means that for each 1st July we should fish out those persons in the DM data that are alive at the date and with a diagnosis of DM before the data, so for the year 2005, we refer to the midpoint of the year as 2005.5 — well not quite:

```
> cal.yr( as.Date("2005-07-01") )
  [1] 2005.496
  attr(,"class")
  [1] "cal.yr"  "numeric"
```

For the midpoint of 2005, we take the relevant subset:

```
> p2005 <- subset( DM, doDM<2005.5 & (dod>2005.5 | is.na(dod) ) )
> p2005 <- with( p2005, as.data.frame( table( sex, simd,
+                                             age=floor(2005.5-dob) ) ) )
> str( p2005 )
  'data.frame':        2080 obs. of  4 variables:
   $ sex : Factor w/ 2 levels "F","M": 1 2 1 2 1 2 1 2 1 2 ...
   $ simd: Factor w/ 10 levels "1","2","3","4",..: 1 1 2 2 3 3 4 4 5 5 ...
   $ age : Factor w/ 104 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
   $ Freq: int  0 0 1 0 0 0 0 0 1 0 ...
> head( p2005 )
    sex simd age Freq
  1   F    1   1    0
  2   M    1   1    0
  3   F    2   1    1
  4   M    2   1    0
  5   F    3   1    0
  6   M    3   1    0
```

In order to get a collected dataframe that we can match with the population data, we do this in loop over the years. Note that we start out with a `NULL` structure to which we can `rbind` the data from the single years. For the sake of exploration we also compute the prevalences at the middle of 2011, alhtough we expect them to be biased towards 0:

```
> prv <- NULL
> for( y in 2005:2011 )
+    {
+    my <- y + 0.5 # Mid-year date
+    sb <- subset( DM, doDM < my & ( dod > my | is.na(dod) ) )
+    prv <- rbind( prv,
+                  cbind( per = y,
+                         with( sb, as.data.frame( table( sex,
+                                                         simd,
+                                                  age=floor(my-dob) ) ) ) ) )
+    }
> str( prv )
  'data.frame':        14600 obs. of  5 variables:
   $ per : int  2005 2005 2005 2005 2005 2005 2005 2005 2005 2005 ...
   $ sex : Factor w/ 2 levels "F","M": 1 2 1 2 1 2 1 2 1 2 ...
   $ simd: Factor w/ 10 levels "1","2","3","4",..: 1 1 2 2 3 3 4 4 5 5 ...
   $ age : Factor w/ 106 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
   $ Freq: int  0 0 1 0 0 0 0 0 1 0 ...
> names( prv )[5] <- "X"
> prv <- transform( prv, age = as.numeric( as.character( age  ) ),
+                        simd = as.numeric( as.character( simd ) ) )
> str( prv )
```

```
  'data.frame':        14600 obs. of  5 variables:
   $ per : int  2005 2005 2005 2005 2005 2005 2005 2005 2005 2005 ...
   $ sex : Factor w/ 2 levels "F","M": 1 2 1 2 1 2 1 2 1 2 ...
   $ simd: num  1 1 2 2 3 3 4 4 5 5 ...
   $ age : num  1 1 1 1 1 1 1 1 1 1 ...
   $ X   : int  0 0 1 0 0 0 0 0 1 0 ...
> str( pop )
  'data.frame':        14560 obs. of  6 variables:
   $ per : int  2005 2005 2005 2005 2005 2005 2005 2005 2005 2005 ...
   $ age : int  0 0 0 0 0 0 0 0 0 0 ...
   $ sex : Factor w/ 2 levels "M","F": 1 1 1 1 1 1 1 1 1 1 ...
   $ simd: int  1 2 3 4 5 6 7 8 9 10 ...
   $ D   : num  21 17 22 20 13 10 20 14 9 8 ...
   $ N   : int  3823 3240 2907 2754 2639 2588 2600 2542 2529 2471 ...
```

Since the pop dataframe only has 90 age-classes (0–89) — the last is 90+ — we only merge the datasets for these age-classes. Note that we have made sure that the reevant variables have the same names, so we an immediately merge the two data frames:

```
> prv <- merge( subset( prv, age<90 ),
+               subset( pop, age<90 & per<2012 ) )
> summary( prv )
      per       sex           simd           age           X                D
 Min.   :2005   F:6230   Min.   : 1.0   Min.   : 1   Min.   :  0.0   Min.   :  0.00
 1st Qu.:2006   M:6230   1st Qu.: 3.0   1st Qu.:23   1st Qu.: 20.0   1st Qu.:  1.00
 Median :2008            Median : 5.5   Median :45   Median : 75.0   Median :  9.00
 Mean   :2008            Mean   : 5.5   Mean   :45   Mean   :115.8   Mean   : 26.32
 3rd Qu.:2010            3rd Qu.: 8.0   3rd Qu.:67   3rd Qu.:200.0   3rd Qu.: 43.00
 Max.   :2011            Max.   :10.0   Max.   :89   Max.   :501.0   Max.   :171.00
       N
 Min.   : 157
 1st Qu.:2454
 Median :3030
 Mean   :2873
 3rd Qu.:3571
 Max.   :4682
> save( prv, file="../data/prv.Rda" )
> load( file="../data/prv.Rda" )
```

4.2.0.3 Analysis of prevalences

Formally we cannot analyze prevalence figures from different years in a single model, since the same persons contribute to the numerator in several successive years. We shall however ignore this and analyse prevalences by a binomial model with log-link (so that the reslts correspond to relative proportions instead of odds-ratios as we would get from ordinary logistic regression).

In the first instance we model data for men and women separately, using a cubic spline to model the age-effct and a categorical varibale to model the effect of depracation index as a factor with 10 levels, and we also let the dates of prevalences vary by a factor.

In order to set up a cubic spline for the age-effect we must define a set of knots, which we for convenience take a equally speced between 5 and 85 with distance 10 years. We fit the model separately for men and women:

```
> library( splines )
> a.kn <- seq( 5, 85, 10 )
> m0 <- glm( cbind(X,N-X) ~ Ns( age+0.5, knots=a.kn ) +
+                           factor( per ) +
```

```
+                            factor( simd ),
+            family = binomial(link="log"),
+              data = subset(prv,sex=="M") )
> f0 <- update( m0, data = subset(prv,sex=="F") )
> round( cbind(  ci.exp( m0 ),
+                ci.exp( f0 ) ), 3 )
```

	exp(Est.)	2.5%	97.5%	exp(Est.)	2.5%	97.5%
(Intercept)	0.001	0.001	0.001	0.001	0.001	0.001
Ns(age + 0.5, knots = a.kn)1	7.135	6.757	7.534	4.878	4.613	5.158
Ns(age + 0.5, knots = a.kn)2	15.646	14.755	16.590	11.206	10.550	11.902
Ns(age + 0.5, knots = a.kn)3	38.230	36.259	40.309	23.268	22.053	24.551
Ns(age + 0.5, knots = a.kn)4	93.314	88.485	98.408	54.026	51.203	57.005
Ns(age + 0.5, knots = a.kn)5	145.101	137.708	152.890	91.669	86.983	96.607
Ns(age + 0.5, knots = a.kn)6	125.750	121.459	130.194	90.498	87.417	93.688
Ns(age + 0.5, knots = a.kn)7	1230.110	1097.878	1378.268	861.172	767.949	965.712
Ns(age + 0.5, knots = a.kn)8	50.791	49.556	52.057	36.494	35.600	37.410
factor(per)2006	1.049	1.040	1.058	1.052	1.042	1.062
factor(per)2007	1.096	1.087	1.105	1.095	1.085	1.105
factor(per)2008	1.137	1.128	1.146	1.136	1.126	1.147
factor(per)2009	1.184	1.174	1.194	1.179	1.169	1.190
factor(per)2010	1.228	1.218	1.238	1.221	1.211	1.232
factor(per)2011	1.255	1.245	1.264	1.249	1.238	1.260
factor(simd)2	0.996	0.988	1.005	0.940	0.932	0.949
factor(simd)3	0.952	0.944	0.961	0.875	0.866	0.883
factor(simd)4	0.914	0.906	0.923	0.848	0.840	0.857
factor(simd)5	0.873	0.865	0.881	0.777	0.769	0.785
factor(simd)6	0.808	0.801	0.816	0.729	0.722	0.736
factor(simd)7	0.801	0.794	0.809	0.691	0.684	0.698
factor(simd)8	0.780	0.772	0.787	0.657	0.650	0.664
factor(simd)9	0.751	0.744	0.758	0.615	0.609	0.622
factor(simd)10	0.649	0.643	0.656	0.501	0.495	0.507

Thus, both for men and women we see an increase by calendar year and a decrease by soical class.

We can get a quick overview og the relatve sizes for the effects of tiem and deprication by plotting the relvant parameters:

```
> mests <- ci.exp( m0, subset="fact" )
> fests <- ci.exp( f0, subset="fact" )
> rownames( mests ) <- gsub( "factor\\(per\\)", "", rownames( mests ) )
> rownames( mests ) <- gsub( "factor\\(simd\\)", "", rownames( mests ) )
> mests <- rbind( 1, mests[1:6,], 1, mests[7:15,] )
> fests <- rbind( 1, fests[1:6,], 1, fests[7:15,] )
> rownames( mests )[c(1,8)] <- c("Year 2005", "Social class 1")
> mests
```

	exp(Est.)	2.5%	97.5%
Year 2005	1.0000000	1.0000000	1.0000000
2006	1.0488624	1.0400680	1.0577311
2007	1.0962704	1.0872179	1.1053983
2008	1.1368172	1.1275490	1.1461616
2009	1.1839493	1.1744256	1.1935502
2010	1.2278864	1.2181273	1.2377237
2011	1.2545256	1.2446381	1.2644918
Social class 1	1.0000000	1.0000000	1.0000000
2	0.9963585	0.9875104	1.0052858
3	0.9523125	0.9438277	0.9608736
4	0.9143434	0.9061926	0.9225674
5	0.8725615	0.8646942	0.8805003
6	0.8081264	0.8007256	0.8155956
7	0.8012129	0.7938888	0.8086047
8	0.7795455	0.7722731	0.7868864
9	0.7513165	0.7442116	0.7584892
10	0.6494794	0.6430243	0.6559993

```
> par( mar=c(3,3,1,1), mgp=c(3,1,0)/1.6 )
>  plotEst( mests, y=17:1+0.1, lwd=3, cex=0.5, xlog=TRUE, vref=1, col="blue",
+                  xtic=c(c(5,7)/10,1,1.1,1.3), grid=5:13/10,
+                  xlab="Relative prevalence" )
> linesEst( fests, y=17:1-0.1, lwd=3, cex=0.5, col="red" )
```

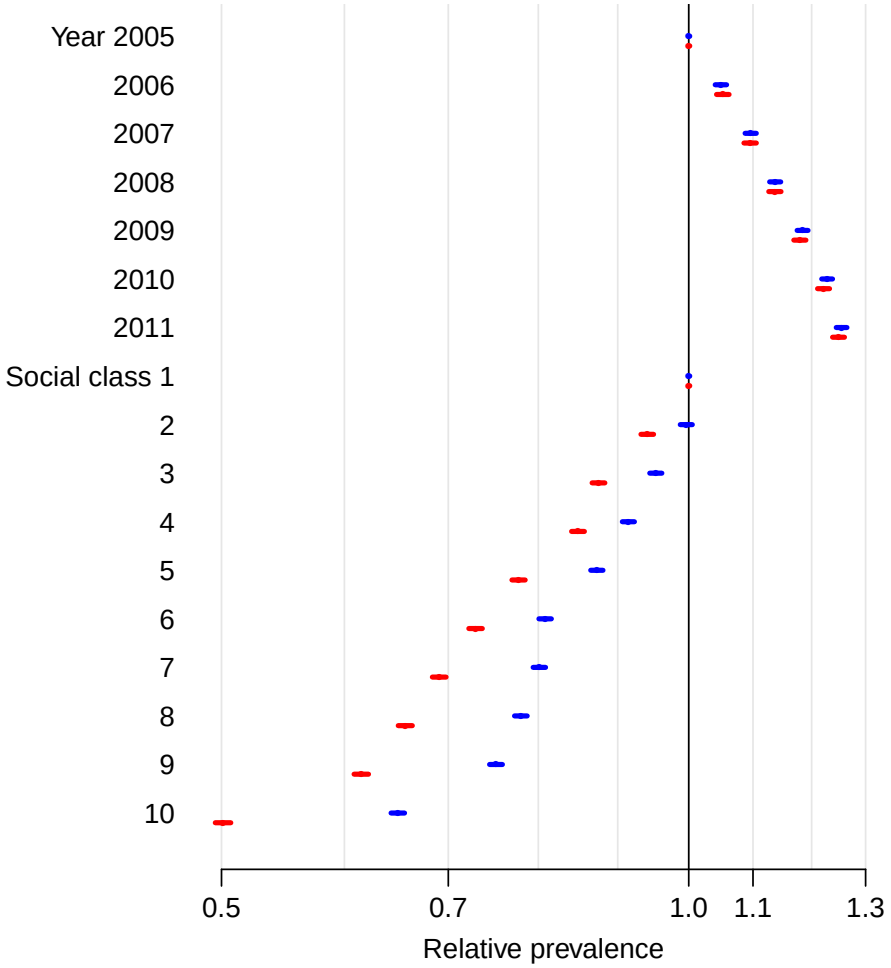

Figure 4.2: *Estimated effects of date and social class on prevalence. Reference level is social class 1 (lowest) at 1.7.2005.*

From the figure 4.2 we see the overall tendency that prevalence increases roughly linarly by time (on the log-scale) at the same pace for men and women, but also with a slightly smaller increase in 2011, as ecpected from the histograms. The social gradient in prevalence of diabetes is much stronger for women than for men. Also we see that the confidence intervals are tiny due to the large material.

For the sake of simplicity we also fit a model with linear effetcs of social class and date to summarize the differences, but excluding the 2011 data because they are likely to be biased:

```
> prv <- subset( prv, per<2011 )
> m0 <- update( m0, data = subset(prv,sex=="M") )
> f0 <- update( f0, data = subset(prv,sex=="F") )
> ml <- update( m0, . ~ . - factor(simd) - factor(per)
+                         + simd + I(per-2008) )
```

```
> fl <- update( ml, data = subset(prv,sex=="F") )
> round( (cbind( ci.exp( ml, subset=c("simd","per") ),
+                ci.exp( fl, subset=c("simd","per") ) ) - 1 )*100, 3 )
                exp(Est.)   2.5%  97.5% exp(Est.)   2.5%  97.5%
simd             -4.229 -4.306 -4.151    -6.363 -6.448 -6.278
I(per - 2008)     4.149  4.010  4.289     4.007  3.852  4.162
```

Thus we see that the average *relative* change in prevalence per social index value is −4.2% for men and −6.4% forwomen, whereas the average annual increase is 4.1% for men and 4.0% for women.

There is a very clear linear trend of increase over time and decrease by increasing social status. However these are under the assumption that the shape of the age-effect is the same over time and across deprivation strata. We can test this in two ways; either by including rather detailed interactions between these two factors and the age-effect or more specifically only including a linear term for `per` or `simd` in the interaction. The latter means a parametric interaction on 8 df., but the former an interaction of 32 (or 64) df.; way too complicated for reporting:

```
> mpl <- update( m0, . ~ . +  Ns( age+0.5, knots=a.kn ):I(per-2005) )
> mpi <- update( m0, . ~ . +  Ns( age+0.5, knots=a.kn ):factor(per) )
> msl <- update( m0, . ~ . +  Ns( age+0.5, knots=a.kn ):I(simd-2005) )
> msi <- update( m0, . ~ . +  Ns( age+0.5, knots=a.kn ):factor(simd) )
> fpl <- update( f0, . ~ . +  Ns( age+0.5, knots=a.kn ):I(per-2005) )
> fpi <- update( f0, . ~ . +  Ns( age+0.5, knots=a.kn ):factor(per) )
> fsl <- update( f0, . ~ . +  Ns( age+0.5, knots=a.kn ):I(simd-2005) )
> fsi <- update( f0, . ~ . +  Ns( age+0.5, knots=a.kn ):factor(simd) )
> anova( mpi, mpl, m0, msl, msi, test="Chisq" )
Analysis of Deviance Table

Model 1: cbind(X, N - X) ~ Ns(age + 0.5, knots = a.kn) + factor(per) +
    factor(simd) + Ns(age + 0.5, knots = a.kn):factor(per)
Model 2: cbind(X, N - X) ~ Ns(age + 0.5, knots = a.kn) + factor(per) +
    factor(simd) + Ns(age + 0.5, knots = a.kn):I(per - 2005)
Model 3: cbind(X, N - X) ~ Ns(age + 0.5, knots = a.kn) + factor(per) +
    factor(simd)
Model 4: cbind(X, N - X) ~ Ns(age + 0.5, knots = a.kn) + factor(per) +
    factor(simd) + Ns(age + 0.5, knots = a.kn):I(simd - 2005)
Model 5: cbind(X, N - X) ~ Ns(age + 0.5, knots = a.kn) + factor(per) +
    factor(simd) + Ns(age + 0.5, knots = a.kn):factor(simd)
  Resid. Df Resid. Dev  Df Deviance Pr(>Chi)
1      5277     9915.8
2      5309     9926.8 -32    -11.0   0.9998
3      5317    10213.7  -8   -287.0   <2e-16
4      5309     6869.8   8   3343.9   <2e-16
5      5245     6478.6  64    391.2   <2e-16
> anova( fpi, fpl, f0, fsl, fsi, test="Chisq" )
Analysis of Deviance Table

Model 1: cbind(X, N - X) ~ Ns(age + 0.5, knots = a.kn) + factor(per) +
    factor(simd) + Ns(age + 0.5, knots = a.kn):factor(per)
Model 2: cbind(X, N - X) ~ Ns(age + 0.5, knots = a.kn) + factor(per) +
    factor(simd) + Ns(age + 0.5, knots = a.kn):I(per - 2005)
Model 3: cbind(X, N - X) ~ Ns(age + 0.5, knots = a.kn) + factor(per) +
    factor(simd)
Model 4: cbind(X, N - X) ~ Ns(age + 0.5, knots = a.kn) + factor(per) +
    factor(simd) + Ns(age + 0.5, knots = a.kn):I(simd - 2005)
Model 5: cbind(X, N - X) ~ Ns(age + 0.5, knots = a.kn) + factor(per) +
    factor(simd) + Ns(age + 0.5, knots = a.kn):factor(simd)
  Resid. Df Resid. Dev  Df Deviance Pr(>Chi)
1      5277    10572.0
2      5309    10580.5 -32     -8.5        1
```

```
  3      5317    10987.5  -8   -407.0   <2e-16
  4      5309     6988.9   8   3998.6   <2e-16
  5      5245     6587.6  64    401.3   <2e-16
```

Clearly, the linear interactions have by far the largest impacts on the estimated prevalences, hence we try a model where both are included:

```
> mspl <- update( msl, . ~ . + Ns( age+0.5, knots=a.kn ):I(per-2005) )
> fspl <- update( fsl, . ~ . + Ns( age+0.5, knots=a.kn ):I(per-2005) )
> anova( mpl, mspl, msl, test="Chisq" )
  Analysis of Deviance Table

  Model 1: cbind(X, N - X) ~ Ns(age + 0.5, knots = a.kn) + factor(per) +
      factor(simd) + Ns(age + 0.5, knots = a.kn):I(per - 2005)
  Model 2: cbind(X, N - X) ~ Ns(age + 0.5, knots = a.kn) + factor(per) +
      factor(simd) + Ns(age + 0.5, knots = a.kn):I(simd - 2005) +
      Ns(age + 0.5, knots = a.kn):I(per - 2005)
  Model 3: cbind(X, N - X) ~ Ns(age + 0.5, knots = a.kn) + factor(per) +
      factor(simd) + Ns(age + 0.5, knots = a.kn):I(simd - 2005)
    Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
  1      5309     9926.8
  2      5301     6594.7  8   3332.1 < 2.2e-16
  3      5309     6869.8 -8   -275.1 < 2.2e-16

> anova( fpl, fspl, fsl, test="Chisq" )
  Analysis of Deviance Table

  Model 1: cbind(X, N - X) ~ Ns(age + 0.5, knots = a.kn) + factor(per) +
      factor(simd) + Ns(age + 0.5, knots = a.kn):I(per - 2005)
  Model 2: cbind(X, N - X) ~ Ns(age + 0.5, knots = a.kn) + factor(per) +
      factor(simd) + Ns(age + 0.5, knots = a.kn):I(simd - 2005) +
      Ns(age + 0.5, knots = a.kn):I(per - 2005)
  Model 3: cbind(X, N - X) ~ Ns(age + 0.5, knots = a.kn) + factor(per) +
      factor(simd) + Ns(age + 0.5, knots = a.kn):I(simd - 2005)
    Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
  1      5309    10580.5
  2      5301     6607.4  8   3973.1 < 2.2e-16
  3      5309     6988.9 -8   -381.5 < 2.2e-16
```

and we see there is a substantial influence of both even if the other is in the model.

With this interaction model in place we will of course need to know *how* the interaction looks as a function of age and deprivation index. Hence we derive predictions for the date 1 July 2008 across deprivation strata, and for deprivation index 5 across years:

```
> nd <- data.frame( age = 0:90,
+                   per = 2008,
+                  simd = 5 )
> mpr2008 <- fpr2008 <- NULL
> for( sc in 1:10 )
+    {
+    mpr2008 <- cbind( mpr2008, ci.pred( mspl, newdata = transform(nd,simd=sc) ) )
+    fpr2008 <- cbind( fpr2008, ci.pred( fspl, newdata = transform(nd,simd=sc) ) )
+    }
> dim( mpr2008 )
  [1] 91 30

> mprcl5 <- fprcl5 <- NULL
> for( yy in 2005+0:5 )
+    {
+    mprcl5 <- cbind( mprcl5, ci.pred( mspl, newdata = transform(nd,per=yy) ) )
+    fprcl5 <- cbind( fprcl5, ci.pred( fspl, newdata = transform(nd,per=yy) ) )
+    }
> dim( mprcl5 )
```

```
  [1] 91 18
> par( mfrow=c(2,2), mar=c(3,3,1,1), mgp=c(3,1,0)/1.6, bty="n" )
> matplot( nd$age, mpr2008[,1+0:9*3]*100,
+          ylim=c(0,19), xlab="Age",
+          ylab="Prevalence at 1.1.2008",
+          type="l", lty=1, lwd=3, col=gray(3:12/15) )
> matplot( nd$age, fpr2008[,1+0:9*3]*100,
+          ylim=c(0,19), xlab="Age",
+          ylab="Prevalence at 1.1.2008",
+          type="l", lty=1, lwd=3, col=gray(3:12/15) )
> matplot( nd$age, mprcl5[,1+0:5*3]*100,
+          ylim=c(0,19), xlab="Age",
+          ylab="Prevalence in class 5",
+          type="l", lty=1, lwd=3, col=gray(8:3/11) )
> matplot( nd$age, fprcl5[,1+0:5*3]*100,
+          ylim=c(0,19), xlab="Age",
+          ylab="Prevalence in class 5",
+         type="l", lty=1, lwd=3, col=gray(8:3/11) )
```

.

From Figure 4.3 we see that the *shape* of the inetraction is more pronounced across social classes than across periods; the most interesting feature is that social class 1 has a markedly smaller prevalence of diabetes than the other classes, and that the differences between socal classes seem to vanish in ages over 80, except for class 1 among women where the prevalences seem to be smaller througout the age-range.

4.3 Follow-up data

For the anaysis of mortality and incidence rates we need the number of new diabetes cases, resp. deaths, which we can derive from the `DM` dataset, but since we from the prevalence dataset saw that up to 20% in ages around 70 suffer from diabetes, it is essential to subtract the risk time among the diabetes patients from the total population i order to get the risk time in the non-diabetes part of the population.

Ultimately we want to set up a datatset classified by sex, age and calendar time of follow-up and social class (social depravation index). The tabulated responses in this dataset must be:

- For incidence analysis:
  - No. of incident diabetes cases
  - Person-years among non-diabetic persons
- For mortality analysis
  - No. of deaths among diabetes patients
  - No. of deaths among non-diabetic persons
  - Person-years among diabetes patients
  - Person-years among non-diabetic persons

Alternatively, the same data could be set up in a dataset classified by sex, age and calendar time of follow-up, social class *and* diabetes status, and in this dataset we would only need the number of incident cases of DM (which would be `NA` for diabetes status = DM).

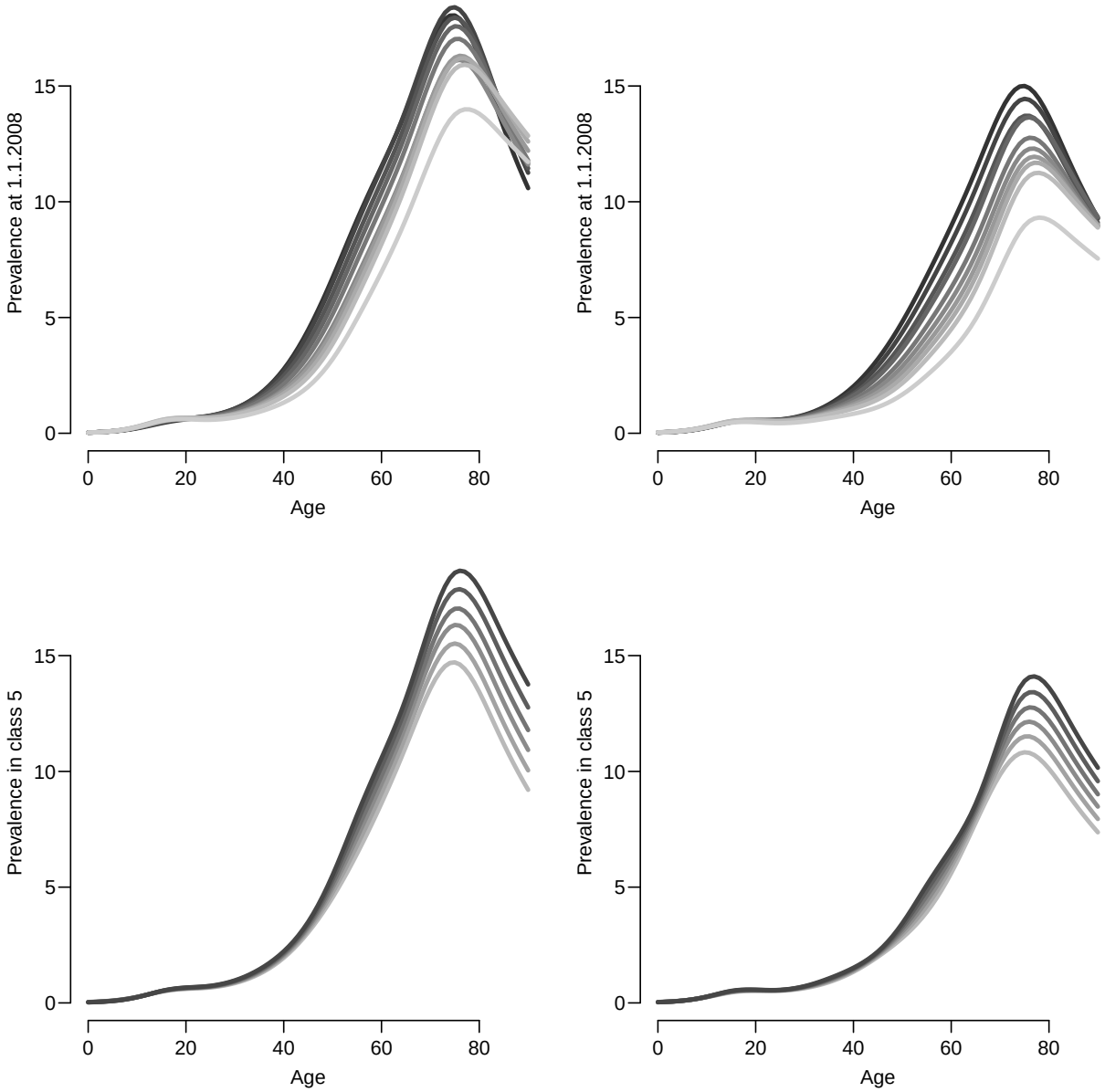

Figure 4.3: *The estimated age-specific prevalences from the model with interaction between age and linear period and linear social class. Dark colours correspond to the lowest social class, resp. latest date*

We shall see that we are essentially forced to set op the first "wide" version of the dataset first, but also that the latter, "long" form of the dataset is useful for comparative mortality analyses.

4.3.1 A `Lexis` object of follow-up

To this end we set up the follow-up in a Lexis object, a helping date `dox` the last date of follow-up for death, is computed for convenience:

```
> options( width=120 )
> summary( DM )
```

```
      simd          sex            DMtype           dod             dob            doDM
 Min.   : 1.000   F:135820   Min.   :1.000   Min.   :2005   Min.   :1900   Min.   :1916
 1st Qu.: 3.000   M:163298   1st Qu.:2.000   1st Qu.:2007   1st Qu.:1933   1st Qu.:1998
 Median : 5.000              Median :2.000   Median :2009   Median :1943   Median :2004
 Mean   : 5.098              Mean   :1.894   Mean   :2009   Mean   :1945   Mean   :2002
 3rd Qu.: 7.000              3rd Qu.:2.000   3rd Qu.:2011   3rd Qu.:1954   3rd Qu.:2008
 Max.   :10.000              Max.   :2.000   Max.   :2012   Max.   :2010   Max.   :2012
 NA's   :4780                                NA's   :238270
```

```
> ( dox <- cal.yr( as.Date("2012-05-18") ) )
```

```
[1] 2012.376
attr(,"class")
[1] "cal.yr"  "numeric"
```

```
> LD <- Lexis( entry = list( per = pmax( 2005, doDM ),
+                            age = pmax( 2005, doDM ) - dob ),
+               exit = list( per = pmin( dox, dod, na.rm=TRUE ) ),
+        exit.status = pmin( dod, dox, na.rm=TRUE ) < dox,
+             states = c("ALive","Dead"),
+               data = DM )
> summary( LD )
```

```
Transitions:
     To
From     ALive  Dead  Records:  Events: Risk time:  Persons:
  ALive 238280 60838    299118    60838    1575840    299118
```

This dataset represents the follow-up of all diabetes patients from data of diagnosis till detah in the calendar time window 2005-01-01 to 2012-05-18. In the previously read data set `pop` we have the tabulated follow-up (person-years and deaths) for the `entire` Scottish population, so quantities for the non-diabetic part of the population must be obtained by subtraction.

4.3.1.1 Time-splitting and tabulation

The strategy now is to tabulate the DM-cases, deaths and person-years in 1-year classes of age and calendar time in the same format as the population data. However since we have 1.5 million person-years, and the intervals on average will be half a year long we will end up with a dataset of about 3 million records, which is a bit large for an ordinary computer and also quite slow (we shall look into that too).

Hence what we do is to split small chunks of the data frame `LD` at a time, and then tabulate deaths and person-years by age and data, and add this to a master table that eventually will contain all deaths and person-years:

First decide the number of chunks and then the starting and ending records of the chunks. But we first restrict the data to those diagnosed prior to 2011-01-01:

```
> table( entry(LD,"per")<2011, useNA="ifany" )
```

```
 FALSE   TRUE
 12314 286804
```

```
> LD <- subset( LD, entry(LD,"per")<2011 )
> nch <- 20
> ( ll <- round( seq(0,nrow(LD),,nch+1) ) )
```

```
 [1]      0  14340  28680  43021  57361  71701  86041 100381 114722 129062
[11] 143402 157742 172082 186423 200763 215103 229443 243783 258124 272464
[21] 286804
```

```
> diff( ll )
```

```
 [1] 14340 14340 14341 14340 14340 14340 14340 14341 14340 14340 14340 14340
[13] 14341 14340 14340 14340 14340 14341 14340 14340
```

Then we can split follow-up by age and calendar time within each chunk of data. For the sake of illustration we start with the first chunk:

First we compute which rows from `LD` should be used for splitting by age and calendar time, we put the row names in the vector `whr`:

```
> i <- 1
> range( whr <- (ll[i]+1):ll[i+1] )
  [1]     1 14340
```

Note that this works because we started `ll` with 0 rather than 1, so that the first record in each chunk has number `ll[i]+1`.

We then split the follow-up time of the persons in the chosen rows by first calendar time then age (the order is immaterial):

```
> sl <- splitLexis( LD[whr,], 1990:2015, "per" )
> sl <- splitLexis( sl      ,    0:150 , "age" )
```

We the aggregate deaths and person-years by sex, social class age and period. Note that we put person-years and deaths in variables `y` and `d` (lowercase)

```
> agg <- with( sl, aggregate( cbind( y = lex.dur,
+                                    d = (lex.Xst=="Dead") ),
+                         by = list( sex = sex,
+                                    A = floor(age),
+                                    P = floor(per),
+                                   sC = simd ),
+                                  FUN = sum ) )
> str( agg )
  'data.frame':        2515 obs. of  6 variables:
   $ sex: Factor w/ 2 levels "F","M": 1 1 2 2 1 2 1 2 2 2 ...
   $ A  : num  17 18 19 20 26 26 27 27 29 30 ...
   $ P  : num  2005 2005 2005 2005 2005 ...
   $ sC : int  1 1 1 1 1 1 1 1 1 1 ...
   $ y  : num  0.13895 0.86105 0.00616 0.99384 0.10404 ...
   $ d  : num  0 0 0 0 0 0 0 1 0 0 ...
```

This aggregated data frame has one record per combination of values of the variables mentioned in the `by=` argument to `aggregate` in the split data frame `sl`.

This is now merged into the master data frame `DMtab`, which we specify with the column names for classification and for holding the aggregated number of person-years and deaths among diabetes patients. Besides it must have the same variables as `agg`, so we set it up by expanding agg by the two desired columns:

```
> DMtab <- cbind( agg, Y.dm=0, D.dm=0 )
> str( DMtab )
  'data.frame':        2515 obs. of  8 variables:
   $ sex : Factor w/ 2 levels "F","M": 1 1 2 2 1 2 1 2 2 2 ...
   $ A   : num  17 18 19 20 26 26 27 27 29 30 ...
   $ P   : num  2005 2005 2005 2005 2005 ...
   $ sC  : int  1 1 1 1 1 1 1 1 1 1 ...
   $ y   : num  0.13895 0.86105 0.00616 0.99384 0.10404 ...
   $ d   : num  0 0 0 0 0 0 0 1 0 0 ...
   $ Y.dm: num  0 0 0 0 0 0 0 0 0 0 ...
   $ D.dm: num  0 0 0 0 0 0 0 0 0 0 ...
```

We must now add the amount of person-years and number of deaths from `agg` (that is from the latest chunk of the `Lexis` object `DL`) to the aggregated numbers in `DMtab` which we represent in `Y.dm` and `D.dm`.

Note that we must use the construction `pmax(y,0,na.rm=TRUE)`, because units in the merged data frame where there is no contribution from `agg` have missing values for `y` and `d`, and units with no contribution from `DMtab` have missing values for `Y.dm` and `D.dm`. Finally, we strip the variables `y` and `d` from the result, so that we can merge them in again afresh from next chunk:

```
> DMtab <- transform( DMtab, Y.dm = pmax( Y.dm, 0, na.rm=TRUE ) +
+                                   pmax( y   , 0, na.rm=TRUE ),
+                            D.dm = pmax( D.dm, 0, na.rm=TRUE ) +
+                                   pmax( d   , 0, na.rm=TRUE ) )[
+          c("sex","A","P","sC","Y.dm","D.dm") ]
> str( DMtab )
  'data.frame':        2515 obs. of  6 variables:
   $ sex : Factor w/ 2 levels "F","M": 1 1 2 2 1 2 1 2 2 2 ...
   $ A   : num  17 18 19 20 26 26 27 27 29 30 ...
   $ P   : num  2005 2005 2005 2005 2005 ...
   $ sC  : int  1 1 1 1 1 1 1 1 1 1 ...
   $ Y.dm: num  0.13895 0.86105 0.00616 0.99384 0.10404 ...
   $ D.dm: num  0 0 0 0 0 0 0 1 0 0 ...
```

We can now collect this in loop over the remaining chunks of data:

```
> for( i in 2:nch )
+ {
+ whr <- (ll[i]+1):ll[i+1]
+ sl <- splitLexis( LD[whr,], 1990:2015, "per" )
+ sl <- splitLexis( sl      ,    0:150 , "age" )
+ agg <- with( sl, aggregate( cbind( y = lex.dur,
+                                    d = (lex.Xst=="Dead") ),
+                            list( sex = sex,
+                                    A = floor(age),
+                                    P = floor(per),
+                                   sC = simd ),
+                                  FUN = sum ) )
+ DMtab <- merge( DMtab, agg, all=TRUE )
+ DMtab <- transform( DMtab, Y.dm = pmax( Y.dm, 0, na.rm=TRUE ) +
+                                   pmax( y   , 0, na.rm=TRUE ),
+                            D.dm = pmax( D.dm, 0, na.rm=TRUE ) +
+                                   pmax( d   , 0, na.rm=TRUE ) )[
+                            c("sex","A","P","sC","Y.dm","D.dm") ]
+ cat( "Merged in chunk", i, "now", nrow(DMtab), "rows, at",
+      format(Sys.time(),format="%Y-%m-%d %H:%M:%S"), "\n" )
+ flush.console()
+ }
  Merged in chunk 2 now 5325 rows, at 2014-08-19 18:55:40
  Merged in chunk 3 now 8262 rows, at 2014-08-19 18:56:00
  Merged in chunk 4 now 10404 rows, at 2014-08-19 18:56:25
  Merged in chunk 5 now 14482 rows, at 2014-08-19 18:56:53
  Merged in chunk 6 now 15306 rows, at 2014-08-19 18:57:20
  Merged in chunk 7 now 15534 rows, at 2014-08-19 18:57:53
  Merged in chunk 8 now 15661 rows, at 2014-08-19 18:58:25
  Merged in chunk 9 now 15716 rows, at 2014-08-19 18:58:53
  Merged in chunk 10 now 15766 rows, at 2014-08-19 18:59:26
  Merged in chunk 11 now 15797 rows, at 2014-08-19 18:59:54
  Merged in chunk 12 now 15826 rows, at 2014-08-19 19:00:31
  Merged in chunk 13 now 15853 rows, at 2014-08-19 19:01:03
  Merged in chunk 14 now 15866 rows, at 2014-08-19 19:01:41
  Merged in chunk 15 now 15902 rows, at 2014-08-19 19:02:17
  Merged in chunk 16 now 15925 rows, at 2014-08-19 19:02:50
  Merged in chunk 17 now 15941 rows, at 2014-08-19 19:03:25
  Merged in chunk 18 now 15959 rows, at 2014-08-19 19:04:03
  Merged in chunk 19 now 15968 rows, at 2014-08-19 19:04:35
  Merged in chunk 20 now 15984 rows, at 2014-08-19 19:05:14
```

```
> str( DMtab )

  'data.frame':        15984 obs. of  6 variables:
   $ sex : Factor w/ 2 levels "F","M": 1 1 1 1 1 1 1 1 1 ...
   $ A   : num  0 0 0 0 0 0 0 1 1 1 ...
   $ P   : num  2005 2005 2006 2006 2007 ...
   $ sC  : int  3 7 2 3 2 5 5 2 4 5 ...
   $ Y.dm: num  0.4162 0.0363 0.1567 0.2389 0.1971 ...
   $ D.dm: num  0 0 0 0 0 0 0 0 0 0 ...

> save( DMtab, file="../data/DMtab.Rda" )
```

Now we have all the deaths and risk-time (person-years) among diabetes patients in `DMtab`, classified by sex, social class and age and date of follow-up in 1-year classes. Exactly as the risk time and deaths in the Scottish population.

4.3.2 Merging tabulated diabetes data with population data

So we can merge the two, but we must specify which variables are to be paired up as the variable names in the two data frames are not the same:

```
> load( file="../data/DMtab.Rda" )
> str( DMtab )
  'data.frame':        15984 obs. of  6 variables:
   $ sex : Factor w/ 2 levels "F","M": 1 1 1 1 1 1 1 1 1 ...
   $ A   : num  0 0 0 0 0 0 0 1 1 1 ...
   $ P   : num  2005 2005 2006 2006 2007 ...
   $ sC  : int  3 7 2 3 2 5 5 2 4 5 ...
   $ Y.dm: num  0.4162 0.0363 0.1567 0.2389 0.1971 ...
   $ D.dm: num  0 0 0 0 0 0 0 0 0 0 ...

> str( pop )

  'data.frame':        14560 obs. of  6 variables:
   $ per : int  2005 2005 2005 2005 2005 2005 2005 2005 2005 2005 ...
   $ age : int  0 0 0 0 0 0 0 0 0 0 ...
   $ sex : Factor w/ 2 levels "M","F": 1 1 1 1 1 1 1 1 1 1 ...
   $ simd: int  1 2 3 4 5 6 7 8 9 10 ...
   $ D   : num  21 17 22 20 13 10 20 14 9 8 ...
   $ N   : int  3823 3240 2907 2754 2639 2588 2600 2542 2529 2471 ...

> Atab <- merge( subset( DMtab, A<90 ),
+                subset( pop  , age<90 & simd<11 ),
+                by.x = c("sex","sC"  ,"A"  ,"P"  ),
+                by.y = c("sex","simd","age","per"),
+                all = TRUE )
> summary( Atab )

   sex            sC              A               P             Y.dm                D.dm              D
   F:7200   Min.   : 1.0   Min.   : 0.0   Min.   :2005   Min.   :  0.0027   Min.   : 0.00   Min.   :
   M:7200   1st Qu.: 3.0   1st Qu.:22.0   1st Qu.:2007   1st Qu.: 18.2473   1st Qu.: 0.00   1st Qu.:
            Median : 5.5   Median :44.5   Median :2008   Median : 67.1961   Median : 1.00   Median :
            Mean   : 5.5   Mean   :44.5   Mean   :2008   Mean   :107.4974   Mean   : 3.88   Mean   :
            3rd Qu.: 8.0   3rd Qu.:67.0   3rd Qu.:2010   3rd Qu.:181.0674   3rd Qu.: 6.00   3rd Qu.:
            Max.   :10.0   Max.   :89.0   Max.   :2012   Max.   :476.5975   Max.   :36.00   Max.   :
                                                         NA's   :282        NA's   :282

       N
   Min.   : 157
   1st Qu.:2470
   Median :3031
   Mean   :2880
   3rd Qu.:3576
   Max.   :4720
```

Note that the column names of the resulting dataframe is that of the *first* ("`x`") mentioned in the call to `merge`.

We now also want the number of incident cases of DM from the original `Lexis` dataset, `LD`. Note that we here exploit the fact the the timescale variables (in this case `age` and `per` are coded as the *entry* into the study, and that there is only one record per person in `LD`:

```
> head( LD )
```

```
     per      age       lex.dur lex.Cst lex.Xst lex.id simd sex DMtype      dod      dob      doDM
  1 2005 75.82957 0.0006844627   ALive    Dead      1    5   M      2 2005.001 1929.170 2000.815
  2 2005 79.93908 0.0006844627   ALive    Dead      2    4   M      2 2005.001 1925.061 1996.538
  3 2005 76.74127 0.0006844627   ALive    Dead      3    4   M      2 2005.001 1928.259 1997.387
  4 2005 68.81520 0.0006844627   ALive    Dead      4    8   F      2 2005.001 1936.185 1994.942
  5 2005 61.97057 0.0006844627   ALive    Dead      5    2   F      2 2005.001 1943.029 2000.133
  6 2005 80.74401 0.0006844627   ALive    Dead      6    3   F      2 2005.001 1924.256 2003.856
```

```
> DMinc <- with( subset(LD,entry(LD,"per")<2011),
+                aggregate( !is.na(doDM),
+                           list( A = floor(age),
+                                 P = floor(per),
+                               sex = sex,
+                                sC = simd ),
+                           FUN = sum ) )
> names( DMinc )[5] <- "I.dm"
> str( DMinc )
```

```
  'data.frame':        10163 obs. of  5 variables:
   $ A   : num  2 3 4 5 6 7 8 9 10 11 ...
   $ P   : num  2005 2005 2005 2005 2005 ...
   $ sex : Factor w/ 2 levels "F","M": 1 1 1 1 1 1 1 1 1 1 ...
   $ sC  : int  1 1 1 1 1 1 1 1 1 1 ...
   $ I.dm: int  4 5 4 5 5 8 6 5 4 16 ...
```

```
> Atab <- merge( Atab, subset( DMinc, A < 90 & P < 2012 ), all=TRUE )
> head( Atab )
```

```
    sex sC A    P Y.dm D.dm  D    N I.dm
  1   F  1 0 2005   NA   NA 22 3608   NA
  2   F  1 0 2006   NA   NA 14 3612   NA
  3   F  1 0 2007   NA   NA 29 3701   NA
  4   F  1 0 2008   NA   NA 19 4049   NA
  5   F  1 0 2009   NA   NA 13 3964   NA
  6   F  1 0 2010   NA   NA 18 3864   NA
```

Note that there are units from the populatin data, `pop`, that may not have any match in DMtab and DMinc, and the corresonding counts should therefor be set equal to 0:

```
> Atab <- transform( Atab, Y.dm = pmax( 0, Y.dm, na.rm=TRUE ),
+                          D.dm = pmax( 0, D.dm, na.rm=TRUE ),
+                          I.dm = pmax( 0, I.dm, na.rm=TRUE ) )
```

The `Atab` now has the person-years among diabetes patients (`Y.dm`), and the mid-year population for each single year (`N`). By multiplying the latter by 1 year we get a reasonable approximation to the person-years in the population, and so we can get the person-year in the non-diabetic part of the population by subtraction. Similarly, the number of deaths in the non-diabetic part of the population can be computed by subtraction:

```
> Atab <- transform( Atab, Y.nd = pmax( 0, N-Y.dm, na.rm=TRUE ),
+                          D.nd = pmax( 0, D-D.dm, na.rm=TRUE ) )
> str( Atab )
```

```
  'data.frame':        14400 obs. of  11 variables:
   $ sex : Factor w/ 2 levels "F","M": 1 1 1 1 1 1 1 1 1 1 ...
   $ sC  : int  1 1 1 1 1 1 1 1 1 1 ...
   $ A   : num  0 0 0 0 0 0 0 0 1 1 ...
```

```
   $ P   : num  2005 2006 2007 2008 2009 ...
   $ Y.dm: num  0 0 0 0 0 ...
   $ D.dm: num  0 0 0 0 0 0 0 0 0 0 ...
   $ D   : num  22 14 29 19 13 18 21 8 3 3 ...
   $ N   : int  3608 3612 3701 4049 3964 3864 3973 3892 3397 3522 ...
   $ I.dm: num  0 0 0 0 0 0 0 0 0 1 ...
   $ Y.nd: num  3608 3612 3701 4049 3964 ...
   $ D.nd: num  22 14 29 19 13 18 21 8 3 3 ...
> summary( Atab )
   sex            sC              A               P             Y.dm             D.dm              D
   F:7200   Min.   : 1.0   Min.   : 0.0   Min.   :2005   Min.   :  0.00   Min.   : 0.000   Min.   :
   M:7200   1st Qu.: 3.0   1st Qu.:22.0   1st Qu.:2007   1st Qu.: 17.15   1st Qu.: 0.000   1st Qu.:
            Median : 5.5   Median :44.5   Median :2008   Median : 63.18   Median : 0.000   Median :
            Mean   : 5.5   Mean   :44.5   Mean   :2008   Mean   :105.39   Mean   : 3.804   Mean   :
            3rd Qu.: 8.0   3rd Qu.:67.0   3rd Qu.:2010   3rd Qu.:177.91   3rd Qu.: 6.000   3rd Qu.:
            Max.   :10.0   Max.   :89.0   Max.   :2012   Max.   :476.60   Max.   :36.000   Max.   :1
      I.dm             Y.nd             D.nd
   Min.   :  0.00   Min.   : 140.8   Min.   :  0.0
   1st Qu.:  0.00   1st Qu.:2276.1   1st Qu.:  1.0
   Median :  3.00   Median :2969.0   Median :  8.0
   Mean   : 19.42   Mean   :2774.9   Mean   : 22.3
   3rd Qu.: 18.00   3rd Qu.:3478.8   3rd Qu.: 35.0
   Max.   :367.00   Max.   :4712.4   Max.   :148.0
> save( Atab, file="../data/Atab" )
```

`Atab` now contains the follow-up data tabulated by social class, sex, age and calendar time; for analysis of:

- Incidence of DM, by using (`I.dm`,`Y.nd`)
- Mortality after DM, by using (`D.dm`,`Y.dm`)
- Relative mortality after DM, by using (`D.dm`,`Y.dm`), and compare with (`D.nd`,`Y.nd`).

4.4 Incidence rates of DM

First we (re-)load the tabulated follow-up data

```
> library( Epi )
> library( splines )
> load( file="../data/Atab" )
> summary( Atab )
   sex            sC              A               P             Y.dm             D.dm
   F:7200   Min.   : 1.0   Min.   : 0.0   Min.   :2005   Min.   :  0.00   Min.   : 0.000
   M:7200   1st Qu.: 3.0   1st Qu.:22.0   1st Qu.:2007   1st Qu.: 17.15   1st Qu.: 0.000
            Median : 5.5   Median :44.5   Median :2008   Median : 63.18   Median : 0.000
            Mean   : 5.5   Mean   :44.5   Mean   :2008   Mean   :105.39   Mean   : 3.804
            3rd Qu.: 8.0   3rd Qu.:67.0   3rd Qu.:2010   3rd Qu.:177.91   3rd Qu.: 6.000
            Max.   :10.0   Max.   :89.0   Max.   :2012   Max.   :476.60   Max.   :36.000
        D                N              I.dm             Y.nd             D.nd
   Min.   :  0.0   Min.   : 157   Min.   :  0.00   Min.   : 140.8   Min.   :  0.0
   1st Qu.:  1.0   1st Qu.:2470   1st Qu.:  0.00   1st Qu.:2276.1   1st Qu.:  1.0
   Median :  9.0   Median :3031   Median :  3.00   Median :2969.0   Median :  8.0
   Mean   : 26.1   Mean   :2880   Mean   : 19.42   Mean   :2774.9   Mean   : 22.3
   3rd Qu.: 42.0   3rd Qu.:3576   3rd Qu.: 18.00   3rd Qu.:3478.8   3rd Qu.: 35.0
   Max.   :171.0   Max.   :4720   Max.   :367.00   Max.   :4712.4   Max.   :148.0
> tt <- addmargins( xtabs( cbind(I.dm,Y.nd/1000) ~ A + P,
+                          data=Atab ),
+                   margin = 1 )
> str( tt )
```

```
   table [1:91, 1:8, 1:2] 3 15 32 43 64 82 115 128 172 172 ...
   - attr(*, "dimnames")=List of 3
    ..$ A: chr [1:91] "0" "1" "2" "3" ...
    ..$ P: chr [1:8] "2005" "2006" "2007" "2008" ...
    ..$  : chr [1:2] "I.dm" "V2"
   - attr(*, "class")= chr [1:2] "table" "array"
> cbind( round(tt[,,1]), round(tt[,,2],1) )
```

	2005	2006	2007	2008	2009	2010	2011	2012	2005	2006	2007	2008	2009	2010	201
0	3	1	0	1	1	0	0	0	54.5	55.1	57.0	59.5	59.7	59.3	60.
1	15	12	10	1	9	7	0	0	53.9	54.6	55.4	57.3	59.5	59.5	57.
2	32	12	10	8	5	17	0	0	52.2	53.9	55.0	55.6	57.2	59.3	59.
3	43	11	4	14	15	12	0	0	51.7	52.3	54.2	55.2	55.7	57.0	59.
4	64	13	13	16	21	9	0	0	53.1	51.9	52.5	54.3	55.3	55.7	56.
5	82	21	13	17	12	8	0	0	54.3	53.3	52.1	52.6	54.3	55.4	55.
6	115	16	15	18	22	22	0	0	56.6	54.6	53.7	52.3	52.6	54.3	55.
7	128	18	19	13	25	19	0	0	57.9	56.8	55.0	54.0	52.4	52.6	54.
8	172	28	21	28	25	24	0	0	59.6	58.1	57.2	55.3	54.2	52.5	52.
9	172	26	29	21	25	27	0	0	59.2	59.9	58.4	57.5	55.5	54.4	52.
10	200	39	25	33	32	31	0	0	60.0	59.5	60.3	58.7	57.7	55.7	54.
11	255	22	29	31	35	37	0	0	61.8	60.1	59.9	60.6	58.9	57.9	55.
12	285	37	43	37	25	20	0	0	62.8	62.1	60.4	60.1	60.8	59.1	58.
13	284	33	31	26	25	29	0	0	64.9	62.7	62.2	60.8	60.5	61.1	59.
14	346	23	15	24	21	21	0	0	64.7	64.9	62.9	62.4	61.0	60.7	61.
15	316	19	30	7	21	23	0	0	62.7	64.5	65.1	63.3	62.7	61.3	61.
16	357	12	21	28	17	35	0	0	62.8	62.8	64.6	65.1	63.8	62.9	61.
17	355	18	23	25	17	23	0	0	64.7	63.2	64.1	64.9	68.1	64.1	63.
18	353	21	25	22	25	35	0	0	63.8	65.5	64.5	65.4	68.6	69.6	66.
19	362	21	19	25	26	33	0	0	67.1	67.4	69.2	68.8	67.4	72.1	72.
20	350	20	15	19	17	26	0	0	68.1	69.0	69.5	71.5	68.1	69.9	74.
21	370	19	29	20	35	28	0	0	65.8	69.3	68.6	68.7	70.5	70.0	72.
22	369	21	33	22	23	21	0	0	66.0	65.6	68.8	69.3	69.6	71.9	72.
23	394	19	31	21	29	20	0	0	66.4	66.0	66.3	68.9	69.6	70.1	73.
24	398	27	26	30	31	30	0	0	67.6	66.7	66.6	66.3	68.8	69.6	70.
25	396	30	32	24	35	24	0	0	65.9	67.7	67.6	67.2	66.5	69.0	70.
26	439	25	32	34	37	36	0	0	62.1	65.8	68.1	68.0	67.5	66.6	69.
27	411	40	43	38	32	40	0	0	57.4	62.5	66.7	68.3	68.3	67.8	67.
28	468	33	37	41	48	40	0	0	55.8	57.8	63.3	66.8	68.5	68.5	68.
29	465	40	43	52	45	54	0	0	59.5	56.3	58.6	63.8	66.9	68.5	69.
30	585	53	44	46	61	53	0	0	60.3	59.9	57.2	59.3	64.4	67.0	68.
31	627	50	55	55	64	53	0	0	61.3	60.7	60.7	57.9	59.7	64.7	67.
32	713	41	55	58	63	71	0	0	65.6	61.8	61.5	61.3	58.2	60.0	65.
33	783	72	50	73	74	56	0	0	70.0	65.9	62.5	61.9	61.7	58.6	60.
34	836	81	73	76	77	93	0	0	73.7	70.4	66.7	63.1	62.4	62.2	59.
35	892	89	83	91	73	106	0	0	73.1	74.0	71.0	67.2	63.5	62.7	62.
36	987	101	92	108	99	84	0	0	76.2	73.5	74.5	71.5	67.5	63.8	63.
37	1166	115	123	140	125	94	0	0	77.9	76.5	73.7	74.8	71.9	67.8	64.
38	1167	129	125	124	125	112	0	0	79.3	78.1	76.9	74.1	75.1	72.2	68.
39	1330	159	134	179	156	156	0	0	78.7	79.4	78.5	77.1	74.4	75.2	72.
40	1463	137	165	174	191	175	0	0	81.2	78.7	79.6	78.7	77.1	74.4	75.
41	1567	177	176	209	187	199	0	0	80.8	81.3	78.8	79.6	78.7	77.1	74.
42	1736	214	181	199	212	204	0	0	80.4	80.7	81.5	79.0	79.4	78.6	77.
43	1869	201	227	203	205	216	0	0	78.8	80.4	80.8	81.5	78.9	79.3	78.
44	2021	244	222	267	302	252	0	0	77.3	78.7	80.3	80.7	81.4	78.7	79.
45	2063	257	270	256	281	267	0	0	74.8	77.0	78.6	80.3	80.5	81.2	78.
46	2227	285	283	284	294	323	0	0	74.2	74.4	76.8	78.4	80.0	80.2	81.
47	2490	306	284	311	319	331	0	0	72.2	73.9	74.2	76.6	78.2	79.8	80.
48	2573	307	314	333	352	362	0	0	70.4	71.9	73.8	73.9	76.2	77.8	79.
49	2694	330	358	338	372	340	0	0	68.5	70.1	71.6	73.4	73.5	75.8	77.
50	2884	325	349	394	399	377	0	0	65.5	68.1	69.8	71.3	73.0	73.0	75.
51	2918	366	378	345	395	391	0	0	64.9	65.1	67.8	69.4	70.8	72.5	72.
52	3162	414	379	442	410	431	0	0	63.6	64.4	64.7	67.3	68.8	70.3	72.
53	3202	361	403	397	407	426	0	0	61.2	63.2	63.9	64.3	66.7	68.3	70.
54	3387	383	364	468	452	453	0	0	62.6	60.8	62.7	63.4	63.7	66.0	68.
55	3826	426	404	420	452	417	0	0	63.0	62.0	60.3	62.2	62.7	63.0	65.
56	4098	459	414	471	460	441	0	0	64.8	62.4	61.5	59.8	61.6	62.1	62.
57	4767	499	434	437	476	480	0	0	66.8	64.1	61.7	60.9	59.1	60.9	61.

58	4708	540	419	442	463	466	0	0	72.2	66.0	63.4	60.9	60.1	58.4	60.
59	4000	531	489	463	491	458	0	0	54.9	71.4	65.3	62.7	60.2	59.3	58.
60	4412	527	544	585	562	501	0	0	52.9	54.1	70.6	64.5	61.8	59.4	58.
61	4567	459	534	630	546	545	0	0	54.0	52.1	53.3	69.8	63.4	60.9	58.
62	4644	468	434	536	622	508	0	0	52.3	53.1	51.2	52.4	68.7	62.5	60.
63	4726	462	531	421	548	600	0	0	48.2	51.3	52.1	50.4	51.4	67.6	61.
64	4868	478	514	468	492	541	0	0	45.1	47.2	50.4	51.0	49.4	50.3	66.
65	5149	426	481	506	477	430	0	0	46.4	44.1	46.3	49.3	50.0	48.4	49.
66	5574	471	450	458	542	480	0	0	45.5	45.3	43.1	45.3	48.2	48.9	47.
67	5499	453	431	434	491	448	0	0	44.3	44.4	44.2	42.0	44.2	47.1	48.
68	5621	475	473	468	507	452	0	0	42.8	43.1	43.2	43.1	40.9	43.0	46.
69	5603	464	428	432	441	417	0	0	41.6	41.5	42.0	42.0	41.9	39.8	42.
70	5553	465	415	461	455	445	0	0	40.3	40.3	40.3	40.7	40.8	40.6	38.
71	5302	455	462	441	436	424	0	0	38.1	38.9	38.9	39.1	39.5	39.5	39.
72	5166	414	407	387	435	415	0	0	36.6	36.6	37.6	37.6	37.7	38.2	38.
73	5031	392	381	407	444	365	0	0	36.7	35.2	35.2	36.2	36.3	36.3	37.
74	5144	361	355	391	376	390	0	0	35.0	35.2	33.8	33.8	34.8	35.0	35.
75	4528	394	308	344	387	327	0	0	33.2	33.4	33.7	32.4	32.4	33.4	33.
76	4357	336	310	319	357	291	0	0	31.6	31.5	31.8	32.2	30.9	31.0	32.
77	3840	296	308	298	313	303	0	0	28.5	29.8	29.8	30.3	30.6	29.4	29.
78	3764	255	253	298	274	270	0	0	27.6	26.8	28.2	28.0	28.7	29.1	28.
79	3483	256	242	270	269	268	0	0	26.7	25.9	25.2	26.5	26.4	27.1	27.
80	3109	188	199	227	238	222	0	0	24.2	24.9	24.2	23.7	24.8	24.6	25.
81	2639	202	187	170	198	199	0	0	22.9	22.4	23.1	22.5	22.0	23.1	23.
82	2357	163	166	187	171	156	0	0	20.3	21.1	20.7	21.3	20.8	20.5	21.
83	2192	138	135	133	159	149	0	0	19.8	18.5	19.2	18.8	19.5	19.2	18.
84	2091	128	118	137	146	154	0	0	19.4	17.9	16.9	17.4	17.1	17.8	17.
85	1414	103	100	122	109	111	0	0	18.1	17.5	16.1	15.1	15.7	15.5	16.
86	1042	100	72	91	83	89	0	0	11.1	16.2	15.6	14.3	13.5	14.0	14.
87	907	54	88	86	63	80	0	0	9.3	9.8	14.3	13.8	12.6	12.0	12.
88	765	49	42	82	56	58	0	0	8.5	8.0	8.4	12.4	12.1	11.1	10.
89	675	39	43	41	57	50	0	0	7.7	7.3	6.9	7.2	10.7	10.5	9.
Sum	188762	17800	17474	18329	19027	18326	0	0	4905.9	4917.7	4944.8	4968.2	4986.5	5003.1	5038.

	2012
0	58.7
1	60.6
2	57.8
3	59.5
4	59.3
5	57.0
6	55.8
7	55.4
8	54.4
9	52.7
10	52.6
11	54.7
12	56.1
13	58.3
14	59.5
15	61.6
16	61.4
17	62.2
18	65.4
19	68.6
20	74.4
21	76.2
22	73.6
23	72.6
24	73.0
25	70.7
26	70.1
27	69.7
28	67.3
29	68.3
30	69.3
31	68.9

```
32    67.6
33    65.3
34    60.7
35    59.6
36    63.3
37    63.8
38    64.9
39    68.8
40    73.2
41    76.2
42    75.3
43    78.1
44    79.7
45    80.4
46    79.8
47    82.6
48    81.6
49    81.3
50    79.4
51    77.5
52    74.7
53    74.3
54    72.2
55    70.3
56    68.0
57    65.1
58    64.3
59    63.1
60    60.8
61    61.5
62    61.5
63    63.3
64    65.0
65    70.5
66    53.1
67    50.3
68    51.1
69    49.1
70    45.1
71    41.7
72    42.3
73    41.2
74    40.0
75    37.8
76    36.3
77    34.4
78    31.9
79    29.8
80    29.2
81    26.8
82    24.2
83    22.1
84    19.3
85    17.6
86    15.9
87    13.6
88    11.9
89     9.9
Sum 5193.8
```

Thus we see that there are no incident cases recorded in the years 2011 and 2012, so for the incidence analysis we restrict the data to the 6 year period 2005–2010, and we also recode the age and period variables to represent the midpoint of the intervals:

```
> Iana <- transform( subset( Atab, A<2011 ),
+                    A = A + 0.5,
+                    p = P + 0.5 )
```

We start by setting up a simple model with age, calendar time and social status, and we expect to see similar effects as for prevalence because incidence rates are the main drivers of prevalence. So the model will look a lot like the model for prevalences, but while the prevalence is modelled using the binomial distribution for fractions, incidence rates are modelled using the Poisson distribution (or more precisely the Poisson likelihood):

```
> a.kn <- seq(5,85,,10)
> p.kn <- c(2006.5,2008,2009.5)
> im1 <- glm( I.dm ~ Ns(A,kn=a.kn) + Ns(P,kn=p.kn) + factor(sC),
+                    offset = log(Y.nd),
+                    family = poisson,
+                      data = subset(Iana,sex=="M") )
> if1 <- update( im1,  data = subset(Iana,sex=="F") )

> round( cbind( ci.exp( im1 ), ci.exp( if1 ) ), 3 )
                    exp(Est.)    2.5%   97.5% exp(Est.)    2.5%   97.5%
  (Intercept)           0.000   0.000   0.000     0.000   0.000   0.000
  Ns(A, kn = a.kn)1     2.635   2.376   2.924     1.807   1.618   2.019
  Ns(A, kn = a.kn)2     7.630   6.901   8.436     5.621   5.055   6.250
  Ns(A, kn = a.kn)3    13.896  12.744  15.152     8.218   7.506   8.998
  Ns(A, kn = a.kn)4    38.354  35.267  41.711    22.475  20.610  24.509
  Ns(A, kn = a.kn)5    55.829  51.456  60.574    32.945  30.315  35.803
  Ns(A, kn = a.kn)6    95.513  88.035 103.626    65.400  60.237  71.007
  Ns(A, kn = a.kn)7    68.475  64.559  72.629    48.976  46.176  51.947
  Ns(A, kn = a.kn)8   280.596 234.953 335.106   197.959 165.449 236.857
  Ns(A, kn = a.kn)9    28.936  27.434  30.520    21.958  20.818  23.161
  Ns(P, kn = p.kn)1     0.079   0.078   0.080     0.072   0.071   0.073
  Ns(P, kn = p.kn)2     0.348   0.344   0.351     0.348   0.343   0.352
  factor(sC)2           0.992   0.971   1.013     0.941   0.920   0.962
  factor(sC)3           0.951   0.930   0.971     0.875   0.856   0.895
  factor(sC)4           0.913   0.894   0.933     0.848   0.829   0.867
  factor(sC)5           0.870   0.852   0.889     0.777   0.759   0.795
  factor(sC)6           0.798   0.780   0.816     0.725   0.708   0.743
  factor(sC)7           0.791   0.773   0.808     0.692   0.676   0.709
  factor(sC)8           0.767   0.750   0.784     0.658   0.642   0.675
  factor(sC)9           0.729   0.713   0.746     0.607   0.592   0.623
  factor(sC)10          0.611   0.597   0.626     0.478   0.465   0.491
```

As for the prevalences we can see the clear decline in RR relative to the most depraved areas (`sC` 1), and a stronger effect among women than among men.

However, we want to show how the age-specific incidence rates of diabetes in Scotland looks, but in order to do this we must decide on reference points for year and social class. Then we can produce separate curves for men and women.

As before we set up a prediction data frame and use that for extraction of the rates. Now note that we now need to give a value for the person-years (`Y.nd`) too, in order to get the rates in the right units, in this case as events per 1000 PY.

```
> nd <- data.frame( A = 0:90,
+                   P = 2008,
+                  sC = 5,
+                Y.nd = 1000 )
> minc2008 <- ci.pred( im1, newdata = nd )
> finc2008 <- ci.pred( if1, newdata = nd )
```

Having collected the incidence rates separately for men and women we can plot the together:

```
> par( mar=c(3,4,1,1) )
> matplot( nd$A, cbind( minc2008,
+                       finc2008 ),
+          lwd=c(3,1,1), col=rep(c("blue","red"),each=3), lty=1, type="l",
+          log="y", xlab="Age (years)", ylab=" ")
> mtext( "DM incidence rate (per 1000 PY)", side=2, line=2.5, las=0 )
```

4.4.1 Age by social class interaction

If we want to explore *if* there is an interaction between age and social class and how it looks we make an interaction term where the age-effect (*i.e.* the `differences` in age-effects), have fewer degrees of freedom than the overall age-effect we have modelled:

```
> r.kn <- seq(2,88,,4)
> imi <- update( im1, . ~ . + factor(sC):Ns(A,knots=r.kn) )
> ifi <- update( if1, . ~ . + factor(sC):Ns(A,knots=r.kn) )
> summary( ifi )
  Call:
  glm(formula = I.dm ~ Ns(A, kn = a.kn) + Ns(P, kn = p.kn) + factor(sC) +
      factor(sC):Ns(A, knots = r.kn), family = poisson, data = subset(Iana,
      sex == "F"), offset = log(Y.nd))

  Deviance Residuals:
       Min         1Q     Median         3Q        Max
  -10.1601    -1.8776    -0.4784     1.5887     7.2991

  Coefficients: (1 not defined because of singularities)
                                      Estimate Std. Error  z value Pr(>|z|)
  (Intercept)                        -3.666e+01  3.864e+00   -9.487  < 2e-16
  Ns(A, kn = a.kn)1                  -1.508e+02  2.073e+01   -7.272 3.55e-13
  Ns(A, kn = a.kn)2                  -1.965e+02  2.720e+01   -7.225 5.00e-13
  Ns(A, kn = a.kn)3                  -2.124e+02  2.955e+01   -7.187 6.63e-13
  Ns(A, kn = a.kn)4                  -2.073e+02  2.916e+01   -7.108 1.18e-12
  Ns(A, kn = a.kn)5                  -1.904e+02  2.705e+01   -7.040 1.92e-12
  Ns(A, kn = a.kn)6                  -1.704e+02  2.434e+01   -7.000 2.55e-12
  Ns(A, kn = a.kn)7                  -1.265e+02  1.812e+01   -6.982 2.91e-12
  Ns(A, kn = a.kn)8                  -2.035e+02  2.865e+01   -7.101 1.24e-12
  Ns(A, kn = a.kn)9                  -8.207e+01  1.170e+01   -7.015 2.30e-12
  Ns(P, kn = p.kn)1                  -2.626e+00  6.932e-03 -378.750  < 2e-16
  Ns(P, kn = p.kn)2                  -1.057e+00  6.102e-03 -173.278  < 2e-16
  factor(sC)2                         2.144e-01  1.348e-01    1.591 0.111631
  factor(sC)3                         4.607e-01  1.335e-01    3.451 0.000559
  factor(sC)4                         3.345e-01  1.369e-01    2.444 0.014541
  factor(sC)5                         4.508e-01  1.361e-01    3.313 0.000922
  factor(sC)6                         4.603e-01  1.370e-01    3.360 0.000780
  factor(sC)7                         5.400e-01  1.358e-01    3.977 6.98e-05
  factor(sC)8                         6.713e-01  1.349e-01    4.977 6.46e-07
  factor(sC)9                         6.251e-01  1.366e-01    4.577 4.72e-06
  factor(sC)10                        7.030e-01  1.369e-01    5.136 2.81e-07
  factor(sC)1:Ns(A, knots = r.kn)1    1.481e+02  2.126e+01    6.968 3.23e-12
  factor(sC)2:Ns(A, knots = r.kn)1    1.479e+02  2.126e+01    6.957 3.47e-12
  factor(sC)3:Ns(A, knots = r.kn)1    1.477e+02  2.126e+01    6.949 3.67e-12
  factor(sC)4:Ns(A, knots = r.kn)1    1.476e+02  2.126e+01    6.945 3.80e-12
  factor(sC)5:Ns(A, knots = r.kn)1    1.474e+02  2.126e+01    6.936 4.03e-12
  factor(sC)6:Ns(A, knots = r.kn)1    1.473e+02  2.126e+01    6.931 4.17e-12
  factor(sC)7:Ns(A, knots = r.kn)1    1.472e+02  2.126e+01    6.926 4.33e-12
  factor(sC)8:Ns(A, knots = r.kn)1    1.470e+02  2.126e+01    6.915 4.67e-12
  factor(sC)9:Ns(A, knots = r.kn)1    1.470e+02  2.126e+01    6.917 4.60e-12
  factor(sC)10:Ns(A, knots = r.kn)1   1.468e+02  2.126e+01    6.905 5.01e-12
  factor(sC)1:Ns(A, knots = r.kn)2    4.061e+02  5.554e+01    7.312 2.63e-13
  factor(sC)2:Ns(A, knots = r.kn)2    4.056e+02  5.554e+01    7.302 2.83e-13
  factor(sC)3:Ns(A, knots = r.kn)2    4.049e+02  5.554e+01    7.290 3.09e-13
  factor(sC)4:Ns(A, knots = r.kn)2    4.052e+02  5.554e+01    7.296 2.97e-13
```

```
factor(sC)5:Ns(A, knots = r.kn)2   4.048e+02  5.554e+01    7.289 3.12e-13
factor(sC)6:Ns(A, knots = r.kn)2   4.047e+02  5.554e+01    7.286 3.18e-13
factor(sC)7:Ns(A, knots = r.kn)2   4.044e+02  5.554e+01    7.282 3.29e-13
factor(sC)8:Ns(A, knots = r.kn)2   4.041e+02  5.554e+01    7.276 3.43e-13
factor(sC)9:Ns(A, knots = r.kn)2   4.040e+02  5.554e+01    7.274 3.49e-13
factor(sC)10:Ns(A, knots = r.kn)2  4.034e+02  5.554e+01    7.264 3.76e-13
factor(sC)1:Ns(A, knots = r.kn)3  -8.884e-02  6.420e-02   -1.384 0.166419
factor(sC)2:Ns(A, knots = r.kn)3  -1.112e-01  6.358e-02   -1.749 0.080235
factor(sC)3:Ns(A, knots = r.kn)3  -1.797e-01  6.318e-02   -2.845 0.004442
factor(sC)4:Ns(A, knots = r.kn)3  -1.148e-02  6.394e-02   -0.179 0.857579
factor(sC)5:Ns(A, knots = r.kn)3  -3.370e-02  6.427e-02   -0.524 0.599989
factor(sC)6:Ns(A, knots = r.kn)3   5.792e-02  6.488e-02    0.893 0.371954
factor(sC)7:Ns(A, knots = r.kn)3   1.007e-01  6.483e-02    1.554 0.120190
factor(sC)8:Ns(A, knots = r.kn)3   2.690e-01  6.451e-02    4.170 3.05e-05
factor(sC)9:Ns(A, knots = r.kn)3   2.401e-01  6.598e-02    3.638 0.000274
factor(sC)10:Ns(A, knots = r.kn)3         NA         NA       NA       NA

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 396949  on 7199  degrees of freedom
Residual deviance:  52001  on 7150  degrees of freedom
AIC: 71729

Number of Fisher Scoring iterations: 7
```

Note that the last parameter is `NA`; this is because it is *aliased* — the natural spline basis `Ns(A,knots=r.kn)` includes a *linear* term in `A`, which is also included in the original spline term for `A`, and hence only is estimable for 9 out of the 10 social class strata. This will give a warning when we do prediction, but this type of aliasing will give the correct predictions anyway.

But we just take a look at the formal significance of the interaction, we see that it is massive:

```
> anova( imi, im1, test="Chisq" )
Analysis of Deviance Table

Model 1: I.dm ~ Ns(A, kn = a.kn) + Ns(P, kn = p.kn) + factor(sC) + factor(sC):Ns(A,
    knots = r.kn)
Model 2: I.dm ~ Ns(A, kn = a.kn) + Ns(P, kn = p.kn) + factor(sC)
  Resid. Df Resid. Dev  Df Deviance  Pr(>Chi)
1      7150      63368
2      7179      64414 -29  -1045.3 < 2.2e-16
> anova( ifi, if1, test="Chisq" )
Analysis of Deviance Table

Model 1: I.dm ~ Ns(A, kn = a.kn) + Ns(P, kn = p.kn) + factor(sC) + factor(sC):Ns(A,
    knots = r.kn)
Model 2: I.dm ~ Ns(A, kn = a.kn) + Ns(P, kn = p.kn) + factor(sC)
  Resid. Df Resid. Dev  Df Deviance  Pr(>Chi)
1      7150      52001
2      7179      53213 -29    -1212 < 2.2e-16
```

However the main interest is in the *shape* of the interactions, so we predict the incidence rates separately for each sex and social class and plot the. To this end we first set up a 3-dimensional array to hold the predictions:

```
> ii <- NArray( list( A = nd$A,
+                    sex = c("M","F"),
+                     sC = 1:10 ) )
> str( ii )
```

```
   logi [1:91, 1:2, 1:10] NA NA NA NA NA NA ...
   - attr(*, "dimnames")=List of 3
    ..$ A  : chr [1:91] "0" "1" "2" "3" ...
    ..$ sex: chr [1:2] "M" "F"
    ..$ sC : chr [1:10] "1" "2" "3" "4" ...
```

With this in place we can fill in the array:

```
> for( sc in 1:10 )
+    {
+  ii["M",sc] <- ci.pred( imi, newdata = transform( nd, sC=sc ) )[,1]
+  ii["F",sc] <- ci.pred( ifi, newdata = transform( nd, sC=sc ) )[,1]
+    }
```

Then we can plot the estimated incidence rates in different strata separately for men and women:

```
> par( mfrow=c(1,2), mar=c(3,1,1,1), oma=c(0,4,0,0), mgp=c(3,1,0)/1.6,
+      las=1, bty="n" )
> matplot( nd$A, ii["M",],
+          lwd=2:3, col=gray(1:10/14), lty=1, type="l",
+          log="y", xlab="Age (years)", ylab=" ", ylim=c(5/199,10) )
> matplot( nd$A, ii["F",],
+          lwd=2:3, col=gray(1:10/14), lty=1, type="l",
+          log="y", xlab="Age (years)", ylab=" ", ylim=c(5/199,10) )
> mtext( "DM incidence rate (per 1000 PY)", side=2, line=2.5, las=0, outer=TRUE )
```

From figure 4.4 It is seen that the social gradient crosses over at age about 15, and largely disappears after age 75. The qualitative pattern in the interaction is similar among men and women, but clearly the gradient much more pronounced in young ages (< 15) among men, and more pronounced among women in ages over 25. Because the latter age-range has by far the most cases, it is only larger social gradient among women that is seen in the non-interaction models.

4.5 Mortality rates in Scottish diabetes patients

As for the incidence data we (re-)load the tabulated follow-up data

```
> library( Epi )
> library( splines )
> load( file="../data/Atab" )
> summary( Atab )
   sex            sC              A               P              Y.dm             D.dm
   F:7200   Min.   : 1.0   Min.   : 0.0   Min.   :2005   Min.   :  0.00   Min.   : 0.000
   M:7200   1st Qu.: 3.0   1st Qu.:22.0   1st Qu.:2007   1st Qu.: 17.15   1st Qu.: 0.000
            Median : 5.5   Median :44.5   Median :2008   Median : 63.18   Median : 0.000
            Mean   : 5.5   Mean   :44.5   Mean   :2008   Mean   :105.39   Mean   : 3.804
            3rd Qu.: 8.0   3rd Qu.:67.0   3rd Qu.:2010   3rd Qu.:177.91   3rd Qu.: 6.000
            Max.   :10.0   Max.   :89.0   Max.   :2012   Max.   :476.60   Max.   :36.000
        D               N             I.dm             Y.nd             D.nd
   Min.   :  0.0   Min.   : 157   Min.   :  0.00   Min.   : 140.8   Min.   :  0.0
   1st Qu.:  1.0   1st Qu.:2470   1st Qu.:  0.00   1st Qu.:2276.1   1st Qu.:  1.0
   Median :  9.0   Median :3031   Median :  3.00   Median :2969.0   Median :  8.0
   Mean   : 26.1   Mean   :2880   Mean   : 19.42   Mean   :2774.9   Mean   : 22.3
   3rd Qu.: 42.0   3rd Qu.:3576   3rd Qu.: 18.00   3rd Qu.:3478.8   3rd Qu.: 35.0
   Max.   :171.0   Max.   :4720   Max.   :367.00   Max.   :4712.4   Max.   :148.0
```

```
> tt <- addmargins( xtabs( cbind(D.dm,Y.dm/1000) ~ A + P,
+                          data=Atab ),
+                   margin = 1 )
> str( tt )
```

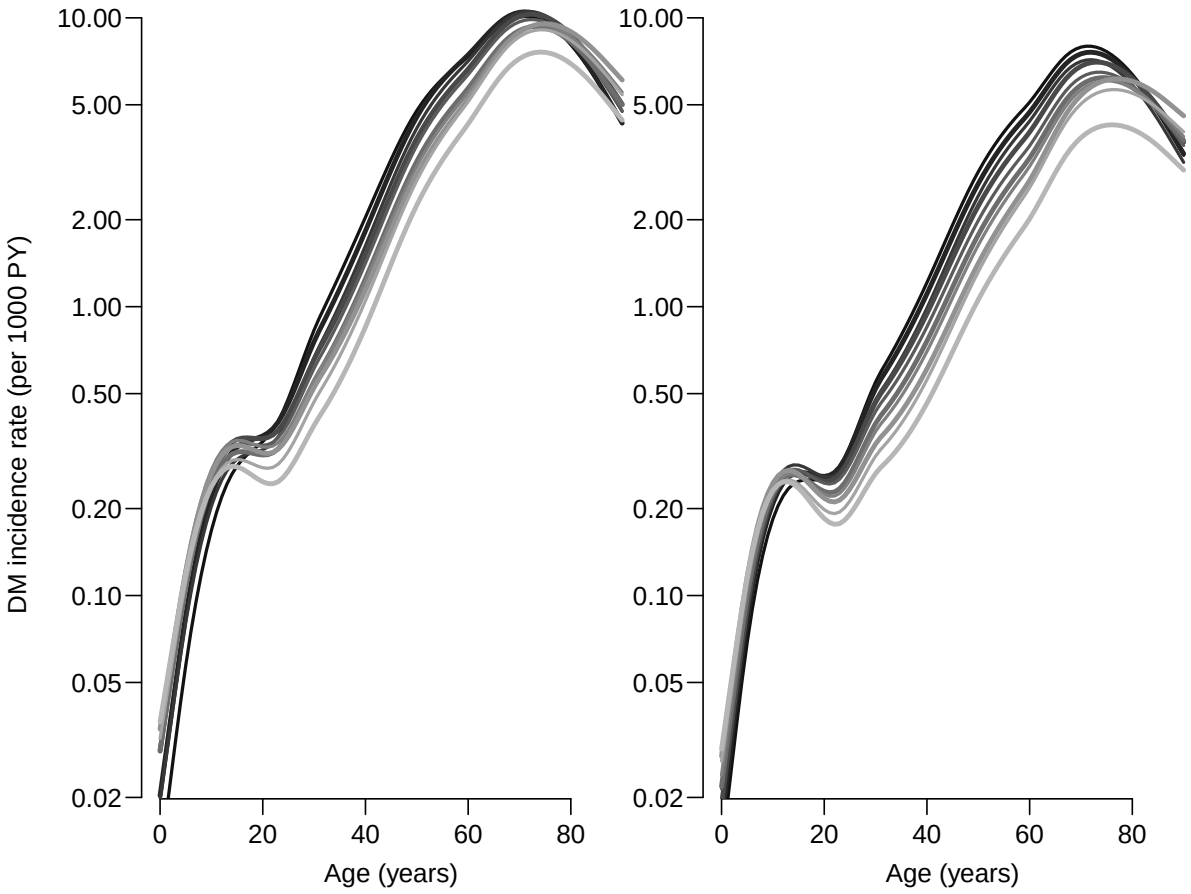

Figure 4.4: *Predicted incidence rates of DM in Scotland for social classes 1–10 (light to dark).*

```
  table [1:91, 1:8, 1:2] 0 0 0 0 0 0 0 0 0 0 ...
  - attr(*, "dimnames")=List of 3
   ..$ A: chr [1:91] "0" "1" "2" "3" ...
   ..$ P: chr [1:8] "2005" "2006" "2007" "2008" ...
   ..$  : chr [1:2] "D.dm" "V2"
  - attr(*, "class")= chr [1:2] "table" "array"
```

```
> cbind( round(tt[,,1]), round(tt[,,2],1) )
```

	2005	2006	2007	2008	2009	2010	2011	2012	2005	2006	2007	2008	2009	2010	2011	2012
0	0	0	0	0	0	0	0	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0	0	0	0	0	0	0	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0	0	0	0	0	0	0	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0	0	0	0	0	0	0	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0	0	0	0	0	0	0	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0	0	0	0	0	0	0	0	0.1	0.1	0.0	0.1	0.0	0.1	0.0	0.0
6	0	0	0	0	0	0	0	0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
7	0	0	0	0	0	0	0	0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
8	0	0	0	0	0	0	0	0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
9	0	0	0	0	0	0	0	0	0.2	0.2	0.2	0.1	0.1	0.1	0.1	0.0
10	0	0	0	0	0	0	0	0	0.2	0.2	0.2	0.2	0.2	0.2	0.1	0.0
11	0	0	0	0	0	0	0	0	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.1
12	0	0	0	0	0	1	0	0	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.1
13	0	0	1	1	0	0	1	0	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.1
14	0	0	0	0	0	0	0	0	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.1
15	0	0	1	0	0	1	0	0	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.1
16	1	0	0	0	0	0	0	0	0.3	0.4	0.3	0.3	0.3	0.3	0.3	0.1

17	2	1	0	2	0	0	0	1	0.4	0.3	0.4	0.4	0.4	0.3	0.3	0.1
18	0	1	0	2	3	1	1	0	0.3	0.4	0.3	0.4	0.4	0.4	0.4	0.1
19	0	2	3	1	2	2	0	1	0.3	0.4	0.4	0.4	0.4	0.4	0.4	0.1
20	0	3	0	4	0	0	2	1	0.3	0.4	0.4	0.4	0.4	0.4	0.4	0.2
21	3	2	2	1	1	1	1	0	0.3	0.4	0.4	0.4	0.4	0.4	0.5	0.2
22	2	1	1	3	3	3	0	0	0.4	0.4	0.4	0.4	0.4	0.5	0.4	0.2
23	1	1	2	2	2	1	2	1	0.4	0.4	0.4	0.4	0.4	0.4	0.5	0.2
24	1	3	0	4	1	1	2	1	0.4	0.4	0.4	0.4	0.4	0.5	0.4	0.2
25	0	2	2	0	0	0	2	0	0.4	0.4	0.4	0.4	0.4	0.5	0.5	0.2
26	0	2	4	1	0	0	0	0	0.4	0.4	0.4	0.5	0.5	0.5	0.5	0.2
27	1	4	2	2	2	1	3	0	0.4	0.4	0.4	0.5	0.5	0.5	0.5	0.2
28	1	3	0	2	2	1	2	1	0.4	0.5	0.4	0.5	0.5	0.5	0.5	0.2
29	2	2	2	2	6	1	3	2	0.5	0.4	0.5	0.5	0.5	0.6	0.5	0.2
30	2	1	2	2	2	1	4	0	0.5	0.5	0.5	0.5	0.5	0.6	0.6	0.2
31	3	4	3	4	9	1	2	1	0.6	0.5	0.5	0.5	0.6	0.6	0.6	0.2
32	3	4	5	2	1	5	4	0	0.6	0.6	0.5	0.6	0.6	0.6	0.6	0.2
33	3	8	4	2	1	2	3	4	0.7	0.7	0.7	0.6	0.7	0.7	0.7	0.2
34	6	1	0	1	4	6	1	0	0.8	0.8	0.7	0.7	0.7	0.7	0.7	0.3
35	3	4	3	2	5	3	6	1	0.8	0.8	0.9	0.8	0.8	0.8	0.8	0.3
36	1	3	2	5	5	6	3	1	0.9	0.9	0.9	0.9	0.9	0.9	0.8	0.3
37	3	6	9	6	7	7	6	3	1.0	1.0	1.0	1.0	1.1	1.0	0.9	0.3
38	7	6	7	8	4	7	8	1	1.1	1.1	1.1	1.1	1.2	1.2	1.0	0.3
39	3	9	9	9	4	14	5	3	1.1	1.2	1.2	1.2	1.3	1.3	1.2	0.4
40	6	9	11	7	10	14	10	4	1.3	1.3	1.4	1.4	1.4	1.4	1.4	0.5
41	8	7	11	12	10	11	10	2	1.4	1.4	1.4	1.5	1.5	1.6	1.5	0.5
42	7	7	9	8	13	11	5	3	1.5	1.6	1.6	1.6	1.7	1.7	1.7	0.6
43	9	12	9	13	13	9	10	1	1.6	1.7	1.8	1.8	1.8	1.9	1.8	0.6
44	10	16	8	13	10	10	16	4	1.8	1.9	1.9	2.0	2.1	2.0	2.0	0.7
45	16	11	12	14	19	11	9	2	1.9	2.0	2.1	2.2	2.3	2.3	2.1	0.8
46	18	15	16	15	16	21	13	5	2.0	2.1	2.3	2.4	2.4	2.5	2.4	0.8
47	21	21	23	16	22	19	15	6	2.1	2.2	2.4	2.5	2.7	2.7	2.7	0.9
48	22	18	18	15	21	21	17	6	2.3	2.4	2.5	2.7	2.8	3.0	2.9	1.0
49	18	27	27	26	27	22	22	9	2.4	2.6	2.7	2.8	3.1	3.2	3.1	1.1
50	22	23	30	43	20	28	33	9	2.6	2.7	3.0	3.1	3.1	3.4	3.3	1.2
51	25	33	29	34	30	32	36	6	2.7	2.9	3.1	3.3	3.4	3.5	3.6	1.3
52	30	34	28	35	27	34	32	15	2.8	3.0	3.2	3.4	3.7	3.8	3.7	1.3
53	22	36	35	41	38	43	31	8	3.0	3.1	3.4	3.6	3.8	4.0	4.0	1.4
54	41	36	51	39	34	54	43	22	3.0	3.3	3.5	3.8	3.9	4.2	4.2	1.5
55	48	39	51	37	60	56	46	19	3.2	3.4	3.6	3.8	4.2	4.3	4.4	1.6
56	46	31	53	49	49	53	52	15	3.6	3.6	3.8	4.0	4.2	4.6	4.5	1.7
57	77	60	62	46	61	82	49	18	4.0	4.0	4.0	4.2	4.5	4.6	4.8	1.7
58	78	66	59	61	49	56	68	15	4.5	4.4	4.4	4.4	4.6	4.9	4.8	1.8
59	74	67	84	69	68	71	66	37	4.2	5.0	4.8	4.8	4.8	5.0	5.0	1.8
60	62	95	91	94	69	85	83	27	3.7	4.7	5.5	5.3	5.2	5.3	5.2	1.9
61	99	88	95	113	105	100	82	38	4.2	4.1	5.1	6.0	5.7	5.7	5.4	1.9
62	91	98	80	84	101	111	110	36	4.3	4.6	4.5	5.6	6.5	6.2	5.8	2.1
63	88	110	107	111	118	139	104	49	4.4	4.7	5.0	4.8	6.0	7.0	6.3	2.2
64	103	118	122	131	111	151	145	43	4.4	4.7	5.1	5.3	5.1	6.5	7.1	2.4
65	144	129	133	135	129	116	139	44	4.6	4.8	5.1	5.4	5.7	5.5	6.5	2.6
66	133	104	136	158	121	159	120	43	4.9	5.0	5.1	5.4	5.8	6.0	5.6	2.3
67	183	150	143	148	161	151	159	54	5.2	5.3	5.3	5.4	5.8	6.2	6.1	2.1
68	169	176	179	152	148	190	163	56	5.1	5.6	5.6	5.6	5.7	6.1	6.2	2.3
69	193	186	205	176	199	195	200	67	5.3	5.4	5.8	5.8	5.9	6.0	6.1	2.3
70	192	199	200	210	207	172	218	78	5.2	5.6	5.7	6.1	6.1	6.1	6.0	2.2
71	211	225	236	211	217	262	222	72	5.2	5.5	5.8	5.9	6.3	6.3	6.1	2.2
72	200	220	237	288	231	259	252	90	4.9	5.5	5.7	6.0	6.1	6.5	6.3	2.2
73	236	246	269	255	279	285	240	101	4.8	5.1	5.6	5.8	6.1	6.3	6.4	2.3
74	249	255	262	309	246	311	286	109	4.7	5.0	5.2	5.7	5.9	6.3	6.2	2.3
75	286	275	323	325	322	285	319	115	4.7	4.9	5.1	5.3	5.8	6.0	6.1	2.2
76	255	269	289	325	290	307	319	127	4.2	4.8	4.9	5.1	5.3	5.8	5.9	2.2
77	301	278	297	283	304	327	325	125	4.0	4.2	4.8	4.9	5.1	5.3	5.6	2.1
78	305	322	286	348	327	372	302	142	3.5	4.0	4.2	4.8	4.9	5.1	5.2	2.0
79	270	250	296	338	339	335	341	138	3.4	3.5	3.9	4.2	4.7	4.8	4.9	1.8
80	265	307	327	370	331	360	357	116	3.1	3.4	3.5	3.8	4.1	4.6	4.6	1.7
81	295	315	308	310	336	347	404	137	2.8	3.0	3.2	3.3	3.7	4.0	4.3	1.6
82	249	314	283	321	319	348	355	148	2.4	2.7	2.9	3.1	3.2	3.5	3.7	1.5

```
  83   259  257  313  295  333  304  336  153   2.1   2.3   2.5   2.8   3.0   3.0   3.3  1.3
  84   268  270  260  275  287  330  334  125   2.0   2.0   2.1   2.4   2.6   2.8   2.8  1.1
  85   250  238  284  280  286  335  321  118   1.8   1.8   1.8   1.9   2.2   2.5   2.5  0.9
  86   182  228  246  241  269  258  332  116   1.2   1.7   1.7   1.7   1.8   2.0   2.2  0.8
  87   173  184  271  260  225  251  264  128   0.9   1.1   1.6   1.5   1.5   1.6   1.8  0.7
  88   139  134  183  245  244  227  237  112   0.8   0.8   0.9   1.4   1.3   1.3   1.4  0.6
  89   124  124  129  149  227  202  200   86   0.7   0.7   0.7   0.8   1.2   1.1   1.2  0.4
  Sum 6620 6805 7280 7601 7542 7998 7913 3021 175.8 186.4 196.5 206.2 216.7 226.8 226.5 82.9
```

Thus we see that there are about half as many deaths recorded in 2012 as in previous years, consistent with the follow-up for death only till 18 May 2012, so for the mortality analysis we restrict the data to the 7 year period 2005–2011, as well as only the units where we actually do have follow-up. We also recode the age and period variables to represent the midpoint of the intervals:

```
> Mana <- transform( subset( Atab, A<2012 & Y.dm>0 ),
+                    A = A + 0.5,
+                    p = P + 0.5 )
```

We start by setting up a simple model with age, calendar time and social status:

```
> a.kn <- c(10,20,40,seq(50,85,,5))
> p.kn <- c(2006.5,2008.5,2010.5)
> mm1 <- glm( D.dm ~ Ns(A,kn=a.kn) + Ns(P,kn=p.kn) + factor(sC),
+                    offset = log(Y.dm),
+                    family = poisson,
+                      data = subset(Mana,sex=="M") )
> mf1 <- update( mm1,  data = subset(Mana,sex=="F") )

> round( cbind( ci.exp( mm1 ), ci.exp( mf1 ) ), 3 )
                    exp(Est.)    2.5%     97.5% exp(Est.)    2.5%     97.5%
  (Intercept)           0.000   0.000     0.001     0.000   0.000     0.001
  Ns(A, kn = a.kn)1    14.900   5.614    39.541     7.845   3.038    20.261
  Ns(A, kn = a.kn)2    21.875   7.545    63.420    20.001   6.987    57.250
  Ns(A, kn = a.kn)3    42.947  15.329   120.325    33.826  12.363    92.552
  Ns(A, kn = a.kn)4    80.438  28.462   227.328    61.579  22.290   170.116
  Ns(A, kn = a.kn)5    84.898  43.483   165.757    53.995  28.421   102.580
  Ns(A, kn = a.kn)6  2807.086 314.078 25088.467  3044.560 342.099 27095.540
  Ns(A, kn = a.kn)7    94.204  65.378   135.739    53.821  37.998    76.232
  Ns(P, kn = p.kn)1     0.870   0.843     0.899     0.895   0.864     0.927
  Ns(P, kn = p.kn)2     0.918   0.899     0.938     0.945   0.923     0.968
  factor(sC)2           0.909   0.868     0.951     0.935   0.892     0.981
  factor(sC)3           0.834   0.796     0.874     0.846   0.805     0.888
  factor(sC)4           0.794   0.758     0.833     0.826   0.786     0.868
  factor(sC)5           0.774   0.738     0.812     0.797   0.757     0.839
  factor(sC)6           0.723   0.688     0.759     0.794   0.754     0.837
  factor(sC)7           0.672   0.639     0.706     0.741   0.702     0.782
  factor(sC)8           0.659   0.626     0.693     0.714   0.675     0.754
  factor(sC)9           0.644   0.612     0.679     0.712   0.672     0.754
  factor(sC)10          0.583   0.551     0.616     0.605   0.567     0.646
```

We can see a clear decline in RR relative to the most depraved areas (`sC` 1), but the effect is quite similar between man and women.

Again in parallel to the analyses of incidence rates we show how mortality rates among diabetes patients look as a function of age, so we set up a prediction data frame and use that for extraction of the rates.

```
> nd <- data.frame( A = 0:90,
+                   P = 2008,
+                  sC = 5,
+                Y.dm = 1000 )
> mmort2008 <- ci.pred( mm1, newdata = nd )
> fmort2008 <- ci.pred( mf1, newdata = nd )
```

Having collected the incidence rates separately for men and women we can plot the together:

```
> par( mar=c(3,4,1,1) )
> matplot( nd$A, cbind( mmort2008,
+                       fmort2008 ),
+          lwd=c(3,1,1), col=rep(c("blue","red"),each=3), lty=1, type="l",
+          log="y", xlab="Age (years)", ylab=" ",ylim=c(1,300))
> mtext( "Mortalty rate in DM patients (per 1000 PY)", side=2, line=2.5, las=0 )
```

4.5.1 Age by social class interaction

If we want to explore *if* there is an interaction between age and social class and how it looks we make an interaction term where the age-effect (*i.e.* the `differences` in age-effects), have fewer degrees of freedom than the overall age-effect we have modelled:

```
> r.kn <- seq(30,85,,4)
> mmi <- update( mm1, . ~ . + factor(sC):Ns(A,knots=r.kn) )
> mfi <- update( mf1, . ~ . + factor(sC):Ns(A,knots=r.kn) )
> summary( mfi )
  Call:
  glm(formula = D.dm ~ Ns(A, kn = a.kn) + Ns(P, kn = p.kn) + factor(sC) +
      factor(sC):Ns(A, knots = r.kn), family = poisson, data = subset(Mana,
      sex == "F"), offset = log(Y.dm))

  Deviance Residuals:
      Min       1Q   Median       3Q      Max
  -3.1777  -0.5673  -0.2126   0.1039   4.2243

  Coefficients: (1 not defined because of singularities)
                                    Estimate Std. Error z value Pr(>|z|)
  (Intercept)                        32.36130   27.08453   1.195  0.23216
  Ns(A, kn = a.kn)1                 -52.10774   36.78089  -1.417  0.15657
  Ns(A, kn = a.kn)2                 -73.80170   52.01071  -1.419  0.15591
  Ns(A, kn = a.kn)3                 -84.29271   59.53992  -1.416  0.15685
  Ns(A, kn = a.kn)4                 -84.96540   60.54724  -1.403  0.16053
  Ns(A, kn = a.kn)5                 -66.58256   47.99465  -1.387  0.16535
  Ns(A, kn = a.kn)6                 -79.73230   59.61948  -1.337  0.18111
  Ns(A, kn = a.kn)7                 -42.04601   31.48564  -1.335  0.18174
  Ns(P, kn = p.kn)1                  -0.11002    0.01776  -6.196 5.80e-10
  Ns(P, kn = p.kn)2                  -0.05577    0.01201  -4.643 3.44e-06
  factor(sC)2                         0.12618    0.23567   0.535  0.59235
  factor(sC)3                        -0.14141    0.25907  -0.546  0.58518
  factor(sC)4                         0.15131    0.23578   0.642  0.52105
  factor(sC)5                        -0.30547    0.27607  -1.106  0.26852
  factor(sC)6                        -0.61476    0.32298  -1.903  0.05698
  factor(sC)7                        -0.28495    0.28174  -1.011  0.31183
  factor(sC)8                        -0.61139    0.32957  -1.855  0.06358
  factor(sC)9                        -1.47917    0.49632  -2.980  0.00288
  factor(sC)10                       -0.65066    0.34094  -1.908  0.05634
  factor(sC)1:Ns(A, knots = r.kn)1   46.49753   32.73194   1.421  0.15545
  factor(sC)2:Ns(A, knots = r.kn)1   46.30997   32.73117   1.415  0.15711
  factor(sC)3:Ns(A, knots = r.kn)1   46.48183   32.73094   1.420  0.15557
  factor(sC)4:Ns(A, knots = r.kn)1   46.21074   32.73113   1.412  0.15800
  factor(sC)5:Ns(A, knots = r.kn)1   46.58672   32.73131   1.423  0.15465
  factor(sC)6:Ns(A, knots = r.kn)1   46.75670   32.73089   1.429  0.15314
  factor(sC)7:Ns(A, knots = r.kn)1   46.46471   32.73082   1.420  0.15572
  factor(sC)8:Ns(A, knots = r.kn)1   46.59601   32.72976   1.424  0.15455
  factor(sC)9:Ns(A, knots = r.kn)1   47.17864   32.73138   1.441  0.14947
  factor(sC)10:Ns(A, knots = r.kn)1  46.17024   32.72908   1.411  0.15834
  factor(sC)1:Ns(A, knots = r.kn)2   63.91691   43.39158   1.473  0.14074
  factor(sC)2:Ns(A, knots = r.kn)2   63.46674   43.39044   1.463  0.14355
  factor(sC)3:Ns(A, knots = r.kn)2   63.81264   43.38988   1.471  0.14138
```

```
  factor(sC)4:Ns(A, knots = r.kn)2   63.15402   43.39219   1.455  0.14555
  factor(sC)5:Ns(A, knots = r.kn)2   63.84081   43.39384   1.471  0.14124
  factor(sC)6:Ns(A, knots = r.kn)2   64.45804   43.39234   1.485  0.13742
  factor(sC)7:Ns(A, knots = r.kn)2   63.47334   43.39187   1.463  0.14352
  factor(sC)8:Ns(A, knots = r.kn)2   64.03745   43.39119   1.476  0.13999
  factor(sC)9:Ns(A, knots = r.kn)2   65.55159   43.39864   1.510  0.13093
  factor(sC)10:Ns(A, knots = r.kn)2  63.81923   43.39300   1.471  0.14137
  factor(sC)1:Ns(A, knots = r.kn)3   -0.59881    0.19802  -3.024  0.00249
  factor(sC)2:Ns(A, knots = r.kn)3   -0.53802    0.19908  -2.702  0.00688
  factor(sC)3:Ns(A, knots = r.kn)3   -0.52286    0.20440  -2.558  0.01053
  factor(sC)4:Ns(A, knots = r.kn)3   -0.56488    0.20263  -2.788  0.00531
  factor(sC)5:Ns(A, knots = r.kn)3   -0.30275    0.21220  -1.427  0.15366
  factor(sC)6:Ns(A, knots = r.kn)3   -0.19415    0.22103  -0.878  0.37974
  factor(sC)7:Ns(A, knots = r.kn)3   -0.13106    0.22080  -0.594  0.55280
  factor(sC)8:Ns(A, knots = r.kn)3   -0.01746    0.23296  -0.075  0.94024
  factor(sC)9:Ns(A, knots = r.kn)3    0.47660    0.27178   1.754  0.07949
  factor(sC)10:Ns(A, knots = r.kn)3        NA         NA      NA       NA

  (Dispersion parameter for poisson family taken to be 1)

      Null deviance: 25466.5  on 7050  degrees of freedom
  Residual deviance:  4821.1  on 7003  degrees of freedom
  AIC: 16444

  Number of Fisher Scoring iterations: 8
```

Note that the last parameter is `NA`; this is because it is *aliased* — the natural spline basis `Ns(A,knots=r.kn)` includes a *linear* term in `A`, which is also included in the original spline term for `A`, and hence only is estimable for 9 out of the 10 social class strata. This will give a warning when we do prediction, but this type of aliasing will give the correct predictions anyway.

But we just take a look at the formal significance of the interaction, we see that it is massive:

```
> anova( mmi, mm1, test="Chisq" )
  Analysis of Deviance Table

  Model 1: D.dm ~ Ns(A, kn = a.kn) + Ns(P, kn = p.kn) + factor(sC) + factor(sC):Ns(A,
      knots = r.kn)
  Model 2: D.dm ~ Ns(A, kn = a.kn) + Ns(P, kn = p.kn) + factor(sC)
    Resid. Df Resid. Dev  Df Deviance  Pr(>Chi)
  1      7019     5098.6
  2      7048     5276.4 -29  -177.79 < 2.2e-16
> anova( mfi, mf1, test="Chisq" )
  Analysis of Deviance Table

  Model 1: D.dm ~ Ns(A, kn = a.kn) + Ns(P, kn = p.kn) + factor(sC) + factor(sC):Ns(A,
      knots = r.kn)
  Model 2: D.dm ~ Ns(A, kn = a.kn) + Ns(P, kn = p.kn) + factor(sC)
    Resid. Df Resid. Dev  Df Deviance  Pr(>Chi)
  1      7003     4821.1
  2      7032     5000.5 -29  -179.41 < 2.2e-16
```

However the main interest is in the *shape* of the interactions, so we predict the incidence rates separately for each sex and social class and plot the. To this end we first set up a 3-dimensional array to hold the predictions:

```
> mi <- NArray( list( A = nd$A,
+                     sex = c("M","F"),
+                      sC = 1:10 ) )
> str( mi )
```

```
 logi [1:91, 1:2, 1:10] NA NA NA NA NA NA ...
 - attr(*, "dimnames")=List of 3
  ..$ A  : chr [1:91] "0" "1" "2" "3" ...
  ..$ sex: chr [1:2] "M" "F"
  ..$ sC : chr [1:10] "1" "2" "3" "4" ...
```

With this in place we can fill in the array:

```
> for( sc in 1:10 )
+    {
+  mi[,"M",sc] <- ci.pred( mmi, newdata = transform( nd, sC=sc ) )[,1]
+  mi[,"F",sc] <- ci.pred( mfi, newdata = transform( nd, sC=sc ) )[,1]
+    }
```

Then we can plot the estimated incidence rates in different strata separately for men and women:

```
> par( mfrow=c(1,2), mar=c(3,1,1,1), oma=c(0,4,0,0), mgp=c(3,1,0)/1.6,
+      las=1, bty="n" )
> matplot( nd$A, mi[,"M",],
+          lwd=2:3, col=gray(1:10/14), lty=1, type="l",
+          log="y", xlab="Age (years)", ylab=" ", ylim=c(1/100,250) )
> matplot( nd$A, mi[,"F",],
+          lwd=2:3, col=gray(1:10/14), lty=1, type="l",
+          log="y", xlab="Age (years)", ylab=" ", ylim=c(1/100,250) )
> mtext( "Mortality rate among DM pateints (per 1000 PY)",
+        side=2, line=2.5, las=0, outer=TRUE )
```

From figure 4.5 we see that mortality rates are certainly not proportional, but the social class gradient has the same `direction` across the age-span, albeit not the same `size`; essentially the mortality rates are converging (in relative terms) by age.

```
> par( mfrow=c(1,2), mar=c(3,1,1,1), oma=c(0,4,0,0), mgp=c(3,1,0)/1.6,
+      las=1, bty="n" )
> matplot( nd$A, mi[,"M",],
+          lwd=2:3, col=gray(1:10/14), lty=1, type="l",
+          xlab="Age (years)", ylab=" ", ylim=c(0,250) )
> matplot( nd$A, mi[,"F",],
+          lwd=2:3, col=gray(1:10/14), lty=1, type="l",
+          xlab="Age (years)", ylab=" ", ylim=c(0,250) )
> mtext( "Mortality rate among DM pateints (per 1000 PY)",
+        side=2, line=2.5, las=0, outer=TRUE )
```

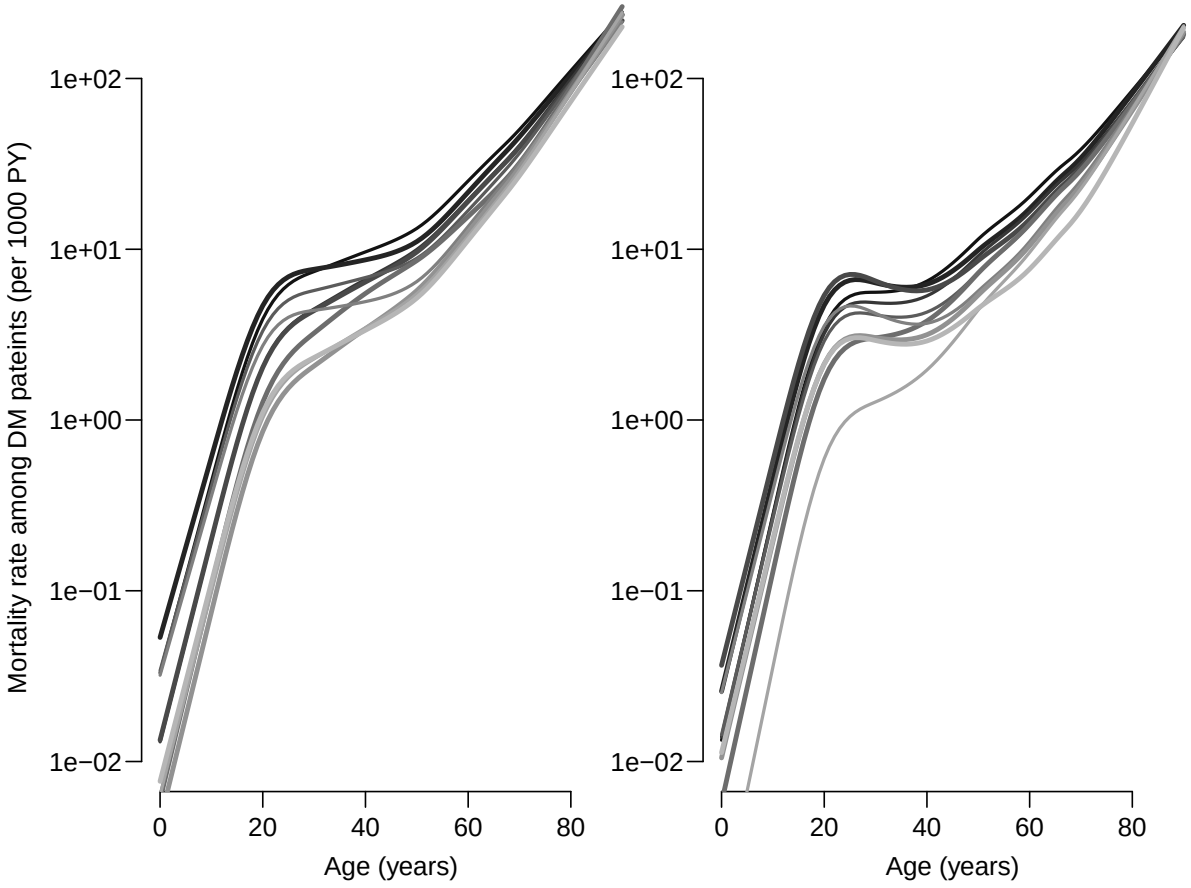

Figure 4.5: *Predicted mortality rates among DM patients in Scotland for social classes 1–10 (light to dark).*

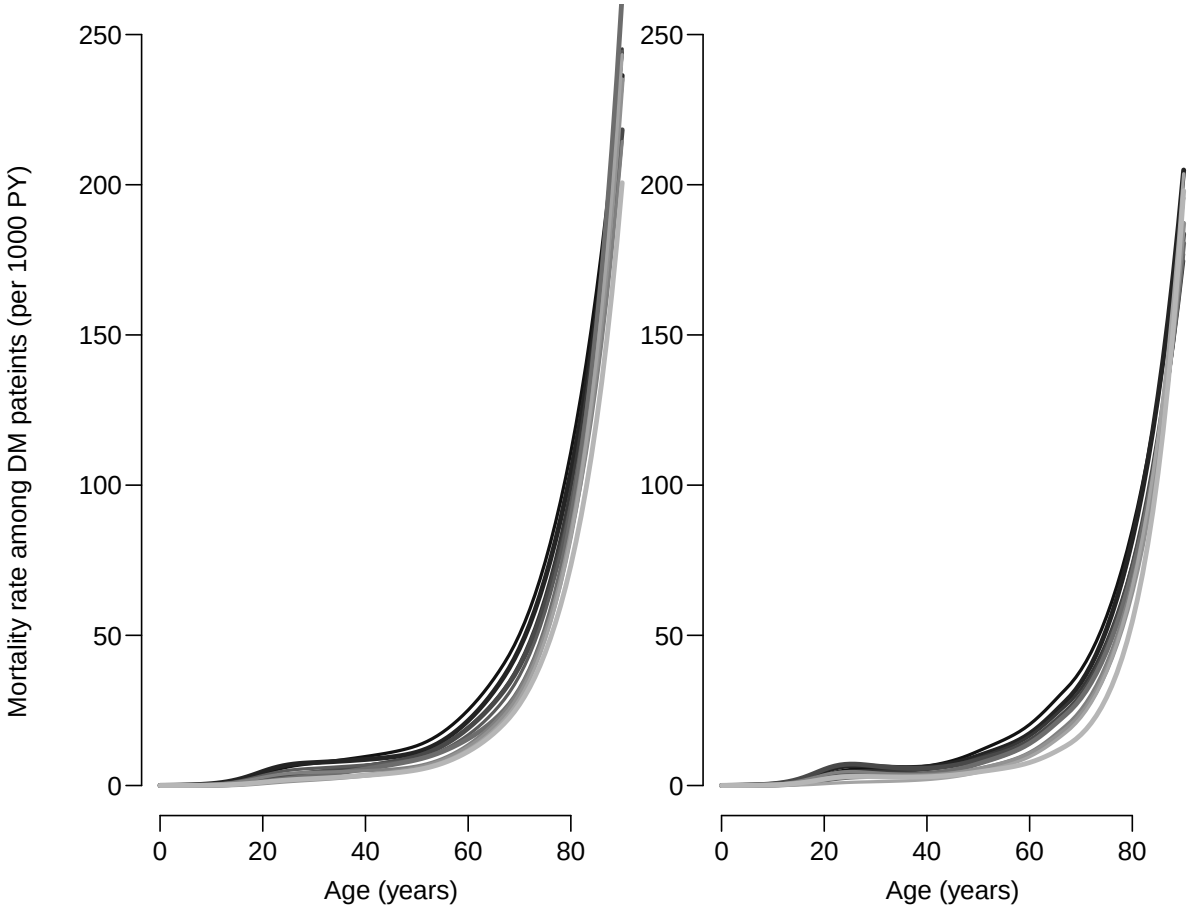

Figure 4.6: *Predicted mortality rates among DM patients in Scotland for social classes 1–10 (light to dark).*