Modern Demographic Methods in Epidemiology with R Practical exercises

University of Melbourne
23 November 2015
http://BendixCarstensen/AdvCoh/courses/Melb-2015
Version 2.1

Compiled Thursday 19th November, 2015, 02:39
from: `/home/bendix/teach/AdvCoh/courses/Melb.2015/pracs/pracs.tex`

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

Contents

Course program

The course is centered around practical calculations in R, illustrating the concepts through analysis of data. All sessions will be alternating between lectures and practicals, most followed by a walk-through of the computing issues.

Please note the details of the computing requirements on the course web-site, http://bendixcarstensen.com/AdvCoh/Melb-2015/, including download of datasets and programs for the practicals.

Monday 23 November 2015

08:45 – 09:00	Arrival & introduction
09:00 – 10:00	Brief introduction to R.
	Rates and survival.
	P: Computing rates, RRs and RDs (2.1, 2.2)
10:00 – 10:40	Representation of follow-up data.
	P: Lexis diagrams and Lexis objects (2.3).
10:40 – 11:00	**Coffee**
11:00 – 12:45	Kaplan-Meier, Cox and Lexis.
	P: Fitting a Cox model and a Poisson model and comparing (2.4).
12:45 – 13:30	**Lunch**
13:30 – 15:00	Estimating — and drawing — a smooth curve.
	Multiple time scales.
	P: Estimating a curved effect: Testis cancer in DK (2.5).
15:00 – 15:30	**Afternoon Tea**
15:30 – 17:30	Life expectancy and life lost to disease.
	Multistate models.
	P: Modeling rates and computing life lost (2.6).
	P: Lifetime risk of diabetes (to appear).
17:30 – 18:00	Summary of the day.

Chapter 1

Fundamental relations in demography and survival analysis

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, mortality/morbidity rate or simply "rate".

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

Relationships between terms:

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

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

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

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

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

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

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

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 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

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

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 and that the mortality rate from this cause was the same as when the other causes were present too.

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

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

Subdistribution hazard Fine and Gray defined models for the so-called subdistribution hazard. 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 risk. 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:

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

where f is the density of the distribution of lifetimes.

The relation between the density f and the survival function S is $f(a) = -S'(a)$, 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.

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

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):

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

Chapter 2

Practical exercises

2.1 Life table

Fill in the empty columns in the life-table calculations for Stage II cervix cancer patients

Year (t)	N	D	L	N$-\frac{1}{2}$L	P(F)	P(S)	S(t)
1	234	24	3				
2	207	27	11				
3	169	31	9				
4	129	17	7				
5	105	7	13				
6	85	6	6				
7	73	5	6				
8	62	3	10				
9	49	2	13				
10	34	4	6				

2.2 Calculation of rates, RR and RD

This exercise is `very` prescriptive, so you should make an effort to really understand everything you type into R.

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. Now try to achieve this estimate and c.i. using a Poisson model. Use the number of events as the response 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` or `ci.exp` functions from `Epi`:

   ```
   > ci.lin( mm )
   > ci.exp( mm )
   ```

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.exp( mmx )
   ```

   Verify that this give the same results as above.

5. The advantage of this latter approach is that it will also make sense to use an identity link — the response is the same but the parameter 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` or `ci.exp` to produce the estimate and the confidence intervals from this model:

   ```
   > ci.lin( ma )
   > ci.exp( ma, Exp=FALSE )
   ```

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 whether 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`.

What is the deviance good for?

9. 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:

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

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

10. 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?

You can extract the exponentiated parameters by:

```
> ci.exp( mm )
```

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

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

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()
```

12. 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.exp( ma, Exp=FALSE )
```

13. Normally one would like to get both the rates and the ratio between them. This can be achieved in one go using the `ctr.mat` argument to `ci.exp`. 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.exp( mm, ctr.mat=CM )
> round( ci.exp( mm, ctr.mat=CM ) )
```

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

15. 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.exp( ma, ctr.mat=CM, Exp=FALSE )
```

0

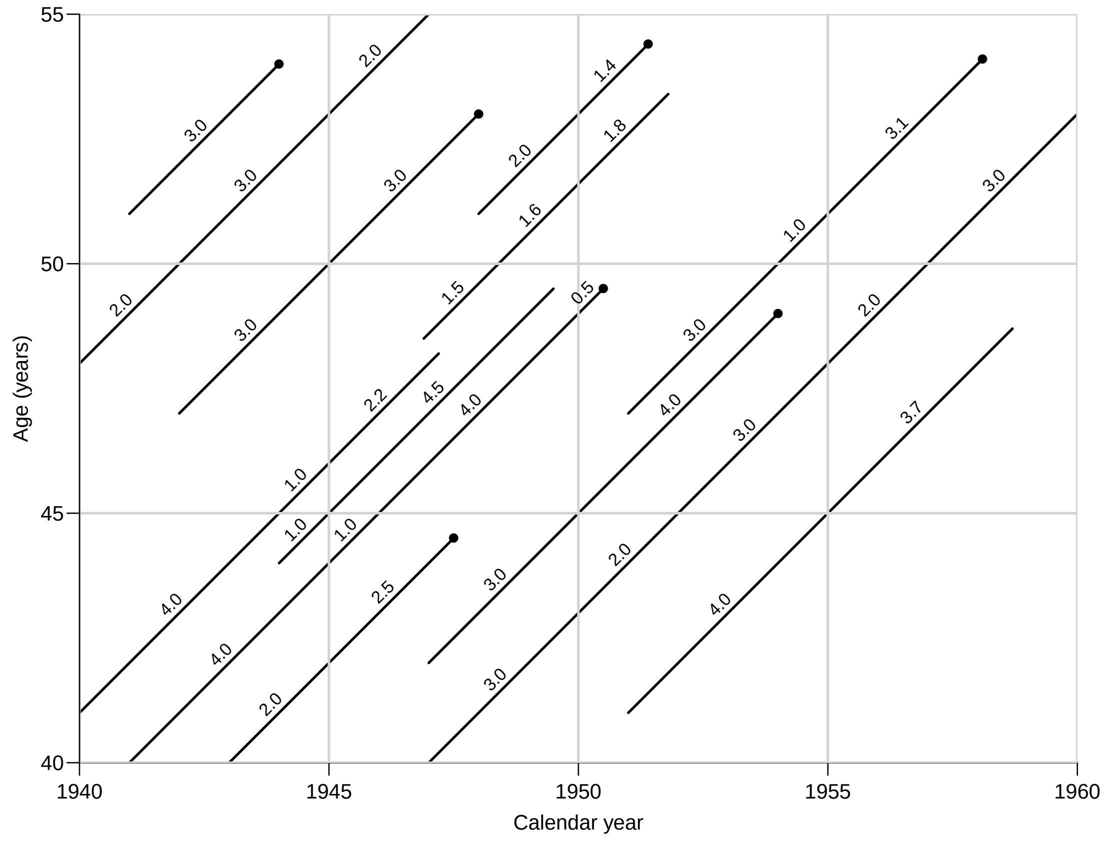

Figure 2.1: *Lexis diagram of a small occupational cohort.*

2.3 Lexis diagram

In the Lexis diagram below is shown follow-up times of a small occupational cohort over the years 1940–1959 and the age range 40–54 years (this example is from **B&D**). Each line runs from the entry to follow-up until either the diagnosis of cancer (●), or censoring or withdrawal (no symbol) due to death from other causes or migration.

1. Calculate the number of new cases of cancer, and person-years at risk in all the three 5-year age bands: 40–44, 45–49, and 50–54 years for each of the 5-year calendar periods 1940–44, 1945–49, and 1950–54 separately.

   Hint: The data set is available as an example dataset, `occup`, in the `Epi` package. Try:

   ```
   > library( Epi )
   > data( occup )
   > str( occup )
   > occup
   > example( occup )
   ```

2. Calculate the numbers of new cases of cancer, person-years at risk in the three 5-year age groups: 40–44, 45–49, and 50–54 years for a *birth cohort* born in 1902–11. *Hint:* You can use the function `splitLexis` to subdivide follow-up in age- and calendar time bands.

3. Continuing from 2, estimate the cumulative rate and the cumulative risk over the whole 15-year age range for the chosen birth cohort.

4. Now suppose the age-specific incidences (per 100,000 person-years) in the three 5-year age-groups during 1940–60 in the whole population of the country were 100, 200, and 400, respectively, so there was no variation between the sub-periods. Assuming that this is an appropriate reference population, calculate the expected number of cases for the index occupational cohort for the same period. Compare the observed and expected number of cases by standardized incidence ratio, SIR.

   Comment on the result.

2.4 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.

2.4.1 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:

```
> # If you use windows thins might be a good idea;
> # 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?

2.4.2 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) )
```

2.4.3 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") )
    ```

2.4.4 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?

2.5 Estimation and reporting of linear and curved effects

The purpose of this exercise is to take you through models with curved effects of age and calendar time, in order to show you how to report 1) a curved effect of a main effect such as age and 2) a curved effect of a *relative* effect where a reference point is needed. In the exercise we will use the `testisDK` data from the `Epi` package, which contains the number of cases of testis cancer in Denmark 1943–96:

1. First load the Danish testis cancer data, and inspect the dataset:

    ```
    library( Epi )
    sessionInfo()
    data( testisDK )
    str( testisDK )
    head( testisDK )
    ```

    Tabulate both events and person-years in say 10-year age-groups and 10-year periods of follow-up. Use for example `xtabs`. In which ages are the age-specific testis cancer rates highest?

2. Now fit a Poisson-model for the mortality rates with a linear term for age at follow-up (current age, attained age):

    ```
    ml <- glm( D ~ A, offset=log(Y), family=poisson, data=testisDK )
    ci.exp( ml )
    ```

    What do the parameters mean?

3. Work out the the predicted log-mortality rates for ages 25 to 45, say, by doing a hand-calculation based on the coefficients:

    ```
    ( cf <- coef( ml ) )
    ```

4. However, we do not have the standard errors of these mortality rates, and hence neither the confidence intervals. This is implemented in `ci.exp`; if we provide the argument `ctr.mat=` as a matrix where each row corresponds to a prediction point and each column to a parameter from the model. Look at the help page for `ci.exp` and then try:

```
( CM <- cbind( 1, 25:45 ) )
round( ci.exp( ml, ctr.mat=CM )*10^5, 3 )
```

5. Use this machinery to derive and plot the mortality rates over the range from 15 to 65 years, say:

```
C1 <- cbind( 1, 15:65 )
matplot( 15:65, ci.exp( ml, ctr.mat=C1 )*10^5,
         log="y", xlab="Age", ylab="Testis cancer incidence rate per 100,000 PY",
         type="l", lty=1, lwd=c(3,1,1), col="black" )
```

6. Now check if the mortality rates really are exponentially increasing by age (that is linearly on the log-scale), by adding a quadratic term to the model. Note that you must use the expression `I(A^2)` in the modeling in order to avoid that the "^" is interpreted as part of the model formula:

```
mq <- glm( D ~ A + I(A^2), offset=log(Y), family=poisson, data=testisDK )
ci.exp( mq, Exp=F )
```

   Then plot the estimated rates under the quadratic model.

```
aa <- 15:65
C2 <- cbind( 1, aa, aa^2 )
matplot( aa, ci.exp( mq, ctr.mat=C2 )*10^5,
         log="y", xlab="Age", ylab="Testis cancer incidence rate per 100,000 PY",
         type="l", lty=1, lwd=c(3,1,1), col="black" )
```

   Try to overlay the estimated rates from the model with linear effect of age — you will need the function `matlines`.

7. Repeat the same using a 3rd degree polynomial.

8. Instead of continuing with higher powers of age we could use fractions of powers, or we could use splines, piece wise polynomial curves, that fit nicely together at join points (knots). This is implemented in the `splines` package, in the function `ns`, which returns a matrix. There is a wrapper `Ns` in the `Epi`-package that automatically designate the smallest and largest knots as *boundary knots*, beyond which the resulting curve is linear:

```
library( splines )
ms <- glm( D ~ Ns(A,knots=seq(15,65,10)), offset=log(Y),
           family=poisson, data=testisDK )
```

   In order to extract the estimated effects, construct a contrast matrix that correspond to the parameters of the model, you can try the following (you can skip some of the fancy stuff in the `plot` statement).

```
As <- Ns( aa, knots=seq(15,65,10) )
matplot( aa, ci.exp( ms, ctr.mat=cbind(1,As) )*10^5, log="y",
         type="l", lty=1, lwd=c(3,1,1), col="black",
         xlab="Age", ylab="Testis cancer incidence rate per 100,000 PY" )
```

9. Now add a linear term in calendar time `P` to the model, and make a prediction of the incidence rates in 1970. You would need to take a look at the parameter of the model in order to devise the contrast matrix:

```
msp <- glm( D ~ Ns(A,knots=seq(15,65,10)) + P, offset=log(Y), family=poisson, data=testisDK )
ci.exp( msp )
matplot( aa, ci.exp( msp, ctr.mat=cbind(1,As,1970) )*10^5,
         log="y", xlab="Age", ylab="Testis cancer incidence rate per 100,000 PY",
         type="l", lty=1, lwd=c(3,1,1), col="black" )
```

   Note that `cbind` automatically will expand the 1 and the 1970 to match the number of rows of `As`.

10. Extract the RR relative to 1970, by using the `subset` argument to `ci.exp`:

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

   What is the annual relative increase in the testis cancer incidence rates? Show the RR of testis cancer by year relative to 1970 by multiplying the log-RR for period with the distance form 1970, such as:

```
yy  <- 1943:1996
Cp1 <- cbind( yy - 1970 )
matplot( yy, ci.exp( msp, ctr.mat=Cp1, subset="P" ),
         log="y", xlab="Date", ylab="RR of Testis cancer",
         type="l", lty=1, lwd=c(3,1,1), col="black" )
abline( h=1 )
```

11. Try to add a quadratic term to the period effect, and plot the resulting RR relative to 1970.
    Hint: In order to extract the quadratic effects relative to 1970, you must form the matrix of linear and quadratic period, and a corresponding matrix where all rows are identical to the 1970 row:

```
msp <- glm( D ~ Ns(A,knots=seq(15,65,10)) + P + I(P^2),
                offset=log(Y), family=poisson, data=testisDK )
Cq <- cbind( yy, yy^2 ) - cbind( rep(1970,length(yy)), 1970^2 )
```

   Use this matrix as argument to `ci.exp`

12. Now investigate if there is any non-linearity in period beyond the quadratic, by fitting fit a spline for (`P`) as well, and comparing the models. Plot the resulting RR by year, relative to 1970 too. You must define a contrast matrix corresponding to the years where the prediction is made, as well as a matrix with the same number of rows, but with all rows identical to the one corresponding to the reference year. You must use the difference of these two as the argument to `ctr.mat` in `ci.exp`.

13. Plot the estimated age-specific rates in 1970 from this model. Note that you need a reference matrix for the period with all rows identical to the 1970 row, but this time with the same number of rows as the *age*-prediction points.

14. Collect these steps in a general outline, where you first define the knots, and the points of age and period prediction, and then fit the model and do the two plots.

15. Form a new variable in the data frame, `B=P-A`, the data of birth, and repeat the last analysis with this variable instead of `P`.

2.6 Diabetes in Denmark

This exercise is using data from the National Danish Diabetes register. Theer 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. The exercise is mainly about assessing how mortality depends age, and how to understand and compute years of life lost to diabetes by comparing with the population mortality.

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. 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.

4. Now estimate 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 ),
               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 modelling statement.

5. However, when we are working with event data, the `ns` machinery does not necessarity choose knots for splines sensibly, so it is better to explicitly allocate theknots so that the number of *events* is the same between knots; try:

```
( a.kn <- with( subset(SL,lex.Xst=="Dead"),
                quantile( A+lex.dur, (1:10-0.5)/10 ) ) )
```

   These are the locations of knots that places 10% of events between each succesive pair of knots, and 5% beyond the outer knots. If we use these as knots in the fuction `Ns` we automatically get the smallest and the largest as boundary knots, beyond which the splines are linear:

```
r.m <- glm( (lex.Xst=="Dead") ~ Ns( A, knots=a.kn ),
            offset = log( lex.dur ),
            family = poisson,
              data = subset( SL, sex=="M" ) )
r.f <- update( r.m, data = subset( SL, sex=="F" ) )
```

6. With these objects you can get the estimated log-rates by using `ci.pred`, 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 <- ci.pred( r.m, newdata = nd )
p.f <- ci.pred( r.f, newdata = nd )
str( p.m )
```

   Plot the two sets of estimated rates (men and women).
   (Hint: use `matplot`)

2.6.1 Comparison with the population rates

7. 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…

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

```
R.m <- glm( D ~ Ns( A, knots=seq(10,90,10) ),
            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 a new dataset for prediction, because in this original dataset the person-years are called `Y` in the dataset with the patient follow-up person-years were in a variable called `lex.dur`. Add the curves with predicted rates to the plot of the patient mortality rates.

2.6.2 Life expectancy

9. Recall from the section of fundamental concepts that the expected lifetime is the area under the survival curve, and remember the relationship between the mortality rates and the survival curve:

$$S(a) = \exp\left(-\Lambda(a)\right) = \exp\left(-\int_o^a \lambda(s)\,\mathrm{d}s\right)$$

The $\lambda(s)$ is the smooth function of age we just estimated in the models for the diabetes population and for the general population. Now, an integral is merely a sum; we can compute it by approximating the area under the curve with a histogram with very narrow intervals. Now, compute λ (by `ci.pred`) at the middle of, say, 1000 intervals between 0 and 100 years, multiply each value by the width of the interval, and compute the cumulative integral Λ:

```
mid.pt <- 0:999/10 + 1/20
mid.pt[1:5]
nd <- data.frame( A = mid.pt, Y = 1 )
```

Note that we devise a dataframe `nd` where the person-years is 1, so that we get the predicted rates in the units of "events per year". We use `cumsum`, and then the exponential to get the survival curves:

```
S.m <- exp( -cumsum( ci.pred(R.m,newdata=nd)[,1]*1/10 ) )
S.f <- exp( -cumsum( ci.pred(R.f,newdata=nd)[,1]*1/10 ) )
```

Note that we have multiplied the estimated rate (calculated in units of events per 1 year) by the interval length, 1/10 year.

10. Plot the survival curves for men and women

11. Compute the expected lifetime as the area under thse curves; recall that we have the survival curve evaluated at points 0.1, 0.2, ... 99.9, 100 years. So if we take the sum of these values of the survival function and multiply by 0.1, we get the area under the curve. What is is the expected lifetime (at birth) of men and women, respectively?

12. Make the same calculations for diabetes patients — remember that you must use `lex.dur` rather than the variable `Y` in the prediction data frame.

    What is the expected lifetime of diabetes pateinst at birth? Is this relly a meningful quantity?

2.6.3 Life lost to diabetes

13. What is the difference in life expectancy between diabetes patients and the general population?

14. Are these numbers sensible? What scenario are they referring to?

15. Instead take a look at the life-expectancy of persons, say, 50 years old. To this end we need the *conditional* survival curves *given* that aperson is alive at age 50. But these are just the survival curves from age 50, *divided* the probability of surviving till 50. Compute this and the expected residual life time from age 50 for the genral population and for diabets patients.

16. Make this calculation general by wrapping it in an R-function that takes the age as argiment and returns the years of life lost calculated at that age. Compute the years of life lost to diabetes at ages 40–80 and plot these for men and women as functions of age.

2.6.4 Changes in life lost to diabetes

17. The previous calculations just used the crude age-specific mortality rates for the entire period 1995–2009 (incl.). Expand the models for the mortality rates for DM patients and the population with a term of calendar time. A first approximation could be just a linear effect of calendar time.

18. How much is the average change in mortality among diabetes patients and in the general population?

19. Predict the mortality rates, compute survival function, expected residual life and years of life lost to diabetes, using mortality rates from 1995, 2000, 2005 and 2010. To this end you must predict age-specific mortality rates for each of the dates 1.1.1995 etc. and make the previous calculations for each. You may want to store rates in an array, see the help page for the function `NArray`.

20. Plot a curve of life lost to diabetes for each date as a function of age. What are the assumptions behind these curves?

21. (Very long-winded) How could we go about including duration of diabetes in the mortality models, and how would you report the results

2.7 Practical reporting of multistate models

This exercise is merely intended as a walk-through of the facilities associated with the function `simLexis`.

Therefore, take a copy of the example code on the help-page for `simLexis`, and walk your way through it. There is a slightly more elaborate account of this example in a so-called vignette that comes with the Epi package: If you start the HTML-help by typing `help.start()` go to `Packages` and finding the `Epi` package. Under `User guides, package vignettes and other documentation` you will find a document under the link

`Simulation in multistate models with multiple timescales`, that explains this example in more depth.

Chapter 3

Solutions to exercises:

This section contains solutions to the exercises generated by including all the output from the suggested R-commnds (and a few more) from the corresponding `.Rout`-file generated from running the R-code. At the end of each section is shown the graphs generated by the programs.

3.1 Life table

The filled-in life table looks like this:

Year (t)	N	D	L	N$-\frac{1}{2}$L	P(F)	P(S)	S(t)
1	234	24	3	232.5	0.10323	0.89677	0.89677
2	207	27	11	201.5	0.13400	0.86600	0.77661
3	169	31	9	164.5	0.18845	0.81155	0.63026
4	129	17	7	125.5	0.13546	0.86454	0.54489
5	105	7	13	98.5	0.07107	0.92893	0.50616
6	85	6	6	82.0	0.07317	0.92683	0.46913
7	73	5	6	70.0	0.07143	0.92857	0.43562
8	62	3	10	57.0	0.05263	0.94737	0.41269
9	49	2	13	42.5	0.04706	0.95294	0.39327
10	34	4	6	31.0	0.12903	0.87097	0.34252

The calculations can be done quite simply in R by putting the three columns into vectors, and just plugging data into the formulae:

```
> N <- c(234, 207, 169, 129, 105,  85,  73,  62,  49, 34)
> D <- c(24, 27, 31, 17, 7, 6, 5, 3, 2, 4)
> L <- c( 3, 11, 9, 7,13, 6, 6, 10, 13, 6)
> res <- cbind( N, D, L, N-L/2,
+                     D/(N-L/2),
+                   1-D/(N-L/2),
+           cumprod(1-D/(N-L/2)) )
> colnames(res)[4:7] <- c("eff.N","mort","surv","Surv")
> round( res, 5 )
        N  D  L eff.N    mort    surv    Surv
 [1,] 234 24  3 232.5 0.10323 0.89677 0.89677
 [2,] 207 27 11 201.5 0.13400 0.86600 0.77661
 [3,] 169 31  9 164.5 0.18845 0.81155 0.63026
 [4,] 129 17  7 125.5 0.13546 0.86454 0.54489
 [5,] 105  7 13  98.5 0.07107 0.92893 0.50616
 [6,]  85  6  6  82.0 0.07317 0.92683 0.46913
 [7,]  73  5  6  70.0 0.07143 0.92857 0.43562
 [8,]  62  3 10  57.0 0.05263 0.94737 0.41269
 [9,]  49  2 13  42.5 0.04706 0.95294 0.39327
[10,]  34  4  6  31.0 0.12903 0.87097 0.34252
```

Also, assigning the survival probabilities to a vector, `S`, enables a simple plot of the estimated survival function in figure **??**; note we plot times 0 to 10, the first point corresponding to a survival probability of 1 at time 0 added separately:

```
> S <- cumprod(1-D/(N-L/2))
> plot( 0:10, c(1,S), pch=16, type="b",
+       ylim=0:1, yaxs="i",
+       ylab="Survival", xlab="Time since diagnosis" )
```

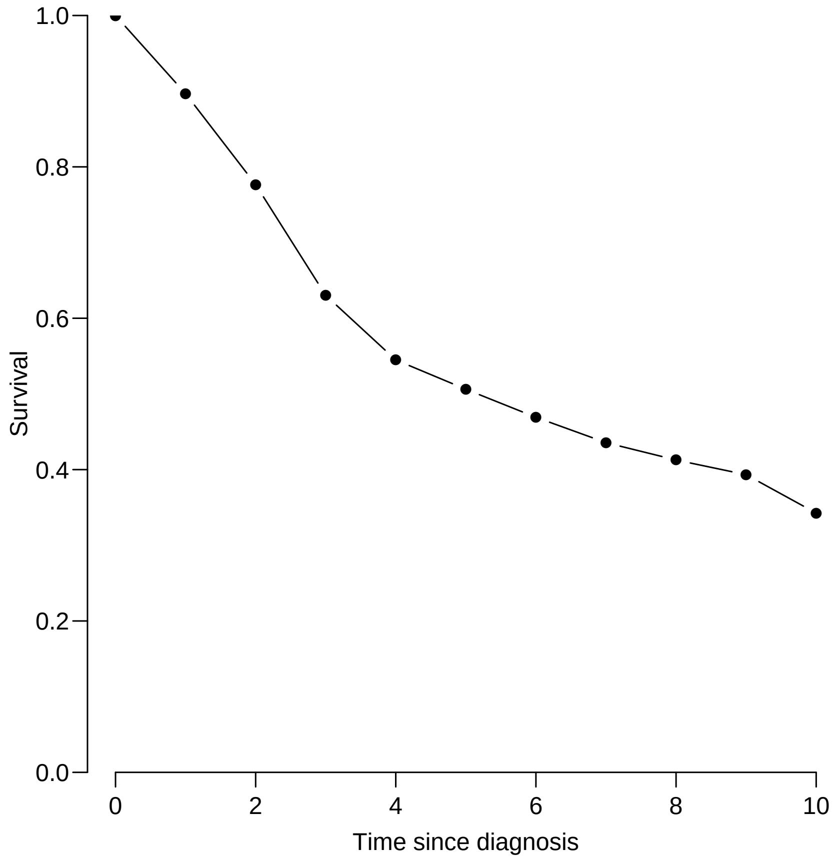

Figure 3.1: *The estimated survival function — liftable estimator.*

3.2 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 we use R as a simple desk calculator to derive the rate and a confidence interval (note that you can stick several R-commands on one line if you separate them by ";"):

   ```
   > 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
   ```

   The function `ci.mat()` returns a 2 by 3 matrix, 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. Now we use a Poisson model to estimate a rate and its confidence interval. We use the number of events as the response 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
   ```

The default link used is the log link, and we are using the log-person-years as offset, so the model is:

$$\log\big(\mathrm{E}(D)\big) = \mu + \log(Y) \quad \Leftrightarrow \quad \log\big(\mathrm{E}(D)/Y\big) = \mu$$

The parameter μ in this model is therefore the log of the rate.

3. A confidence interval can be extracted directly from the model with the `ci.lin()` or `ci.exp()`-function from `Epi`; note that the `Exp=TRUE` argument will transform the estimate and the confidence interval to the rate-scale — normally we would only want this, and so subset the output from `ci.lin`

```
> ci.lin( mm )
```

```
             Estimate    StdErr         z             P      2.5%     97.5%
(Intercept) -5.910254 0.2581989 -22.89032 5.801722e-116 -6.416315 -5.404194
```

```
> ci.exp( mm )
```

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

```
> round( ci.exp( mm ), 5 )
```

```
            exp(Est.)    2.5%  97.5%
(Intercept)   0.00271 0.00163 0.0045
```

4. The 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 )
> round( ci.exp( mmx ), 5 )
```

```
            exp(Est.)    2.5%  97.5%
(Intercept)   0.00271 0.00163 0.0045
```

We see that this gives 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 parameter estimated is now the rate, not the log-rate:

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

The intercept in this model is now the rate itself, because of the identity link.

We see that we get the same estimate as before:

```
> log( coef(ma) )
```

```
(Intercept)
  -5.910254
```

6. We can then use `ci.lin` (or `ci.exp` with argument `Exp=FALSE`) to produce the estimate and the confidence intervals from this model:

```
> ci.lin( ma )
```

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

```
> ci.exp( ma, Exp=FALSE )
```

```
               Estimate        2.5%       97.5%
(Intercept) 0.002711497 0.001339315 0.004083678
```

```
> round( ci.exp( ma, Exp=FALSE ), 5 )
```

```
            Estimate    2.5%   97.5%
(Intercept)  0.00271 0.00134 0.00408
```

The confidence limits from this model are based on the 2nd derivative of the log-likelihood with respect to the *rate*, and not as before with respect to the *log rate*, and therefor they are different — they are symmetrical on the rate-scale and not on the log-rate scale:

$$\ell(\lambda) = D\ln(\lambda) - \lambda Y \qquad \ell'(\lambda) = D/\lambda - Y \qquad \ell''(\lambda) = -D/\lambda^2\big|_{\lambda=D/Y} = -Y^2/D$$

Thus the observed information is Y^2/D and hence the approximate standard deviation of the rate is square root of the inverse of this, $\sqrt{D}/Y$, which is exactly the standard deviation you got from the model:

```
> c( sqrt(D)/Y, ci.lin( ma )[,2] )
[1] 0.0007001054 0.0007001054
```

7. If we assume that the events and person years are collected over three time periods, which we for convenience number 1 to 3:

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

If we fit the same model as before to the data from the separate periods, we get the same estimates, because the Poisson log-likelihood for three independent observations with the same relationship between mean and person-years is identical to the likelihood for the sum of the observations with an exp-offset equal to the sum of the exp-offsets:

$$\sum_i \big(D_i \log(\lambda) - \lambda Y_i\big) = \big(\sum_i D_i\big)\log(\lambda) - \lambda\big(\sum_i Y_i\big)$$

— basically this is a consequence of the fact that the likelihood for follow-up data with constant rate is additive *both* in the no. events *and* the person-time.

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

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

8. With separate observations from three periods we can test whether the rates are the same in the three periods; we just fit a model with the period as a factor:

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

and compare the two models via a log-likelihood ratio test 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
```

We note that the test statistic is the same as deviance of the model `mp`. This is because the deviance of a model is the log-likelihood ratio test statistic of the model versus the saturated model; i.e. the model with one parameter per observation, in this case the model `mp`.

9. 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:

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

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

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

If we have 15 events during 5532 person-years in an unexposed group and 28 events during 4783 person-years in an exposed group, we can then 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
```

10. But this can also be achieved 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 )
> summary( mm )
```

```
Call:
glm(formula = D ~ factor(xpos), family = poisson, offset = log(Y))

Deviance Residuals:
[1]  0  0

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    -5.9103     0.2582 -22.890   <2e-16
factor(xpos)1   0.7696     0.3200   2.405   0.0162

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 6.1110e+00  on 1  degrees of freedom
Residual deviance: 1.7764e-15  on 0  degrees of freedom
AIC: 13.733

Number of Fisher Scoring iterations: 3
```

The parameters in this model are:

`(Intercept)`: the log rate in the reference group, scaled to the units of `Y`

`factor(xpos)1`: the log RR between group 1 and 0.

We can extract the exponentiated parameters, corresponding to the rate and the rate-ratio by:

```
> ci.exp( mm )

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

11. If we instead wanted the rate-difference as a comparative measure, we just subtract the rates, and the variance of the difference is (since the rates are based on independent samples) just the sum of the variances:

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

When we use this formula to compute the rate difference and a 95% confidence interval for it we get:

```
> 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
```

12. This is also the confidence interval we get when you fit an additive model with exposure as factor. Note that since the model is a model where the rates and rate differences are parameters, we shall *not* us the `Exp=TRUE` argument to `ci.lin`:

```
> ma <- glm( D/Y ~ factor(xpos), weight=Y,
+                  family=poisson(link=identity) )
> ci.exp( ma, Exp=FALSE )
```

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

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

```
> 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.exp( mm )

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

> round( ci.exp( mm, ctr.mat=CM ), 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
```

14. If we want the rates in units of cases per 1000, we just use `Y/1000` as the person time:

```
> mm <- glm( D ~ factor(xpos),
+                 offset=log(Y/1000), family=poisson )
> ci.exp( mm, ctr.mat=CM )

           exp(Est.)     2.5%    97.5%
rate 0      2.711497 1.634669 4.497678
rate 1      5.854066 4.041994 8.478512
RR 1 vs. 0  2.158980 1.153160 4.042106

> round( ci.exp( mm, ctr.mat=CM ), 3 )

           exp(Est.)  2.5% 97.5%
rate 0         2.711 1.635 4.498
rate 1         5.854 4.042 8.479
RR 1 vs. 0     2.159 1.153 4.042
```

15. The same machinery can be used to the additive model to get the rates and the rate-difference in one go. We want the rates per 1000, so we rescale; also note that the annotation of the resulting estimates are via the row-names of the contrast matrix.

```
> rownames( CM ) <- c("rate 0","rate 1","RD 1 vs. 0")
> ma <- glm( D/(Y/1000) ~ factor(xpos), weight=Y/1000,
+                   family=poisson(link=identity) )
> round( ci.exp( ma, ctr.mat=CM, Exp=FALSE ), 2 )
           Estimate 2.5% 97.5%
rate 0         2.71 1.34  4.08
rate 1         5.85 3.69  8.02
RD 1 vs. 0     3.14 0.58  5.71
```

3.3 Lexis diagram

Here are the cases and person-years split by age in three bands, and by period in 3 bands and lo for the 1902–11 birth cohort:

	period 1940–44		period 1945–49		period 1950–54		1902–11 cohort	
Age (y)	Cases	P-years	Cases	P-years	Cases	P-years	Cases	P-years
40-44	-	11	1	9.5	-	6	1	16.5
45-49	-	6	-	12.2	2	10.5	1	15.7
50-54	1	6	1	8.5	1	4.2	1	7.1

1. We can load the dataset from the `Epi` package by:

```
> library( Epi )
> data( occup )
> occup

    AoE    DoE    DoX Xst
1  51.0 1941.0 1944.0   D
2  48.0 1940.0 1947.0   X
3  47.0 1942.0 1948.0   D
4  51.0 1948.0 1951.4   D
5  48.5 1946.9 1951.8   W
6  41.0 1940.0 1947.2   W
7  44.0 1944.0 1949.5   W
8  40.0 1941.0 1950.5   D
9  40.0 1943.0 1947.5   D
10 47.0 1951.0 1958.1   D
11 42.0 1947.0 1954.0   D
12 40.0 1947.0 1960.0   X
13 41.0 1951.0 1958.7   W
```

In order to compute the cases and person-years we set up a `Lexis` object:

```
> oL <- Lexis( entry = list( age=AoE, per=DoE ),
+               exit = list(          per=DoX ),
+       entry.status = factor( rep("W",nrow(occup)) ),
+        exit.status = factor( Xst ),
+               data = occup )

Incompatible factor levels in entry.status and exit.status:
 both lex.Cst and lex.Xst now have levels:
 W D X

> summary( oL )

Transitions:
     To
From W D X  Records:  Events: Risk time:  Persons:
   W 4 7 2        13        9       85.8        13
```

Exit status `X` and `W` are synonymous. If we want to classify the follow-up (person-years and events) by age and calendar time we must first subdivide by the two timescales; this is done by `splitLexis`:

```
> sL <- splitLexis( oL, time="age", breaks=seq(0,100,5) )
> sL <- splitLexis( sL, time="per", breaks=seq(0,100,5)+1900 )
> sL[order(sL$lex.id,sL$age),]
```

```
   lex.id  age    per lex.dur lex.Cst lex.Xst  AoE    DoE    DoX Xst
1       1 51.0 1941.0     3.0       W       D 51.0 1941.0 1944.0   D
2       2 48.0 1940.0     2.0       W       W 48.0 1940.0 1947.0   X
3       2 50.0 1942.0     3.0       W       W 48.0 1940.0 1947.0   X
4       2 53.0 1945.0     2.0       W       X 48.0 1940.0 1947.0   X
5       3 47.0 1942.0     3.0       W       W 47.0 1942.0 1948.0   D
6       3 50.0 1945.0     3.0       W       D 47.0 1942.0 1948.0   D
7       4 51.0 1948.0     2.0       W       W 51.0 1948.0 1951.4   D
8       4 53.0 1950.0     1.4       W       D 51.0 1948.0 1951.4   D
9       5 48.5 1946.9     1.5       W       W 48.5 1946.9 1951.8   W
10      5 50.0 1948.4     1.6       W       W 48.5 1946.9 1951.8   W
11      5 51.6 1950.0     1.8       W       W 48.5 1946.9 1951.8   W
12      6 41.0 1940.0     4.0       W       W 41.0 1940.0 1947.2   W
13      6 45.0 1944.0     1.0       W       W 41.0 1940.0 1947.2   W
14      6 46.0 1945.0     2.2       W       W 41.0 1940.0 1947.2   W
15      7 44.0 1944.0     1.0       W       W 44.0 1944.0 1949.5   W
16      7 45.0 1945.0     4.5       W       W 44.0 1944.0 1949.5   W
17      8 40.0 1941.0     4.0       W       W 40.0 1941.0 1950.5   D
18      8 44.0 1945.0     1.0       W       W 40.0 1941.0 1950.5   D
19      8 45.0 1946.0     4.0       W       W 40.0 1941.0 1950.5   D
20      8 49.0 1950.0     0.5       W       D 40.0 1941.0 1950.5   D
21      9 40.0 1943.0     2.0       W       W 40.0 1943.0 1947.5   D
22      9 42.0 1945.0     2.5       W       D 40.0 1943.0 1947.5   D
23     10 47.0 1951.0     3.0       W       W 47.0 1951.0 1958.1   D
24     10 50.0 1954.0     1.0       W       W 47.0 1951.0 1958.1   D
25     10 51.0 1955.0     3.1       W       D 47.0 1951.0 1958.1   D
26     11 42.0 1947.0     3.0       W       W 42.0 1947.0 1954.0   D
27     11 45.0 1950.0     4.0       W       D 42.0 1947.0 1954.0   D
28     12 40.0 1947.0     3.0       W       W 40.0 1947.0 1960.0   X
29     12 43.0 1950.0     2.0       W       W 40.0 1947.0 1960.0   X
30     12 45.0 1952.0     3.0       W       W 40.0 1947.0 1960.0   X
31     12 48.0 1955.0     2.0       W       W 40.0 1947.0 1960.0   X
32     12 50.0 1957.0     3.0       W       X 40.0 1947.0 1960.0   X
33     13 41.0 1951.0     4.0       W       W 41.0 1951.0 1958.7   W
34     13 45.0 1955.0     3.7       W       W 41.0 1951.0 1958.7   W
```

Having split the follow-up we can make a tabulation of the follow-up using the utility function `timeBand`:

```
> table( timeBand(sL,"age","left"), timeBand(sL,"per","left"))
```

```
     1940 1945 1950 1955
  40    4    4    2    0
  45    3    4    4    2
  50    2    4    3    2
```

Also it is now straight-forward to show the follow-up in a Lexis diagram:

However we do not want the number of observations (lines) in the dataset, we want the number of person-years (`lex.dur`) and the number of deaths (`lex.Xst=="D"`), so we set up a matrix with these as columns, and define the two classification variables:

```
> FU <- with( sL, cbind(lex.Xst=="D",lex.dur) )
> colnames(FU) <- c("D","Y")
> Age    <- timeBand(sL,"age","left")
> Period <- timeBand(sL,"per","left")
```

This enables us to use `xtabs` to simultaneously tabulate person-years and deaths

```
> FUtab <- xtabs( FU ~ Age + Period )
> ftable( FUtab, col.vars=2:3 )
```

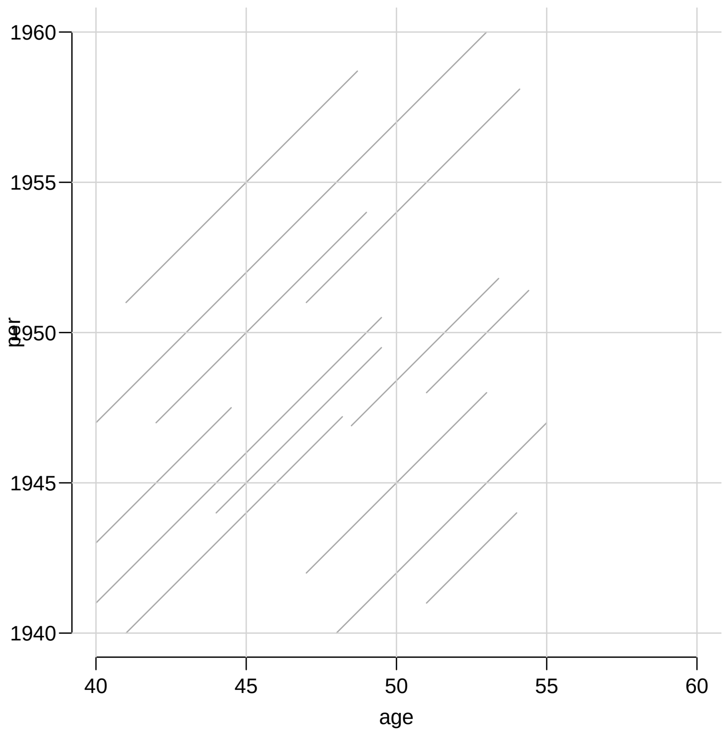

Figure 3.2: *Default plot of a Lexis object.*

```
    Period 1940       1945       1950       1955
              D    Y    D    Y    D    Y    D    Y
Age
40          0.0 11.0  1.0  9.5  0.0  6.0  0.0  0.0
45          0.0  6.0  0.0 12.2  2.0 10.5  0.0  5.7
50          1.0  6.0  1.0  8.6  1.0  4.2  1.0  6.1
```

2. If we want the tabulation by age for the birth cohort 1902–11, we simply restrict the dataset to his group, i.e. the persons where `per` − `age` is between 1029 and 1912:

```
> BC <- subset(sL,per-age>1902 & per-age<1912)
> FU <- with( BC, cbind(lex.Xst=="D",lex.dur) )
> colnames(FU) <- c("D","Y")
> Age    <- timeBand(BC,"age","left")
> FUctab <- xtabs( FU ~ Age )
> FUctab
```

```
Age     D    Y
  40  1.0 16.5
  45  1.0 15.7
  50  1.0  7.1
```

3. The cumulative rate for the cohort are defined theoretically as:

$$5 \times \left(\frac{1}{16.5} + \frac{1}{15.7} + \frac{1}{7.1} \right) = 1.32, \quad 1 - \exp(-1.32) = 0.73$$

or in terms of the just computed:

```
> sum( FUctab[,1] / FUctab[,2]*5 )
```

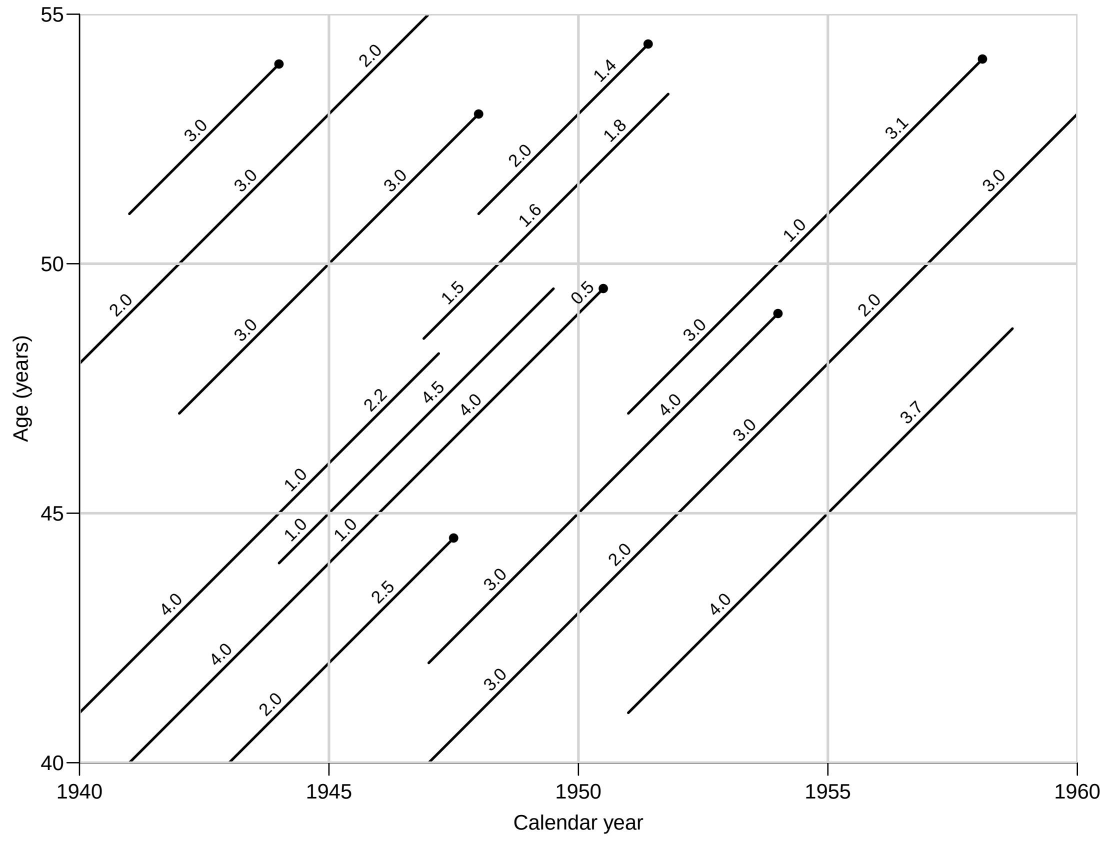

Figure 3.3: *Lexis diagram of the occupational cohort using a few bells and whistles.*

```
[1] 1.325727
```

4. The expected number of cases is computed by taking the person-years and multiplying with the reference rates, in this case 100, 200 and 400 per 100,000 PY for the three age classes throughout the follow-up:

$$E = \frac{100}{10^5\text{y}}\times(11+9.5+6+0)\text{ y}+\frac{200}{10^5\text{y}}\times(6+12.2+10.5+5.7)\text{ y}+\frac{400}{10^5\text{y}}\times(6+8.5+4.2+6.1)\text{ y} = 0.1949$$

The observed number of cases is $O = 7$, so the standardized incidence ratio is $7/0.1949 = 35.9$. Quite a risky occupation!

Note that the point of subdividing the follow-up by age and calendar time is to make it possible to apply population rates to the follow-up — the population rates vary by age and calendar time. So what is formally done is to match the population rates to the follow-up dataset:

```
> p.rates <- data.frame( rate=c(100,200,400), Age=c(40,45,50) )
> sL$Age <- timeBand(sL,"age","left")
> sL <- merge( sL, p.rates)
> sL
```

```
  Age lex.id  age    per lex.dur lex.Cst lex.Xst  AoE    DoE    DoX Xst rate
1  40      8 40.0 1941.0     4.0       W       W 40.0 1941.0 1950.5   D  100
2  40      9 40.0 1943.0     2.0       W       W 40.0 1943.0 1947.5   D  100
```

```
3   40       8 44.0 1945.0     1.0        W       W 40.0 1941.0 1950.5   D  100
4   40       6 41.0 1940.0     4.0        W       W 41.0 1940.0 1947.2   W  100
5   40      12 43.0 1950.0     2.0        W       W 40.0 1947.0 1960.0   X  100
6   40       9 42.0 1945.0     2.5        W       D 40.0 1943.0 1947.5   D  100
7   40       7 44.0 1944.0     1.0        W       W 44.0 1944.0 1949.5   W  100
8   40      12 40.0 1947.0     3.0        W       W 40.0 1947.0 1960.0   X  100
9   40      13 41.0 1951.0     4.0        W       W 41.0 1951.0 1958.7   W  100
10  40      11 42.0 1947.0     3.0        W       W 42.0 1947.0 1954.0   D  100
11  45       3 47.0 1942.0     3.0        W       W 47.0 1942.0 1948.0   D  200
12  45       2 48.0 1940.0     2.0        W       W 48.0 1940.0 1947.0   X  200
13  45       5 48.5 1946.9     1.5        W       W 48.5 1946.9 1951.8   W  200
14  45       6 46.0 1945.0     2.2        W       W 41.0 1940.0 1947.2   W  200
15  45       8 45.0 1946.0     4.0        W       W 40.0 1941.0 1950.5   D  200
16  45       6 45.0 1944.0     1.0        W       W 41.0 1940.0 1947.2   W  200
17  45      12 45.0 1952.0     3.0        W       W 40.0 1947.0 1960.0   X  200
18  45      10 47.0 1951.0     3.0        W       W 47.0 1951.0 1958.1   D  200
19  45       7 45.0 1945.0     4.5        W       W 44.0 1944.0 1949.5   W  200
20  45      13 45.0 1955.0     3.7        W       W 41.0 1951.0 1958.7   W  200
21  45      11 45.0 1950.0     4.0        W       D 42.0 1947.0 1954.0   D  200
22  45       8 49.0 1950.0     0.5        W       D 40.0 1941.0 1950.5   D  200
23  45      12 48.0 1955.0     2.0        W       W 40.0 1947.0 1960.0   X  200
24  50       1 51.0 1941.0     3.0        W       D 51.0 1941.0 1944.0   D  400
25  50       3 50.0 1945.0     3.0        W       D 47.0 1942.0 1948.0   D  400
26  50       2 50.0 1942.0     3.0        W       W 48.0 1940.0 1947.0   X  400
27  50       2 53.0 1945.0     2.0        W       X 48.0 1940.0 1947.0   X  400
28  50      10 51.0 1955.0     3.1        W       D 47.0 1951.0 1958.1   D  400
29  50       5 50.0 1948.4     1.6        W       W 48.5 1946.9 1951.8   W  400
30  50       4 51.0 1948.0     2.0        W       W 51.0 1948.0 1951.4   D  400
31  50       4 53.0 1950.0     1.4        W       D 51.0 1948.0 1951.4   D  400
32  50       5 51.6 1950.0     1.8        W       W 48.5 1946.9 1951.8   W  400
33  50      10 50.0 1954.0     1.0        W       W 47.0 1951.0 1958.1   D  400
34  50      12 50.0 1957.0     3.0        W       X 40.0 1947.0 1960.0   X  400
```

With this we can now compute the observed and expected cases:

```
> O <- with( sL, sum( lex.Xst=="D" ) )
> E <- with( sL, sum( lex.dur*rate/10^5 ) )
> c(O,E,O/E)
```

```
[1]  7.00000  0.19490 35.91585
```

Usually, we will use smaller intervals, as well as population rates that actually *do* vary by calendar time, but that would require proper statistical modeling of the rates.

3.4 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.

3.4.1 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] Inf
```

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

```
R version 3.2.2 (2015-08-14)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 14.04.3 LTS

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

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

other attached packages:
[1] survival_2.38-3 Epi_1.1.71

loaded via a namespace (and not attached):
[1] cmprsk_2.2-7    MASS_7.3-44     parallel_3.2.2  etm_0.6-2       splines_3.2.2
[6] grid_3.2.2      lattice_0.20-31
```

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.

3.4.2 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
  0.008   0.000   0.008

 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
```

3.4.3 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 
 0.130   0.003   0.134 
```

. . . 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 
14.703   0.025  14.722 

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
  0.215   0.004   0.219

 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
```

3.4.4 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 )
```

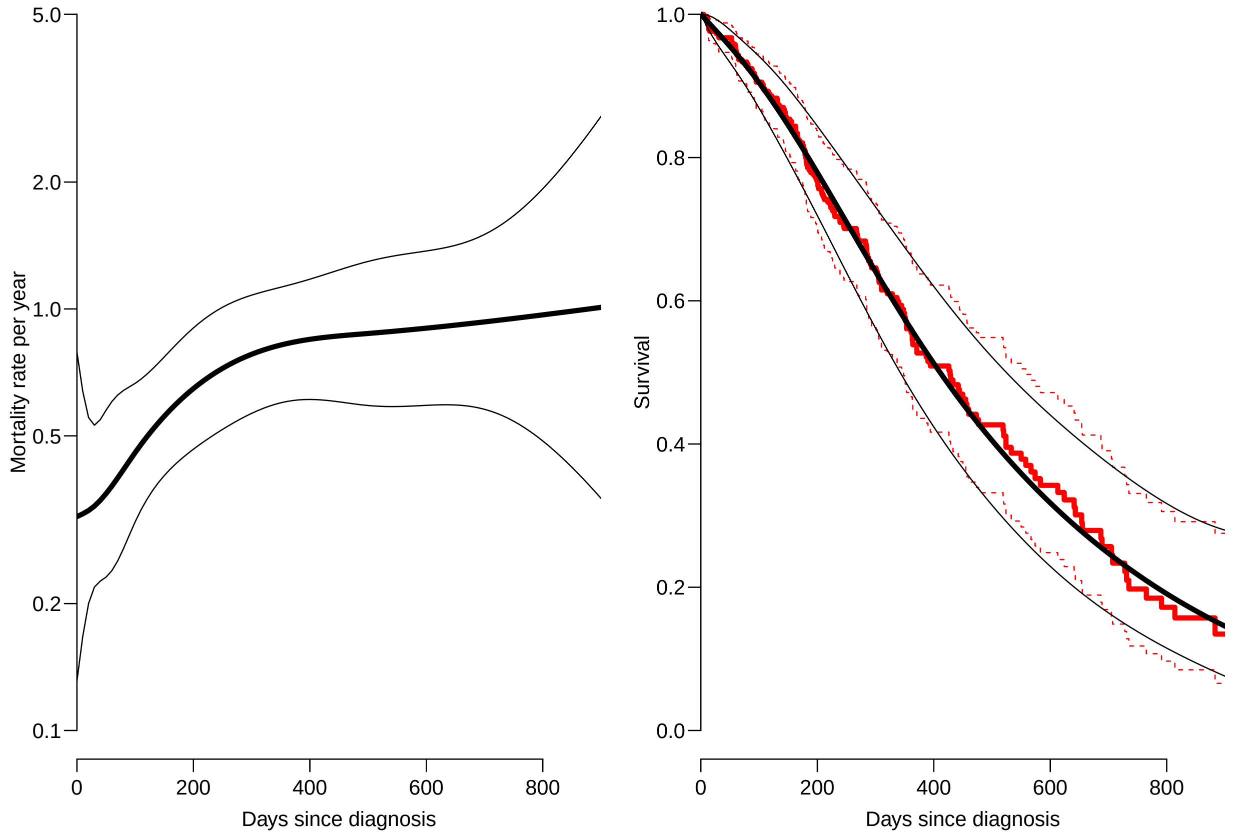

Figure 3.4: *Left: Hazard function for 60-year old men from spline model with 95% c.i. Right: Survival curve for 60 year old men; black from spline model, red from Cox-model.*

3.5 Estimation and reporting of linear and curved effects

The purpose of this exercise is to take you through models with curved effects of age and calendar time, in order to show you how to report 1) a curved effect of a main effect such as age and 2) a curved effect of a *relative* effect where a reference point is needed.

In the exercise we will use the `testisDK` data from the `Epi` package, which contains the number of cases of testis cancer in Denmark 1943–96:

1. First we load the Danish testis cancer data, and inspect the dataset:

```
> library( Epi )
> sessionInfo()

R version 3.2.2 (2015-08-14)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 14.04.3 LTS

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

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

other attached packages:
[1] Epi_1.1.71

loaded via a namespace (and not attached):
[1] cmprsk_2.2-7    MASS_7.3-44     parallel_3.2.2  survival_2.38-3 etm_0.6-2
[6] splines_3.2.2   grid_3.2.2      lattice_0.20-31

> data( testisDK )
> str( testisDK )

'data.frame':        4860 obs. of  4 variables:
 $ A: num  0 1 2 3 4 5 6 7 8 9 ...
 $ P: num  1943 1943 1943 1943 1943 ...
 $ D: num  1 1 0 1 0 0 0 0 0 0 ...
 $ Y: num  39650 36943 34588 33267 32614 ...

> head( testisDK )

  A    P D        Y
1 0 1943 1 39649.50
2 1 1943 1 36942.83
3 2 1943 0 34588.33
4 3 1943 1 33267.00
5 4 1943 0 32614.00
6 5 1943 0 32020.33
```

We can tabulate both events (testis cancer diagnoses) and person-years using either `xtabs` or `stat.table`, the latter is a bit more versatile, because we can get rates too:

```
> round( ftable( xtabs( cbind(D,PY=Y/1000) ~ I(floor(A/10)*10) +
+                                            I(floor(P/10)*10),
+                       data=testisDK ),
+                row.vars=c(3,1) ), 1 )
```

```
                       I(floor(P/10) * 10)   1940   1950   1960   1970   1980   1990
   I(floor(A/10) * 10)
D  0                                         10.0    7.0   16.0   18.0    9.0   10.0
   10                                        13.0   27.0   37.0   72.0   97.0   75.0
   20                                       124.0  221.0  280.0  535.0  724.0  557.0
   30                                       149.0  288.0  377.0  624.0  771.0  744.0
   40                                        95.0  198.0  230.0  334.0  432.0  360.0
   50                                        40.0   79.0  140.0  151.0  193.0  155.0
   60                                        29.0   43.0   54.0   83.0   82.0   44.0
   70                                        18.0   26.0   35.0   41.0   40.0   32.0
   80                                         7.0    9.0   13.0   19.0   18.0   21.0
PY 0                                       2604.7 4037.3 3885.0 3820.9 3070.9 2165.5
   10                                      2135.7 3505.2 4004.1 3906.1 3847.4 2261.0
   20                                      2225.5 2923.2 3401.6 4028.6 3941.2 2824.6
   30                                      2195.2 3058.8 2856.2 3410.6 3968.8 2728.4
   40                                      1874.9 2980.1 2986.8 2823.1 3322.6 2757.7
   50                                      1442.8 2426.5 2796.6 2813.3 2635.0 2069.2
   60                                      1041.9 1711.8 2055.1 2358.1 2357.3 1565.0
   70                                       537.6  967.9 1136.1 1336.9 1538.0 1100.9
   80                                       133.6  261.6  346.3  423.5  504.2  414.6
```

Note that for this type of cancer the peak age-specific rates are in the 30es.

2. We then fit a Poisson-model for the mortality rates with a linear term for age:

```
> ml <- glm( D ~ A, offset=log(Y), family=poisson, data=testisDK )
> ci.exp( ml )
```

```
                exp(Est.)         2.5%        97.5%
(Intercept) 5.682883e-05 0.0000545697 0.0000591815
A           1.005499e+00 1.0045507062 1.0064479370
```

The parameter labeled A gives the annual increase in mortality by age (0.55%/year), but the intercept parameter is meaningless; it is the predicted mortality per 1 person-year (because we used Y in the offset, and this is in units of person-years), but for a 0 year old male.

3. We can work out the predicted log-mortality rates for ages 25 to 45, say, by doing a hand-calculation based on the coefficients:

```
> ( cf <- coef( ml ) )
```

```
 (Intercept)            A
-9.775466746  0.005483811
```

We now have the intercept (the log-rate) and the slopes for age and calendar time, so to get the age-specific rates in ages 50 to 60 we just take the intercept and add the slope multiplied by the vector of ages.

```
> round( cbind( 25:45, exp( cf[1] + cf[2]*(25:45) )*10^5 ), 3 )
```

```
      [,1]  [,2]
 [1,]   25 6.518
 [2,]   26 6.554
 [3,]   27 6.590
 [4,]   28 6.626
 [5,]   29 6.662
 [6,]   30 6.699
 [7,]   31 6.736
 [8,]   32 6.773
 [9,]   33 6.810
```

```
[10,]   34 6.848
[11,]   35 6.885
[12,]   36 6.923
[13,]   37 6.961
[14,]   38 7.000
[15,]   39 7.038
[16,]   40 7.077
[17,]   41 7.116
[18,]   42 7.155
[19,]   43 7.194
[20,]   44 7.234
[21,]   45 7.273
```

Note that we also multiplied by 10^5 in order to get the rates in units of cases per 100,000 person-years.

4. But we do not have the standard errors of these mortality rates, and hence neither the confidence intervals. This is implemented in `ci.exp`; if we provide the argument `ctr.mat=` as a matrix where each row corresponds to a prediction point and each column to a parameter from the model.

   Thus for each age we need the corresponding multipliers for the coefficients:

```
> ( CM <- cbind( 1, 25:45 ) )

      [,1] [,2]
 [1,]    1   25
 [2,]    1   26
 [3,]    1   27
 [4,]    1   28
 [5,]    1   29
 [6,]    1   30
 [7,]    1   31
 [8,]    1   32
 [9,]    1   33
[10,]    1   34
[11,]    1   35
[12,]    1   36
[13,]    1   37
[14,]    1   38
[15,]    1   39
[16,]    1   40
[17,]    1   41
[18,]    1   42
[19,]    1   43
[20,]    1   44
[21,]    1   45

> round( ci.exp( ml, ctr.mat=CM )*10^5, 3 )

      exp(Est.)  2.5% 97.5%
 [1,]     6.518 6.365 6.674
 [2,]     6.554 6.403 6.708
 [3,]     6.590 6.441 6.742
 [4,]     6.626 6.479 6.777
 [5,]     6.662 6.516 6.812
 [6,]     6.699 6.554 6.847
 [7,]     6.736 6.592 6.883
 [8,]     6.773 6.630 6.919
 [9,]     6.810 6.667 6.956
[10,]     6.848 6.705 6.993
[11,]     6.885 6.743 7.031
[12,]     6.923 6.780 7.069
[13,]     6.961 6.817 7.108
```

```
[14,]     7.000 6.855 7.147
[15,]     7.038 6.892 7.187
[16,]     7.077 6.929 7.228
[17,]     7.116 6.966 7.268
[18,]     7.155 7.003 7.310
[19,]     7.194 7.040 7.352
[20,]     7.234 7.077 7.394
[21,]     7.273 7.113 7.437
```

5. We can now use this machinery to plot the mortality rates over the range from 15 to 65 years:

```
> C1 <- cbind( 1, 15:65 )
> matplot( 15:65, ci.exp( ml, ctr.mat=C1 )*10^5,
+          log="y", xlab="Age", ylab="Testis cancer incidence rate per 100,000 PY",
+          type="l", lty=1, lwd=c(3,1,1), col="black" )
```

6. Now suppose we want to see if the mortality rates really are exponentially increasing by age (that is linearly on the log-scale), we could add a quadratic term to the model:

```
> mq <- glm( D ~ A + I(A^2), offset=log(Y), family=poisson, data=testisDK )
> ci.exp( mq, Exp=F )
                 Estimate          2.5%         97.5%
(Intercept) -12.365625166 -12.482504296 -12.248746037
A             0.180595889   0.174140158   0.187051619
I(A^2)       -0.002325937  -0.002410829  -0.002241045
```

Note that we must use the function `I()` to prevent the "^" to be interpreted as part of the model formula.

We can then plot the estimated rates using the same machinery, but now with 3 columns in the matrix:

```
> aa <- 15:65
> C2 <- cbind( 1, aa, aa^2 )
> matplot( aa, ci.exp( mq, ctr.mat=C2 )*10^5,
+          log="y", xlab="Age", ylab="Testis cancer incidence rate per 100,000 PY",
+          type="l", lty=1, lwd=c(3,1,1), col="black" )
> matlines( aa, ci.exp( ml, ctr.mat=C1 )*10^5,
+           type="l", lty=1, lwd=c(3,1,1), col="blue" )
```

Which indeed is dramatically different — we see that the model with quadratic effect gives a much better fit; a deviance of 4800 on 1 d.f.:

```
> anova( mq, ml, test="Chisq" )
Analysis of Deviance Table

Model 1: D ~ A + I(A^2)
Model 2: D ~ A
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1      4857     7815.8
2      4858    12648.0 -1  -4832.1 < 2.2e-16
```

7. We could do the same using a 3rd degree polynomial:

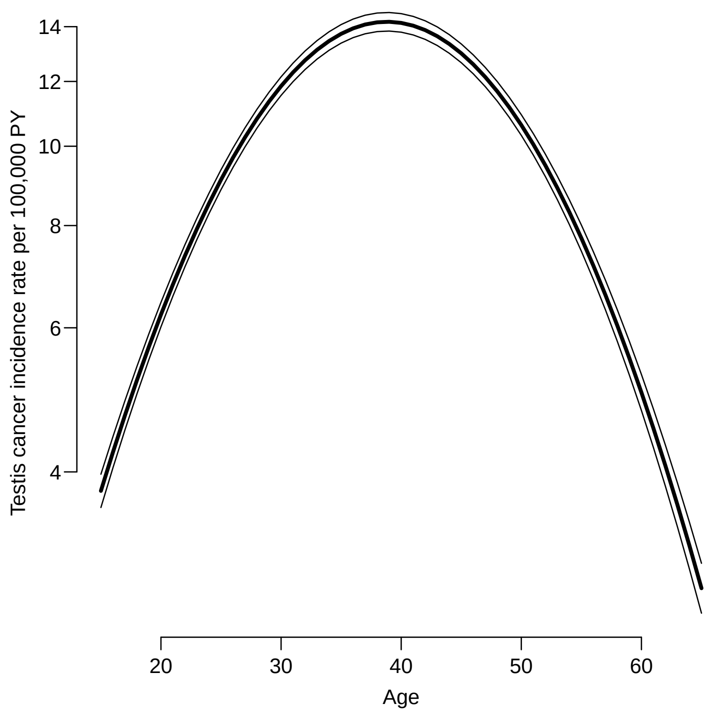

Figure 3.5: *Testis cancer incidence rates overall, modelled by 2nd degree polynomial, overlaid by the estimated linear estimate.*

```
> mc <- glm( D ~ A + I(A^2) + I(A^3), offset=log(Y), family=poisson, data=testisDK )
> C3 <- cbind( 1, aa, aa^2, aa^3 )
> matplot( aa, ci.exp( mc, ctr.mat=C3 )*10^5,
+          log="y", xlab="Age", ylab="Testis cancer incidence rate per 100,000 PY",
+          type="l", lty=1, lwd=c(3,1,1), col="black" )
```

8. Instead of continuing with higher powers of age we could use fractions of powers ("fractional polynomials"), or we could use splines, which are piece wise polynomial curves, that fit nicely together at join points (knots). This is implemented in the `splines` package, in the function `ns`, which returns a matrix. There is a wrapper `Ns` in the `Epi`-package that automatically designate the smallest and largest knots as *boundary knots*, beyond which the resulting curve is linear:

```
> library( splines )
> ms <- glm( D ~ Ns(A,knots=seq(15,65,10)), offset=log(Y), family=poisson, data=testisDK )
> As <- Ns( aa, knots=seq(15,65,10) )
> matplot( aa, ci.exp( ms, ctr.mat=cbind(1,As) )*10^5,
+          log="y", xlab="Age", ylab="Testis cancer incidence rate per 100,000 PY",
+          type="l", lty=1, lwd=c(3,1,1), col="black" )
```

9. Now in addition to this we would like to see how the dependence on calendar was, so we add a linear term to the model, and make a prediction for 1970, say:

```
> msp <- glm( D ~ Ns(A,knots=seq(15,65,10)) + P, offset=log(Y), family=poisson, data=testisDK
> CM <- cbind( 1, Ns( aa, knots=seq(15,65,10) ), 1970 )
> matplot( aa, ci.exp( msp, ctr.mat=CM )*10^5,
+          log="y", xlab="Age", ylab="Testis cancer incidence rate per 100,000 PY",
+          type="l", lty=1, lwd=c(3,1,1), col="black" )
```

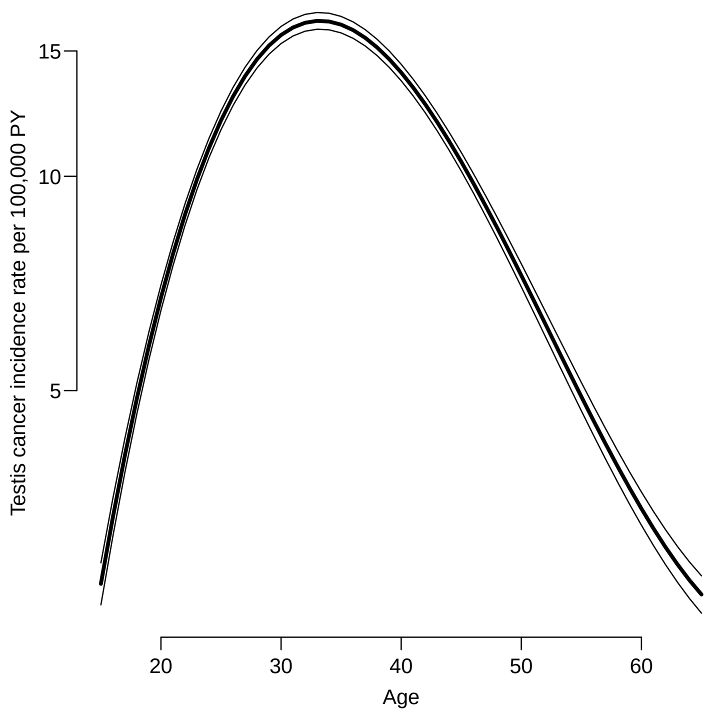

Figure 3.6: *Testis cancer incidence rates overall, modelled by 3rd degree polynomial.*

10. We would also like to see how the RR relative to 1970 is, so we select only the period parameter, using the `subset` argument:

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

```
  exp(Est.)     2.5%    97.5%
P  1.024235 1.022769 1.025704
```

    So we have an increase of 2.4%

    To get the RR relative to 1970 for the years 1943 to 1996 we must multiply the log-RR for period with the distance form 1970, such as:

```
> yy  <- 1943:1996
> Cp1 <- cbind( yy - 1970 )
> matplot( yy, ci.exp( msp, ctr.mat=Cp1, subset="P" ),
+          log="y", xlab="Date", ylab="RR of Testis cancer",
+          type="l", lty=1, lwd=c(3,1,1), col="black" )
> abline( h=1 )
```

11. As above we might like to see how it looks if we add a quadratic to the period effect:

```
> msp <- glm( D ~ Ns(A,knots=seq(15,65,10)) + P + I(P^2),
+                offset=log(Y), family=poisson, data=testisDK )
> Cq <- cbind( yy, yy^2 ) - cbind( rep(1970,length(yy)), 1970^2 )
> matplot( yy, ci.exp( msp, ctr.mat=Cq, subset="P" ),
+          log="y", xlab="Age", ylab="Testis cancer incidence RR",
+          type="l", lty=1, lwd=c(3,1,1), col="black" )
> abline( h=1, v=1970 )
```

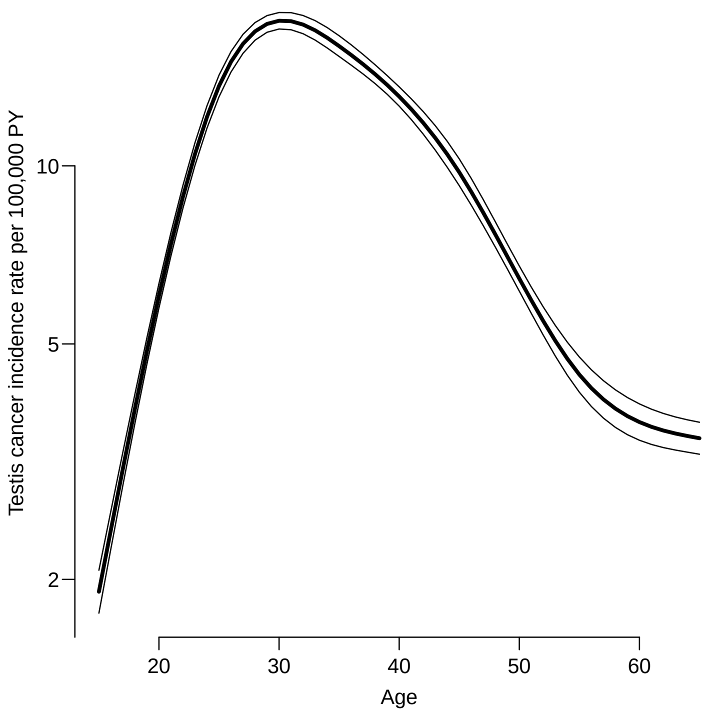

Figure 3.7: *Testis cancer incidence rates overall, modelled by splines.*

12. But we would like also to see if there were some non-linearity beyond the quadratic, with period as well, so we fit a spline for period (`P`) as well

```
> mssp <- glm( D ~ Ns(A,knots=seq(15,65,10)) +
+                  Ns(P,knots=seq(1950,1990,10)),
+                  offset=log(Y), family=poisson, data=testisDK )
> anova( mssp, msp, test="Chisq" )
Analysis of Deviance Table

Model 1: D ~ Ns(A, knots = seq(15, 65, 10)) + Ns(P, knots = seq(1950,
    1990, 10))
Model 2: D ~ Ns(A, knots = seq(15, 65, 10)) + P + I(P^2)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1      4850     4787.9
2      4852     4792.3 -2   -4.488    0.106
```

But as above we must compute the *difference* in the contribution from period in year y and in the reference year, here 1970. So every row of the contrast matrix must have the corresponding contribution from the reference year subtracted

```
> Ps <- Ns(                yy   , knots=seq(1950,1990,10) )
> Pr <- Ns( rep(1970,length(yy)), knots=seq(1950,1990,10) )
> matplot( yy, ci.exp( mssp, ctr.mat=Ps-Pr, subset="P" ),
+          log="y", xlab="Age", ylab="Testis cancer incidence RR",
+          type="l", lty=1, lwd=c(3,1,1), col="black" )
```

13. But for this model we would also like to see the estimated age-specific rates in say 1970.

To this end we need a reference matrix for the year with a number of rows equal to the number of age-prediction points:

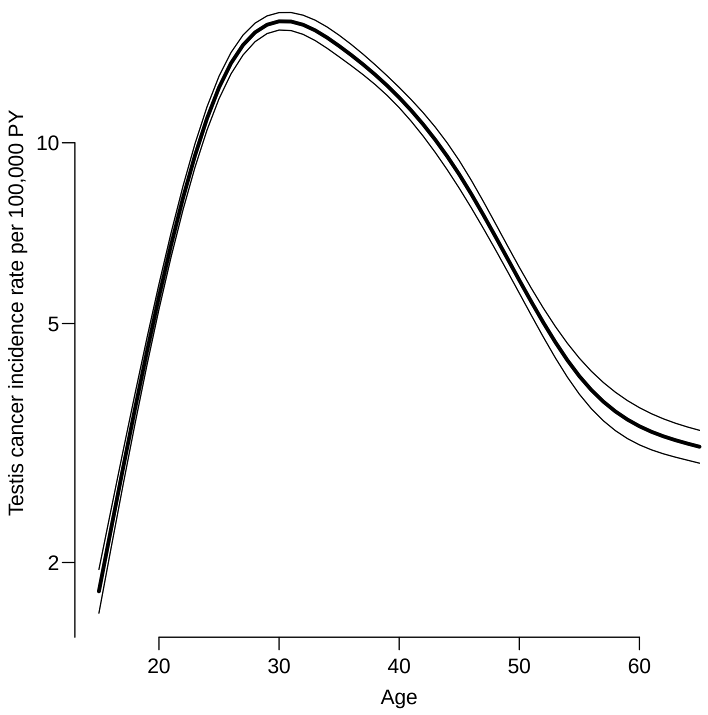

Figure 3.8: *Testis cancer incidence rate in 1970. Note the different level of the rates relative to the overall plot above.*

```
> Ar <- Ns( rep(1970,length(aa)), knots=seq(1950,1990,10) )
> matplot( aa, ci.exp( mssp, ctr.mat=cbind(1,As,Ar) )*10^5,
+          log="y", xlab="Age", ylab="Testis cancer incidence RR",
+          type="l", lty=1, lwd=c(3,1,1), col="black" )
```

14. In order to all do this in one go where we have overview of what we do, what is needed is:

    - the knots for age and period
    - the prediction points for age and period
    - the reference point for period

    — then we can derive everything else from this:

```
> a.kn <- seq(15,65,10)
> p.kn <- seq(1950,1990,10)
> a.pt <- 10:65
> p.pt <- 1945:1993
> p.ref <- 1970
> na <- length(a.pt)
> np <- length(p.pt)
> As <- Ns( a.pt, knots=a.kn )
> Ps <- Ns( p.pt, knots=p.kn )
> Prp <- Ns( rep(p.ref,np), knots=p.kn )
> Pra <- Ns( rep(p.ref,na), knots=p.kn )
> mAP <- glm( D ~ Ns(A,knots=a.kn) + Ns(P,knots=p.kn),
+                 offset=log(Y), family=poisson, data=testisDK )
> par( mfrow=c(1,2) )
> matplot( a.pt, ci.exp( mAP, ctr.mat=cbind(1,As,Pra) )*10^5,
```

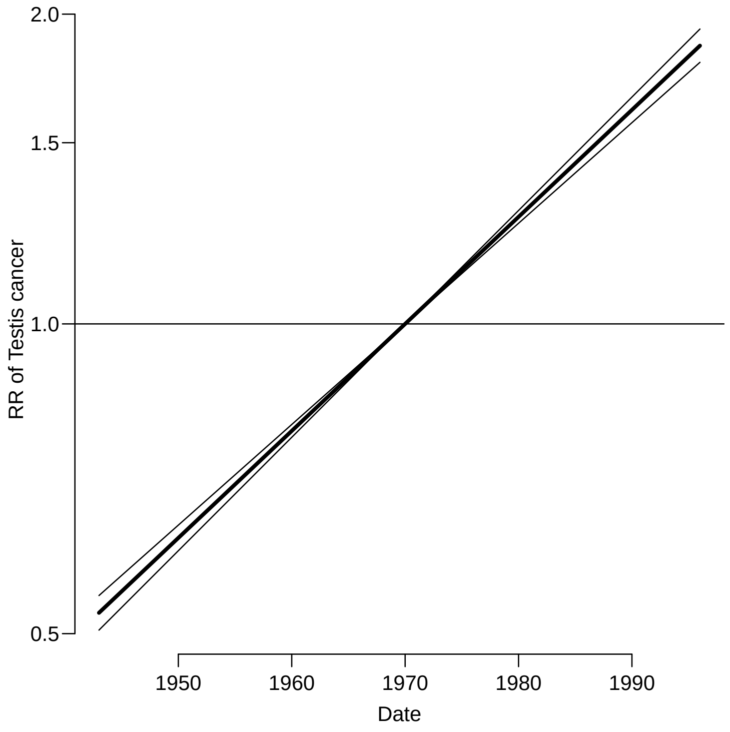

Figure 3.9: *Testis cancer incidence rate-ratio relative to 1970.*

```
+           log="y", xlab="Age",
+           ylab=paste( "Testis cancer incidence per 100,000 PY, in", p.ref ),
+           type="l", lty=1, lwd=c(3,1,1), col="black",
+           ylim=c(1,20) )
> matplot( p.pt, ci.exp( mAP, ctr.mat=Ps-Prp, subset="P" ),
+           log="y", xlab="Age", ylab="Testis cancer incidence RR",
+           type="l", lty=1, lwd=c(3,1,1), col="black",
+           ylim=c(1,20)/4 )
> abline( h=1, v=p.ref )
```

15. Finally with this in place we could do the same for a model where we had replaced `P`, the data of follow-up by the the date of birth, `B=P-A`:

```
> testisDK <- transform( testisDK, B = P-A )
> # with( testisDK, hist( rep(B,D), breaks=100, col="black" ) )
> a.kn <- seq(15,65,5)
> b.kn <- seq(1900,1970,5)
> a.pt <- 10:65
> b.pt <- 1890:1970
> b.ref <- 1940
> na <- length(a.pt)
> nb <- length(b.pt)
> As <- Ns( a.pt, knots=a.kn )
> Bs <- Ns( b.pt, knots=b.kn )
> Brb <- Ns( rep(b.ref,nb), knots=b.kn )
> Bra <- Ns( rep(b.ref,na), knots=b.kn )
> mAB <- glm( D ~ Ns(A,knots=a.kn) + Ns(B,knots=b.kn),
+                offset=log(Y), family=poisson, data=testisDK )
> par( mfrow=c(1,2) )
> matplot( a.pt, ci.exp( mAB, ctr.mat=cbind(1,As,Bra) )*10^5,
+           log="y", xlab="Age",
+           ylab=paste( "Testis cancer incidence per 100,000 PY, in", b.ref, "birth cohort"),
+           type="l", lty=1, lwd=c(3,1,1), col="black",
```

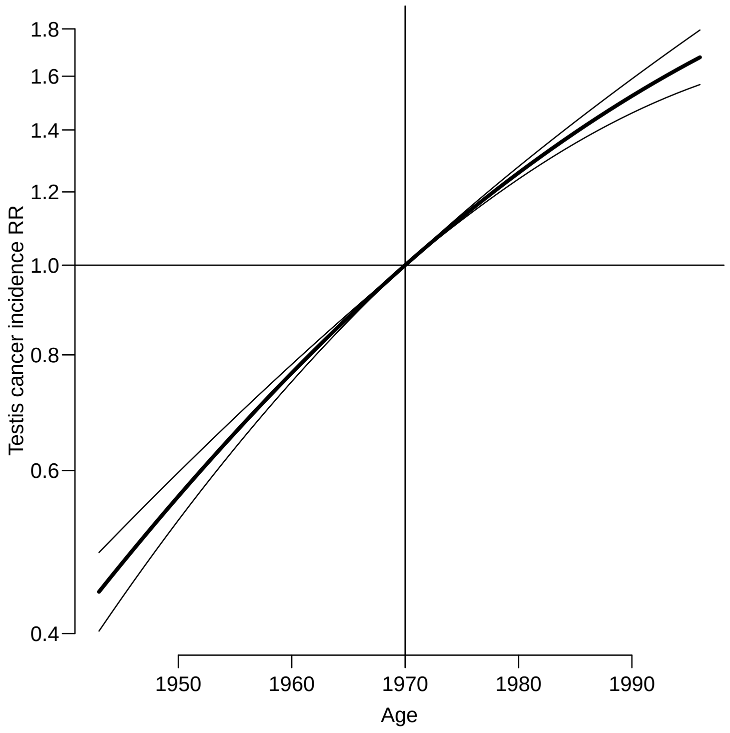

Figure 3.10: *Testis cancer incidence rate-ratio relative to 1970.*

```
+          ylim=c(1,20) )
> matplot( b.pt, ci.exp( mAB, ctr.mat=Bs-Brb, subset="B" ),
+          log="y", xlab="Age", ylab="Testis cancer incidence RR",
+          type="l", lty=1, lwd=c(3,1,1), col="black",
+          ylim=c(1,20)/4 )
> abline( h=1, v=b.ref )
```

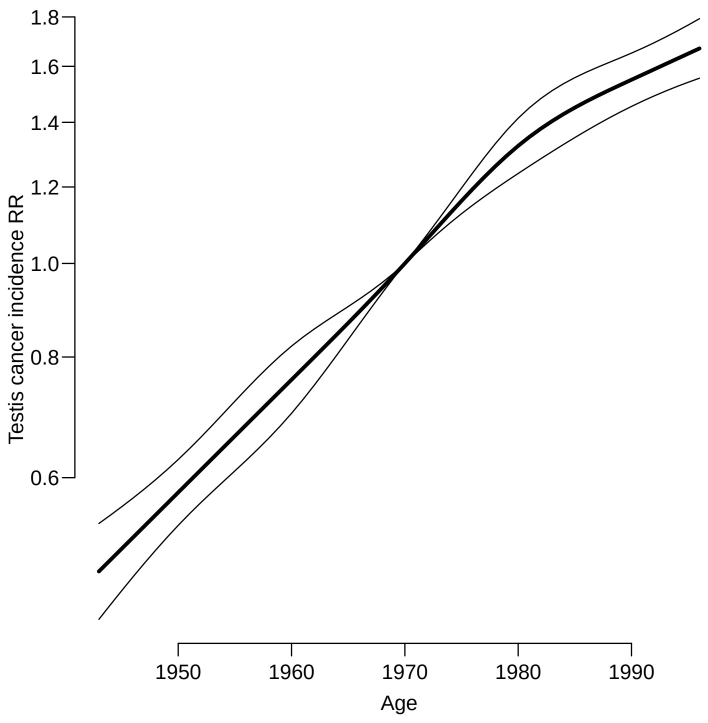

Figure 3.11: *Incidence rates of testis cancer in 1950 per 100,000 PY.*

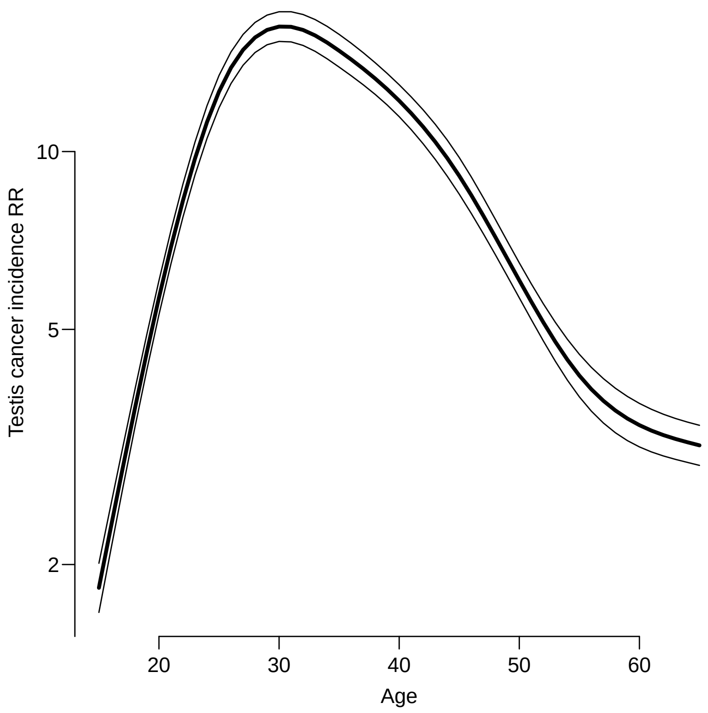

Figure 3.12: *Relative risk by date of follow-up.*

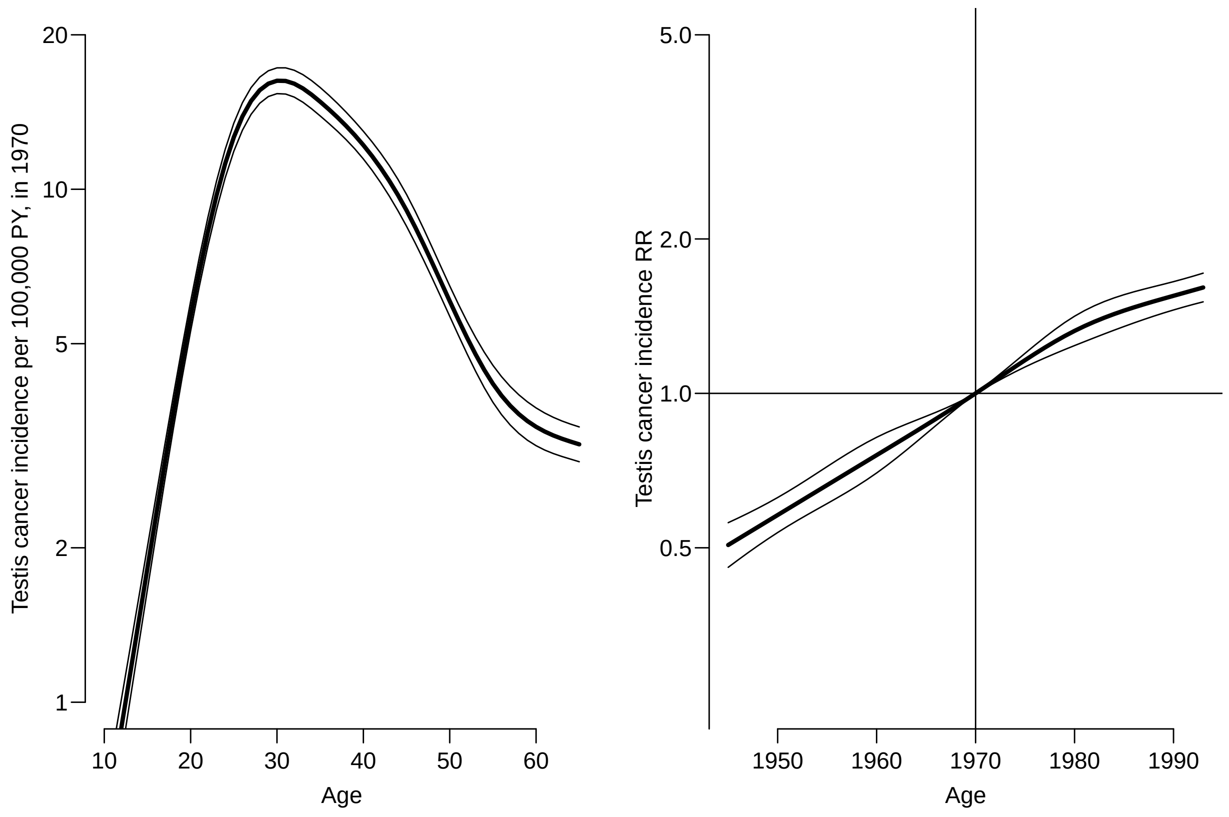

Figure 3.13: *Incidence rates of testis cancer in 1970, and RR relative to this.*

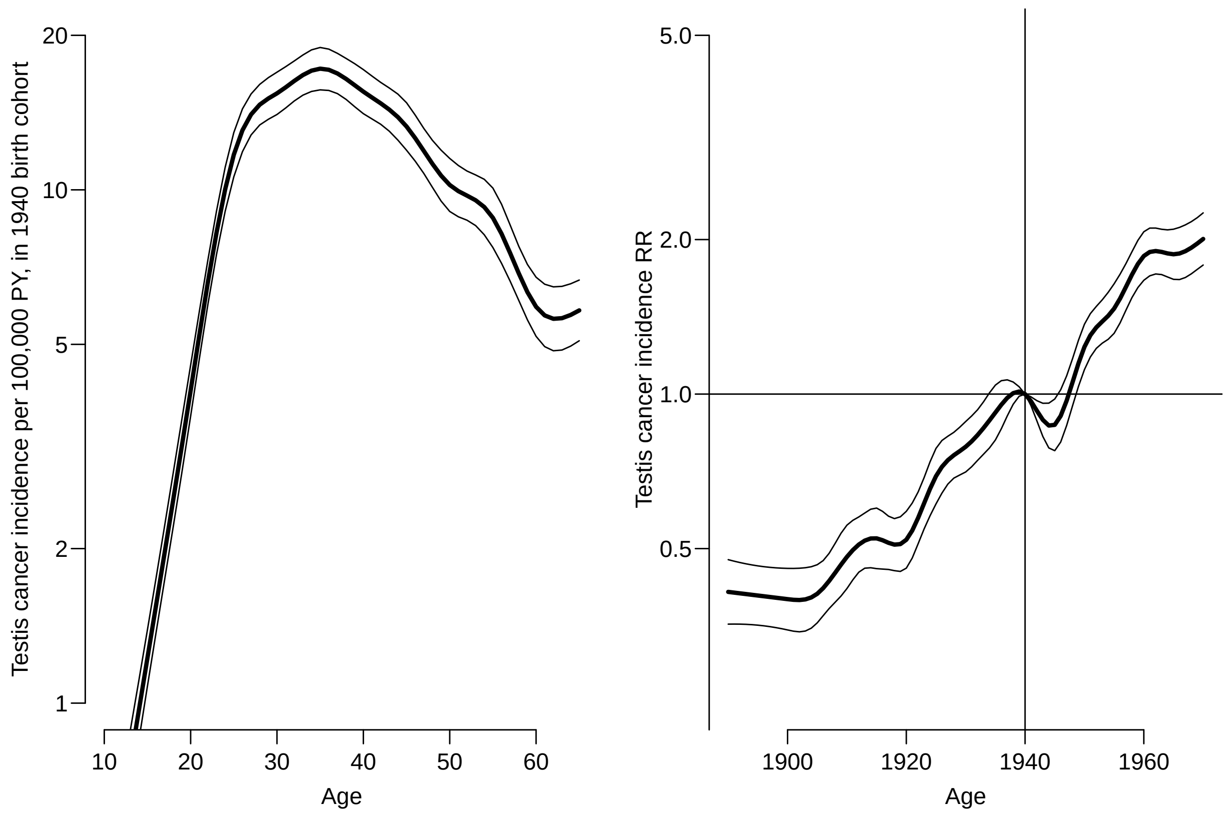

Figure 3.14: *Incidence rates of testis cancer in the 1940 birth cohort, and RR relative to this. We see that there is a considerable effect of birth cohort — there seems to be an effect of being born during the 1st or 2nd world war.*

3.6 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. 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 mainly about assessing how mortality depends age, and how to understand and compute years of life lost to diabetes by comparing with the population mortality.

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     
 M:5185   Min.   :1898   Min.   :1995   Min.   :1995   Min.   :1995   Min.   :1995  
 F:4815   1st Qu.:1930   1st Qu.:2000   1st Qu.:2002   1st Qu.:2001   1st Qu.:2001  
          Median :1941   Median :2004   Median :2005   Median :2004   Median :2005  
          Mean   :1942   Mean   :2003   Mean   :2005   Mean   :2004   Mean   :2004  
          3rd Qu.:1951   3rd Qu.:2007   3rd Qu.:2008   3rd Qu.:2007   3rd Qu.:2007  
          Max.   :2008   Max.   :2010   Max.   :2010   Max.   :2010   Max.   :2010  
                                         NA's   :7497   NA's   :4503   NA's   :8209  
      dox      
 Min.   :1995  
 1st Qu.:2010  
 Median :2010  
 Mean   :2009  
 3rd Qu.:2010  
 Max.   :2010  
```

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.

```
> 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. 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
```

4. We then estimate a crude age-specific mortality curves 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.

5. However, when we are working with event data, the `ns` machinery does not necessarily choose knots for splines sensibly, so it is better to explicitly allocate knots so that the number of *events* is the same between knots, here we use `quantile` on the subset of data with events — note that we add `lex.dur` so that we get the actual event times:

```
> ( a.kn <- with( subset(SL,lex.Xst=="Dead"),
+                 quantile( A+lex.dur, (1:10-0.5)/10 ) ) )

      5%      15%      25%      35%      45%      55%      65%      75%      85%      95%
56.02519 63.67995 69.06092 73.25311 76.29021 79.03847 81.42094 84.27242 87.66598 92.27406
```

   These are the locations of knots that places 10% of events between each successive pair of knots, and 5% beyond the outer knots. If we use these as knots in the function `Ns` we automatically get the smallest and the largest as boundary knots, beyond which the splines are linear:

```
> r.m <- glm( (lex.Xst=="Dead") ~ Ns( A, knots=a.kn ),
+             offset = log( lex.dur ),
+             family = poisson,
+               data = subset( SL, sex=="M" ) )
> r.f <- update( r.m, data = subset( SL, sex=="F" ) )
```

6. With these objects we can get the estimated log-rates by using `ci.pred`, supplying a data frame of prediction points, and finally use the wrapper `ci.pred` to get the rates with CIs:

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

and then we can plot the two sets of estimated rates:

```
> matplot( seq(10,90,0.5), cbind(p.m,p.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" )
```

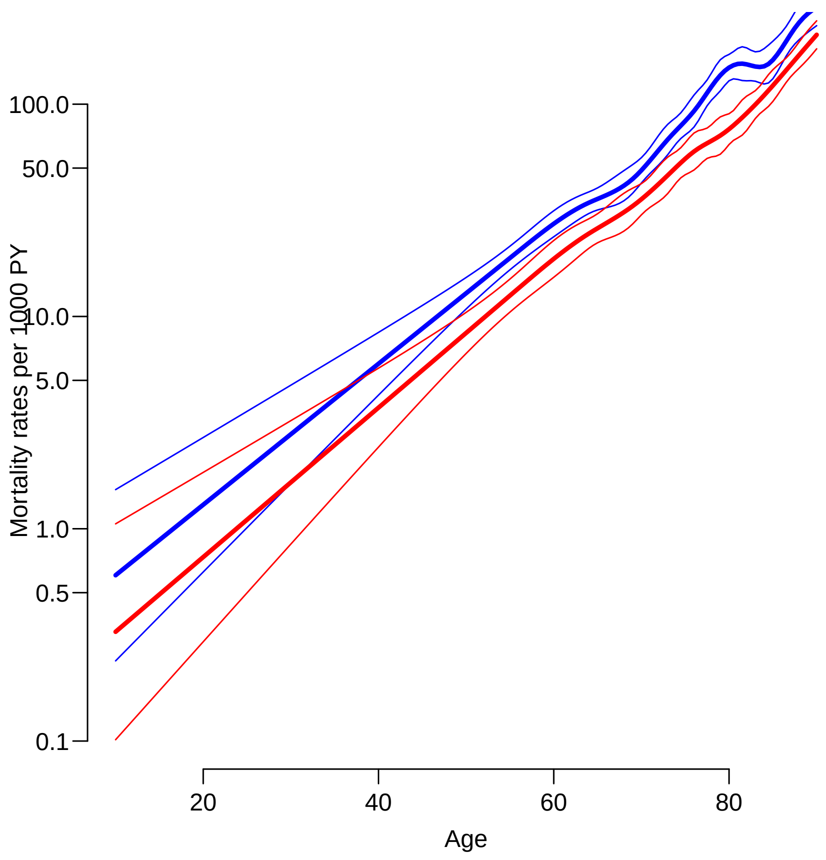

Figure 3.15: *Age-specific mortality rates for Danish diabetes patients as estimated from a model with only age. Blue: men, red: women.*

3.6.1 Comparison with the population rates

7. 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 ) )
```

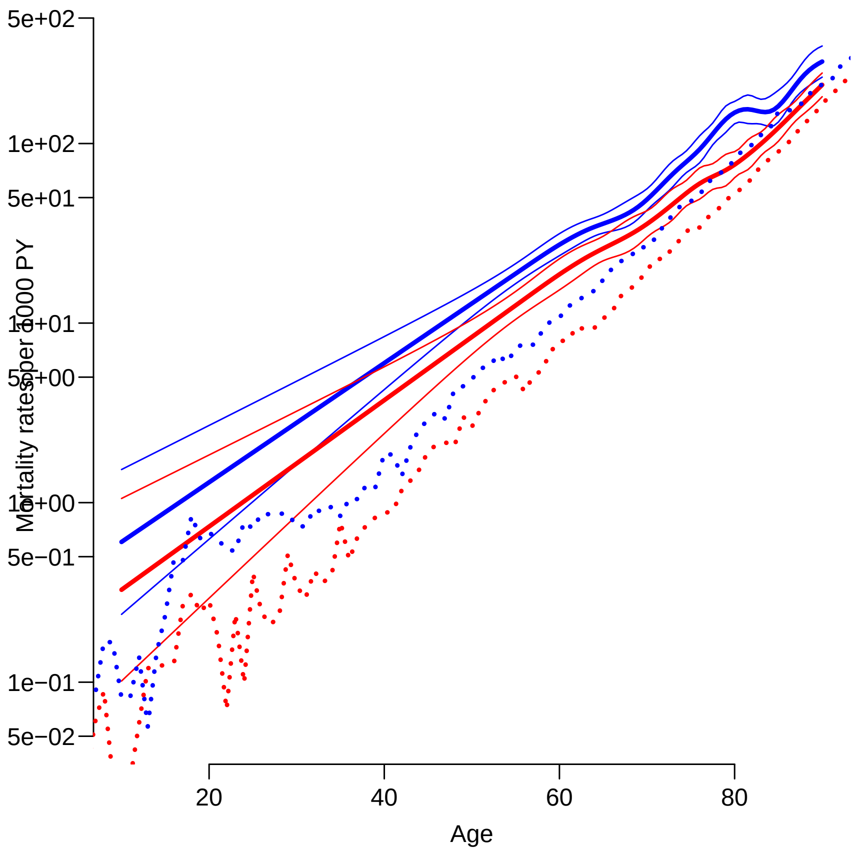

Figure 3.16: *Age-specific mortality rates for Danish diabetes patients as estimated from a model with only age. Broken lines are empirical rates from 2005. Blue: men, red: women.*

8. It would however be more prudent to model these rates in a similar fashion as the diabetes mortality — note that we now supply a column `Y` for the person-years in order to get rates per 1000 PY:

```
> R.m <- glm( D ~ Ns( A, knots=seq(10,90,10) ),
+             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 <- ci.pred( R.m, newdata = nd )
> P.f <- ci.pred( R.f, 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), cbind(p.m,p.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), cbind(P.m,P.f), lty="12",
+           col=c("blue","red"), lwd=c(3,1,1) )
```

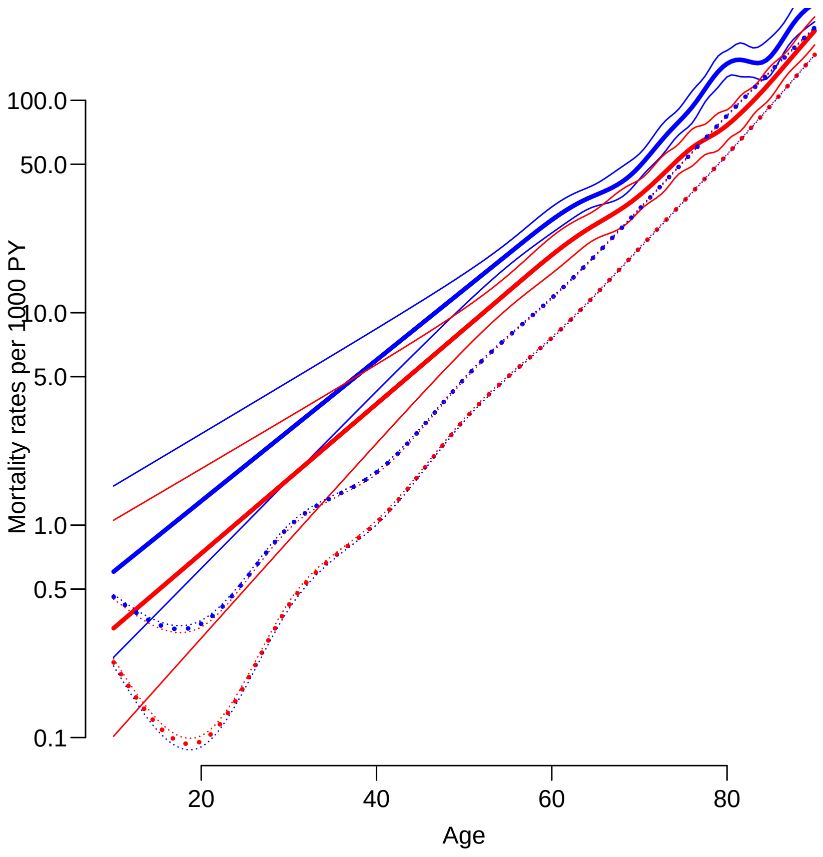

Figure 3.17: *Age-specific mortality rates for Danish diabetes patients as estimated from a model with only age. Broken lines are modeled population rates 1995–2010. Blue: men, red: women.*

3.6.2 Life expectancy

9. Recall from the section of fundamental concepts that the expected lifetime is the area under the survival curve, and remember the relationship between the mortality rates

and the survival curve:

$$S(t) = \exp\left(-\int_o^t \lambda(s)\,\mathrm{d}s\right)$$

The $\lambda(s)$ is the smooth function of age we just estimated in the models for the diabetes population and for the general population.

Now, an integral is merely a sum; we can compute it by approximating the area under the curve with a histogram with very narrow intervals. So we compute the λ in the middle of 1000 intervals between 0 and 100 years, each 1/10 year long. We then multiply each value by the width of the interval to get the integral:

```
> mid.pt <- 0:999/10 + 1/20
> mid.pt[1:5]

[1] 0.05 0.15 0.25 0.35 0.45

> nd <- data.frame( A = mid.pt, Y = 1 )
```

Note that we devise a data frame `nd` where the person-years is 1, so that we get the predicted rates in the units of "events per year". Recall that `ci.pred` automatically gives us the rates, so in order to get the cumulative rates at each of the ages,

$$\Lambda(t) = \int_O^t \lambda(s)\,\mathrm{d}s$$

we use `cumsum`, and then the exponential to get the survival curves:

```
> S.m <- exp( -cumsum( ci.pred(R.m,newdata=nd)[,1]*1/10 ) )
> S.f <- exp( -cumsum( ci.pred(R.f,newdata=nd)[,1]*1/10 ) )
```

Note that we have multiplied the estimated rate (calculated in units of events per 1 year in `ci.pred`) by the interval length 1/10.

10. We now have the population survival curves for men and women, so easily plotted:

```
> matplot( mid.pt+1/20, cbind(S.m,S.f),
+          type="l", lty=1, col=c("blue","red"), lwd=3,
+          ylim=c(0,1), xlim=c(0,100),
+          ylab="Survival probability", xlab="Age (years)" )
```

11. We can compute the expected lifetime as the area under these curves as a simple sum; recall that we have the survival curve evaluated at points 0.1, 0.2, ... 99.9, 100 years. So if we take the sum of these values of the survival function and multiply by 0.1, we get the area under the curve:

```
> ( EL.m <- sum(S.m)/10 )

[1] 74.84376

> ( EL.f <- sum(S.f)/10 )

[1] 79.40901
```

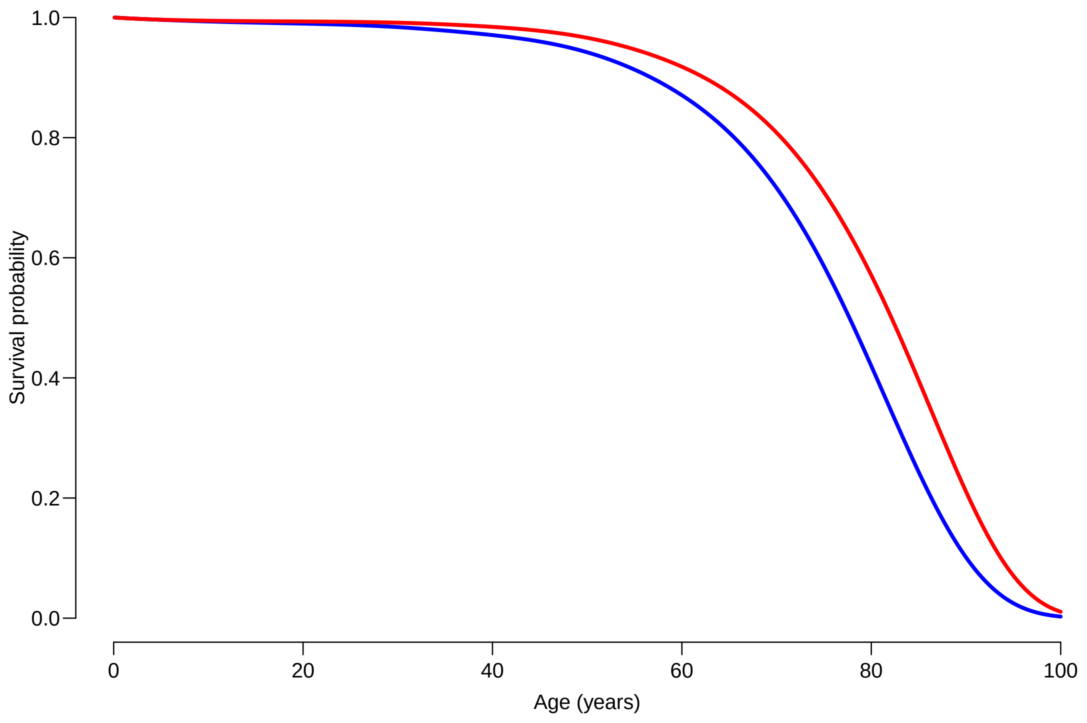

Figure 3.18: *Survival curves for the Danish population*

Formally, what we have done is to sum the values of the survival curves at the points 1/10, 2/10, . . . — because the survival function we computed is evaluated at the *end* of each interval. If we assume that each values corresponds to the midpoint of an interval, the first interval is really the interval (1/20;3/20), so what we are missing is the first 1/20 long interval so the calculation should really be (taking the survival function to be 1 in this interval):

```
> ( EL.m <- sum(S.m)/10+1/20 )
[1] 74.89376
> ( EL.f <- sum(S.f)/10+1/20 )
[1] 79.45901
```

12. We can make the same calculations for diabetes patients, but now we need the variable `lex.dur` rather than the variable `Y`:

```
> nd <- data.frame( A = mid.pt, lex.dur = 1 )
> s.m <- exp( -cumsum( ci.pred(r.m,newdata=nd)[,1]*1/10 ) )
> s.f <- exp( -cumsum( ci.pred(r.f,newdata=nd)[,1]*1/10 ) )
> ( el.m <- sum(s.m)/10 )
[1] 66.39297
> ( el.f <- sum(s.f)/10 )
[1] 71.82342
```

These expected lifetimes are not really meaningful; the formal interpretation is the expected lifetime of persons that through life experience mortality rates as persons with diabetes — notably persons that get diabetes at different times through life. So one essential assumption is that mortality among diabetes patients does not depend on duration of diabetes. . .

3.6.3 Life lost to diabetes

13. Thus wee see that diabetes patients have a smaller life expectancy than the general population, the differences are what is usually termed the years of life lost to diabetes:

```
> EL.m - el.m
[1] 8.500787
> EL.f - el.f
[1] 7.635589
```

14. These numbers are not really sensible, as we are pretending to look at persons with diabetes at birth, assuming that they have the same mortality as persons with diabetes diagnosed later in life. So we are ignoring the effect of diabetes duration on mortality.

15. Instead we might look at the life-expectancy of persons, say, 50 years old. To this end we need the *conditional* survival curves *given* that a person is alive at age 50. But these are just the survival curves from age 50, *divided* the probability of surviving till 50:

```
> S50.m <- S.m[500:1000]/S.m[500]
> S50.f <- S.f[500:1000]/S.f[500]
> s50.m <- s.m[500:1000]/s.m[500]
> s50.f <- s.f[500:1000]/s.f[500]
```

    With these conditional survival curves we can now compute the years of life lost to diabetes at age 50:

```
> ( EL50.m <- sum(S50.m)/10 )
[1] 27.40997
> ( EL50.f <- sum(S50.f)/10 )
[1] 31.07231
> ( el50.m <- sum(s50.m)/10 )
[1] 21.65778
> ( el50.f <- sum(s50.f)/10 )
[1] 25.51076
> EL50.m - el50.m
[1] 5.752186
> EL50.f - el50.f
[1] 5.561547
```

    So we see that years of life lost at age 50 is about 5 to 6 years for both sexes.

16. We can make this a bit more general by wrapping the calculation in a function that takes the conditioning age as input:

```
> YLL <- function( A )
+  {
+  SA.m <- S.m[(A*10):1000]/S.m[(A*10)]
+  SA.f <- S.f[(A*10):1000]/S.f[(A*10)]
+  sA.m <- s.m[(A*10):1000]/s.m[(A*10)]
+  sA.f <- s.f[(A*10):1000]/s.f[(A*10)]
+  ELA.m <- sum(SA.m)/10
+  ELA.f <- sum(SA.f)/10
+  elA.m <- sum(sA.m)/10
+  elA.f <- sum(sA.f)/10
+  c( Men=ELA.m - elA.m,
+   Women=ELA.f - elA.f )
+  }
> YLL( 50 )
```

```
     Men    Women
5.752186 5.561547
```

```
> YLL( 60 )
```

```
     Men    Women
4.100607 4.156742
```

Finally we can compute the number of years lost for ages 40 through 80:

```
> yll <- matrix( NA, 41, 2 )
> rownames( yll ) <- 40:80
> colnames( yll ) <- c("Men","Women")
> t(yll)
```

```
      40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67
Men   NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
Women NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
      68 69 70 71 72 73 74 75 76 77 78 79 80
Men   NA NA NA NA NA NA NA NA NA NA NA NA NA
Women NA NA NA NA NA NA NA NA NA NA NA NA NA
```

```
> for( a in 40:80 ) yll[a-39,] <- YLL(a)
> round( t(yll), 1 )
```

```
       40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
Men   7.1 6.9 6.8 6.7 6.6 6.4 6.3 6.2 6.0 5.9 5.8 5.6 5.5 5.3 5.1 5.0 4.8 4.6 4.5 4.3 4.1
Women 6.6 6.5 6.4 6.3 6.2 6.1 6.0 5.9 5.8 5.7 5.6 5.4 5.3 5.2 5.1 4.9 4.8 4.6 4.5 4.3 4.2
       61  62  63  64  65  66 67  68  69  70  71  72  73  74  75  76  77  78  79  80
Men   3.9 3.7 3.5 3.4 3.2 3.1  3 2.9 2.8 2.7 2.6 2.5 2.4 2.3 2.2 2.0 1.9 1.8 1.6 1.3
Women 4.0 3.8 3.6 3.5 3.3 3.1  3 2.8 2.7 2.5 2.4 2.3 2.1 2.0 1.8 1.7 1.5 1.4 1.3 1.2
```

Finally we can plot the years of life lost to diabetes for men and women at different ages:

```
> matplot( 40:80, yll,
+          type="l", lty=1, lwd=3, col=c("blue","red"),
+          ylim=c(0,7),
+          ylab="Years of life lost to DM", xlab="Age at evaluation" )
```

3.6.4 Changes in life lost to diabetes

17. In the previous calculations we only used the crude age-specific mortality rates for the entire period 1995–2009 (incl.).

    We now expand the models for the mortality rates with a term in calendar time. A first approximation could be just a linear effect of calendar time. Note that everything *not* mentioned in the `update` statement is kept from the original model.

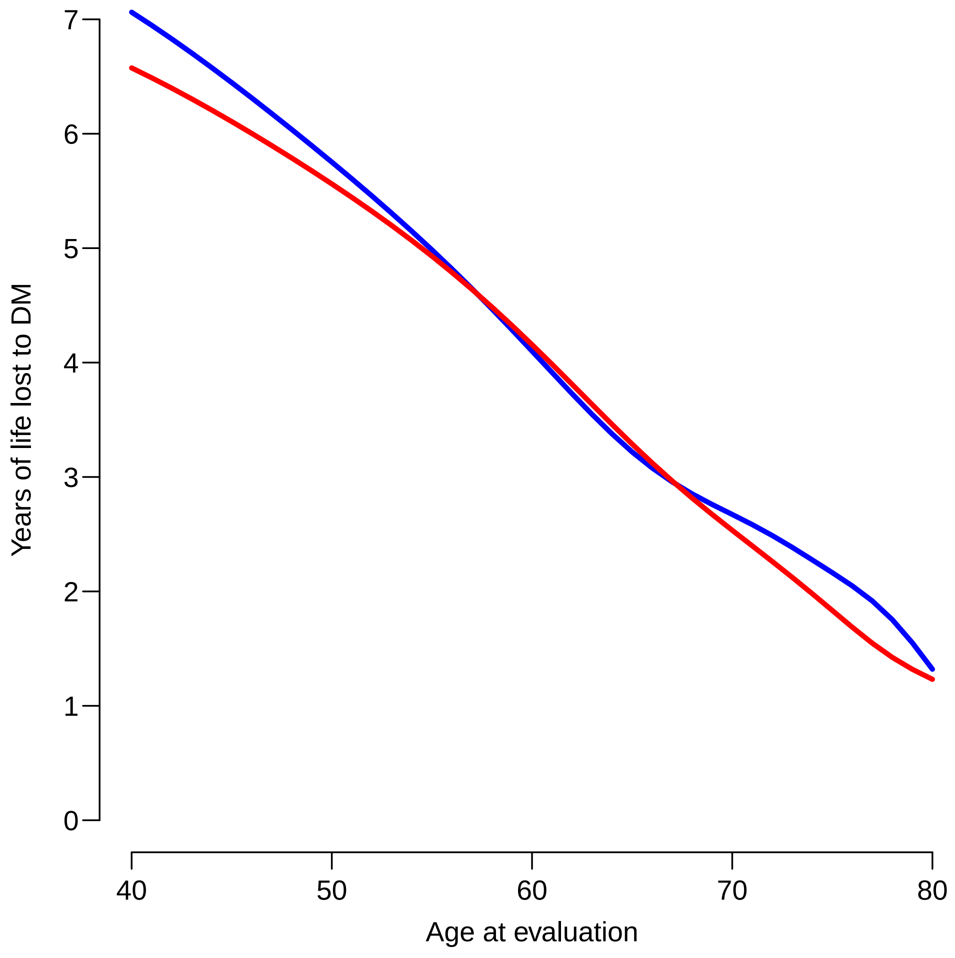

Figure 3.19: *Years of life lost to diabetes at different ages.*

```
> rp.m <- update( r.m, . ~ . + P )
> rp.f <- update( r.f, . ~ . + P )
> Rp.m <- update( R.m, . ~ . + P )
> Rp.f <- update( R.f, . ~ . + P )
```

18. In order to extract the trend by period we use the `subset` argument to `ci.exp` — that will give us the annual `relative` change in rates:

```
> ci.exp( rp.m, subset="P" )

  exp(Est.)      2.5%     97.5%
P 0.9636571 0.9493981 0.9781303
```

If we subtract 1 and multiply by 100 we get the annual trend in rates:

```
> chg <-
+ cbind( rbind( ci.exp(rp.m,subset="P"),
+               ci.exp(rp.f,subset="P") ),
+        rbind( ci.exp(Rp.m,subset="P"),
+               ci.exp(Rp.f,subset="P") ) )
> rownames( chg ) <- c("DM","Pop")
> colnames( chg ) <- paste( c("M:","","","F:","",""), colnames( chg ) )
> round( (chg-1)*100, 1 )
```

```
    M: exp(Est.)  2.5%  97.5% F: exp(Est.)  2.5%  97.5%
DM          -3.6  -5.1   -2.2         -2.3  -2.4   -2.3
Pop         -3.5  -5.1   -2.0         -1.7  -1.8   -1.7
```

We see that mortality rates among diabetes patients decline faster than in the general population, and faster among men; but also that there is virtually no difference in mortality decline between DM and no DM for men.

19. We start by predicting the mortality rates. Since we will need the mortality rates in ages 0-100, and for all combinations of age, sex, date and DM / population, we store the predictions in an array so that we can easily access them for further calculations:

```
> a.pt <- 0:999/10 + 1/20
> mort <- NArray( list( A = a.pt,
+                       P = seq(1995,2010,5),
+                     sex = c("M","W"),
+                    type = c("DM","Pop") ) )
> str( mort )
 logi [1:1000, 1:4, 1:2, 1:2] NA NA NA NA NA NA ...
 - attr(*, "dimnames")=List of 4
  ..$ A   : chr [1:1000] "0.05" "0.15" "0.25" "0.35" ...
  ..$ P   : chr [1:4] "1995" "2000" "2005" "2010"
  ..$ sex : chr [1:2] "M" "W"
  ..$ type: chr [1:2] "DM" "Pop"
```

Once the array is set up, we can fill in the predicted mortalities (in units of cases per 1 PY) at ages spaced 1/10 year apart:

```
> nd <- data.frame( A = a.pt,
+                   Y = 1,
+             lex.dur = 1 )
> for( ip in seq(1995,2010,5) )
+    {
+    nd$P <- ip
+    mort[,paste(ip),"M","DM"]  <- ci.pred( rp.m, newdata=nd )[,1]
+    mort[,paste(ip),"W","DM"]  <- ci.pred( rp.f, newdata=nd )[,1]
+    mort[,paste(ip),"M","Pop"] <- ci.pred( Rp.m, newdata=nd )[,1]
+    mort[,paste(ip),"W","Pop"] <- ci.pred( Rp.f, newdata=nd )[,1]
+    }
> round( ftable( mort[1:5+600,,,]*1000,
+                row.vars=c(4,1),
+                col.vars=c(3,2) ), 1 )
           sex    M                       W
           P    1995 2000 2005 2010 1995 2000 2005 2010
type A
DM   60.05      39.1 32.5 27.0 22.4 26.3 22.0 18.4 15.4
     60.15      39.4 32.7 27.2 22.6 26.5 22.2 18.5 15.5
     60.25      39.6 32.9 27.4 22.7 26.7 22.3 18.7 15.6
     60.35      39.9 33.2 27.6 22.9 26.9 22.5 18.8 15.7
     60.45      40.2 33.4 27.7 23.1 27.1 22.7 19.0 15.8
Pop  60.05      14.5 12.9 11.5 10.2  9.0  8.2  7.5  6.9
     60.15      14.7 13.0 11.6 10.3  9.0  8.3  7.6  6.9
     60.25      14.8 13.2 11.7 10.4  9.1  8.4  7.7  7.0
     60.35      14.9 13.3 11.8 10.5  9.2  8.4  7.7  7.1
     60.45      15.1 13.4 11.9 10.6  9.3  8.5  7.8  7.1
```

From this we can devise an array of the cumulative rates by simply summing along the age-dimension — using apply, but dividing by 10 because every interval is 1/10 year long:

```
> Mort <- apply( mort, 2:4, cumsum )/10
> str( Mort )
```

```
 num [1:1000, 1:4, 1:2, 1:2] 3.72e-05 7.48e-05 1.13e-04 1.51e-04 1.89e-04 ...
 - attr(*, "dimnames")=List of 4
  ..$ A    : chr [1:1000] "0.05" "0.15" "0.25" "0.35" ...
  ..$ P    : chr [1:4] "1995" "2000" "2005" "2010"
  ..$ sex : chr [1:2] "M" "W"
  ..$ type: chr [1:2] "DM" "Pop"
```

```
> round( ftable( Mort[200+1:5,,,],
+                row.vars=c(4,1),
+                col.vars=c(3,2) ), 5 )
```

```
          sex        M                                   W
          P       1995    2000    2005    2010    1995    2000    2005    2010
type A
DM   20.05     0.01800 0.01496 0.01243 0.01033 0.00999 0.00835 0.00697 0.00583
     20.15     0.01817 0.01510 0.01255 0.01043 0.01009 0.00843 0.00704 0.00589
     20.25     0.01835 0.01525 0.01267 0.01053 0.01019 0.00852 0.00712 0.00595
     20.35     0.01853 0.01540 0.01280 0.01064 0.01029 0.00860 0.00719 0.00601
     20.45     0.01872 0.01555 0.01292 0.01074 0.01040 0.00869 0.00726 0.00607
Pop  20.05     0.01250 0.01111 0.00988 0.00878 0.00768 0.00703 0.00644 0.00589
     20.15     0.01254 0.01115 0.00991 0.00881 0.00769 0.00704 0.00645 0.00590
     20.25     0.01258 0.01118 0.00994 0.00884 0.00770 0.00705 0.00646 0.00591
     20.35     0.01262 0.01122 0.00998 0.00887 0.00771 0.00706 0.00647 0.00592
     20.45     0.01267 0.01126 0.01001 0.00890 0.00772 0.00707 0.00648 0.00593
```

Note that this *only* gives an array of the same structure because the dimension we are applying a function over (`A`) is the first, and because the function applied returns a vector of the same length as input. The survival curve is then simply computed by taking $\exp(-\Lambda)$; we can plot the survival curves for men with and without diabetes for each of the 4 dates of evaluation:

```
> Surv <- exp( -Mort )
> matplot( a.pt, cbind( Surv[,,"M","DM"],
+                       Surv[,,"M","Pop"] ),
+          type="l", lwd=3, lty=1, col=heat.colors(6)[1:4] )
```

If we now want to compute the expected residual life from age 40, say, we need the *conditional* survival function from age 40:

```
> S40 <- Surv[400:1000,,,]/Surv[rep(400,601),,,]
> matplot( a.pt[400:1000], cbind( S40[,,"M","DM"],
+                                 S40[,,"M","Pop"] ),
+          type="l", lwd=3, lty=1, col=heat.colors(6)[1:4] )
```

The expected residual life time from age 40 is then the sum of these multiplied by the interval length:

```
> ERL40 <- apply( S40, 2:4, sum )/10
> str( ERL40 )
```

```
 num [1:4, 1:2, 1:2] 25.7 27.7 29.6 31.7 30.1 ...
 - attr(*, "dimnames")=List of 3
  ..$ P    : chr [1:4] "1995" "2000" "2005" "2010"
  ..$ sex : chr [1:2] "M" "W"
  ..$ type: chr [1:2] "DM" "Pop"
```

```
> ftable( ERL40 )
```

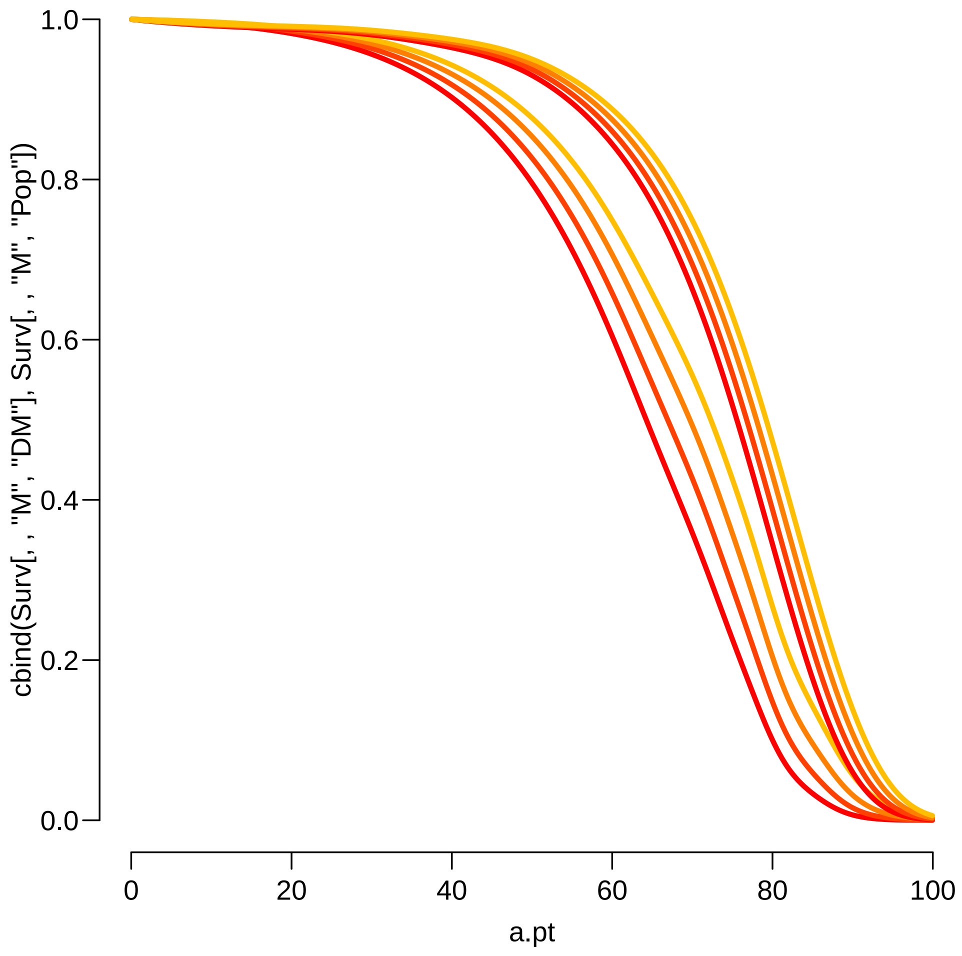

Figure 3.20: *Estimated survival curves for men, based on mortality rates as of 1 January 1995, 2000, 2005 and 2010, for the general population (upper curves) and the diabetic population (lower curves). Survival is increasing by calendar time.*

```
         type       DM      Pop
P    sex
1995 M        25.71040 34.54566
     W        30.14586 39.02936
2000 M        27.65551 35.67264
     W        32.11185 39.85356
2005 M        29.64429 36.80467
     W        34.09393 40.67512
2010 M        31.67231 37.94021
     W        36.08057 41.49283
```

We can expand this to show the years of life lost at age 40:

```
> ERL40 <- ERL40[,,c(1,2,2)]
> dimnames(ERL40)[[3]][3] <- "YLL"
> ERL40[,,"YLL"] <- ERL40[,,"Pop"] - ERL40[,,"DM"]
> ftable( ERL40, row.vars=2:1 )
         type        DM       Pop       YLL
sex P
M   1995      25.710395 34.545655  8.835260
    2000      27.655508 35.672643  8.017135
```

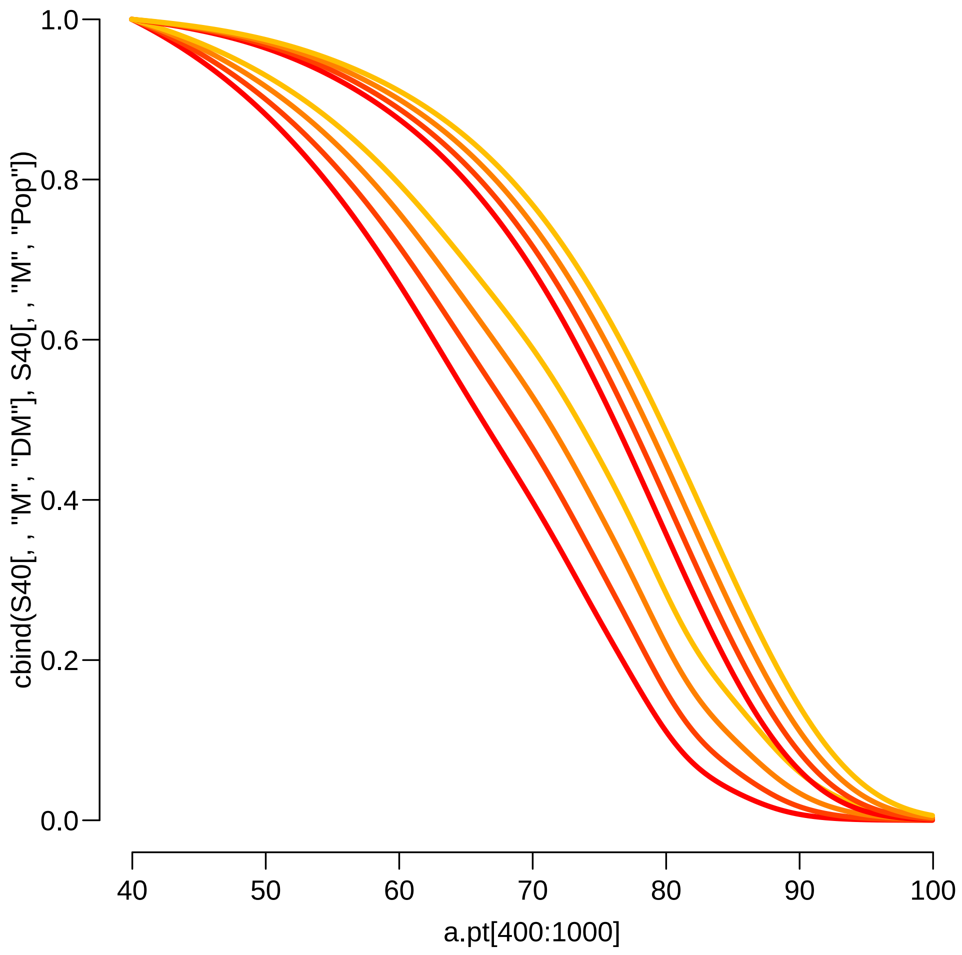

Figure 3.21: *Estimated survival curves for men,* given *survival till age 40, based on mortality rates as of 1 January 1995, 2000, 2005 and 2010, for the general population (upper curves) and the diabetic population (lower curves). Survival is increasing by calendar time.*

```
      2005      29.644288 36.804673  7.160385
      2010      31.672307 37.940215  6.267908
   W  1995      30.145865 39.029357  8.883492
      2000      32.111851 39.853562  7.741710
      2005      34.093932 40.675116  6.581185
      2010      36.080569 41.492830  5.412262
```

20. If we want a curve of life lost to diabetes as a function of age, we should formalize what we do above; when we compute the residual life time, we compute the integral from a given age to infinity (well, 100 years), and then divide by the survival function at the given age. We can make this in one go for all ages by cumulating the survival function from the top of the age-scale. So wee need a function that cumulates from the end of a vector rather than from the beginning of a vector (`rev` is the function that puts a vector in reverse order):

```
> musmuc <- function(x) rev(cumsum(rev(x)))
```

We the apply this to `Surv` along the age-scale, divide by the interval length, and finally divide by `Surv`, to rescale to the integrals of the *conditional* survival functions:

```
> ERL <- ( apply( Surv, 2:4, musmuc )/10 ) / Surv
> LL <- ERL[,,,c(1,2,2)]
> dimnames( LL )[[4]][3] <- "YLL"
> LL[,,,3] <- LL[,,,"Pop"]-LL[,,,"DM"]
> str( LL )
```

```
 num [1:1000, 1:4, 1:2, 1:3] 62 61.9 61.8 61.7 61.6 ...
 - attr(*, "dimnames")=List of 4
  ..$ A   : chr [1:1000] "0.05" "0.15" "0.25" "0.35" ...
  ..$ P   : chr [1:4] "1995" "2000" "2005" "2010"
  ..$ sex : chr [1:2] "M" "W"
  ..$ type: chr [1:3] "DM" "Pop" "YLL"
```

The years of life lost at different ages can then be plotted for each of the years where we did the evaluation:

```
> matplot( a.pt, cbind( LL[,,"M","YLL"], LL[,,"W","YLL"] ),
+          type="l", lty=1, lwd=1:4, col=rep(c("blue","red"),each=4),
+          xlim=c(40,80), ylim=c(0,10), yaxs="i",
+          xlab="Age alive", ylab="Years of life lost to DM" )
> abline( h=1:10, v=4:8*10, col=gray(0.8) )
```

We see in figure 3.22 that the decrease in the years of life lost to diabetes has been more dramatic among women than among men, consonant with the fact that the mortality rates among DM men have been dropping at about the same rate as among the rest of the population, whereas the mortality among DM women have been decreasing considerably faster

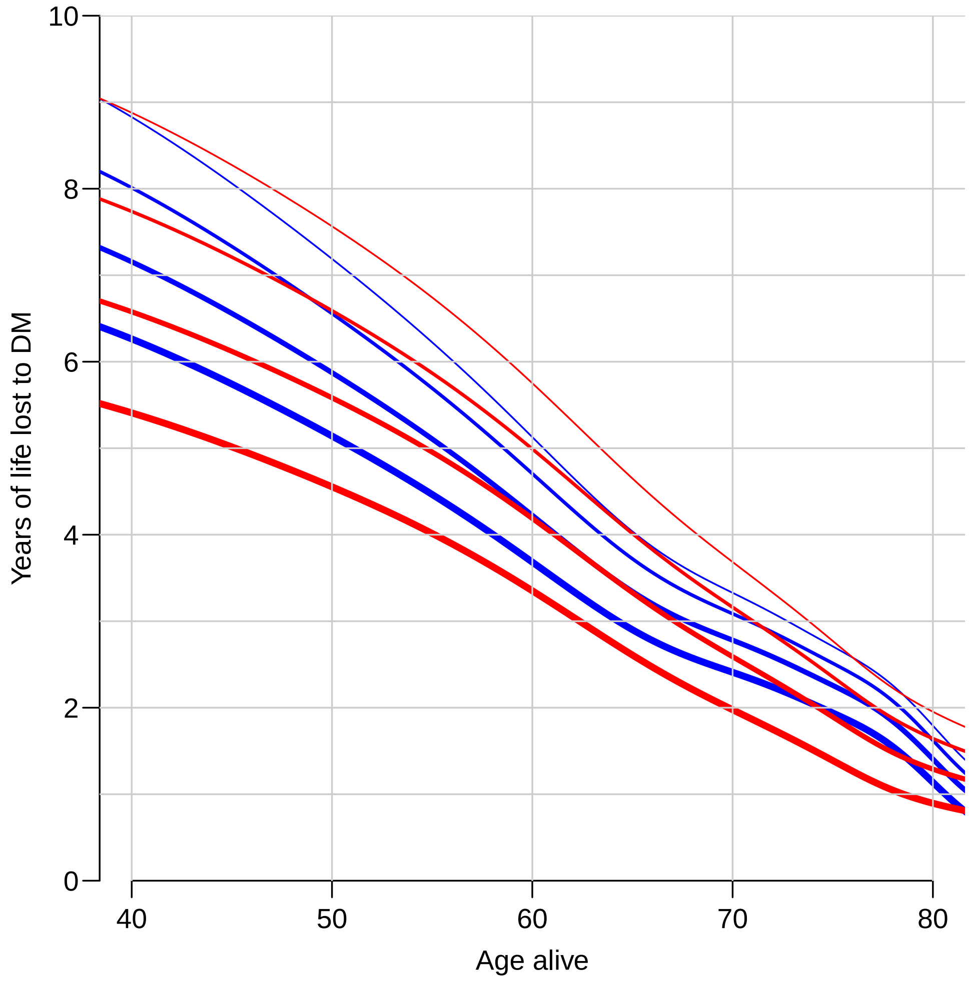

Figure 3.22: *Years of life lost to DM at different ages for men (blue) and women (red) evaluated on the basis of mortality rates at 1 January 1995, 2000, 2005, 2010 (thin to thick).*