Modern Demographic Methods in Epidemiology

Computer practicals

1 - 3 June 2010

St. Andrews

http://www.biostat.ku.dk/~bxc/AdvCoh/StAn-2010

(compiled Monday 31st May, 2010, 22:57)

Bendix Carstensen Steno Diabetes Center, Gentofte, Denmark
& Department of Biostatistics, University of Copenhagen
bxc@steno.dk
http://staff.pubhealth.ku.dk/~bxc/

EINLEITUNG

IN DIE

THEORIE

DER

BEVÖLKERUNGSSTATISTIK

VON

W. LEXIS

DR. DER STAATSWISSENSCHAFTEN UND DER PHILOSOPHIE,
O. PROFESSOR DER STATISTIK IN DORPAT.

STRASSBURG

KARL J. TRÜBNER

1875.

Contents

Course program

Tuesday 1st June 2010

09:00 – 09:15	**Welcome**
09:15 – 10:15	**L**: Introduction to time-to-event data. Relationships between rates and probabilities. Life-tables and likelihood for rates. Estimating rates / Estimating the survivor function.
10:15 – 10:45	**Morning Tea**
10:45 – 12:00	**P**: Calculation of rates, RR and RD.
12:00 – 13:00	**Lunch**
13:00 – 14:15	**L**: Classical estimation of survival curve. The Cox model. The Lexis diagram: Representation of follow-up on multiple timesacles Representation of follow-up data using `Lexis` in the `Epi` package.
14:15 – 16:00	**P**: Simple analysis of Estonian stroke data Estonian stroke study using the Cox model.

Wednesday 2nd June 2010

09:00 – 09:30	Recap of Tuesday's practicals.
09:30 – 10:15	**L**: Modelling the baseline hazard – it's just another variable. Who need the Cox model anyway?
10:15 – 10:45	**Coffee**
10:45 – 12:00	**P**: Cox model and time-splitting using Estonian stroke data Estonian stroke study using with Poisson, comparing with Cox.
12:00 – 13:00	**Lunch**
13:00 – 13:45	**L**: SMR and Poisson-modelling.
13:45 – 14:15	**Afternoon Tea**
14:15 – 16:00	**P**: Time-splitting and SMR: Thorotrast

Thursday 3rd June 2010

09:00 – 09:30	Recap of Wednesday's practicals.
09:30 – 10:15	**L**: Interaction and timescales.
10:15 – 10:45	**Coffee**
10:45 – 12:00	**P**: Renal complications.
12:00 – 13:00	**Lunch**
13:00 – 13:45	**L**: Multi-state models and competing risks (little boxes).
13:45 – 14:15	**Afternoon Tea**
14:15 – 16:00	**P**: Competing risks: The thorotrast study.
16:00 – 16:30	Recap of the day's practicals.
16:30 – 16:45	Evaluation, feedback, closing remarks and farewell.

Chapter 1

Fundamental relations in survival

1.1 Concepts in survival studies

This section briefly summarizes relations between various quantities used in analysis of follow-up studies. They are used all the time in the analysis and reporting of results. Hence it is important to be familiar with all of them.

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:

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

Density function of death times:

$$f(t) = \mathrm{P}\left\{\text{death in } (t, t + \mathrm{d}t)\right\} / \mathrm{d}t$$

Intensity:

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

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

Relationships between terms:

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

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

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

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

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

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

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

Likelihood from one person:

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

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

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

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

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

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

Log-likelihood for several perons. Adding log-likelihoods from a group of persons (assuming identical and constant rates) gives:

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

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

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

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

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

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

A linear model for the log-rate, $\log(\lambda) = X\beta$ implies that $\lambda Y = \exp\big(\log(\lambda) + \log(Y)\big) = \exp\big(X\beta + \log(Y)\big)$. Therefore, in order to get a linear model we must require that $\log(Y)$ appear as a variable in the model for the log-rate with the regression coefficient fixed to 1, a so-called offset-term in the linear predictor.

Competing risks: If there is more than one cause of death, occurring with (cause-specific) rates λ_1, λ_2, λ_3, the survival function is:

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

The probability of dying from cause 1 before time t is:

$$\int_0^t \lambda_1(u)S(u)\,\mathrm{d}u \neq 1 - \exp\left(-\int_0^t \lambda_1(u)\,\mathrm{d}u\right)$$

The second part of the term on the right hand side (sometimes referred to as the "cause-specific survival") does not have any probabilistic interpretation.

Chapter 2

Introduction to computing

2.1 Getting the R software

You can obtain a copy of R from the R-homepage, http://www.r-project.org/, use the link to CRAN (pronounced: see-ran), the Comprehensive R Archive Network, http://cran.r-project.org/mirrors.html where you choose a mirror (i.e. a server somewhere in the world. After this you do:

Download and Install R → Windows → base → Download R 2.11.0 for Windows

This gives you an executable file (self-extracting zip) which you can run or download and run. Anyway, this will install R for you. You can decide which folder, so if you have trouble installing R in the default place `c:\Program Files\R` because of restrictions on your computer, just choose another folder where you *do* have access rights — R will work anyway.

Once this is completed, you should fire up R by clicking on the icon. You then must install the `Epi` package: click on "Packages" → "Install paksage(s)" and you will be asked to choose a mirror (a server somewhere in the world where R is available) and the to choose "Epi" among the some 2300 other packages.

You can also install the `Epi`-package by writing:

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

It is important that the first letter in "Epi" is capitalized, and also that the name is in quotes. This will cause R to locate the package and download and install it for you.

2.2 What is where in the R installation?

When you have installed R on your computer, you will have a folder called `R`, where some of the subfolders are:

```
R-2.11.0
- bin
- doc
- etc
  - rgb.txt
  - Rdevga
  - repositories
  - Makeconf
  - Rconsole
  - Rprofile.site
- include
- library
  - base
  - boot
```

```
  - class
  - cluster
  - codetools
  - datasets
  - foreign
  - graphics
  - grDevices
  - grid
  - KernSmooth
  - lattice
  - ...
- modules
- share
- src
- Tcl
```

In the folder `etc` you will be particularly interested in the file `Rconsole`. If you open this file in Tinn, Notepad or some similar editor you will see that it specifies how your R-console looks: Fonts colors etc. By changing this you can customize how R looks when you start it.

The file `Rprofile` contains R-commands that are executed whenever you start R, so here you can insert various commands that loads packages that you always use, e.g. a command like `library(Epi)`.

2.2.1 Non-official packages

Some R-packages are not on CRAN, but available as a `.zip`-file. If you want to install such packages you should get the `.zip`-file and unpack it in the folder `c:\Program Files\R\R-2.11.0\library` — remember to tick the box "Use folder names". If you take a look in the folder `c:\Program Files\R\R-2.11.0\library` you will see that there is a sub-folder for each installed package. This is the general idea behnd how R installs packages: The program code and the help files are just stuffed into a special folder structure.

A simple and safe approach is to use the menu in R:"Packages" → "Install paksage(s) from local zip files...".

2.3 Software versions

For this course you should have the latest version of R installed, i.e. 2.11.0.

The exercises require that you have access to the `Epi` package, version 1.1.14.

Once you installed it, you should check that you have the right version by starting R and say:

```
library(Epi)
library(help=Epi)
installed.packages()["Epi","Version"]
```

2.4 The script editor

The practicals are fairly detailed w.r.t. the R-commands that will produce parts of the solutions.

You are strongly encouraged to write your R commands in a script file using either Tinn-R (fancy, with syntax highlighting) or the built-in script editor in R — File → New script (or Open script) (basic, primitive and no-nonsense). You can submit part of a script by selecting the relevant lines of code and pressing Ctrl-R.

If noting is selected the current line is submitted and the cursor moved to the next line. This way you can aubmit muliple lines of R-code by repeatedly pressing Ctrl-R.

2.5 Help on R-commands you need

To benefit maximally from the practicals you should consult the help pages for the R functions that you use. **All of them!**. You can use the `help.search()` function. For example try:

```
?plot
help.search("Cox model")
```

Alternatively you may start the html-help pages which have links between functions, try:

```
help.start()
```

This has nice links between functions so you can surf around and get wiser. Keep it open all the time.

Finally you can run the examples from the help-page for a function by typing (e.g for the function `stat.table` from the Epi package):

```
example(stat.table)
```

These are the ways you will have to get help once the course is over, so you better get used to it. It will have the side effect that you will read a lot of things irrelevant to your task, but also that you occasionally will learn something useful you never anticipated!

2.6 Data sets

Data sets for the practials are either accessible as files on the website `www.biotsta.ku.dk/~bxc/AdvCoh/data`, or as built-in datasets in the `Epi` package. The latter loaded into the workspace by:

```
data( thoro )
```

The former are read according to the instructions in the pracaticals, mostly using `read.table()`.

If you want to read in your own datasets you may need the package `foreign`, which has functions to read from SAS, Stata and SPSS. You do not need to install `foreign`, because it is a part of the default R installation, but you need to attach it by `library(foreign)` before you can use it.

2.7 Solutions

The last chapter of this document contains a complete set of R-scripts and the results of running these, including the graphs. The R-files will also be available on the course web-site. `http://staff.pubhealth.ku.dk/~bxc/AdvCoh/StAn-2010`. They may be useful for you to consult, but you will get more out of the course if you try on your own first.

There is no guarantee that you will not be able to find a smarter solutions than the ones provided.

Chapter 3

Practical exercises

3.1 Calculation of rates, RR and RD

Recall that the standard error of log-rate: $1/\sqrt{D}$, 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. Suppose you have 15 events during 5532 person-years. Now use R as a simple desk calculator to derive the rate and a confidence interval:

```
> library( Epi )

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

[1] 0.002711497 0.001634654 0.004497720

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

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

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

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

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

Deviance Residuals:
[1]  0

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

```
(Dispersion parameter for poisson family taken to be 1)

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

Number of Fisher Scoring iterations: 3
```

What does the parameters of this model mean?

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

```
> ci.lin( mm, E=T)

             Estimate    StdErr         z P   exp(Est.)        2.5%       97.5%
(Intercept) -5.910254 0.2581989 -22.89032 0 0.002711497 0.001634669 0.004497678

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

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

4. There is an alternative way to fit a Poisson model, using the rates a the Poisson response, and the person-years as weights instead:

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

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

Verify that this give the same results as above.

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

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

What is the meaning of the intercept in this model?

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

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

```
> ci.lin( ma )

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

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

   Estimate        2.5%       97.5%
0.002711497 0.001339315 0.004083678
```

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

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.

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

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

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

```
> Dx <- rep(0:1,c(5517,15))
> Yx <- rep(1,5532)
```

10. 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 ratio of the rates, RR, is:

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

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

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

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

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

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

[1] 2.158980 1.153146 4.042153

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

     Estimate     2.5%    97.5%
[1,] 2.158980 1.153160 4.042106
```

11. This can also be achieved in 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?

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

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

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

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

    Use thsi formual to compute the rate difference and a 95% confidence interval for it.

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

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

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

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

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

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

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

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

> round( ci.lin( mm, ctr.mat=CM, E=T), 3 )

           Estimate StdErr       z     P exp(Est.)  2.5% 97.5%
rate 0       -5.910  0.258 -22.890 0.000     0.003 0.002 0.004
rate 1       -5.141  0.189 -27.202 0.000     0.006 0.004 0.008
RR 1 vs. 0    0.770  0.320   2.405 0.016     2.159 1.153 4.042
```

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 estiamtes are via the column-names of the contrast matrix.

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

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

> round( ci.lin( ma, ctr.mat=CM ), 3 )

           Estimate StdErr     z     P  2.5% 97.5%
rate 0        0.003  0.001 3.873 0.000 0.001 0.004
rate 1        0.006  0.001 5.292 0.000 0.004 0.008
RD 1 vs. 0    0.003  0.001 2.400 0.016 0.001 0.006
```

3.2 Simple analysis of Estonian stroke data

```
> library(Epi)
```

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

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

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

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

   Calculate an id variable in the dataframe.

   ```
   > stroke <- read.table( url("http://www.biostat.ku.dk/~bxc/AdvCoh/data/stroke.csv"),
   +                       sep=";", header=TRUE, na.strings="." )
   > stroke$id <- 1:nrow(stroke)
   > str( stroke )
   > head( stroke )
   ```

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

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

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

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

   How many deaths occurred after 1 January 1996?

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

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

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

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

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

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

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

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

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

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

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

8. Plot the result as 4 curves.

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

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

   Do the hazards look proportional?

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

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

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

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

    What do you conclude?

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

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

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

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

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

```
> summary( Lst )
```

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

    ```
    > plot( Lst )
    ```

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

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

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

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

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

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

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

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

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

3.3 Cox model and time-splitting using Estonian stroke data

```
> library(Epi)
```

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

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

Alternatively you must read the data afresh, transform etc.

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

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

   Are there any differences?

   What is the underlying time scale used here?

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

8. Now take a look at the estimated coefficients:

   ```
   > coef( mP )
   ```

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

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

   Are there any major differences?

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

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

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

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

    Compare the parameter estimates with the previous models.

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

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

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

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

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

    Verify that you get the same estimates:

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

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

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

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

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

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

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

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

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

3.4 Time-splitting and SMR: Thorotrast

In the period 1935–50 a contrast medium called Thorotrast was used for cerebral angiography (X-ray imaging of the brain). This contrast medium contained ^{232}Th, thorium. It turns out that thorium is not excreted from the body, it is permanently deposited, some 60% in the liver, 20% in the spleen and some 10% in the bone marrow, and a very small fraction in other organs.

Thorium is an α-emitting radionuclide, i.e. it emits α-rays (i.e. He-nuclei) which is ionizing, but not particularly penetrating; it only penetrates 2–3 cell-layers. The half-life of ^{232}Th is 1.4×10^{10} years, so the patients that have been injected with Thorotrast exposed are exposed to a constant, small α-radiation for life.

In the study is 990 Thorotrast patients who had a cerebral angiography in the period 1935–50 and 1480 controls who have had a cerebral angiography in the period 1946–63, on similar indications as the Thorotrast patients, but with another contrast medium.

Persons undergoing cerebral angiography are in may cases seriously ill, they are suspected of cerebral malformations or tumors, so both the Thorotrast group and the control group have very high mortality rates, and a pattern of causes of death that differ substantially from the general population. Especially during the first year after diagnosis there is a very high mortality among the patients, which is entirely associated to the conditions that have lead to to the cerebral angiography. Therefore, follow-up of both Thorotrast patients and control patients is only relevant from one year after angiography.

3.4.1 The data sets

There are two sources of data for this exercise, the cohort data and the mortality rates from Denmark. Both are available as internal datasets in the `Epi` package. The dataset with the cohort is loaded by `data(thoro)`; and you can get an explanation by typing `?thoro`. The relevant cause-specific mortality figures for Denmark are loaded by `data(gmortDK)`. As well as overall mortality (`rt`), the file also contains the mortality for all cancers, etc. For a complete explanation, use `?gmortDK`.

1. First load the Thorotrast dataset from the `Epi` package:

```
> library( Epi )
> data( thoro )
```

   Then take a look at the cohort data by e.g. `head(thoro)` and/or `summary(thoro)`.

   Note that the date variables are of class "`Date`", i.e. they are stored as days since 1 January 1970, you may want to try for example:

```
> bd <- thoro$birthdat[1:5]
> bd
> as.numeric(bd)
> cal.yr(bd)
> (cal.yr(bd)-1970)*365.25
```

   Don't forget to use `?cal.yr`.

   The most convenient is to use `cal.yr` on the dataset, which will convert all date variables in the dataset to `cal.yr`:

```
> thoro <- cal.yr( thoro )
> str( thoro )
```

   You would also want a decent annotation of the primary effect variable `contrast`, so define it as a factor with appropriate labels:

```
> thoro$contrast <- factor( thoro$contrast, labels=c("Thoro","Contr") )
```

2. Declare the follow-up timescales for the dataset, using the `Lexis` command, e.g.:

```
> thL <- Lexis( entry = list(per=injecdat,
+                            age=injecdat-birthdat,
+                            tfi=0),
+               exit = list(per=exitdat),
+        exit.status = (exitstat==1),
+                 id = id,
+               data = thoro )
> str( thL )
> head( thL )
```

   Explain the meaning of the variables added by `Lexis`, and how they relate to the data variables.

   Get an overview of the `Lexis` object (the dataset) using `summary`:

```
> summary( thL )
```

3. Note that `thL` has got class "`Lexis`". Now make a Lexis diagram using the defined object `thL`:

```
>  plot( thL )
```

This really uses the function `plot.Lexis` to make the plot. Use `?plot.Lexis` to find the available options for this command, and try to improve the plots with indications of the exit-status of the persons in the cohort.

4. Try to make the life-lines of Thorotrast patients and controls different color. Hint: use the indexing facility for a character vector with color names: `col=c("red","blue")[contrast]`.

3.4.2 Rates

5. The first analytical task is to look at overall mortality by contrast medium (`contrast`).

   Tabulate the number of deaths and person-years from the study by group using `stat.table()`:

```
> stat.table( contrast,
+             list( D=sum(lex.Xst),
+                   Y=sum(lex.dur),
+                rate=ratio(lex.Xst,lex.dur,1000) ),
+             data=thL )
```

6. However, we want to to start follow-up one year after angiography (`injecdat`) and exclude persons without follow-up beyond one year. Note you need to redefine the entry date to the study on all three timescales:

```
> thL <- Lexis( entry = list(per=injecdat+1,
+                            age=injecdat+1-birthdat,
+                            tfi=1),
+               exit = list(per=exitdat),
+         exit.status = exitstat==1,
+                id = id,
+               data = subset( thoro, (injecdat+1) < exitdat ) )
> summary( thL )
```

   Use the new dataframe to produce the same table by `stat.table`.

7. Compute 95% confidence intervals for the overall rates and for the rate-ratio between the two groups.

   Try to do this also by fitting a Poisson-model with `glm` and subsequently use `ci.lin` to compute the rates and the RR.

8. (*Optional*) It is well known that Thorotrast causes liver cancer; try to tabulate the number of liver cancers by patient group.

   One may argue that the deaths caused by liver cancer should not be counted, so repeat the mortality calculations above after censoring patients at date of liver cancer diagnosis.

9. An important question is how the mortality rates in the two groups varies with time since injection.

   In order to see if the mortality changes the same way in the two groups, split the follow-up time by time since injection using `splitLexis`. Split follow-up in intervals of 1 year during the first years say 5 years from time since angiography and subsequently every 5 years, e.g. by:

```
>  thx <- splitLexis(thL, "tfi", breaks=c(0:4,seq(5,55,5)) )
```

   Take a look at the split data for example by listing the observations with `id==1`, (use for example `subset(thx,id==1)`). Make sure that you understand how they relate to the original record.

10. Compute mortality rates in each interval separately for the two groups, using `stat.table`. How do the rates in the two groups of patients behave by time since injection?

11. Try to show it in a graph. Hint: Assign the result of `stat.table()` to an object and take a look at the `dimnames()` of this object. Then use `matplot()` to plot the two sets of rates by taking appropriate subsets of the object.

12. The next step is to model the mortality and the rate-ratio in the two groups by a smooth function. Therefore, we split the follow-up in small intervals, and fit a model using natural splines for the mortality as a function of time.

```
> thxx <- splitLexis(thL, "tfi", breaks=c(0,seq(1,100,0.5)) )
> dim( thxx )
> summary( thxx )
```

    In order to do so we need a *quantitative* variable for each of the intervals, giving the midpoints of the intervals, as well as a failure indicator:

```
> thxx$m.tfi <- timeBand( thxx, "tfi", "middle" )
```

    Remember to consult the help page for `timeBand`.

13. A Poisson model can now be used to fit a model for the mortality as a function of time since injection using natural splines. The point is to fit a separate mortality curve for each contrast group as a function of time since injection:

    - Splines are available in the `splines` package, which is loaded by `library(splines)`.
    - The definition of splines requires the definition of internal knots (`knots`) and boundary knots (`Boundary.knots`, abbreviated `Bo`). These are most conveniently defined before the splines.
    - If you want separate splines for each level of `contrast`, use the interaction operator ":".
    - To get the parametrization as log-rates in each of the groups we remove the overall intercept from the model by "`-1`", and include an intercept with the splines by `intercept=TRUE` (or `i=T`).
    - Finally we scale the person-years by 1000, in order to get results in rates per 1000 person-years.

    Now, put these points together in the model specification:

```
> library( splines )
> kn <- c(4,8,seq(10,40,10))
> bk <- c(1,50)
> m1 <- glm( lex.Xst ~ -1 + contrast:ns( m.tfi, knots=kn, Bo=bk, i=T ) +
+                       offset( log(lex.dur/1000) ),
+            family=poisson, data=thxx )
```

    Note that we use the midpoint `m.tfi` as defined above as the regression variable in the model, and we deliberately omit the intercept (`-1`), because we want the two spline terms (one for controls and one for Thorotrast) to carry the rate dimension.

14. How would you include (current) age in the model?

15. Construct a contrast matrix to multiply with (some of) the coefficients of the model, so that you get the estimated mortality rates at a set of points between 1 and 40 years, say.

You can extract the parameters and multiply them with the contrast matrix in one go by using the facilities of `ci.lin` — remember `?ci.lin`. This will give you estimated mortalities at each of the time-points in `tpt`.

First try to see what the parameters are called using `ci.lin`, then see which ones you extract (and in which order!) when using the `subset` parameter:

```
>   ci.lin( m1 )[,1:2]
>   ci.lin( m1, subset="Thoro:ns" )[,1:2]
>   ci.lin( m1, subset="Contr:ns" )[,1:2]
>   ci.lin( m1, subset=c(":ns") )[,1:2]
>   ci.lin( m1, subset=c("Thoro:ns","Contr:ns") )[,1:2]
```

This can be used by pre-multiplying a matrix to the parameters to get estimates of the rates at a number of points, using the `ctr.mat=` argument (`con`trast `mat`rix):

```
>   tpt <- seq(1,40,0.5)
>   CM <- ns( tpt, knots=kn, Bo=bk, i=T )
>   mort1 <- ci.lin( m1, ctr.mat=CM, subset="Thoro:ns", Exp=TRUE )
>   mort2 <- ci.lin( m1, ctr.mat=CM, subset="Contr:ns", Exp=TRUE )
```

Because the contrast matrix `CM` is constructed using the `ns` with `tpt` as argument, the result `ci.lin` will be log-rates estimated at each of the time points in `tpt`, and transformed to rates by the argument `Exp=TRUE`. Now, plot the two sets of estimated mortality rates as nice curves with confidence intervals, as a function of `tpt`.

16. Use the contrast matrix to construct estimates of the rate-ratio between the groups at the same time points. You can use the relevant contrast matrix to multiply on these parameters to get the rate-ratio between the two patient groups:

```
>   RR <- ci.lin( m1, ctr.mat=cbind(CM,-CM), subset=c("Thoro:ns","Contr:ns"), Exp=TRUE )
```

3.4.3 SMR

The follow-up of the two groups of patients are in very different time periods and they have differing age-distributions. Therefore it is desirable to control for age and calendar time. This could be done by making an internal comparison of the two contrast groups controlled for age, sex, and calendar period. However, because of the different calendar periods of follow-up, some information would be lost; the effect of calendar time would be almost perfectly confounded with group. Instead, the comparison can be partly standardized for age, sex, and period, using SMR, i.e. by modelling the mortality *relative* to the population mortality, and assuming that this does not depend on calendar time.

17. Explain why the latter assumption is crucial.

18. The Danish mortality figures are in the dataframe `gmortDK`. Load it by `data(gmortDK)` and inspect it using `?gmortDK`. In order to be able to match up the Danish population mortality rates to the follow-up data these must first be split by current age and calendar time. The names and coding of the age and period variables must be chosen so that they are the same in `gmortDK` and in the split cohort data.

19. Split the dataset, now also along current age and period using cutpoints that correspond to those from the population data; note the splitting is successive; each split is taking the previously split dataset as input:

```
>   tha   <- splitLexis(thL , "age", breaks=seq(0,90,5) )
>   thap  <- splitLexis(tha , "per", breaks=seq(1938,2038,5) )
>   thapt <- splitLexis(thap, "tfi", breaks=seq(0,55,0.5) )
```

20. Unlike other packages there is no need in R to sort the dataframe by variables we merge on, or to name them explicitly — R will merge on all variables common to the two dataframes, and only include records in the result that have contributions from both dataframes.

    But you must make sure that variables have common names, so define `agr` and `pgr` in the cohort data:

```
>   data( gmortDK )
>   names( gmortDK )
>   thapt$agr <- timeBand(thapt, "age", "left")
>   thapt$pgr <- timeBand(thapt, "per", "left")
```

    It is important to make this exercise, because the time-scale variables `age` and `per` will not be equal to the left endpoint of the age/period interval for all units in the dataset. (Why?).

21. Then make `pgr` in the population mortality data match the coding in the cohort data:

```
>   gmortDK$pgr <- gmortDK$per + 1900
```

    Now you can merge the the population data with the follow-up data on the variables `agr`, `pgr` and `sex` (only taking the relevant columns from `gmortDK`):

```
>   th1ap <- merge(thapt, gmortDK[,c("agr","pgr","sex","rt")],
+                  by=c("agr","pgr","sex"))
```

22. The variable `rt` from `gmortDK` has the population mortality rate in cases per 1000 person-years. Multiply this with the person-years (`lex.dur`) to form the expected number of cases, `E`, say.

23. Compute the observed and expected number of cases as well as the ratio (SMR) by group using `stat.table`. Further tabulate this by time since injection.

24. Now use the log of `E` as offset-variable to estimate in a model where the SMR in each of the two groups of patients are assumed to depend smoothly on time since injection. Plot the SMR for each of the groups, and the ratio of SMRs as a function of time since injection. (This is parallel to what you did with the rates).

25. Do the ratios of SMRs differ substantially from the rate ratios obtained without using the reference rates?

3.5 Renal complications: Time-dependent variables and multiple states

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

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

You can find a `.pdf`-version of the paper on the course homepage.

Table 3.1: *Variables in* `renal.dta`.

`id`	Patient id
`sex`	1=male, 2=female
`dob`	Date of birth
`doe`	Date of entry into the study (2.5 years after NRA)
`dor`	Date of remission. Missing if no remission has occurred
`dox`	Date of exit from study
`event`	Exit status: 1,2,3=event (end stage renal disease, ESRD), 0=censored

3.5.1 The renal failure dataset

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

1. The dataset is in Stata-format. Read the dataset using `read.dta` from the `foreign` package, and take a look at the first 20 records:

```
> library( foreign )
> renal <- read.dta( "../data/renal.dta" )
> head( renal )
```

2. First do a simple survival curve for the cohort. The event we are interested in is given by codes 1,2 or 3 in the variable `event`

   The function `Surv()` defines a survival object, used as the response variable in a survival analysis. If called with two arguments the first is the survival time and the second the event indicator. If called with three arguments, the first is the time of entry, the second is the time of exit and the third the event indicator.

```
> library( survival )
> sf <- survfit( Surv( dox-doe, event>0 ) ~ 1, data=renal )
> plot( sf )
```

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

```
> library( Epi )
> Lr <- Lexis( entry = list( per=doe,
+                            age=doe-dob,
+                            tfi=0 ),
+              exit = list( per=dox ),
+       exit.status = factor( event>0, labels=c("NRA","ESRD") ),
+              data = renal )
> str( Lr )
> summary( Lr, scale=1000 )
```

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

```
> plot( Lr, col="black", lwd=3 )
> subset( Lr, age<0 )
```

5. Correct the data, e.g. by:

```
> Lr <- transform( Lr, age=age+100*(dob>2000), dob=dob-100*(dob>2000) )
> str( Lr )
> subset( Lr, id==586 )

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

6. Now try to produce a slightly more fancy Lexis diagram:

```
> par( mai=c(3,3,1,1)/4, mgp=c(3,1,0)/1.6 )
> plot( Lr, 1:2, col=c("blue","red")[Lr$sex], lwd=3, grid=0:20*5,
+       xlim=c(1970,2010), ylim=c(0,80), xaxs="i", yaxs="i", las=1 )
```

7. Make a Cox-regression analysis with the variables sex and age at entry into the study. Give the hazard ratio between males and females and between two persons who differ 10 years in age at entry. Give the 95% confidence intervals for this as well.

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

8. Show the estimated survival function for a male aged 50.

```
> plot( survfit( mc, newdata=data.frame(age=50,sex=1) ), col="black", lwd=3 )
```

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

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

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

   This is accomplished using the `cutLexis` function on the just defined `Lexis` object. Remember to declare the "NRA" state as a precursor state, i.e. a state that is *less* severe than "Remission" in the sense that a person will stay in the "Remission" state unless he goes to the "ESRD" state.

```
> Lc <- cutLexis( Lr, cut=Lr$dor, timescale="per",
+                 new.state="Rem", precursor.states="NRA" )
> summary( Lc )
```

10. Show how the states are connected and the number of transitions between them by using `boxes`. This is an interactive command the requires you to click in the graph window: ■eval=FALSE>= boxes(Lc)

11. Now make a Lexis diagram where different colouring is used for different segments of the follow-up:

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

12. List the first records of the dataframe `Lc` (using `head()`) and make sure that you understand how persons and follow-up time is represented in this dataset.

13. Make a Cox-regression of mortality (i.e. endpoint "ESRD") with sex, age at entry and remission as explanatory variables, and using time since entry as timescale. Note how the variable `lex.Cst` is used as a time-dependent variable:

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

14. What is the assumptions about the rate of ESRD/death between persons in remission and persons not?

15. What is the assumption about the incidence rates of remission? (cf. figure 3.1).

3.5.2 Splitting follow-up time

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

16. Split the follow-up time every 0.5 years after entry, and make sure that the number of events and risk time is the same as before:

```
> sLc <- splitLexis( Lc, "tfi", breaks=seq(0,30,0.5) )
> summary( Lc, scale=1000 )
> summary(sLc, scale=1000 )
```

17. Now try to fit the Poisson-model corresponding to the Cox-model we fitted previously. The function `ns()` produces a model matrix corresponding to a piecewise cubic function.

```
> library( splines )
> mp <- glm( as.numeric(lex.Xst=="ESRD") ~
+                   ns( tfi, df=8 ) +
+                   factor(sex) + I((doe-dob-40)/10) +  (lex.Cst=="Rem") +
+                   offset( log(lex.dur) ),
+            family = poisson, data=sLc )
> summary( mp )
```

Another possibility, which is formally more correct is to use the midpoint of the intervals in which the time is split. These are accessed by the function `timeBand`:

```
> mx <- glm( as.numeric(lex.Xst=="ESRD") ~
+                   ns( timeBand(sLc,"tfi","mid"), df=8 ) +
+                   factor(sex) + I((doe-dob-40)/10) +  (lex.Cst=="Rem") +
+                   offset( log(lex.dur) ),
+            family = poisson, data=sLc )
> summary( mx )
```

Note the similarity of the results.

18. What kind of assumptions are made in this model about the transition rates in the figure 3.1?

19. You can extract the parameters from the models using `ci.lin`, try:

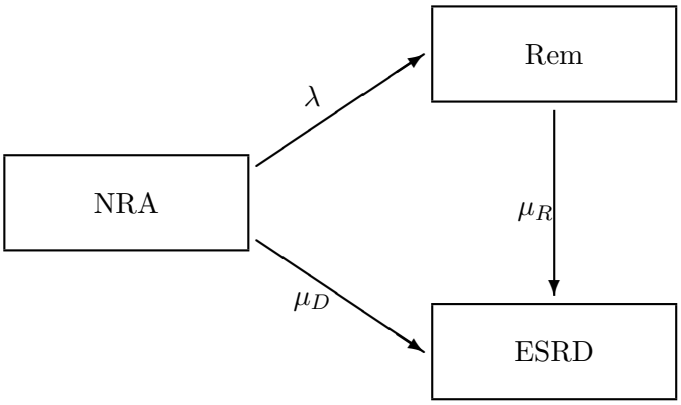

Figure 3.1: *Illness-death model for the renal data.*

```
> ci.lin( mp )
> ci.lin( mp, subset=10:12 )
> ci.lin( mp, subset=10:12, Exp=TRUE )
> ci.lin( mp, subset=c("sex","dob","Cst"), Exp=TRUE )
> ci.lin( mp, subset=10:12, Exp=TRUE )[,5:7]
```

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

```
> ci.lin( m1, Exp=TRUE )
```

3.5.3 Splines and predictions

In the model we have used splines to model the effect of the covariate `tfi`, time since NRA. The model assumes that the rates are constant in 6-month intervals (the units we split into). The spline models how the rates in each of these intervals relate to each other. Spline functions are functions that are cubic between a set of *knots* and constrained to have the same 0th, 1st and 2nd derivative at the knots. The natural splines generated by `ns()` further have the property that they are linear beyond the outermost, so called boundary, knots. Splines are linear combinations of polynomials, and so the models are just usual linear models. The only requirement to generate the columns of the model matrix is fixing the knots. Even if we have 40+ intervals on this time-scale, we only used 8 parameters to model the effect.

To see the effect of time, the estimated coefficients for the time effect must be multiplied with a matrix where each row represents a particular time. This can be done explicitly for chosen values of time by using the `ns` function.

However, the function `ns` chooses the knots based on the data. So if we want to have the same parametrization for a new set of points, we must first extract the knots and boundary knots used in the model. They are stored in the attributes of the `ns` object, and can be found using the function `attr()`. These are then used to generate a model matrix with rows corresponding to a prespecified set of time-points. This matrix is then multiplied with the estimated parameters to get the estimated effects. This is simplest achieved by using `ci.lin`, which also has the facility to select subsets of parameters from a function and to extract standard errors too.

20. Plot the survival function for a male aged 40 at entry (NRA), based on the model `mp`, using the function `ci.cum`.

```
> # Get the knots
> t.kn <- attr( ns( sLc$tfi, df=8 ), "knots" )
> t.Bo <- attr( ns( sLc$tfi, df=8 ), "Boundary.knots" )
```

```
> # Then decide the points of prediction
> t.pt <- seq( 0,20,0.1 )
> nt <- length( t.pt )
> # Then make the contrast matrix to generate log-rates
> MT <- ns( t.pt, knots=t.kn, Bo=t.Bo )
> cmat <- cbind( 1, MT, 1, 0, 0 )
> cinc <- ci.cum( mx, ctr.mat=cmat, intl=0.1 )
> # Note that there is no guarantee that the lower limit of the c.i. for
> # the cumulative incidence is positive:
> cinc[1:20,]
> matplot( t.pt, exp( -cinc )[,-4], type="l", lwd=c(3,1,1), lty=1, col="black" )
```

21. What is the problem with the previous question?

22. Use the extracted knots to generate the relevant model matrix, and use this for plotting effect of time since NRA:

```
> MT <- ns(              t.pt  , knots=t.kn, Bo=t.Bo ) -
+       ns( rep(0,length(t.pt)), knots=t.kn, Bo=t.Bo )
> # The extract, multiply and exponentiate:
> t.eff <- ci.lin( mp, subset="ns", ctr.mat=MT, Exp=T )[,5:7]
> # Then plot the effects
> matplot( t.pt, t.eff, log="y", type="l", lty=1, col="black", lwd=c(3,1,1),
+          xlab="Time since NRA", ylab="Rate ratio" )
```

    What is shown in this plot?
    How relevant is that?

23. The next step is to include a spline effect of current age in the model too. Try:

```
> ma <- glm( as.numeric(lex.Xst=="ESRD") ~
+                   ns( tfi, df=8 ) +
+                   ns( age, df=8 ) +
+                   sex + I((doe-dob)/10) + as.numeric( lex.Cst=="Rem" ) +
+                   offset( log(lex.dur) ),
+            family = poisson, data=sLc )
> summary( ma )
```

    Why is the effect of age at entry (`I((doe-dob)/10)`) set to `NA` in the output from the model?

24. (Esoteric, hard, but quite important) Try to replace the `ns( tfi, df=8 )` with the term `ns( timeBand(sLc,"tfi","mid"), df=8 )` in the model. Why is the age at entry effect not `NA` in this case?

25. In this model (i.e. `ma`) we can show the effect of time since NRA as before, as well as the effect of current age. Try:

```
> # First get the knots in order to generate the splines
> a.kn <- attr( ns( sLc$age, df=8 ), "knots")
> a.Bo <- attr( ns( sLc$age, df=8 ), "Boundary.knots")
> # Now we can make the matrix to multiply with the coefficients:
> a.pt <- seq( 40,70,0.2)
> Ma <- ns(               a.pt  , knots=a.kn, Bo=a.Bo ) -
+       ns( rep(60,length(a.pt)), knots=a.kn, Bo=a.Bo )
> a.eff <- ci.lin( ma, subset="age", ctr.mat=Ma, Exp=T )[,5:7]
> matplot( a.pt, a.eff, log="y", type="l", lty=1, col="black", lwd=c(3,1,1),
+          xlab="Current age", ylab="Rate ratio" )
> abline( h=1 )
```

    How is the rate (as a function of age) behaving relative to what you would expect for the general population?

26. We are however also interested in knowing the absolute magnitude of the rates of ESRD/death for patients without remission. This would include the intercept as well as the rate-ratio function.

The times where we want the rates should be the same as before, but now we must specify an intercept (just a column of 1s) the sex (for males just a column of 1s) and the age. The latter is a matrix of identical rows, each corresponding to the age-effect at 60 years, say. Try:

```
> MTa <- ns( rep(60,length(t.pt)), knots=a.kn, Bo=a.Bo )
> T.inc <- ci.lin( ma, subset=c("Int","tfi","age","sex"),
+                  ctr.mat=cbind(1,MT,MTa,1), Exp=TRUE )[,5:7]
> matplot( t.pt, T.inc, log="y", type="l", lty=1, col="black", lwd=c(5,2,2), las=1,
+          xlab="Time since NRA", ylab="Rate per year", ylim=c(0.01,5) )
```

What is the current age at each time point of this curve?

27. Now try to make the same prediction of rates, but this time for a person that is 50 years *at entry*. Then the rows of the age-effects matrix should correspond to age at entry *plus* time since entry (i.e. current age):

```
> MTa <- ns( 50+t.pt, knots=a.kn, Bo=a.Bo )
> T.inc <- ci.lin( ma, subset=c("Int","tfi","age","sex"),
+                  ctr.mat=cbind(1,MT,MTa,1), Exp=TRUE )[,5:7]
```

Now plot these estimated rates on top of the other:

```
> matplot( t.pt, T.inc, log="y", type="l", lty=1, col="black", lwd=c(5,2,2), las=1,
+          xlab="Time since NRA", ylab="Rate per year", ylim=c(0.01,5) )
> matlines( t.pt, T.inc, type="l", lty=1, col="red", lwd=c(5,2,2) )
> abline( v=10 )
```

What is the meaning of the latter set of rates as opposed to the former?

28. How would you accomplish this analysis with a Cox-model?

29. Apart from the two timescales, time since NRA and current age, a third timescale may be of interest, namely time since remission. However this is only relevant for persons who actually have a remission, so start by checking how many events there are in this group:

```
> summary( sLc )
```

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

30. With this rather limited number of events we can certainly not expect to be able to model anything more complicated than a linear trend with time since remission.

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

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

Make sure that you understand what goes on here.

31. We can now expand the model with this variable. We need not write the entire model statement again, we can just say:

```
> mx <- update( ma, . ~ . + pmax( (per-dor)/10, 0, na.rm=TRUE ) )
> summary( mx )
```

Is the effect significant? Can a substantial effect of time since remission be ruled out?

32. What is your overall conclusion — is it consonant with the conclusion in the paper?

3.6 Competing risks: The Danish Thorotrast study

In the period 1935–50 a contrast medium called Thorotrast was used for cerebral angiography (X-ray imaging of the brain). This contrast medium contained ^{232}Th, thorium. It turns out that thorium is not excreted from the body, it is permanently deposited, some 60% in the liver, 20% in the spleen and some 10% in the bone marrow, and a very small fraction in other organs.

Thorium is an α-emitting radionuclide, i.e. it emits α-rays (i.e. He-nuclei) which is ionizing, but not partcularly penetrating; it only penetrates 2–3 cell-layers. The half-life of ^{232}Th is 1.4×10^{10} years, so the patients that have been injected with Thorotrast exposed are to a constant, small α-radiation for life.

A number of studies of persons subjected to Thorotrast have been conducted (Japan, Germany, Protugal, Sweden and Denmark). The data used in this workshop comes from one of the largest studies, the Danish, which incorporates 999 exposed patients injected with Thorotrast between 1935 and 1947, and 1480 controls who have had a cerebral angiography in the period 1946–63, on similar indications as the Thorotrast patients.

Persons undergoing cerebral angiography are in may cases seriously ill, they are suspected of cerebral malformations or tumors, so both the Thorotrast group and the control group have very high mortality rates, and a pattern of causes of death that differ much from the general population. Especially during the first year after diagnosis, there is a very high mortality among the patients, which is entirely associated to the conditions that have lead to to the cerebral angiography. Therefore, the follow-up of both Thorotrast patients and control patients started one year after the angiograpy, at which time 811 Thorotrast patients and 1236 control patients were alive.

Since the Thorotrast patients recieve a continuous dose to the liver they have very high rates of liver cancer. All 127 liver cancers except 8 have been classified as one of three different subtypes: hepatocellular carcinoma, cholangiocellular carcinoma and haemagiosarcoma.

3.6.1 Cumulative dose

Thorium in the form of Thorotrast has a tendency to form small "lumps" when it deposits in the liver. Because of the limited range of the α-rays this causes the radiation dose per time to be less than proportional to the injected dose, because some of the emitted particles never reach beyond the "lump". It has been estimated that this give rise to the following conversion factors between injected volume and liver dose:

Inj. volume (ml)	Liver dose rate (Gy/year/dl)
1–9	1.40
10–19	1.25
20–29	1.10
30–39	0.95
40–49	0.85
50–59	0.76
60–69	0.72
70–79	0.69
80–99	0.65

The relationship between injected volume v (measured in dl) and effective radiation dose rate ρ (in Gray/year/dl) can be quite well approximated by the function:

$$\rho = 1.502 - 1.937 \times v + 1.109 \times v^2$$

so the annual dose δ (in Gray/per year) is approximately:

$$\delta = (1.502 - 1.937 \times v + 1.109 \times v^2) \times v$$

3.6.2 The data sets

The dataset is available in the Epi package, so you can load the data and inspect it by:

```
> library(Epi)
> data(thoro)
> head(thoro)
> str(thoro)
```

This will load the dataframe `thoro` with information about 2470 cases of cerebral angiography. See the details and variable description on the help page using `?thoro`.

3.6.3 Competing risks: Tumour histology

1. Now, we will look at the incidence rates of the three different histological subtypes of liver cancer. There are no cases of liver cancer in the control group, so this analysis is only of interest for the Thorotrast group, so start by defining a dataset only containing the Thorotrast group (`contrast==1`, for example by:

   ```
   > tht <- thoro[thoro$contrast==1,]
   > with( tht, ftable( liver, hepcc, chola, hmang, col.vars=1 ) )
   ```

   Note that there are some liver cancer cases that it has not been possible to type, hence the number events for `hepcc`, `chola` and `hmang` do not add up to that for `liver`.

   You would want to change the date-variables to numerical calendar-year variables, with the dates as fractions of a year:

   ```
   > tht <- cal.yr(tht)
   > str(tht)
   ```

2. Tabulate the event indicators for these three types of events against the existence of a date of livercancer diagnosis `is.na(liverdat)`. Then define a date of exit, `dox`, for the analysis of these three types of event, using date of death or unknown type of liver cancer as censoring date:

   ```
   > tht$dox <- pmin( tht$liverdat, tht$exitdat, na.rm=T )
   > tht <- subset( tht, dox > injecdat )
   ```

3. Then define the cumulative dose per year:

   ```
   > tht <- transform( tht, dl = volume / 100 )
   > tht <- transform( tht, gpy = (1.502-1.937*dl+1.109*dl^2)*dl )
   ```

4. Now create the dataset needed for analysis of the three competing risks of the three types of liver cancer.

   Set up a `Lexis` object with three different possible event types, that is 4 states: "No cancer", "hepcc", "chola" and "hmang", according to whether an event of the given type has occurred. Note we use the default for Lexis to assign the entry state for eveyone as the firts levels of the state factor defined in `exit.stae`. We also set up two timescales of interest, calendar time and time from injection:

   ```
   > tht.L <- Lexis( entry = list( per = injecdat,
   +                               tfi = 0 ),
   +                 exit = list( per = dox ),
   +           exit.status = factor( 1*hepcc+2*chola+3*hmang,
   +                                 labels=c("No cancer","hepcc","chola","hmang") ),
   +                 data = tht )
   > summary( tht.L )
   ```

   You can visualize the model with number of events and amount of risk time using:

```
> boxes( tht.L )
```

5. Now do a stratified Cox-analysis of the three rates, using time since injection as time and injected dose as covariate.

   To this end you need a stacked object, which you create by `stack.Lexis`:

```
> tht.S <- stack.Lexis( tht.L )
> str( tht.S )
```

6. Now fit two Cox models:

```
> library( survival )
> mi <- coxph( Surv( tfi, tfi+lex.dur, lex.Fail ) ~ volume:lex.Tr + strata( lex.Tr ),
+              data = tht.S )
> m1 <- coxph( Surv( tfi, tfi+lex.dur, lex.Fail ) ~ volume + strata( lex.Tr ),
+              data = tht.S )
> anova( mi, m1, test="Chisq" )
```

   What is the difference between the two models, i.e. what is being tested by `anova`?

7. Why is the `strata(lex.Tr)` term in the model? What would the models mean if we omitted it?

8. What is the differences between the volume effects? (Use a contrats matrix for the `ci.lin` function).

9. Is there any effect of age at entry? Fit the relevant model to answer this question.

10. If we want to assess the effcet of (deterministically) time varying variables we must split time into intervals of length say 1 year. Note that it is immaterial whether we split time before or after we duplicate the dataset for competing risk analysis.

    Then we can split the data along the time since injection and compute the midpoint of the intervals.

```
> thsplit <- splitLexis( tht.L, breaks=0:100, time.scale="tfi" )
> thsplit$m.tfi <- timeBand( thsplit, "tfi", "middle" )
```

    If we want to do a competing risk analysis, we must stack the dataset first:

```
> tht.X <- stack.Lexis(thsplit)
```

    Now make a table that shows when in the follow-up the evenst of the different types of transitions occur, for example:

```
> with( tht.X, tapply( lex.Fail, list(lex.Tr,floor(m.tfi/5)*5), sum ) )
```

11. Now make an analysis equivalent to the Cox-analysis, using splines to model the underlying hazard — be careful where you put the knots, and how many:

```
> Pi <- glm( lex.Fail ~
+            ns( m.tfi, kn=c(15,30), Bo=c(1,50), intercept=T ):lex.Tr +
+            volume:lex.Tr + offset( log(lex.dur) ),
+            family=poisson, data=tht.X )
> P1 <- update( Pi, . ~ . - volume:lex.Tr + volume )
> anova( Pi, P1, test="Chisq" )
```

    How do the conclusions differ from those from the Cox-model?

12. Now, compute the cumulative dose at the beginning of each interval:

```
> tht.X$cdos <- tht.X$m.tfi * tht.X$gpy
```

You may want to also generate the lagged versions, that is variables which at any one time of follow-up give the cumulative dose as it was, say 10 or 20 years earlier:

```
> tht.X$l10dos <- pmax( tht.X$m.tfi-10, 0 ) * tht.X$gpy
```

Fit models that allows you to test whether the cumulative dose has different effects on the three types of liver cancer.

13. In particular, address the question of whether the effect of cumulative dose is proportional between the three types of liver cancer.

3.6.4 Competing risks: Probability of liver cancer.

It may be of interest to estimate how large a fraction of the thorotrast patients actually gets a liver cancer.

1. Set up a `Lexis` object with states "no liver cancer", "liver cancer" and "death" as states.
2. Estimate this proportion by taking the fraction of the patients that actually acquire a liver cancer (look at the variable `liver`).
3. Work out the Nelson-Aalen estimators for the cumulative incidence of liver cancer and the mortality wihtout liver cancer, as a function of time since injection.
4. Use these two to compute the probability of getting liver cancer before time $t = 1, 2, \ldots, 50$ years after injection. (Use the lung cancer example from the lectures).

   Make a plot of it.
5. Use the machinery you applied before to fit a spline model for the rates as function of time since injection.
6. Use the estimated rates to compute the probability of getting liver cancer within t after injection for t between 0 and 50 years.