Epidemiology with R

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

NovoNordisk Epidemiology

August 2015

http://bendixcarstensen.com/Epi/Courses/NNepi/

Introducing R

Epidemiology with R
August 2015
NovoNordisk Epidemiology
http://bendixcarstensen.com/Epi/Courses/NNepi/

data

The best way to learn R

- The best way to learn **R** is to use it!
- This is a short introduction before you sit down in front of a computer.
- **R** is a little different from other packages for statistical analysis.
- These differences make **R** very powerful, but for a new user they can sometimes be confusing.
- Our first job is to help you up the initial learning curve so that you can be comfortable with R.

Nothing is lost or hidden

- Statistical software provides “canned” procedures to address common statistical problems.
- Canned procedures are useful for routine analysis, but they are also limiting.
  - You can only do what the programmer lets you do.
  - You get predetermined output:
    - relevant
    - irrelevant
    - incomprehensible
- In **R**, the results of statistical calculations are always accessible:
  - You can use them for further calculations.
  - You can always see how the calculations were done.

R Packages

- The capabilities of **R** can be extended using “packages”.
- Distributed over the Internet *via* **CRAN**:
  (the **C**omprehensive **R** **A**rchive **N**etwork) and can be downloaded directly from an **R** session.
- There is an **R** package developed during the annual course on “Statistical Practice in Epidemiology using **R**”, called **Epi**.
- Contains special functions for epidemiologists and some data sets useful for illustration.
- There are 6,964 other user contributed packages on CRAN.

Objects and functions

R allows you to build powerful procedures from simple building blocks. These building blocks are **objects** and **functions**.

- All data in **R** is represented by **objects**, for example:
  - A dataset (called data frame in R)
  - A vector of numbers
  - A vector of color names
  - The result of fitting a model to data
- You, the user, call **functions**
- Functions act on objects to create **new objects**:
- Using `glm` on a dataframe (an object) produces a fitted model (another object):
  `m1 <- glm( D ãge+sex, data=dfr )`

Because all is functions. . .

- You will always (almost) use parentheses:
  `> res <- FUN( x, y )`
- . . . which is pronounced
- `res` **gets** ("`<-`") `FUN` **of** `x,y` ("`(x,y)`")

Vectors

One of the simplest objects in **R** is a sequence of numbers, called a **vector**.

You can create a vector in **R** with the collection (`c`) function:

```
> c(1,3,2)
[1] 1 3 2
```

You can save the results of any calculation using the left arrow:

```
> x <- c(1,3,2)
> x
[1] 1 3 2
```

The workspace

- Every time you use `<-`, you create a new object in the **workspace** (or overwrite an old one).
- A list of objects in the workspace can be seen with the `objects` function (synonym: `ls()`):

```
> objects()
[1] "a" "aa" "acz2" "alpha" "b"
[6] "bar" "bb" "bdendo" "beta" "cc"
[11] "Col"
```

- In Epi is a function `lls()` that gives a bit more information on the objects.
- The workspace is held in computer memory and will be lost at the end of the session unless you explicitly save it.

Working Directory

Every **R** session has a **current working directory**, which is the location on the hard disk where files are saved, and the default location from which files are read into R. /pause

- `getwd()` Prints the current working directory
- `setwd("c:/Users/Martyn/Project")` sets the current working directory — note the forward slash (“/”).
- You may also use a Graphical User Interface (GUI) to change directory.

Ending an R session

- To end an **R** session, call the `quit()` function
  - Every time you want to do something in R, you call a function.
- You will be asked “Save workspace image?”
  - Yes saves the workspace to the file “`.RData`” in your current working directory. It will be automatically loaded into **R** the next time you start an **R** session.
  - No does not save the workspace.
  - Cancel continues the current **R** session without saving anything.
- It is recommended you just say “No”.

Always start with a clean workspace

Keeping objects in your workspace from one session to another can be dangerous:

- You forget how they were made
- You cannot easily recreate them if your data changes
- They may not even be from the same project

It is almost always best to start with an empty workspace and use a script file to create the objects you need from scratch:

- You will know from where you read your data
- You will know what you did to get the results

Rectangular Data

Rectangular data sets are common to most statistical packages

”id”	”visit”	”time”	”status”
1	1	0.0	0
1	2	1.5	0
2	1	0.0	0
2	2	1.1	0
2	3	2.3	1

Columns represent variables.
Rows represent individual records.

The world is not a rectangle!

- Most statistical packages used by epidemiologists assume that **all data** can be represented as a rectangular data set.
- **R**allows a much richer set of data structures, represented by **objects** of different **classes**.
- Rectangular data sets are just one type of object that may be in your workspace.
- This class of object is called a **data frame**.

Data Frames

Each column of a data frame is a variable.

Variables may be of different types:

- **vectors:**
  - **numeric:** `c(1,2,3)`
  - **character:** `c("John","Paul","George","Ringo")`
  - **logical:** `c(FALSE,FALSE,TRUE)`
- **factors:** `factor(c("low","medium","high","low", "low"))`

Building your own data frame

Data frames can be constructed from a list of vectors

```
> mydata <- data.frame(x=c(3,6,7),f=c("a","b","a"))
> mydata
  x f
1 3 a
2 6 b
3 7 a
```

Character vectors are automatically converted to factors.

Inspecting data frames

Most data frames are too large to inspect by printing them to the screen, so use:

- `names(x)` returns a vector of variable names.
  - Use `sort(names(x))` to get them in alphabetical order.
  - Use `grep("ch",names(x))` to find names that contain “ch”.
- `head(x)` prints the first few lines, and `tail`...
- `str(x)` prints a brief overview of the **str**ucture of the data frame. Can be used on any object.
- `summary(x)` prints a more comprehensive summary
  - Quantiles for numeric variables
  - Tables for factors

Extracting values from a data frame

Use square brackets to take **subsets** of a data frame (indexing)

- `mydata[1,2]`. The value in row 1, column 2.
- `mydata[1,]`. The whole of the first row.
- `mydata[,2]`. The whole of the second column.

You can also extract a column from a data frame by name:

- `mydata$age`. The column, or variable, named “age”
- `mydata[,"age"]`. The same.

Importing data

- R has good facilities for importing data from other applications:
  - `read.dta` for reading Stata datasets.
  - `read.spss` for reading SPSS datasets.
  - `read.xport` and `read.ssd` for reading SAS-datasets.

Reading Text Files

`read.table` reads data from a text file and returns a data frame:

- `mydata <- read.table("myfile")`
- `myfile` could be
  - A file in the **current working directory**: `fem.dat`
  - A path to a file: `c:/rex/fem.dat`
  - A URL: `url("http://BendixCarstensen.com/Epi/NNepi/data/fem.txt")`
- Note: `myfile` must be enclosed in quotes.
- There are **many** arguments to `read.table` — read the manual page.

`write.table` does the opposite.

R uses a forward slash “`/`” for file paths or double “
”.

Some useful arguments to `read.table`

- `header = TRUE` if first line contains variable names
- `sep=","` if values are comma-separated instead of being space-delimited.
- `as.is = TRUE` to stop strings being converted to factors
- `na.strings = "99"` to denote that 99 means "missing". Default values are:
  - `NA` "Not Available"
  - `NaN` "Not a Number"
- For comma-separated files there is `read.csv` and `read.csv2`

Reading Binary Data

- **R** can read in data in binary (non-text) format from other statistical systems using the foreign extension package.
- **R** is an open source project, and relies on the format for binary files to be well-documented.
- Example:
  - SAS XPORT format has been adopted as a data exchange standard by the US Food and Drug Administration.
  - SAS CPORT format remains a proprietary format.

Some functions in the foreign package

- `read.dta` for Stata (also `write.dta`)
- `read.xport` for SAS XPORT format (not CPORT)
- `read.epiinfo` for EPIINFO
- `read.mtp` for MiniTab Portable Worksheet
- `read.spss` for SPSS

You can write to SAS, SPSS and Stata using `write.foreign`

See the "R Data Import/Export manual" for more details:

```
> RShowDoc("R-data")
```

Accessing databases systems

Microsoft **Access**:

```
> library(RODBC)
> ch <- odbcConnectAccess("../data/theData.mdb")
> bd <- sqlFetch(ch, "aTable" )
> close( ch )
```

Microsoft **Excel**:

```
> library( RODBC )
> cnc <- odbcConnectExcel(paste("../theXel.xls",sep=""))
> sht <- sqlFetch( cnc, "theSheet" )
> close( cnc )
```

Other databases

```
> ?odbcConnect
```

Summary - data

- You can use a data frame to organize your variables
- You can extract variables from a data frame using `$`.
- You can extract variables and observation using indecing `[,]`
- You can read in data using
  - `read.table`
  - tailored function from the `foreign` package
  - database interface from the `RODBC` package

Summary - when it goes wrong

When somthing is fishy with an object `obj`, try to find out what you (accidentally) got, by using:

```
> lls()
> str( obj )
> dim( obj )
> length( obj )
> names( obj )
> head( obj )
> class( obj )
> mode( obj )
```

R language

Epidemiology with R
August 2015
NovoNordisk Epidemiology
http://bendixcarstensen.com/Epi/Courses/NNepi/

lang

Language

- R is a programming language – also on the command line
- This means that there are **syntax rules**
- Print an object by typing its name
- Evaluate an expression by entering it on the command line
- Call a function, giving the arguments in parentheses – possibly empty
- Notice `ls` vs. `ls()`

Objects

- The simplest object type is *vector*
- Modes: numeric, integer, character, generic (list)
- Operations are vectorized: you can add entire vectors with `a + b`
- Recycling of objects: If the lengths don't match, the shorter vector is reused

R expressions

```
x <- rnorm(10, mean=20, sd=5)
m <- mean(x)
sum((x - m)^2)
```

- Object names
- Explicit constants
- Arithmetic operators
- Function calls
- Assignment of results to names

Function calls

Lots of things you do with **R** involve calling functions.
For instance

```
mean(x, na.rm=TRUE)
```

The important parts of this are

- The name of the function
- Arguments: input to the function
- Sometimes, we have named arguments

Function arguments

```
rnorm(10, mean=m, sd=s)
hist(x, main="My histogram")
mean(log(x + 1))
```

Items which may appear as arguments:

- Names of an R objects
- Explicit constants
- Return values from another function call or expression
- Some arguments have **default values**.
- Use `help(`*`function`*`)` or `args(`*`function`*`)` to see the arguments (and their order and default values) that can be given to any function.

Creating simple functions

```
logit <- function(p) log(p/(1-p))
logit(0.5)

simpsum <-
function(x, dec=5)
{ # produces mean and SD of a variable, default value for dec is 5
round(c(mean=mean(x),sd=sd(x)),dec)
}

x <- rnorm(100)
simpsum(x)
simpsum(x,2)
```

The value of a functions is the `last` calculated value.

Indexing

- **R** has several useful indexing mechanisms:
- `a[5]` single element
- `a[5:7]` several elements: 5th, 6th & 7th
- `a[-6]` all except the 6th
- `a[c(1,1,2,1,2)]` some elements repeated
- `a[b>200]` logical index
- `a["well"]` indexing by name

Lists

- Lists are vectors where the elements can have different types
- Functions often return lists
- `lst <- list(A=rnorm(5),B="hello",K=12)`
- Special indexing:
- `lst$A`
- `lst[1:2]` a list with first two first elements (`A` and `B` — NB: single brackets)
- `lst[1]` a list of length 1 which is the first element (codeA — NB: single brackets)
- `lst[[1]]` first element (NB: double brackets) — a vector of length 5 (from `A=rnorm(5)` above).

Classes, generic functions

- R objects have **classes**
- Functions can behave differently depending on the class of an object
- E.g. `summary(x)` or `print(x)` does different things if `x` is numeric, a factor, or a linear model fit

The workspace

- The **global environment** contains R objects created on the command line.
- There is an additional **search path** of loaded packages and attached data frames.
- When you request an object by name, R looks first in the global environment, and if it doesn't find it there, it continues along the search path.
- The search path is maintained by `library()`, `attach()`, and `detach()`
- List the search path by `search()`
- Notice that objects in the global environment may mask objects in packages and attached data frames

Data manipulation and `with`

```
bmi <- stud$weight/(stud$height/100)^2)
bmi <- with(stud, weight/(height/100)^2)
```

uses variables weight and height in the data frame `stud` (not the variables with the same name in the workspace), but creates the variable `bmi` in the global environment (not in the data frame).

To create a new variable in the data frame, you can use:

```
stud$bmi <- with( stud, weight/(height/100)^2 )
bmi <- transform( bmi, stud = weight/(height/100)^2 )
```

Constructors

- Matrices and arrays, constructed by the (surprise) `matrix` and `array` functions:
  - `MM <- matrix( 1:12, 6, 2 )`: 6 rows, 2 columns
  - `AA <- array( 1:36, dim=c(6,2,3) )`: $6 \times 2 \times 3$ array
  - Refer to elements: `AA[4:6,2,2:3]`: 3×2 array
  - Print compacty with `ftable`.
- You can extract and set names with `names(x)`,
- . . . for matrices and data frames also `colnames(x)` and `rownames(x)`, for arrays use `dimnames(x)`.
- You can construct a matrix from its columns using `cbind(x,y,z)`,
- Joining two matrices with equal no of columns (with the same column names) is done with `rbind(aa,bb)`.

Factors (class variables)

- Factors are used to describe groupings.
- Basically, these are just integer codes plus a set of names for the **levels**
- They have class `"factor"` making them (a) print nicely and (b) maintain consistency
- A factor can also be **ordered** (class `"ordered"`), signifying that there is a natural sort order on the levels
- Factors play a fundamental role in statistical models by indicating that a variable should be treated as a classification rather than as a quantitative variable (similar to a CLASS statement in SAS)

The factor function

- Typically used when read.table gets it wrong,
- e.g. group codes read as numeric
- or read as factors, but with levels in the wrong order (e.g. c("rare", "medium", "well-done") sorted alphabetically.)
- Notice that there is a slightly confusing use of levels and labels arguments:
  - levels are the value codes **on input**
  - labels are the value codes **on output** (and becomes the levels of the resulting factor)
  - The levels of a factor is shown by the levels() function.

```
sex <- factor( kon, levels=c(1,2), labels=c("M","F") )
sex <- factor( kon, levels=c(2,1), labels=c("F","M") )
```

Working with Dates

- Dates are usually read as character or factor variables
- Use the as.Date function to convert them to objects of class "Date"
- If data are not in the default ISO format ("yyyy-mm-dd") you need to supply a format specification:

```
> as.Date("11/3-1959",format="%d/%m-%Y")
[1] "1959-03-11"
```

Working with Dates

- Computing the differences between Date objects gives an object of class "difftime", which is number of days between the two dates:

```
> as.numeric(as.Date("2007-5-25")-
             as.Date("1959-3-11"),"days")
[1] 17607
```

- In the Epi package is a function that converts dates to calendar years with decimals:

```
> as.Date("1952-07-14")
[1] "1952-07-14"
> cal.yr( as.Date("1952-07-14") )
[1] 1952.533
attr(,"class")
[1] "cal.yr"  "numeric"
```

Basic graphics

The plot() function is a generic function, producing different plots for different types of arguments. For instance, plot(x) produces:

- a plot of observation index against the observations, when x is a numeric variable
- a bar plot of category frequencies, when x is a factor variable
- a time series plot (interconnected observations) when x is a time series
- a set of diagnostic plots, when x is a fitted regression model
- . . .

Basic graphics

Similarly, the plot(x,y) produces:

- a scatter plot of x is a numeric variable
- a bar plot of category frequencies, when x is a factor variable

Basic graphics

Examples:

```
x <- c(0,1,2,1,2,2,1,1,3,3)
plot(x)
plot(factor(x))
plot(ts(x))   # ts() defines x as time series
y <- c(0,1,3,1,2,1,0,1,4,3)
plot(x,y)
plot(factor(x),y)
```

Basic graphics

More simple plots:

- hist(x) produces a histogram
- barplot(x) produces a bar plot (useful when x contains counts – often one uses barplot(table(x)))
- boxplot(y ~ x) produces a box plot of y by levels of a (factor) variable x.

Rates and Survival

Epidemiology with R
August 2015
NovoNordisk Epidemiology
http://bendixcarstensen.com/Epi/Courses/NNepi/

surv-rate

Survival data

Persons enter the study at some date.

Persons exit at a later date, either dead or alive.

Observation:
Actual time span to death (“event”)
or
Some time alive (“at least this long”)

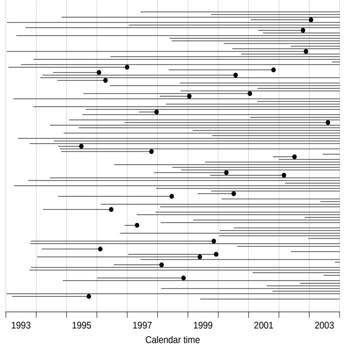
Examples of time-to-event measurements

- Time from diagnosis of cancer to death.
- Time from randomisation to death in a cancer clinical trial
- Time from HIV infection to AIDS.
- Time from marriage to 1st child birth.
- Time from marriage to divorce.
- Time to re-offending after being released from jail

Each line a person

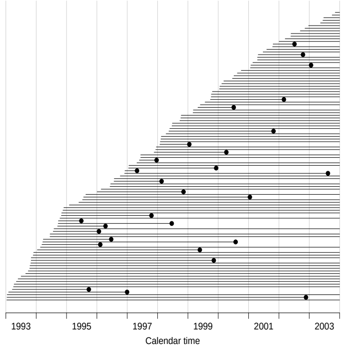
Each blob a death

Study ended at 31 Dec. 2003

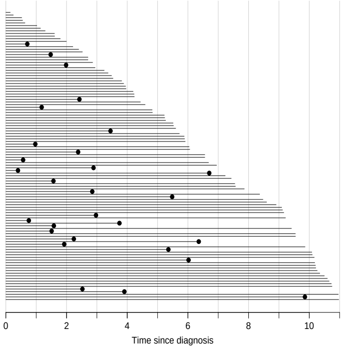


Ordered by date of entry

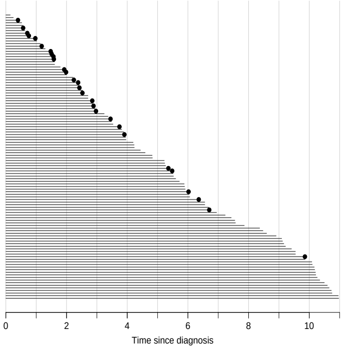
Most likely the order in your database.



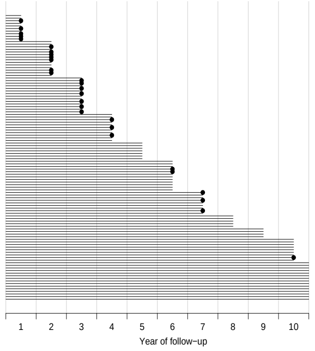
Timescale changed to
“Time since diagnosis”.



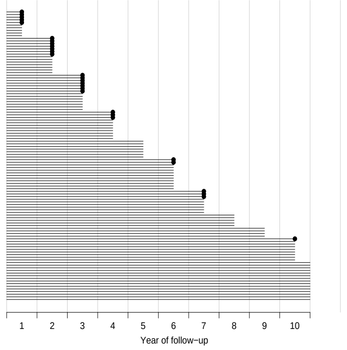
Patients ordered by survival time.



Survival times grouped into bands of survival.



Patients ordered by survival status within each band.



Survival after Cervix cancer

	Stage I			Stage II		
Year	N	D	L	N	D	L
1	110	5	5	234	24	3
2	100	7	7	207	27	11
3	86	7	7	169	31	9
4	72	3	8	129	17	7
5	61	0	7	105	7	13
6	54	2	10	85	6	6
7	42	3	6	73	5	6
8	33	0	5	62	3	10
9	28	0	4	49	2	13
10	24	1	8	34	4	6

Estimated risk in year 1 for Stage I women is $5/107.5 = 0.0465$

Estimated 1 year survival is $1 - 0.0465 = 0.9535$

Life-table estimator.

Survival function

Persons enter at time 0:
Date of birth, date of randomization, date of diagnosis.

How long do they survive?
Survival time T — a stochastic variable.

Distribution is characterized by the survival function:

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

$F(t)$ is the cumulative risk of death before time t.

Intensity or rate

$$P\{\text{event in } (t, t+h] \mid \text{alive at } t\}/h$$

$$\begin{aligned} &= \frac{F(t+h) - F(t)}{S(t) \times h} \\ &= -\frac{S(t+h) - S(t)}{S(t)h} \xrightarrow[h\to 0]{} -\frac{\mathrm{d}\log S(t)}{\mathrm{d}t} \\ &= \lambda(t) \end{aligned}$$

This is the **intensity** or **hazard function** for the distribution.
Characterizes the survival distribution as does f or F.

Theoretical counterpart of a **rate**.

Relationships

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

$\Lambda(t) = \int_0^t \lambda(s)\,\mathrm{d}s$ is called the **integrated intensity**. **Not** an intensity, it is dimensionless.

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

Rate and survival

$$S(t) = \exp\left(-\int_0^t \lambda(s)\,\mathrm{d}s\right) \qquad \lambda(t) = \frac{S'(t)}{S(t)}$$

Survival is a *cumulative* measure, the rate is an *instantaneous* measure.

Note: A cumulative measure requires an origin!

Observed survival and rate

- **Survival studies**: Observation of (right censored) survival time:

  $$X = \min(T, Z), \quad \delta = 1\{X = T\}$$

  — sometimes conditional on $T > t_0$
  (left truncation, delayed entry).
- **Epidemiological studies**:
  Observation of (components of) a rate:

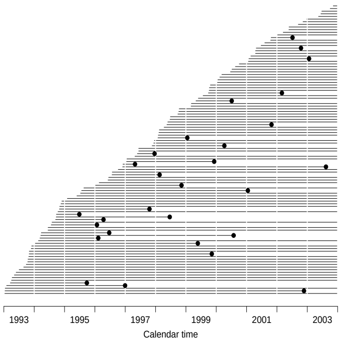
  $$D/Y$$

  D: no. events, Y no of person-years, in a prespecified time-frame.

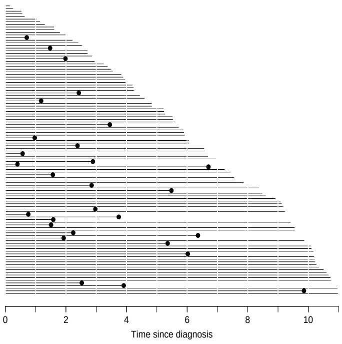
Empirical rates for individuals

- At the *individual* level we introduce the **empirical rate:** (d, y),
  — number of events ($d \in \{0, 1\}$) during y risk time.
- A person contributes several observations of (d, y), with associated covariate values.
- Empirical rates are **responses** in survival analysis.
- The timescale t is a **covariate** — varies within each individual:
  t: age, time since diagnosis, calendar time.
- Don't confuse with y — difference between two points on **any** timescale we may choose.

Empirical rates by calendar time.



Empirical rates by time since diagnosis.



Statistical inference: Likelihood

Two things needed:

- **Data** — what did we actually observe
  Follow-up for each person:
  Entry time, exit time, exit status, covariates
- **Model** — how was data generated
  Rates as a function of time:
  Probability machinery that generated data

Likelihood is the probability of observing the data, assuming the model is correct.

Maximum likelihood estimation is choosing parameters of the model that makes the likelihood maximal.

Likelihood from one person

The likelihood from several empirical rates from one individual is a product of conditional probabilities:

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

Each term refers to one empirical rate (d, y)
— $y = t_i - t_{i-1}$ and mostly $d = 0$.

t_i is the timescale (covariate):
age / time since entry / date of FU / DM duration

Likelihood for an empirical rate

Model: the rate is constant in the interval we are looking at.

The interval should sufficiently small for this assumption to be reasonable:

$$P\{\text{event in } (t, t+h] \mid \text{alive at } t\}/h = \lambda(t)$$

$$\begin{aligned} P\{\text{survive a timespan of } y\} &= \\ P\{\text{survive } n \text{ int's of length } y/n\} &= \left(1 - \lambda(t)\frac{y}{n}\right)^n \end{aligned}$$

$$\text{now, since:} \quad \lim_{n\to\infty}(1 + x/n)^n = \exp(x)$$

$$\Rightarrow \quad (1 - \lambda(t) \times y/n)^n \approx \exp\big(\lambda(t)y\big)$$

Likelihood for an empirical rate

Death probability is: $\pi = 1 - \mathrm{e}^{-\lambda y}$, so for $d = 0, 1$:

$$\begin{aligned} L(\lambda) &= P\{d \text{ events during } y \text{ time}\} = \pi^d(1-\pi)^{1-d} \\ &= (1 - \mathrm{e}^{-\lambda y})^d(\mathrm{e}^{-\lambda y})^{1-d} \\ &= \left(\frac{1 - \mathrm{e}^{-\lambda y}}{\mathrm{e}^{-\lambda y}}\right)^d (\mathrm{e}^{-\lambda y}) \approx (\lambda y)^d \mathrm{e}^{-\lambda y} \end{aligned}$$

since the first term is equal to $\mathrm{e}^{\lambda y} - 1 \approx \lambda y$.

Likelihood for an empirical rate

Log-likelihood:

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

The term $d\log(y)$ does not include λ, so the relevant part of the log-likelihood is:

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

. . . in a model with constant rate, λ

Poisson likelihood

The likelihood contributions from follow-up of **one** individual:

$$d_t \log\big(\lambda(t)\big) - \lambda(t) y_t, \quad t = t_1, \dots, t_n$$

is also the log-likelihood from several independent Poisson observations with mean $\lambda(t) y_t$, i.e. log-mean $\log\big(\lambda(t)\big) + \log(y_t)$

Therefore, analysis of the rates, (λ) can be based on a Poisson model with log-link applied to empirical rates where:

- d is the response variable.
- $\log(\lambda)$ is modelled by covariates
- $\log(y)$ is the offset variable.

Likelihood for follow-up of many subjects

Adding empirical rates over the follow-up of persons:

$$D = \sum d \qquad Y = \sum y \quad \Rightarrow \quad D\log(\lambda) - \lambda Y$$

- Persons are assumed independent
- Contribution from the same person are **conditionally** independent, hence give separate contributions to the log-likelihood.
- No need to correct for dependent observations; the likelihood is a product.

Likelihood

Probability of the data and the parameter:

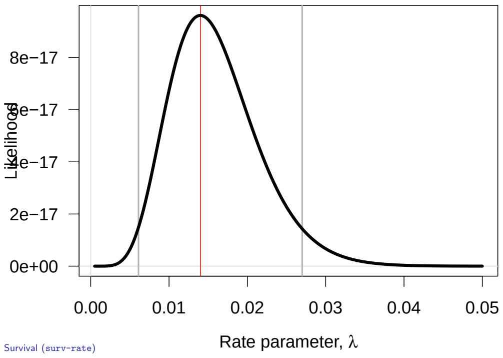
Assuming the rate (intensity) is constant, λ, the probability of observing 7 deaths in the course of 500 person-years:

$$\begin{aligned} P\{D = 7, Y = 500 | \lambda\} &= \lambda^D e^{\lambda Y} \times K \\ &= \lambda^7 e^{\lambda 500} \times K \\ &= L(\lambda | \text{data}) \end{aligned}$$

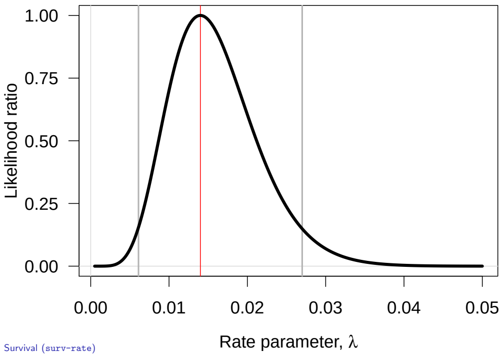
Best guess of λ is where this function is as large as possible.

Confidence interval is where it is not too far from the maximum

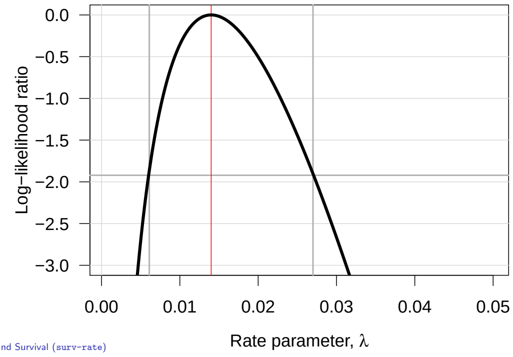
Likelihood function



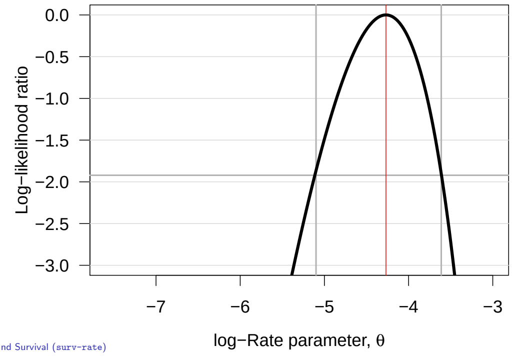
Likelihood function



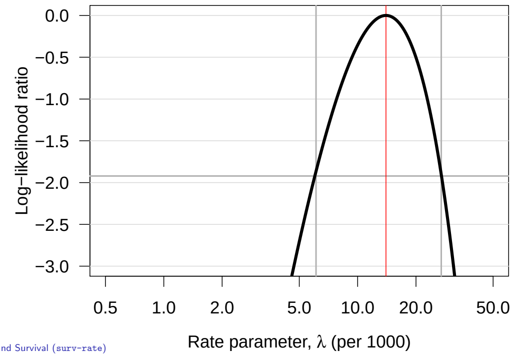
Likelihood function



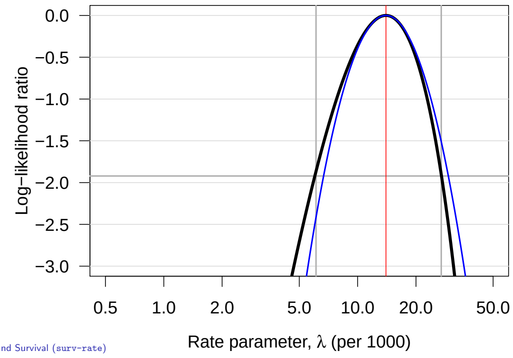
Likelihood function



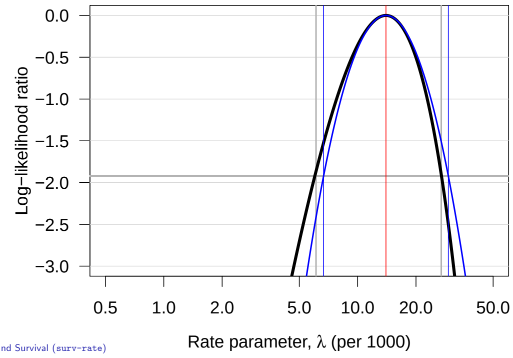
Likelihood function



Likelihood function



Likelihood function



Confidence interval for a rate

A 95% confidence interval for the log of a rate is:

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

Take the exponential to get the confidence interval for the rate:

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

Example

Suppose we have 17 deaths during 843.6 years of follow-up.

The rate is computed as:

$$\hat{\lambda} = D/Y = 17/843.7 = 0.0201 = 20.1 \text{ per 1000 years}$$

The confidence interval is computed as:

$$\hat{\lambda} \overset{\times}{\div} \text{erf} = 20.1 \overset{\times}{\div} \exp(1.96/\sqrt{D}) = (12.5, 32.4)$$

per 1000 person-years.

Ratio of two rates

If we have observations 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-rates, the $\log(\text{RR})$, 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:

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

Example

Suppose we in group 0 have 17 deaths during 843.6 years of follow-up in one group, and in group 1 have 28 deaths during 632.3 years.

The rate-ratio is computed as:

$$\begin{aligned} \text{RR} &= \hat{\lambda}_1/\hat{\lambda}_0 = (D_1/Y_1)/(D_0/Y_0) \\ &= (28/632.3)/(17/843.7) = 0.0443/0.0201 = 2.198 \end{aligned}$$

The 95% confidence interval is computed as:

$$\begin{aligned} \hat{\text{RR}} \overset{\times}{\div} \text{erf} &= 2.198 \overset{\times}{\div} \exp\big(1.96\sqrt{1/17 + 1/28}\big) \\ &= 2.198 \overset{\times}{\div} 1.837 = (1.20, 4.02) \end{aligned}$$

Example using R

Poisson likelihood, for observatio of 17 events in 843.7 PY:

```
 library( Epi )
 D <- 17 ; Y <- 843.7
 m1 <- glm( D ~ 1, offset=log(Y/1000), family=poisson)
 ci.exp( m1 )

            exp(Est.)     2.5%    97.5%
(Intercept)  20.14934 12.52605 32.41213
```

Poisson likelihood, two rates, or one rate and RR:

```
 D <- c(17,28) ; Y <- c(843.7,632.3) ; gg <- factor(0:1)
 m2 <- glm( D ~ gg, offset=log(Y/1000), family=poisson)
 ci.exp( m2 )

            exp(Est.)      2.5%     97.5%
(Intercept) 20.149342 12.526051 32.412130
gg1          2.197728  1.202971  4.015068
```

Example using R

Poisson likelihood, two rates, or one rate and RR:

```
 D <- c(17,28) ; Y <- c(843.7,632.3) ; gg <- factor(0:1)
 m2 <- glm( D ~ gg, offset=log(Y/1000), family=poisson)
 ci.exp( m2 )

            exp(Est.)      2.5%     97.5%
(Intercept) 20.149342 12.526051 32.412130
gg1          2.197728  1.202971  4.015068

 m3 <- glm( D ~ gg - 1, offset=log(Y/1000), family=poisson)
 ci.exp( m3 )

    exp(Est.)     2.5%    97.5%
gg0  20.14934 12.52605 32.41213
gg1  44.28278 30.57545 64.13525
```

You do it!

Survival analysis

- Response variable: Time to event, T
- Censoring time, Z
- We observe $(\min(T, Z), \delta = 1\{T < Z\})$.
- This gives time a special status, and mixes the response variable (risk)time with the covariate time(scale).
- Originates from clinical trials where everyone enters at time 0, and therefore $Y = T - 0 = T$

The life table method

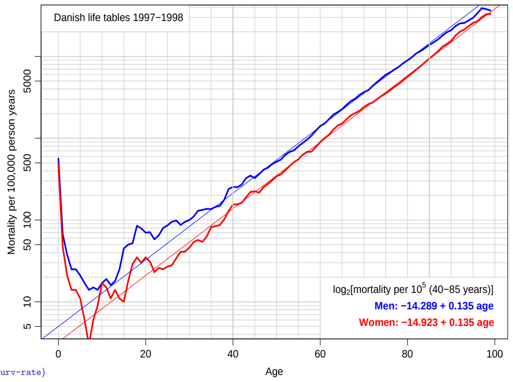
The simplest analysis is by the "life-table method":

interval i	alive n_i	dead d_i	cens. l_i	p_i
1	77	5	2	$5/(77-2/2)$=0.066
2	70	7	4	$7/(70-4/2)$=0.103
3	59	8	1	$8/(59-1/2)$=0.137

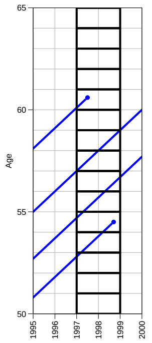
$$\begin{aligned} p_i &= P\,\{\text{death in interval } i\} = 1 - d_i/(n_i - l_i/2) \\ S(t) &= (1-p_1) \times \cdots \times (1-p_t) \end{aligned}$$

Population life table, DK 1997–98

	Men			Women		
a	$S(a)$	$\lambda(a)$	$\text{E}[\ell_{\text{res}}(a)]$	$S(a)$	$\lambda(a)$	$\text{E}[\ell_{\text{res}}(a)]$
0	1.00000	567	73.68	1.00000	474	78.65
1	0.99433	67	73.10	0.99526	47	78.02
2	0.99366	38	72.15	0.99479	21	77.06
3	0.99329	25	71.18	0.99458	14	76.08
4	0.99304	25	70.19	0.99444	14	75.09
5	0.99279	21	69.21	0.99430	11	74.10
6	0.99258	17	68.23	0.99419	6	73.11
7	0.99242	14	67.24	0.99413	3	72.11
8	0.99227	15	66.25	0.99410	6	71.11
9	0.99213	14	65.26	0.99404	9	70.12
10	0.99199	17	64.26	0.99395	17	69.12
11	0.99181	19	63.28	0.99378	15	68.14
12	0.99162	16	62.29	0.99363	11	67.15
13	0.99147	18	61.30	0.99352	14	66.15
14	0.99129	25	60.31	0.99338	11	65.16
15	0.99104	45	59.32	0.99327	10	64.17
16	0.99059	50	58.35	0.99317	18	63.18
17	0.99009	52	57.38	0.99299	29	62.19
18	0.98957	85	56.41	0.99270	35	61.21
19	0.98873	79	55.46	0.99235	30	60.23
20	0.98795	70	54.50	0.99205	35	59.24
21	0.98726	71	53.54	0.99170	31	58.27



Observations for the lifetable



Life table is based on person-years and deaths accumulated in a short period.

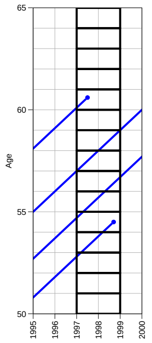
Age-specific rates — cross-sectional!

Survival function:

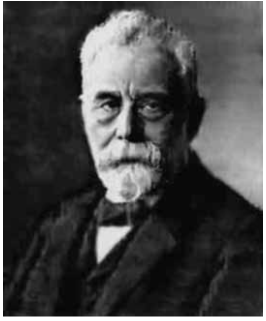
$$S(t) = \mathrm{e}^{-\int_0^t \lambda(a)\,\mathrm{d}a} = \mathrm{e}^{-\sum_0^t \lambda(a)}$$

— assumes stability of rates to be interpretable for actual persons.

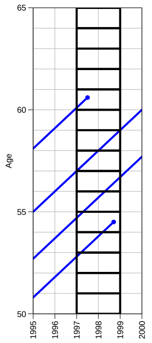
Observations for the lifetable

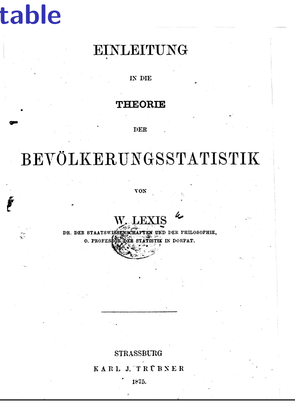


This is a **Lexis** diagram.



Observations for the lifetable





EINLEITUNG

IN DIE

THEORIE

DER

BEVÖLKERUNGSSTATISTIK

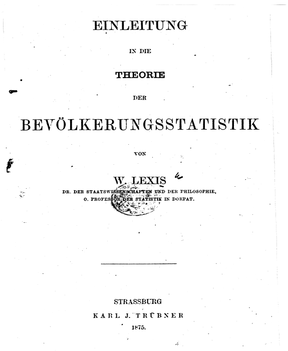
VON

W. LEXIS

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

STRASSBURG
KARL J. TRÜBNER
1875.

Observations for the lifetable



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.

Life table approach

The observation of interest is **not** the survival time of the **individual**.

- It is the **population** experience:
  - D: Deaths (events).
  - Y: Person-years (risk time).
- The classical lifetable analysis compiles these for prespecified intervals of age, and computes age-specific mortality **rates**.
- Data are collected crossectionally, but interpreted longitudinally.
- The **rates** are the basic building bocks — used for construction of:
  - RRs
  - cumulative measures (survival and risk)

Summary

- Follow-up studies observe time to event
- — in the form of **empirical rates**, (d, y) for small interval
- each interval (empirical rate) has covariates attached
- each interval contribute $d\log(\lambda) - \lambda y$
- — like a Poisson observation d with mean λy
- identical covariates: pool obervations to $D = \sum D$, $Y = \sum y$
- — like a Poisson observation D with mean λY
- the result is an **estimate** of the rate λ
- from a **model** where rates are constant within intervals — but varies between intervals.

Non-linear effects

Epidemiology with R
August 2015
NovoNordisk Epidemiology
`http://bendixcarstensen.com/Epi/Courses/NNepi/`

crv-mod

Testis cancer

Testis cancer in Denmark:

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

Cases, PY and rates

```
> stat.table( list(A=floor(A/10)*10,
+                  P=floor(P/10)*10),
+             list( D=sum(D),
+                   Y=sum(Y/1000),
+               rate=ratio(D,Y,10^5) ),
+             margins=TRUE, data=testisDK )

 ----------------------------------------------------------------------
        -------------------------------P-------------------------------
 A           1940      1950      1960      1970      1980      1990     Total
 ----------------------------------------------------------------------
 0          10.00      7.00     16.00     18.00      9.00     10.00     70.00
          2604.66   4037.31   3884.97   3820.88   3070.87   2165.54  19584.22
             0.38      0.17      0.41      0.47      0.29      0.46      0.36

 10         13.00     27.00     37.00     72.00     97.00     75.00    321.00
          2135.73   3505.19   4004.13   3906.08   3847.40   2260.97  19659.48
             0.61      0.77      0.92      1.84      2.52      3.32      1.63

 20        124.00    221.00    280.00    535.00    724.00    557.00   2441.00
          2225.55   2923.22   3401.65   4028.57   3941.18   2824.58  19344.74
             5.57      7.56      8.23     13.28     18.37     19.72     12.62
```

Linear effects in glm

How do rates depend on age?

```
> ml <- glm( D ~ A, offset=log(Y), family=poisson, data=testisDK )
> round( ci.lin( ml ), 4 )

            Estimate StdErr         z P    2.5%   97.5%
(Intercept)  -9.7755 0.0207 -472.3164 0 -9.8160 -9.7349
A             0.0055 0.0005   11.3926 0  0.0045  0.0064

> round( ci.exp( ml ), 4 )

            exp(Est.)   2.5%  97.5%
(Intercept)    0.0001 0.0001 0.0001
A              1.0055 1.0046 1.0064
```

Linear increase of log-rates by age

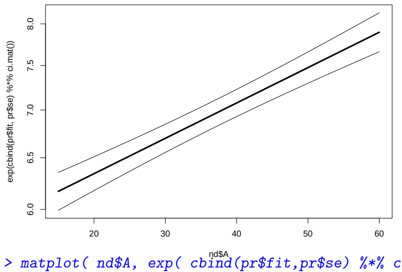
Linear effects in glm

```
> nd <- data.frame( A=15:60, Y=10^5 )
> pr <- predict( ml, newdata=nd, type="link", se.fit=TRUE )
> str( pr )

List of 3
 $ fit           : Named num [1:46] 1.82 1.83 1.83 1.84 1.84 ...
  ..- attr(*, "names")= chr [1:46] "1" "2" "3" "4" ...
 $ se.fit        : Named num [1:46] 0.015 0.0146 0.0143 0.014 0.0137 ...
  ..- attr(*, "names")= chr [1:46] "1" "2" "3" "4" ...
 $ residual.scale: num 1

> ci.mat()

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

> matplot( nd$A, exp( cbind(pr$fit,pr$se) %*% ci.mat() ),
+          type="l", lty=1, lwd=c(3,1,1), col="black", log="y" )
```

Linear effects in glm

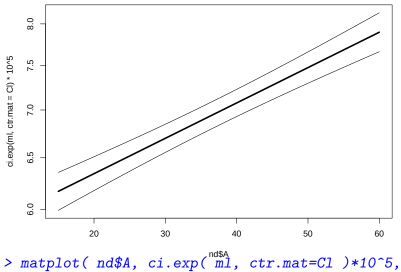
```
> round( ci.lin( ml ), 4 )

            Estimate StdErr         z P    2.5%   97.5%
(Intercept)  -9.7755 0.0207 -472.3164 0 -9.8160 -9.7349
A             0.0055 0.0005   11.3926 0  0.0045  0.0064

> Cl <- cbind( 1, nd$A )
> head( Cl )

     [,1] [,2]
[1,]    1   15
[2,]    1   16
[3,]    1   17
[4,]    1   18
[5,]    1   19
[6,]    1   20

> matplot( nd$A, ci.exp( ml, ctr.mat=Cl ),
+          type="l", lty=1, lwd=c(3,1,1), col="black", log="y" )
```

Linear effects in glm



```
> matplot( nd$A, exp( cbind(pr$fit,pr$se) %*% ci.mat() ),
+          type="l", lty=1, lwd=c(3,1,1), col="black", log="y" )
```

Linear effects in glm



```
> matplot( nd$A, ci.exp( ml, ctr.mat=Cl )*10^5,
+          type="l", lty=1, lwd=c(3,1,1), col="black", log="y" )
```

Quadratic effects in glm

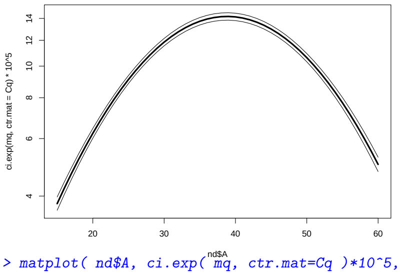
How do rates depend on age?

```
> mq <- glm( D ~ A + I(A^2),
+              offset=log(Y), family=poisson, data=testisDK )
> round( ci.lin( mq ), 4 )

            Estimate StdErr         z P     2.5%    97.5%
(Intercept) -12.3656 0.0596 -207.3611 0 -12.4825 -12.2487
A             0.1806 0.0033   54.8290 0   0.1741   0.1871
I(A^2)       -0.0023 0.0000  -53.7006 0  -0.0024  -0.0022

> round( ci.exp( mq ), 4 )

            exp(Est.)   2.5%  97.5%
(Intercept)    0.0000 0.0000 0.0000
A              1.1979 1.1902 1.2057
I(A^2)         0.9977 0.9976 0.9978
```

Quadratic effect in glm

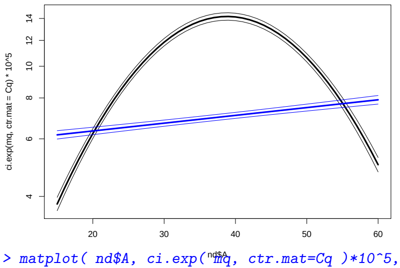
```
> round( ci.lin( mq ), 4 )

            Estimate StdErr         z P     2.5%    97.5%
(Intercept) -12.3656 0.0596 -207.3611 0 -12.4825 -12.2487
A             0.1806 0.0033   54.8290 0   0.1741   0.1871
I(A^2)       -0.0023 0.0000  -53.7006 0  -0.0024  -0.0022

> Cq <- cbind( 1, 15:60, (15:60)^2 )
> head( Cq )

     [,1] [,2] [,3]
[1,]    1   15  225
[2,]    1   16  256
[3,]    1   17  289
[4,]    1   18  324
[5,]    1   19  361
[6,]    1   20  400

> matplot( nd$A, ci.exp( mq, ctr.mat=Cq )*10^5,
+          type="l", lty=1, lwd=c(3,1,1), col="black", log="y" )
```

Quadratic effect in glm



```
> matplot( nd$A, ci.exp( mq, ctr.mat=Cq )*10^5,
+          type="l", lty=1, lwd=c(3,1,1), col="black", log="y" )
```

Quadratic effect in glm

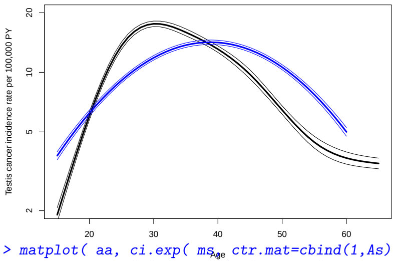


```
> matplot( nd$A, ci.exp( mq, ctr.mat=Cq )*10^5,
+          type="l", lty=1, lwd=c(3,1,1), col="black", log="y" )
> matlines( nd$A, ci.exp( ml, ctr.mat=Cl )*10^5,
+           type="l", lty=1, lwd=c(3,1,1), col="blue" )
```

Spline effects in glm

```
> library( splines )
> aa <- 15:65
> ms <- glm( D ~ Ns(A,knots=seq(15,65,10)),
+                offset=log(Y), family=poisson, data=testisDK )
> round( ci.exp( ms ), 3 )

                                exp(Est.)   2.5%  97.5%
(Intercept)                         0.000  0.000  0.000
Ns(A, knots = seq(15, 65, 10))1     8.548  7.650  9.551
Ns(A, knots = seq(15, 65, 10))2     5.706  4.998  6.514
Ns(A, knots = seq(15, 65, 10))3     1.002  0.890  1.128
Ns(A, knots = seq(15, 65, 10))4    14.402 11.896 17.436
Ns(A, knots = seq(15, 65, 10))5     0.466  0.429  0.505

> As <- Ns( aa, knots=seq(15,65,10) )
> head( As )

                1 2           3          4           5
[1,] 0.0000000000 0  0.00000000 0.00000000  0.00000000
[2,] 0.0001666667 0 -0.02527011 0.07581034 -0.05054022
[3,] 0.0013333333 0 -0.05003313 0.15009940 -0.10006626
[4,] 0.0045000000 0 -0.07378197 0.22134590 -0.14756393
```

Spline effects in glm

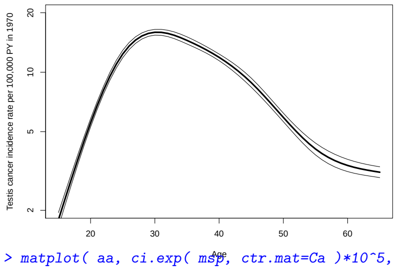


```
> matplot( aa, ci.exp( ms, ctr.mat=cbind(1,As) )*10^5,
+          log="y", xlab="Age", ylab="Testis cancer incidence rate per 100,000 PY
+          type="l", lty=1, lwd=c(3,1,1), col="black", ylim=c(2,20) )
> matlines( nd$A, ci.exp( mq, ctr.mat=Cq )*10^5,
+           type="l", lty=1, lwd=c(3,1,1), col="blue" )
```

Adding a linear period effect

```
> msp <- glm( D ~ Ns(A,knots=seq(15,65,10)) + P,
+                   offset=log(Y), family=poisson, data=testisDK )
> round( ci.lin( msp ), 3 )

                             Estimate StdErr       z     P    2.5%   97.5%
(Intercept)                   -58.105  1.444 -40.229 0.000 -60.935 -55.274
Ns(A, knots = seq(15, 65, 10))1   2.120  0.057  37.444 0.000   2.009   2.231
Ns(A, knots = seq(15, 65, 10))2   1.700  0.068  25.157 0.000   1.567   1.832
Ns(A, knots = seq(15, 65, 10))3   0.007  0.060   0.110 0.913  -0.112   0.125
Ns(A, knots = seq(15, 65, 10))4   2.596  0.097  26.631 0.000   2.405   2.787
Ns(A, knots = seq(15, 65, 10))5  -0.780  0.042 -18.748 0.000  -0.861  -0.698
P                               0.024  0.001  32.761 0.000   0.023   0.025

> Ca <- cbind( 1, Ns( aa, knots=seq(15,65,10) ), 1970 )
> head( Ca )

             1 2           3          4           5
[1,] 1 0.0000000000 0  0.00000000 0.00000000  0.00000000 1970
[2,] 1 0.0001666667 0 -0.02527011 0.07581034 -0.05054022 1970
[3,] 1 0.0013333333 0 -0.05003313 0.15009940 -0.10006626 1970
[4,] 1 0.0045000000 0 -0.07378197 0.22134590 -0.14756393 1970
[5,] 1 0.0106666667 0 -0.09600952 0.28802857 -0.19201905 1970
```

Adding a linear period effect

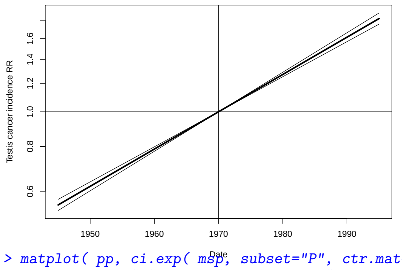


```
> matplot( aa, ci.exp( msp, ctr.mat=Ca )*10^5,
+          log="y", xlab="Age",
+          ylab="Testis cancer incidence rate per 100,000 PY in 1970",
+          type="l", lty=1, lwd=c(3,1,1), col="black", ylim=c(2,20) )
```

The period effect

```
> round( ci.lin( msp ), 3 )

                             Estimate StdErr       z     P    2.5%   97.5%
(Intercept)                   -58.105  1.444 -40.229 0.000 -60.935 -55.274
Ns(A, knots = seq(15, 65, 10))1   2.120  0.057  37.444 0.000   2.009   2.231
Ns(A, knots = seq(15, 65, 10))2   1.700  0.068  25.157 0.000   1.567   1.832
Ns(A, knots = seq(15, 65, 10))3   0.007  0.060   0.110 0.913  -0.112   0.125
Ns(A, knots = seq(15, 65, 10))4   2.596  0.097  26.631 0.000   2.405   2.787
Ns(A, knots = seq(15, 65, 10))5  -0.780  0.042 -18.748 0.000  -0.861  -0.698
P                               0.024  0.001  32.761 0.000   0.023   0.025

> pp <- 1945:1995
> Cp <- cbind( pp ) - 1970
> head( Cp )

      pp
[1,] -25
[2,] -24
[3,] -23
[4,] -22
[5,] -21
[6,] -20
```

Period effect

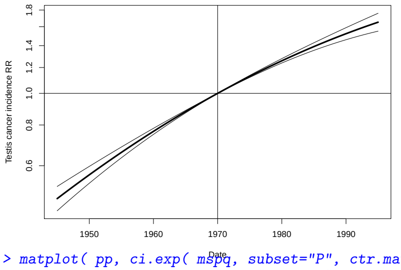


```
> matplot( pp, ci.exp( msp, subset="P", ctr.mat=Cp ),
+          log="y", xlab="Date", ylab="Testis cancer incidence RR",
+          type="l", lty=1, lwd=c(3,1,1), col="black" )
> abline( h=1, v=1970 )
```

A quadratic period effect

```
> mspq <- glm( D ~ Ns(A,knots=seq(15,65,10)) + P + I(P^2),
+                   offset=log(Y), family=poisson, data=testisDK )
> round( ci.exp( mspq ), 3 )

                             exp(Est.)   2.5%  97.5%
(Intercept)                      0.000  0.000  0.000
Ns(A, knots = seq(15, 65, 10))1    8.356  7.478  9.337
Ns(A, knots = seq(15, 65, 10))2    5.513  4.829  6.295
Ns(A, knots = seq(15, 65, 10))3    1.006  0.894  1.133
Ns(A, knots = seq(15, 65, 10))4   13.439 11.101 16.269
Ns(A, knots = seq(15, 65, 10))5    0.458  0.422  0.497
P                                2.189  1.457  3.291
I(P^2)                           1.000  1.000  1.000

> pp <- 1945:1995
> Cq <- cbind( pp-1970, pp^2-1970^2 )
> head( Cq )

     [,1]   [,2]
[1,]  -25 -97875
[2,]  -24 -93984
[3,]  -23 -90091
```

A quadratic period effect



```
> matplot( pp, ci.exp( mspq, subset="P", ctr.mat=Cq ),
+          log="y", xlab="Date", ylab="Testis cancer incidence RR",
+          type="l", lty=1, lwd=c(3,1,1), col="black" )
> abline( h=1, v=1970 )
```

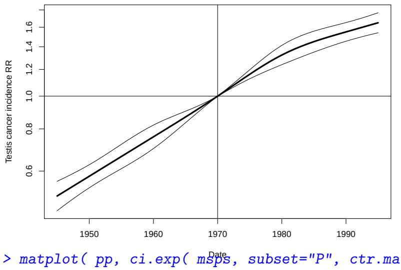
A spline period effect

```
> msps <- glm( D ~ Ns(A,knots=seq(15,65,10)) +
+                  Ns(P,knots=seq(1950,1990,10)),
+                  offset=log(Y), family=poisson, data=testisDK )
> round( ci.exp( msps ), 3 )

                                  exp(Est.)   2.5%  97.5%
(Intercept)                           0.000  0.000  0.000
Ns(A, knots = seq(15, 65, 10))1         8.327  7.452  9.305
Ns(A, knots = seq(15, 65, 10))2         5.528  4.842  6.312
Ns(A, knots = seq(15, 65, 10))3         1.007  0.894  1.133
Ns(A, knots = seq(15, 65, 10))4        13.447 11.107 16.279
Ns(A, knots = seq(15, 65, 10))5         0.458  0.422  0.497
Ns(P, knots = seq(1950, 1990, 10))1     1.711  1.526  1.918
Ns(P, knots = seq(1950, 1990, 10))2     2.190  2.028  2.364
Ns(P, knots = seq(1950, 1990, 10))3     3.222  2.835  3.661
Ns(P, knots = seq(1950, 1990, 10))4     2.299  2.149  2.459
```

A spline period effect

```
> pp <- 1945:1995
> Cs <- Ns(                pp  ,knots=seq(1950,1990,10))
> Cr <- Ns(rep(1970,length(pp)),knots=seq(1950,1990,10))
> head( Cs )

     1          2           3          4
[1,] 0 0.12677314 -0.38031941 0.25354628
[2,] 0 0.10141851 -0.30425553 0.20283702
[3,] 0 0.07606388 -0.22819165 0.15212777
[4,] 0 0.05070926 -0.15212777 0.10141851
[5,] 0 0.02535463 -0.07606388 0.05070926
[6,] 0 0.00000000  0.00000000 0.00000000

> head( Cr )

             1         2         3          4
[1,] 0.6666667 0.1125042 0.1624874 -0.1083249
[2,] 0.6666667 0.1125042 0.1624874 -0.1083249
[3,] 0.6666667 0.1125042 0.1624874 -0.1083249
[4,] 0.6666667 0.1125042 0.1624874 -0.1083249
[5,] 0.6666667 0.1125042 0.1624874 -0.1083249
[6,] 0.6666667 0.1125042 0.1624874 -0.1083249
```

Period effect

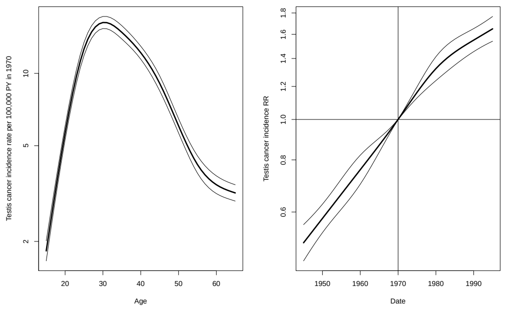


```
> matplot( pp, ci.exp( msps, subset="P", ctr.mat=Cs-Cr ),
+          log="y", xlab="Date", ylab="Testis cancer incidence RR",
+          type="l", lty=1, lwd=c(3,1,1), col="black" )
> abline( h=1, v=1970 )
```

Period effect

```
> par( mfrow=c(1,2) )
> Cap <- cbind( 1, Ns(               aa  ,knots=seq(15,65,10)),
+                  Ns(rep(1970,length(aa)),knots=seq(1950,1990,10)) )
> matplot( aa, ci.exp( msps, ctr.mat=Cap )*10^5,
+          log="y", xlab="Age",
+          ylab="Testis cancer incidence rate per 100,000 PY in 1970",
+          type="l", lty=1, lwd=c(3,1,1), col="black" )
> matplot( msps, subset="P", ctr.mat=Cs-Cr ),
+          log="y", xlab="Date", ylab="Testis cancer incidence RR",
+          type="l", lty=1, lwd=c(3,1,1), col="black" )
> abline( h=1, v=1970 )
```

Age and period effect



Age and period effect with `ci.exp`

- In rate models there is always one term with the **rate** dimension — usually **age**
- But it must refer to a specific **reference** value for all **other** variables (P).
- **All** parameters must be used in computing rates, at reference value.
- For the "other" variables, report the RR **relative** to the reference point.
- Only parameters relevant for the variable (P) used.
- Contrast matrix is a **difference** between prediction points and the reference point.

Classical estimators

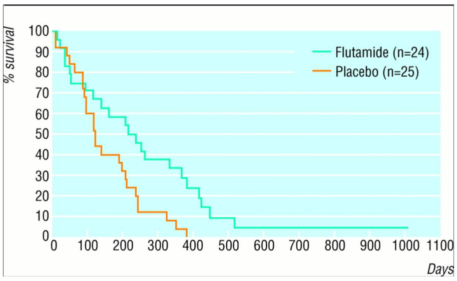
Epidemiology with R
August 2015
NovoNordisk Epidemiology
`http://bendixcarstensen.com/Epi/Courses/NNepi/`

`km-na`

The Kaplan-Meier Method

- The most common method of estimating the survival function.
- A non-parametric method.
- Divides time into small intervals where the intervals are defined by the unique times of failure (death).
- Based on conditional probabilities as we are interested in the probability a subject surviving the next time interval given that they have survived so far.

Example of KM Survival Curve from BMJ

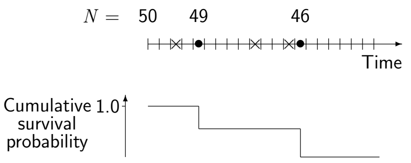


BMJ 1998;316:1935-1938

Kaplan-Meier curve from an RCT of patients with pancreatic cancer

Kaplan–Meier method illustrated

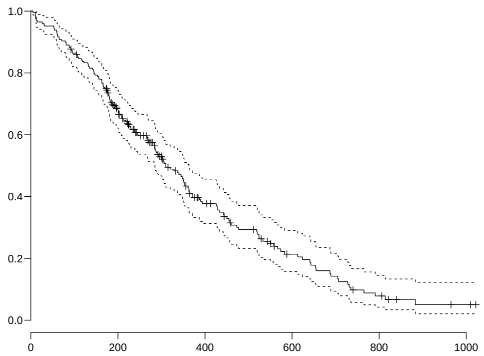
(• = failure and × = censored):

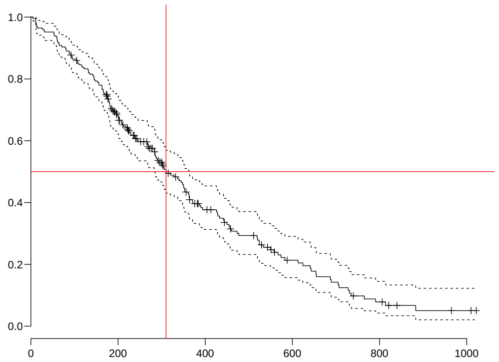


- Steps caused by multiplying by $(1 - 1/49)$ and $(1 - 1/46)$ respectively
- Late entry can also be dealt with

Using R: `Surv()`

```
library( survival )
data( lung )
head( lung, 3 )

  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

with( lung, Surv( time, status==2 ) )[1:10]

 [1]  306   455  1010+  210   883  1022+  310   361   218   166

( s.km <- survfit( Surv( time, status==2 ) ~ 1 , data=lung ) )

Call: survfit(formula = Surv(time, status == 2) ~ 1, data = lung)

      n  events  median 0.95LCL 0.95UCL
    228     165     310     285     363

plot( s.km )
abline( v=310, h=0.5, col="red" )
```





The Cox model

Epidemiology with R
August 2015
NovoNordisk Epidemiology
`http://bendixcarstensen.com/Epi/Courses/NNepi/`

`cox`

Proportional Hazards model

Model hazard rate as function of time (t) and covariates ($\mathbf{x}$)

$$\lambda_i(t, \mathbf{x}_i) = \lambda_0(t) \exp\left(\beta_1 x_{1i} + \beta_2 x_{2i} + \ldots\right)$$

- $\lambda_i(t, \mathbf{x}_i)$ is the hazard rate for the i^{th} person.
- $\mathbf{x}_i = (x_{1i}, \ldots, x_{pi})$ are covariate values for ith person.
- $\lambda_0(t)$ is the **baseline hazard** function
  — a non-linear effect of the **covariate** t.
- $\beta_1 x_{1i} + \beta_2 x_{2i} + \ldots$ is the linear predictor.

Cox-likelihood

The partial likelihood for the regression parameters:

$$\ell(\beta) = \sum_{\text{death times}} \log\left(\frac{e^{x_{\text{death}}\beta}}{\sum_{i \in \mathcal{R}_t} e^{x_i \beta}}\right)$$

- This is David Cox's invention.
- Extremely efficient from a computational point of view.
- The baseline hazard is bypassed (profiled out).
- Ony estimates β — the RR-paraeters

Interpreting regression coefficient

- x_1 binary (only $0/1$ values):

$$\lambda_i(t) = \lambda_0(t) \exp\left(\beta_1 x_{1i}\right)$$

- The hazard rate when $x_1 = 0$ is $\lambda_0(t)$.
- The hazard rate when $x_1 = 1$ is $\lambda_0(t)\exp(\beta_1)$.
- The hazard ratio is therefore

$$\frac{\lambda_0(t)\exp(\beta_1)}{\lambda_0(t)} = \exp(\beta_1)$$

- The $\lambda_0(t)$ cancels: β_1 is the log hazard ratio.
- Exponentiate β_1 to get the hazard ratio (HR, RR).

Interpreting regression coefficient

- x_2 numerical (any values say btw 100 and 200):

$$\lambda_i(t) = \lambda_0(t) \exp\left(\beta_2 x_{2i}\right)$$

- The hazard rate when $x_1 = 130$ is $\lambda_0(t)\exp(\beta_2 \times 130)$.
- The hazard rate when $x_2 = 140$ is $\lambda_0(t)\exp(\beta_2 \times 140)$.
- The hazard ratio is therefore

$$\frac{\lambda_0(t)\exp(\beta_2 \times 140)}{\lambda_0(t)\exp(\beta_2 \times 130)} = \exp(\beta_2 \times 10)$$

- The $\lambda_0(t)$ cancels: $\beta_2 \times 10$ is the log hazard ratio.
- Exponentiate $\beta_2 \times 10$ to get the hazard ratio for a difference of 10 in x_2.

Interpreting regression coefficients

- If x_j is binary $\exp(\beta_j)$ is the estimated hazard ratio for subjects corresponding to $x_j = 1$ compared to those where $x_j = 0$.
- If x_j is continuous $\exp(\beta_j)$ is the estimated increase/decrease in the hazard rate for a unit change in x_j.
- With more than one covariate interpretation is similar, i.e. $\exp(\beta_j)$ is the hazard ratio for subjects who **only** differ with respect to covariate x_j.

Fitting a Cox- model in R

```
 library( survival )
 data(bladder)
 bladder <- subset( bladder, enum<2 )
 head( bladder)

   id rx number size stop event enum
1   1  1      1    3    1     0    1
5   2  1      2    1    4     0    1
9   3  1      1    1    7     0    1
13  4  1      5    1   10     0    1
17  5  1      4    1    6     1    1
21  6  1      1    1   14     0    1
```

```
 ?bladder
```

Fitting a Cox-model in R

```
 c0 <- coxph( Surv(stop,event) ~ number + size + factor(rx), data=bladder )
 c0

Call:
coxph(formula = Surv(stop, event) ~ number + size + factor(rx),
    data = bladder)


               coef exp(coef) se(coef)     z      p
number       0.2382    1.2689   0.0759  3.14 0.0017
size         0.0696    1.0721   0.1016  0.69 0.4931
factor(rx)2 -0.5260    0.5910   0.3158 -1.67 0.0958

Likelihood ratio test=9.92  on 3 df, p=0.0193
n= 85, number of events= 47
```

Plotting the base survival in R

```
 plot( survfit(c0) )
 lines( survfit(c0), conf.int=F, lwd=3 )
```

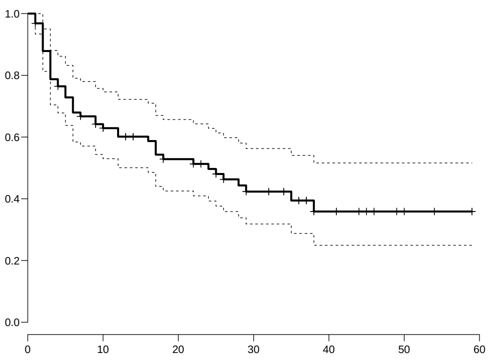
The `plot.coxph` plots the `survival` curve for a person with an **average** covariate value

— which is **not** the average survival for the population considered. . .

— and not necessarily meaningful:

```
 with( bladder, c( nb=mean(number), sz=mean(size), rx=mean(rx) ) )

      nb       sz       rx
2.105882 2.011765 1.447059
```

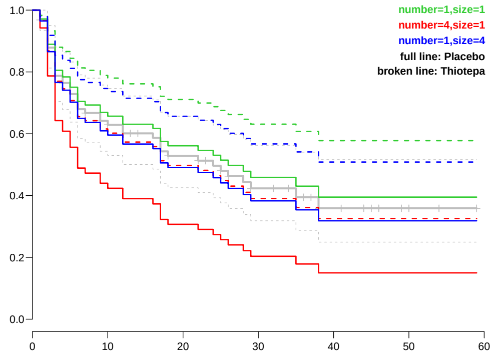
`rx=1`: Placebo,   `rx=2`: Thiotepa



Plotting the base survival in R

You can plot the survival curve for specific values of the covariates, using the `newdata=` argument:

```
 plot( survfit(c0) )
 lines( survfit(c0), conf.int=F, lwd=3 )
 lines( survfit(c0, newdata=data.frame(number=1,size=1,rx=1)),
        lwd=c(3,1,1), lty=c(1,1,1), col="limegreen" )
 text( par("usr")[2]*0.98, 1.00, "number=1,size=1,rx=1",
       col="limegreen", font=2, adj=1 )
```



Who needs the Cox-model anyway?

Epidemiology with R
August 2015
NovoNordisk Epidemiology
http://bendixcarstensen.com/Epi/Courses/NNepi/

WntCma

The proportional hazards model

$$\lambda(t,x) = \lambda_0(t) \times \exp(x'\beta)$$

A model for the rate as a function of t and x.

The covariate t has a special status:

- Computationally, because all individuals contribute to (some of) the range of t.
- Conceptually it is less clear — t is but a covariate that varies within individual.

Cox-likelihood

The (partial) log-likelihood for the regression parameters:

$$\ell(\beta) = \sum_{\text{death times}} \log\left(\frac{e^{\eta_{\text{death}}}}{\sum_{i\in\mathcal{R}_t} e^{\eta_i}}\right)$$

is also a **profile likelihood** in the model where observation time has been subdivided in small pieces (empirical rates) and each small piece provided with its own parameter:

$$\log\big(\lambda(t,x)\big) = \log\big(\lambda_0(t)\big) + x'\beta = \alpha_t + \eta$$

The components are baseline hazard and the RR function

The Cox-likelihood as profile likelihood

- Regression parameters describing the effect of covariates (other than the chosen underlying time scale).
- One parameter per death time to describe the effect of time (i.e. the chosen timescale).

$$\log\big(\lambda(t,x_i)\big) = \log\big(\lambda_0(t)\big) + \beta_1 x_{1i} + \cdots + \beta_p x_{pi} = \alpha_t + \eta_i$$

- Profile likelihood:
  - Derive estimates of α_t as function of data and βs
  - Insert the expressions for α_t in likelihood, now only a function of data and βs
  - Turns out to be Cox's partial likelihood

What the Cox-model really is

Taking the life-table approach *ad absurdum* by:

- dividing time very finely,
- modelling one covariate, the time-scale, with one parameter per distinct value,
- profiling these parameters out and maximizing the profile likelihood,
- regression parameters are the same as in the full model with all the interval-specific parameters
- Subsequently, one may recover the effect of the timescale by smoothing an estimate of the cumulative sum of these.

Sensible modelling

Replace the α_ts by a parmetric function $f(t)$ with a limited number of parameters, for example:

- Piecewise constant
- Splines (linear, quadratic or cubic)
- Fractional polynomials

Use Poisson modelling software on a dataset of empirical rates for small intervals (ys).

Splitting the dataset

- The Poisson approach needs a dataset of empirical rates with small values of y.
- Larger than the original: each individual contributes many empirical rates. From each empirical rate we get:
  - Poisson-response d
  - Risk time y
  - Covariate value for the timescale (time since entry, current age, current date, ...)
  - other covariates

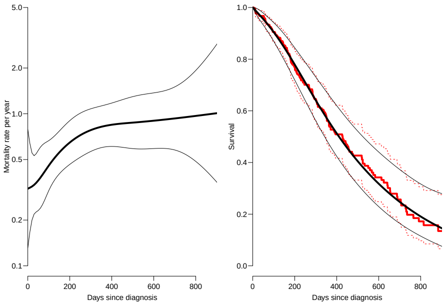
Example: Mayo Clinic lung cancer I

```
> library( survival ) ; library( Epi )
> data( lung )
> head( lung )

  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

> Lx <- Lexis( exit=list( tfd=time), exit.status=(status==2), data=lung )

NOTE: entry is assumed to be 0 on the tfd timescale.

> summary( Lx, scale=365.25 )
```

Example: Mayo Clinic lung cancer II

```
Transitions:
     To
From    FALSE TRUE  Records:  Events: Risk time:  Persons:
  FALSE    63  165       228      165     190.54       228

> Sx <- splitLexis( Lx, "tfd", breaks=c(0,unique(Lx$time)) )
> summary( Sx, scale=365.25 )

Transitions:
     To
From    FALSE TRUE  Records:  Events: Risk time:  Persons:
  FALSE 19857  165     20022      165     190.54       228
```

Smoothing by natural splines with 5 parameters, knots at $0, 25, 100, 500, 1000$ days:



Practical: Cox and Poisson modelling

Follow-up data

Epidemiology with R
August 2015
NovoNordisk Epidemiology
`http://bendixcarstensen.com/Epi/Courses/NNepi/`

`time-split`

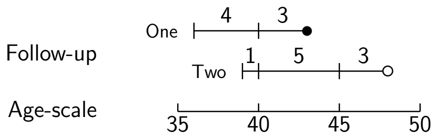
Follow-up and rates

- Follow-up studies:
  - D — events, deaths
  - Y — person-years
  - $\lambda = D/Y$ rates
- Rates differ between persons.
- Rates differ within persons:
  - By age
  - By calendar time
  - By disease duration
  - ...
- Multiple timescales.
- Multiple states (little boxes — later)

Stratification by age

If follow-up is rather short, age at entry is OK for age-stratification.

If follow-up is long, use stratification by categories of **current age**, both for:
No. of events, D, and Risk time, Y.



Representation of follow-up data

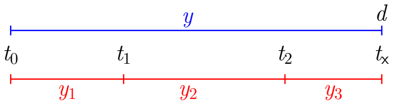
A cohort or follow-up study records:
Events and **Risk time**.

The outcome is thus **bivariate**: (d, y)

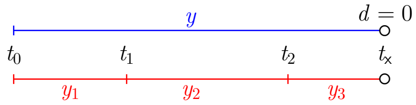
Follow-up **data** for each individual must therefore have (at least) three variables:

Date of entry	`entry`	date variable
Date of exit	`exit`	date variable
Status at exit	`fail`	indicator $(0/1)$

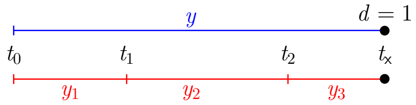
Specific for each **type** of outcome.



Probability	log-Likelihood
$\mathrm{P}(d \text{ at } t_x \mid \text{entry } t_0)$	$d\log(\lambda) - \lambda y$
$= \mathrm{P}(\text{surv } t_0 \to t_1 \mid \text{entry } t_0)$	$= 0\log(\lambda) - \lambda y_1$
$\times\, \mathrm{P}(\text{surv } t_1 \to t_2 \mid \text{entry } t_1)$	$+\, 0\log(\lambda) - \lambda y_2$
$\times\, \mathrm{P}(d \text{ at } t_x \mid \text{entry } t_2)$	$+\, d\log(\lambda) - \lambda y_3$



Probability	log-Likelihood
$\mathrm{P}(\text{surv } t_0 \to t_x \mid \text{entry } t_0)$	$0\log(\lambda) - \lambda y$
$= \mathrm{P}(\text{surv } t_0 \to t_1 \mid \text{entry } t_0)$	$= 0\log(\lambda) - \lambda y_1$
$\times\, \mathrm{P}(\text{surv } t_1 \to t_2 \mid \text{entry } t_1)$	$+\, 0\log(\lambda) - \lambda y_2$
$\times\, \mathrm{P}(\text{surv } t_2 \to t_x \mid \text{entry } t_2)$	$+\, 0\log(\lambda) - \lambda y_3$



Probability	log-Likelihood
$\mathrm{P}(\text{event at } t_x \mid \text{entry } t_0)$	$1\log(\lambda) - \lambda y$
$= \mathrm{P}(\text{surv } t_0 \to t_1 \mid \text{entry } t_0)$	$= 0\log(\lambda) - \lambda y_1$
$\times\, \mathrm{P}(\text{surv } t_1 \to t_2 \mid \text{entry } t_1)$	$+\, 0\log(\lambda) - \lambda y_2$
$\times\, \mathrm{P}(\text{event at } t_x \mid \text{entry } t_2)$	$+\, 1\log(\lambda) - \lambda y_3$

Dividing time into bands:

If we want to put D and Y into intervals on the timescale we must know:

Origin: The date where the time scale is 0:

- Age — 0 at date of birth
- Disease duration — 0 at date of diagnosis
- Occupation exposure — 0 at date of hire

Intervals: How should it be subdivided:

- 1-year classes? 5-year classes?
- Equal length?

Aim: Separate rate in each interval

Example: cohort with 3 persons:

```
Id       Bdate      Entry       Exit  St
 1  14/07/1952 04/08/1965 27/06/1997   1
 2  01/04/1954 08/09/1972 23/05/1995   0
 3  10/06/1987 23/12/1991 24/07/1998   1
```

- Age bands: 10-years intervals of current age.
- Split Y for every subject accordingly
- Treat each segment as a separate unit of observation.
- Keep track of exit status in each interval.

Splitting the follow up

	subj. 1	subj. 2	subj. 3
Age at **E**ntry:	13.06	18.44	4.54
Age at e**X**it:	44.95	41.14	11.12
Status at exit:	Dead	Alive	Dead
Y	31.89	22.70	6.58
D	1	0	1

	subj. 1		subj. 2		subj. 3		$\sum$	
Age	Y	D	Y	D	Y	D	Y	D
0–	0.00	0	0.00	0	5.46	0	5.46	0
10–	6.94	0	1.56	0	1.12	1	8.62	1
20–	10.00	0	10.00	0	0.00	0	20.00	0
30–	10.00	0	10.00	0	0.00	0	20.00	0
40–	4.95	1	1.14	0	0.00	0	6.09	1
$\sum$	31.89	1	22.70	0	6.58	1	60.17	2

Splitting the follow-up

```
id       Bdate       Entry        Exit  St     risk  int

 1  14/07/1952  03/08/1965  14/07/1972   0   6.9432   10
 1  14/07/1952  14/07/1972  14/07/1982   0  10.0000   20
 1  14/07/1952  14/07/1982  14/07/1992   0  10.0000   30
 1  14/07/1952  14/07/1992  27/06/1997   1   4.9528   40
 2  01/04/1954  08/09/1972  01/04/1974   0   1.5606   10
 2  01/04/1954  01/04/1974  31/03/1984   0  10.0000   20
 2  01/04/1954  31/03/1984  01/04/1994   0  10.0000   30
 2  01/04/1954  01/04/1994  23/05/1995   0   1.1417   40
 3  10/06/1987  23/12/1991  09/06/1997   0   5.4634    0
 3  10/06/1987  09/06/1997  24/07/1998   1   1.1211   10
```

Keeping track of calendar time too?

Timescales

- A timescale is a variable that varies **deterministically** *within* each person during follow-up:
  - Age
  - Calendar time
  - Time since treatment
  - Time since relapse
- All timescales advance at the same pace (1 year per year . . .)
- Note: Cumulative exposure is **not** a timescale.

Follow-up on several timescales

- The risk-time is the same on all timescales
- Only need the entry point on each time scale:
  - Age at entry.
  - Date of entry.
  - Time since treatment at entry.
    — if time of treatment is the entry, this is 0 for all.
- Response variable in analysis of rates:

$$(d, y) \qquad (\text{event}, \text{duration})$$

- Covariates in analysis of rates:
  - timescales
  - other (fixed) measurements

Follow-up data in Epi — Lexis objects

A follow-up study:

```
> round( th, 2 )
    id sex birthdat contrast injecdat volume exitdat exitstat
1    1   2  1916.61        1  1938.79     22 1976.79        1
2  640   2  1896.23        1  1945.77     20 1964.37        1
3 3425   1  1886.97        2  1955.18      0 1956.59        1
4 4017   2  1936.81        2  1957.61      0 1992.14        2
...
```

Timescales of interest:

- Age
- Calendar time
- Time since injection

Definition of Lexis object

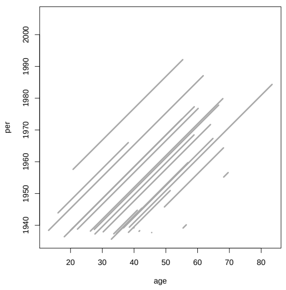
```
> thL <- Lexis( entry = list( age = injecdat-birthdat,
+                             per = injecdat,
+                             tfi = 0 ),
+                exit = list( per = exitdat ),
+         exit.status = as.numeric(exitstat==1),
+                data = th )
```

entry is defined on **three** timescales,
but exit is only defined on **one** timescale:
Follow-up time is the same on all timescales:

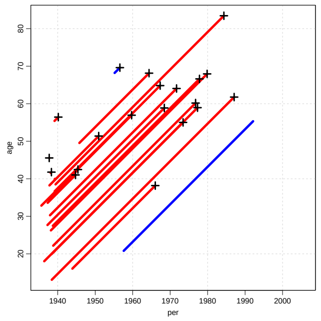
`exitdat - injecdat`

The looks of a Lexis object

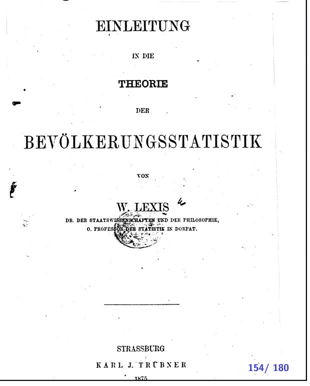
```
> thL[,1:9]
    age     per tfi lex.dur lex.Cst lex.Xst lex.id
1 22.18 1938.79   0   37.99       0       1      1
2 49.54 1945.77   0   18.59       0       1      2
3 68.20 1955.18   0    1.40       0       1      3
4 20.80 1957.61   0   34.52       0       0      4
...

> summary( thL )
Transitions:
     To
From 0  1 Records:  Events:  Risk time:  Persons:
   0 3 20       23       20      512.59        23
```



```
> plot( thL, lwd=3 )
```



Lexis diagram



EINLEITUNG

IN DIE

THEORIE

DER

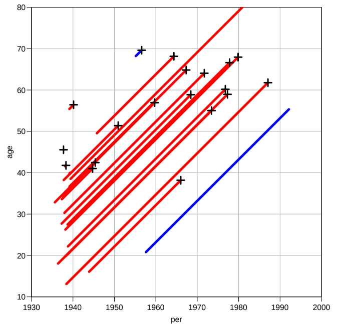
BEVÖLKERUNGSSTATISTIK

VON

W. LEXIS

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

STRASSBURG
KARL J. TRÜBNER
1875



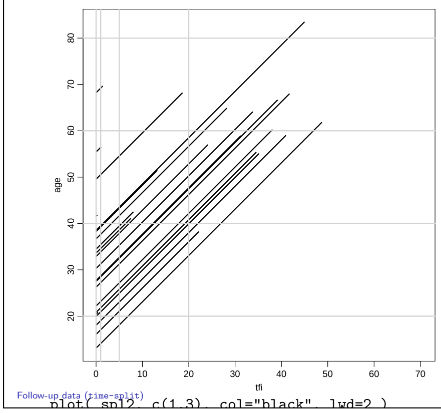
```
> plot( thL, 2:1, lwd=5, col=c("red","blue")[thL$contrast],
+       grid=TRUE, lty.grid=1, col.grid=gray(0.7),
+       xlim=1930+c(0,70), xaxs="i", ylim=  10+c(0,70), yaxs="i", las=1 )
> points( thL, 2:1, pch=c(NA,3)[thL$lex.Xst+1],lwd=3, cex=1.5 )
```

Splitting follow-up time

```
> spl1 <- splitLexis( thL, breaks=seq(0,100,20),
>                          time.scale="age" )
> round(spl1,1)
   age    per  tfi lex.dur lex.Cst lex.Xst   id sex birthdat contrast injecdat vol
1 22.2 1938.8  0.0    17.8       0       0    1   2   1916.6        1   1938.8
2 40.0 1956.6 17.8    20.0       0       0    1   2   1916.6        1   1938.8
3 60.0 1976.6 37.8     0.2       0       1    1   2   1916.6        1   1938.8
4 49.5 1945.8  0.0    10.5       0       0  640   2   1896.2        1   1945.8
5 60.0 1956.2 10.5     8.1       0       1  640   2   1896.2        1   1945.8
6 68.2 1955.2  0.0     1.4       0       1 3425   1   1887.0        2   1955.2
7 20.8 1957.6  0.0    19.2       0       0 4017   2   1936.8        2   1957.6
8 40.0 1976.8 19.2    15.3       0       0 4017   2   1936.8        2   1957.6
...
```

Split on another timescale

```
> spl2 <- splitLexis( spl1, time.scale="tfi",
                            breaks=c(0,1,5,20,100) )
> round( spl2, 1 )
   lex.id  age    per  tfi lex.dur lex.Cst lex.Xst   id sex birthdat contrast inje
1       1 22.2 1938.8  0.0     1.0       0       0    1   2   1916.6        1   19
2       1 23.2 1939.8  1.0     4.0       0       0    1   2   1916.6        1   19
3       1 27.2 1943.8  5.0    12.8       0       0    1   2   1916.6        1   19
4       1 40.0 1956.6 17.8     2.2       0       0    1   2   1916.6        1   19
5       1 42.2 1958.8 20.0    17.8       0       0    1   2   1916.6        1   19
6       1 60.0 1976.6 37.8     0.2       0       1    1   2   1916.6        1   19
7       2 49.5 1945.8  0.0     1.0       0       0  640   2   1896.2        1   19
8       2 50.5 1946.8  1.0     4.0       0       0  640   2   1896.2        1   19
9       2 54.5 1950.8  5.0     5.5       0       0  640   2   1896.2        1   19
10      2 60.0 1956.2 10.5     8.1       0       1  640   2   1896.2        1   19
11      3 68.2 1955.2  0.0     1.0       0       0 3425   1   1887.0        2   19
12      3 69.2 1956.2  1.0     0.4       0       1 3425   1   1887.0        2   19
13      4 20.8 1957.6  0.0     1.0       0       0 4017   2   1936.8        2   19
14      4 21.8 1958.6  1.0     4.0       0       0 4017   2   1936.8        2   19
15      4 25.8 1962.6  5.0    14.2       0       0 4017   2   1936.8        2   19
16      4 40.0 1976.8 19.2     0.8       0       0 4017   2   1936.8        2   19
17      4 40.8 1977.6 20.0    14.5       0       0 4017   2   1936.8        2   19
```



```
 age  tfi lex.dur lex.
22.2  0.0     1.0
23.2  1.0     4.0
27.2  5.0    12.8
40.0 17.8     2.2
42.2 20.0    17.8
60.0 37.8     0.2
```

```
plot( spl2, c(1,3), col="black", lwd=2 )
```

Likelihood for a constant rate

- This setup is for a situation where it is assumed that rates are constant in each of the intervals.
- Each observation in the dataset contributes a term to a “Poisson” likelihood.
- Rates can vary along several timescales simultaneously.
- Models can include fixed covariates, as well as the timescales (the left end-points of the intervals) as continuous variables.

The Poisson likelihood for split data

- Split records (one per **p**erson-**i**nterval (p, i)):

$$D\log(\lambda) - \lambda Y = \sum_{p,i}\big(d_{pi}\log(\lambda) - \lambda y_{pi}\big)$$

- Assuming that the death indicator ($d_{pi} \in \{0, 1\}$) is Poisson, with $\log$-offset y_{pi} will give the same result.
- Model assumes that rates are constant.
- But the split data allows models that assume different rates for different (d_{pi}, y_{pi}), so rates can vary **within** a person's follow-up.

Where is (d_{pi}, y_{pi}) in the split data?

```
> round( spl2, 1 )
   lex.id  age    per  tfi lex.dur lex.Cst lex.Xst   id sex birthdat contrast
1       1 22.2 1938.8  0.0     1.0       0       0    1   2   1916.6        1
2       1 23.2 1939.8  1.0     4.0       0       0    1   2   1916.6        1
3       1 27.2 1943.8  5.0    12.8       0       0    1   2   1916.6        1
4       1 40.0 1956.6 17.8     2.2       0       0    1   2   1916.6        1
5       1 42.2 1958.8 20.0    17.8       0       0    1   2   1916.6        1
6       1 60.0 1976.6 37.8     0.2       0       1    1   2   1916.6        1
7       2 49.5 1945.8  0.0     1.0       0       0  640   2   1896.2        1
8       2 50.5 1946.8  1.0     4.0       0       0  640   2   1896.2        1
9       2 54.5 1950.8  5.0     5.5       0       0  640   2   1896.2        1
10      2 60.0 1956.2 10.5     8.1       0       1  640   2   1896.2        1
...
```

— and what are covariates for the rates?

Analysis of results

- d_{pi} — events in the variable: `lex.Xst`:
  In the model as response: `lex.Xst==1`
- y_{pi} — risk time: `lex.dur` (duration):
  In the model as offset $\log(y)$, `log(lex.dur)`.
- Covariates are:
  - timescales (age, period, time in study)
  - other variables for this person (constant or *assumed* constant in each interval).
- Model rates using the covariates in `glm`:
  — no difference between time-scales and other covariates.

Fitting a simple model

```
> stat.table( contrast,
+             list( D = sum( lex.Xst ),
+                   Y = sum( lex.dur ),
+                Rate = ratio( lex.Xst, lex.dur, 100 ) ),
+             margin = TRUE,
+             data = spl2 )
 ---------------------------------------
 contrast          D        Y     Rate
 ---------------------------------------
 1             19.00   476.67     3.99
 2              1.00    35.93     2.78

 Total         20.00   512.59     3.90
 ---------------------------------------
```

Fitting a simple model

```
 -------------------------
 contrast          D        Y     Rate
 -------------------------
 1             19.00   476.67     3.99
 2              1.00    35.93     2.78

 Total         20.00   512.59     3.90
 -------------------------

> m0 <- glm( lex.Xst ~ factor(contrast) - 1,
                        offset=log(lex.dur/100),
+                       family=poisson, data=spl2 )
> round( ci.exp( m0 ), 2 )

                  exp(Est.) 2.5% 97.5%
factor(contrast)1      3.99 2.54  6.25
factor(contrast)2      2.78 0.39 19.74
```

SMR

Epidemiology with R
August 2015
NovoNordisk Epidemiology
`http://bendixcarstensen.com/Epi/Courses/NNepi/`

SMR

Cohorts where all are exposed

When there is no comparison group we may ask:
Do mortality rates in cohort differ from those of an **external** population, for example:

Rates from:

- Occupational cohorts
- Patient cohorts

compared with reference rates obtained from:

- Population statistics (mortality rates)
- Disease registers (hospital discharge registers)

Log-likelihood

Cohort rates proportional to reference rates:
$\lambda(a) = \theta \times \lambda_R(a)$ — the same in all age-bands.

D_a deaths during Y_a person-years an age-band a gives the likelihood:

$$\begin{aligned} D_a \log(\lambda(a)) - \lambda(a) Y_a &= D_a \log(\theta\lambda_R(a)) - \theta\lambda_R(a) Y_a \\ &= D_a \log(\theta) + D_a \log(\lambda_R(a)) \\ &\quad -\theta(\lambda_R(a) Y_a) \end{aligned}$$

The constant $D_a \log(\lambda_R(a))$ does not involve θ, and so can be dropped.

The term $\lambda_R(a) Y_a = E_a$ is the "expected" number of cases in age a, so the log-likelihood for age a is:

$$D_a \log(\theta) - \theta(\lambda_R(a) Y_a) = D_a \log(\theta) - \theta(E_a)$$

Note: $\lambda_R(a)$ is known for all values of a. The total log-likelihood is:

$$D \log(\theta) - \theta E$$

Therefore:

$$\hat{\theta} = \frac{D}{\lambda_R Y} = \frac{D}{E} = \frac{\text{Observed}}{\text{Expected}} = \text{SMR}$$

SMR is the maximum likelihood estimator of the relative mortality in the cohort.

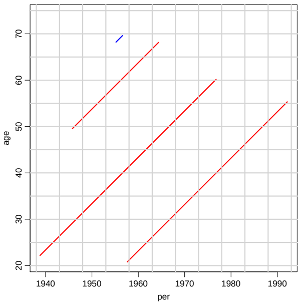
Accounting for age composition

- Compare rates in a study group with a standard set of age–specific rates.
- Reference rates are normally based on large numbers of cases, — assumed known.
- Calculate "expected" number of cases, $E_a = \lambda_R(a) Y_a$, and compare this with the observed number of cases, D:
- SMR is based on a log-likelihood similar to that for a rate — Y is replaced by E:

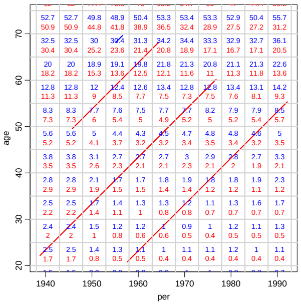
$$\text{SMR} = \frac{D}{E}, \qquad \text{s.d.}\big(\log(\text{SMR})\big) = \frac{1}{\sqrt{D}}$$

Modelling the SMR

- As for the rates, the SMR can be modelled using individual data.
- Response is d_i, the event indicator (`lex.Xst`).
- log-offset is the expected value for each piece of follow-up, $e_i = y_i \times \lambda_R$.
- λ_R is the population rate corresponding to the age, period and sex of the follow-up period y_i.



```
plot( thap, 2:1, col=c("blue","red")[thap$sex], lwd=2 )
```



```
plot( thap, 2:1, col=c("blue","red")[thap$sex], lwd=2 )
...
```

Split the data to fit with population data

```
> # Split the data for SMR-analysis
> tha  <- splitLexis(thL, "age", breaks=seq(0,90,5) )
> thap <- splitLexis(tha, "per", breaks=seq(1938,2038,5) )
> dim( thap )
[1] 41 15
> # Create variables to fit with the population data
> thap$agr <- timeBand( thap, "age", "left" )
> thap$cal <- timeBand( thap, "per", "left" )
> round( thap[,c("lex.id","age","agr","per","cal","lex.dur","lex.Xst","sex")], 2 )
  lex.id   age agr     per  cal lex.dur lex.Xst sex
1      1 22.18  20 1938.79 1938    2.82       0   2
2      1 25.00  25 1941.61 1938    1.39       0   2
3      1 26.39  25 1943.00 1943    3.61       0   2
4      1 30.00  30 1946.61 1943    1.39       0   2
5      1 31.39  30 1948.00 1948    3.61       0   2
6      1 35.00  35 1951.61 1948    1.39       0   2
7      1 36.39  35 1953.00 1953    3.61       0   2
8      1 40.00  40 1956.61 1953    1.39       0   2
```

Merge with population data

```
> thapx <- merge( thap, gmortDK[,c("agr","cal","sex","rt")] )
> str( thapx )
Classes 'Lexis' and 'data.frame': 41 obs. of  18 variables:
 $ sex     : num  1 2 2 2 2 2 2 2 2 2 ...
 $ agr     : num  65 20 20 20 25 25 25 25 30 30 ...
 $ cal     : num  1953 1938 1953 1958 1938 ...
 $ lex.id  : int  3 1 4 4 1 1 4 4 1 1 ...
 $ age     : num  68.2 22.2 20.8 21.2 25.0 ...
 $ per     : num  1955 1939 1958 1958 1942 ...
 $ tfi     : num  0.000 0.000 0.000 0.389 2.818 ...
 $ lex.dur : num  1.405 2.818 0.389 3.806 1.391 ...
 $ lex.Cst : num  0 0 0 0 0 0 0 0 0 0 ...
 $ lex.Xst : num  1 0 0 0 0 0 0 0 0 0 ...
 $ id      : num  3425    1 4017 4017    1 ...
 $ birthdat: num  1887 1917 1937 1937 1917 ...
 $ contrast: num  2 1 2 2 1 1 2 2 1 1 ...
 $ injecdat: num  1955 1939 1958 1958 1939 ...
 $ volume  : num  0 22 0 0 22 22 0 0 22 22 ...
```

Calculation of the SMR

```
> thapx$E <- thapx$lex.dur * thapx$rt / 1000
> stat.table( contrast,
+             list( D = sum( lex.Xst ),
+                   Y = sum( lex.dur ),
+                   E = sum( E ),
+                 SMR = ratio( lex.Xst, E ) ),
+              margin = TRUE,
+                data = thapx )
 ------------------------------------------
 contrast          D       Y       E     SMR
 ------------------------------------------
 1              2.00   56.59    0.33    6.02
 2              1.00   35.93    0.11    8.70

 Total          3.00   92.52    0.45    6.71
 ------------------------------------------
```

Modelling the SMR

```
> m.SMR <- glm( lex.Xst ~ factor(contrast)-1+offset(log(E)),
+               family=poisson, data=thapx )
> round( ci.lin( m.SMR, Exp=TRUE )[,5:7], 3 )
                   exp(Est.)  2.5%  97.5%
factor(contrast)1      6.023 1.506 24.082
factor(contrast)2      8.698 1.225 61.745
```

- Analysis of SMR is like analysis of rates:
- Replace Y with E — that's all!

Danish diabetes data

Epidemiology with R
August 2015
NovoNordisk Epidemiology
http://bendixcarstensen.com/Epi/Courses/NNepi/

DK-DM

Danish diabetes data

- Epi package contains a (bogus) subset of 10,000 incident cases of DM from the Danish Diabetes Register
- . . . including their follow-up till 2009.
- Cannot be used for incidence calculations,
- but useful for illustration of **mortality** among DM patients.
- We shall look at mortality as function of age, calendar time and duration of DM
- . . . as well as **SMR** (standardized mortality ratio) — the mortality relative to the general population.

Example: Danish diabetes data I

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

Example: Danish diabetes data

```
> 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
```

Example: Danish diabetes data

```
> 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.
```

Example: Danish diabetes data

```
> head( LL )

              A        P dur    lex.dur lex.Cst lex.Xst lex.id sex    dobth     d
50185  58.66119 1998.917   0 11.0800821   Alive   Alive      1   F 1940.256 1998.
307563 64.09035 2003.309   0  6.6885695   Alive   Alive      2   M 1939.218 2003.
294104 86.25051 2004.552   0  5.4455852   Alive   Alive      3   F 1918.301 2004.
336439 44.03559 2009.261   0  0.7364819   Alive   Alive      4   F 1965.225 2009.
245651 75.77550 2008.653   0  1.3442847   Alive   Alive      5   M 1932.877 2008.
216824 80.01643 2007.886   0  2.0369610   Alive    Dead      6   F 1927.870 2007.
          dooad doins      dox
50185        NA    NA 2009.997
307563 2007.446    NA 2009.997
294104       NA    NA 2009.997
336439       NA    NA 2009.997
245651       NA    NA 2009.997
216824       NA    NA 2009.923
```

Example: Danish diabetes data

```
> summary( LL )

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

> 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
```

Example: Danish diabetes data

```
> options( digits=5 )
> subset( LL, lex.id %in% c(43,132) )[,1:11]

            A      P dur lex.dur lex.Cst lex.Xst lex.id sex  dobth   dodm  dodth
94480  71.540 2002.9   0  5.2211   Alive    Dead     43   M 1931.3 2002.9 2008.1
352707 67.677 2007.1   0  2.9268   Alive   Alive    132   M 1939.4 2007.1     NA

> subset( SL, lex.id %in% c(43,132) )[,1:11]

    lex.id      A      P     dur lex.dur lex.Cst lex.Xst sex  dobth   dodm  dodth
297     43 71.540 2002.9 0.00000 0.45996   Alive   Alive   M 1931.3 2002.9 2008.1
298     43 72.000 2003.3 0.45996 1.00000   Alive   Alive   M 1931.3 2002.9 2008.1
299     43 73.000 2004.3 1.45996 1.00000   Alive   Alive   M 1931.3 2002.9 2008.1
300     43 74.000 2005.3 2.45996 1.00000   Alive   Alive   M 1931.3 2002.9 2008.1
301     43 75.000 2006.3 3.45996 1.00000   Alive   Alive   M 1931.3 2002.9 2008.1
302     43 76.000 2007.3 4.45996 0.76112   Alive    Dead   M 1931.3 2002.9 2008.1
816    132 67.677 2007.1 0.00000 0.32307   Alive   Alive   M 1939.4 2007.1     NA
817    132 68.000 2007.4 0.32307 1.00000   Alive   Alive   M 1939.4 2007.1     NA
818    132 69.000 2008.4 1.32307 1.00000   Alive   Alive   M 1939.4 2007.1     NA
819    132 70.000 2009.4 2.32307 0.60370   Alive   Alive   M 1939.4 2007.1     NA
```