History and Ecology of R

Martyn Plummer

International Agency for
Research on Cancer

SPE 2018, Lyon

Pre-history

Before there was R, there was S.

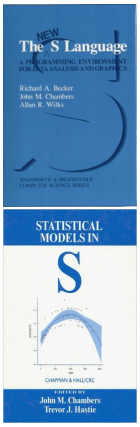
The S language

Developed at AT&T Bell laboratories by Rick Becker, John Chambers, Doug Dunn, Paul Tukey, Graham Wilkinson.

Version 1	1976–1980	Honeywell GCOS, Fortran-based
Version 2	1980–1988	Unix; Macros, Interface Language
	1981–1986	QPE (Quantitative Programming Environment)
	1984–	General outside licensing; books
Version 3	1988-1998	C-based; S functions and objects
	1991–	Statistical models; informal classes and methods
Version 4	1998	Formal class-method model; connections; large objects
	1991–	Interfaces to Java, Corba?

Source: Stages in the Evolution of S `http://ect.bell-labs.com/sl/S/history.html`

The "Blue Book" and the "White Book"



Key features of S version 3 outlined in two books:

- Becker, Chambers and Wilks, *The New S Language: A Programming Environment for Statistical Analysis and Graphics* (1988)
  - Functions and objects
- Chambers and Hastie (Eds), *Statistical Models in S* (1992)
  - Data frames, formulae

These books were later used as a prototype for R.

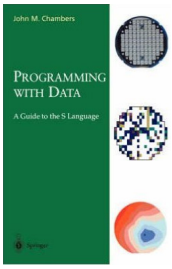
Programming with Data

> *"We wanted users to be able to begin in an interactive environment, where they did not consciously think of themselves as programming. Then as their needs became clearer and their sophistication increased, they should be able to slide gradually into programming." – John Chambers, Stages in the Evolution of S*

This philosophy was later articulated explicitly in *Programming With Data* (Chambers, 1998) as a kind of mission statement for S

> *To turn ideas into software, quickly and faithfully*

The "Green Book"



Key features of S version 4 were outlined in Chambers, *Programming with Data* (1998).

- S as a programming language
- Introduced formal classes and methods, which were later introduced into R by John Chambers himself.

S-PLUS

- AT&T was a regulated monopoly with limited ability to exploit creations of Bell Labs.
- S source code was supplied for free to universities
- After the break up of AT&T in 1984 it became possible for them to sell S.
- S-PLUS was a commercially available form of S licensed to Statistical Sciences (later Mathsoft, later Insightful) with added features:
  - GUI,survival analysis, non-linear mixed effects, Trellis graphics, ...

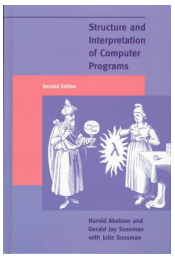
The Rise and Fall of S-PLUS

- 1988. Statistical Science releases first version of S-PLUS.
- 1993. Acquires exclusive license to distribute S. Merges with Mathsoft.
- 2001. Changes name to Insightful.
- 2004. Purchases S language for $2 million.
- 2008. Insightful sold to TIBCO. S-PLUS incorporated into TIBCO Spotfire.

History

How R started, and how it turned into an S clone

The Dawn of R



- Ross Ihaka and Robert Gentlemen at the University of Auckland
- An experimental statistical environment
- Scheme interpreter with S-like syntax
  - Replaced scalar type with vector-based types of S
  - Added lazy evaluation of function arguments
- Announced to *s-news* mailing list in August 1993.

A free software project

- June 1995. Martin Maechler (ETH, Zurich) persuades Ross and Robert to release R under GNU Public License (GPL)
- March 1996. Mailing list *r-testers* mailing list
  - Later split into three *r-announce*, *r-help*, and *r-devel*.
- Mid 1997. Creation of *core team* with access to central repository (CVS)
  - Doug Bates, Peter Dalgaard, Robert Gentleman, Kurt Hornik, Ross Ihaka, Friedrich Leisch, Thomas Lumley, Martin Maechler, Paul Murrell, Heiner Schwarte, Luke Tierney
- 1997. Adopted by the GNU Project as "GNU S".

The draw of S

> *"Early on, the decision was made to use S-like syntax. Once that decision was made, the move toward being more and more like S has been irresistible"*
> *– Ross Ihaka, R: Past and Future History (Interface '98)*

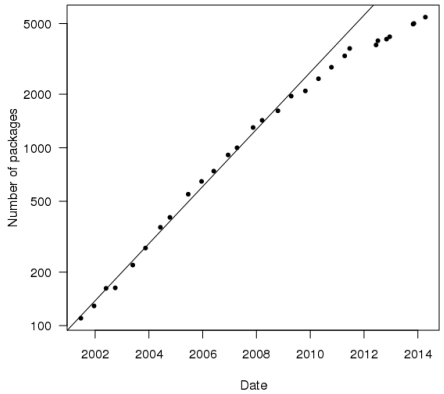
R 1.0.0, a complete and stable implementation of S version 3, was released in 2000.

A Souvenir

Packages

- Comprehensive R Archive Network (CRAN) started in 1997
  - Quality assurance tools built into R
  - Increasingly demanding with each new R release
- Recommended packages distributed with R
  - Third-party packages included with R distribution
  - Provide more complete functionality for the R environment
  - Starting with release 1.3.0 (completely integrated in 1.6.0)

Growth of CRAN



Source: Dataset `CRANpackages` in package `Ecdat`

The present

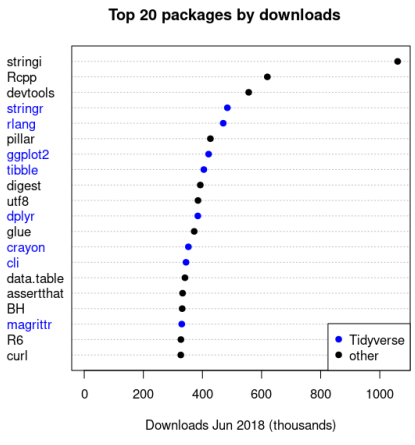
The current era is characterized by

- A mature R community
- Large penetration of R in the commercial world ("data science", "analytics", "big data")
- Increasing interest in the R language from computer scientists.

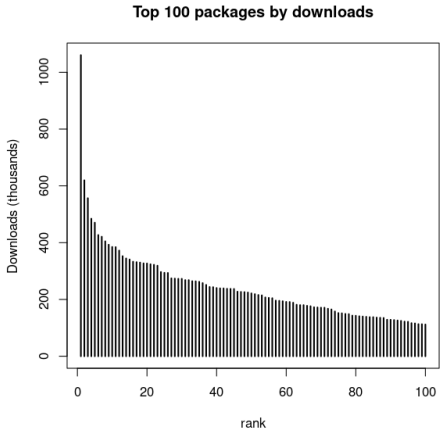
Community

- useR! Annual conference
  - Brisbane (2018), Toulouse (2019), Boston (2020)
- R Journal (`http://journal.r-project.org`)
  - Journal of record, peer-reviewed articles, indexed
  - Journal of Statistical Software (JSS) has many articles dedicated to R packages (`http://jstatsoft.org`)
- Migration to social media
  - Stack Exchange/Overflow, Github, Twitter (#rstats)
  - Follow @_R_Foundation on Twitter

Much important R infrastructure is now in package space



Much important R infrastructure is now in package space



The tidyverse

- Many of the popular packages on CRAN were written by Hadley Wickham.
- These packages became known as the "hadleyverse" until Hadley himself rebranded them the "tidyverse" (`www.tidyverse.org`).
- All packages in the tidyverse have a common design philosophy and work together. Common features are:
  - Non-standard evaluation rules for function calls.
  - Use of the pipe operator `%>%` to pass data transparently from one function call to another.
- The CRAN meta-package `tidyverse` installs all of these packages.

Commercial R

Several commercial organizations provide commercial versions of R including support, consulting, ...

- Revolution Computing, later Revolution Analytics (2007–2014), purchased by Microsoft.
- RStudio (2010–)
- Mango Solutions (2002–)

Validation and Reliability

- *R: Regulatory Compliance and Validation Issues* guidance document by The R Foundation
- ValidR by Mango Solutions
- MRAN (`https://mran.microsoft.com/`), a time-stamped version of CRAN
  - Allows analysis to be re-run with exactly the same package versions at a later date.
  - Used by Microsoft R Open, Microsoft's distribution of R.

Attack of the Clones (and forks)

Name	Implementation	Commercial sponsor	Open source
pqR	C fork		Yes
CXXR	C++ fork	Google	Yes
ORBIT	C fork	Huawei	Yes
Renjin	Java	BeDataDriven	Yes
FastR	Java (Truffle/Graal)	Oracle	Yes
Riposte	C++	Tableau Research	Yes
TERR	C++	TIBCO	No

A number of projects have looked improving the efficiency of R, either by forking the original codebase or by re-implementing R.

The R Foundation for Statistical Computing

A non-profit organization working in the public interest, founded in 2002 in order to:

- Provide support for the R project and other innovations in statistical computing.
- Provide a reference point for individuals, institutions or commercial enterprises that want to support or interact with the R development community.
- Hold and administer the copyright of R software and documentation (This never happened)

The R Consortium

In 2015, a group of organizations created a consortium to support the R ecosystem
Current members (May 2018)

R Foundation A statutory member of The R Consortium

Platinum members IBM, Microsoft, Moore Foundation, RStudio

Gold members TIBCO

Silver members Alteryx, DataCamp, Esri, Google, Mango Solutions, Oracle, ProCogia

The Future

"Prediction is very difficult, especially about the future"
– variously attributed to Niels Bohr, Piet Hein, Yogi Bera

Trends

We cannot make predictions, but some long-term trends are very visible:

- Average age of R Core Team?
- Younger R developers more closely associated with industry than academia
- R Consortium provides mechanism for substantial investment in R infrastructure

> *"R Consortium has invested more that $650,000 USD in over 30 projects that impact the over 2 million R users worldwide" – R Consortium press release 29 May 2018.*

R language versus R implementation

- R has no formal specification
- R language is defined by its implementation ("GNU R")
- Long-term future of R may depend on formal specification of the language, rather than current implementation.

Simply start over and build something better

The x in this function is randomly local or global

```
f = function() {
   if (runif(1) > .5)
      x = 10
   x
}
```

"In the light of this, I've come to the conclusion that rather than "fixing" R, it would be better and much more productive to simply start over and build something better" – Ross Ihaka, Christian Robert's blog, September 13, 2010

Back to the Future

Ross Ihaka and Duncan Temple Lang propose a new language built on top of common lisp with:

- Scalar types
- Type hinting
- Call-by-reference semantics
- Use of multi-cores and parallelism
- More strict license to protect work donated to the commons

Julia (www.julialang.org)

"In Julia, I can build a package that achieves good performance without the need to interface to code written in C, C++ or Fortran – in the sense that my package doesn't need to require compilation of code outside that provided by the language itself.

It is not surprising that the design of R is starting to show its age. Although R has only been around for 15-18 years, its syntax and much of the semantics are based on the design of "S3" which is 25–30 years old"

– Doug Bates, message to R-SIG-mixed-models *list, December 9 2013*

What does all of this mean for the course?

- R incorporates over 40 years of ideas in statistical computing from multiple contributors.
- There is usually more than one way to do something in R.
- Some of the peculiarities of the R language are there for historical reasons.
- The course does not cover some of the recent additions to the R ecosystem.

Resources

- Chambers J, Stages in the Evolution of S
- Becker, R, A Brief History of S
- Chambers R, Evolution of the S language
- Ihaka, R and Gentleman R, R: A language for Data Analysis and Graphics, *J Comp Graph Stat*, **5**, 299–314, 1996.
- Ihaka, R, R: Past and Future History, Interface 98.
- Ihaka, R, Temple Lang, D, Back to the Future: Lisp as a Base for a Statistical Computing System
- Fox, J, Aspects of the Social Organization and Trajectory of the R Project, R Journal, Vol 1/2, 5–13, 2009.

R: language and basic data management

Krista Fischer

Statistical Practice in Epidemiology, Lyon, 2018
(initial slides by P. Dalgaard)

Language

- R is a programming language – also on the command line
- (This means that there are *syntax rules*)

On the command line (or a line in a script) one could:

- Print an object by typing its name
- Evaluate an expression
- Call a function, giving the arguments in parentheses – possibly empty
- Notice `objects` vs. `objects()`

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

Objects

- The simplest object type is *vector*
- Modes: numeric, character, factor, ...
- 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

Example (numeric vectors)

```
> a <- c(2, 8, 3, 1, 0, 7)
> b <- c(3, 4, 1, 4, 5, 2)
> a+b
[1]  5 12  4  5  5  9
> mean(a)
[1] 3.5
> m <- mean(a)
> m
[1] 3.5
> a - m # notice recycling
[1] -1.5  4.5 -0.5 -2.5 -3.5  3.5

> z <- c(1, 2, 3)
> a - z   #recycling!
[1]  1  6  0  0 -2  4
```


Character vectors and factors

- The elements of character vectors are text strings that do not have any numeric value.
- Factors are used to describe groupings – these are just integer codes plus a set of names, as labels for the *levels*
- In model specifications, a factor variable is treated as a classification rather than as a quantitative variable

Example:

```
> x<-c(1,3,3,2,1,3,1)
> fx<-factor(x,labels=c("bad","average","good"))

> fx
[1] bad     good    good    average bad     good

 > levels(fx)
[1] "bad"     "average" "good"
```

Lists

- Lists are vectors where the elements can have different types – thus collections of any elements, gathered into one object
- Functions often return lists
- `lst <- list(A=rnorm(5), B="hello")`
- Special indexing:
- `lst$A`
- `lst[[1]]` first element (NB: double brackets)
- Data frames are special type of lists

Matrices

- A matrix is a rectangular collection of data. All columns of a matrix should be of the same type.

```
> A<-matrix(c(1,4,2,6,7,8),nrow=3,ncol=2,

> A
     [,1] [,2]
[1,]    1    4
[2,]    2    6
[3,]    7    8
```

- One can also construct a matrix from its columns using `cbind`, whereas joining two matrices with equal no of columns (with the same column names) can be done using `rbind`.

Data frames

- Usually a dataset in R is stored in a form of a data frame.
- While reading in data from text files (using `read.table(), read.csv()`), a data frame is created.
- A data frame is similar to a matrix, but can have columns (variables) of different types.
- A variable can be extracted using `dataframe$variable` (as data frames are lists)

```
> D<- data.frame(a=c(8,3,5),b=c("X","Z","Y"))
> D
  a b
1 8 X
2 3 Z
3 5 Y
> D$a
[1] 8 3 5
```

Matrices or data frames?

- A (numeric or character) matrix can be converted to a data frame and vice versa (with `as.data.frame(A)` and `as.matrix(B)`).
- Most R functions for statistical analysis work with data frames, but in some cases it is useful to have a matrix (incl the occasions where you want to use some matrix algebra).
- If you need more dimensions than two, there is also `array`.

How to access variables in the data frame?

Different ways to tell R to use variable X from data frame D:

- As mentioned, you can use the `dataframe$variable` notation

```
summary(D$X)
```

- Use the `with` function

```
with(D, summary(X))
```

- Use the `data` argument (works for some functions only)

```
lm(Y~X, data=D)
```

- Attach the dataframe – DISCOURAGED!
  (seems a convenient solution, but can actually make things more complicated, as it creates a temporary copy of the dataset)

```
attach(D)
summary(X)
detach()
```


Data manipulation

To create a new variable `bmi` in the existing data frame `students`, use either of the two:

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

(notice: you need an assignment, to save the transformed object)

Indexing – extracting elements from objects

Square brackets `[ ]` are used for indexing!

Examples:

- Elements of vectors: `a[5]` (5th element); `a[5:7]` (5th to 7th elements); `a[-6]` (all elements except the 6th)
- Logical index: `a[a<3], a[b>2] , a[is.na(b)]` (elements of `a` corresponding to missing values of `b`)
- In a data frame or matrix – two dimensions, two indexes: `students[5, 7], students[1:10, c(2,5)], students[1, ], students[ ,3]` (entire row/column)

Examples of indexing

```
>  x<- c(2,7,3,1,5,9,0)
> x[c(1,5,7)]
[1] 2 5 0
> x[x<3]
[1] 2 1 0

> NMRimp[1:2,1:4]    #quick look at a large data
  sample.id XXL.VLDL.P XXL.VLDL.L XXL.VLDL.PL
1    V18566   1.46e-04     0.0313     0.00331
2    V36115   9.00e-05     0.0195     0.00178

> fgsa[is.na(fgsa$height),"age"]
  [1] 18 69 52 41 52 44 73 28 66 20 73 63 26
# ages of those with missing height

# equivalent: fgsa$age[is.na(fgsa$height)]
```


Conditional assignment: `ifelse`

- Syntax: `ifelse(expression,A,B)`
  Expression (with values TRUE or FALSE) is a vector, A and B are constants or vectors of the same length.

Examples:

```
> x<-c(1,2,7,3,NA)
> ifelse(x<3,1,2)
[1]  1  1  2  2 NA
> ifelse(is.na(x),0,x) #replace missing values by 0
[1] 1 2 7 3 0
>  y<-c(3,6,1,7,8);  z<-c(0,1,0,2,1)
>  ifelse(z==0,x,y)
[1] 1 6 7 7 8
>  ifelse(is.na(x),0,ifelse(x>3,3,x))
[1] 1 2 3 3 0
```

Naming

- Elements of vectors, rows and columns of matrices and data frames can have names

```
> x <- c(boys=1.2, girls=1.1)
> x
 boys girls
  1.2   1.1
> x["boys"]
boys
 1.2
> D[,"a"]  # works for matrices and data frames
[1] 8 3 5
```

- You can extract and set names with `names(x)`; for matrices and data frames also `colnames(x)` and `rownames(x)`;

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

```
 > summary(x)  # a numeric vector
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
     1       1       2       2       3       3
> summary(fx) # a factor
    bad average    good
      3       1       3
```


Function calls

Round brackets ( ) are used for function calls!

Lots of things you do with R involve calling functions (you have seen that already!).
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

Examples:

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

Items which may appear as arguments:

- Names of R objects
- Explicit constants
- Return values from another function call or expression
- Some arguments have their *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.
- Quite often – first argument is not named, but the others are named

Example

From R-help (`help(t.test)`):

```
t.test(x, y = NULL,
   alternative = c("two.sided","less","greater"),
   mu = 0, paired = FALSE, var.equal = FALSE,
   conf.level = 0.95, ...)
```

- The first argument (x) does not have a default – you have to provide some data!
- The other arguments can be modified, if you need to.

Example (cont.)

The following lines of code are equivalent:

```
t.test(a, b,  alternative="less", paired=TRUE)
t.test(a, b, paired=TRUE, alt="less")

t.test(a, b, p=T, a="l")    #not a good style!
```

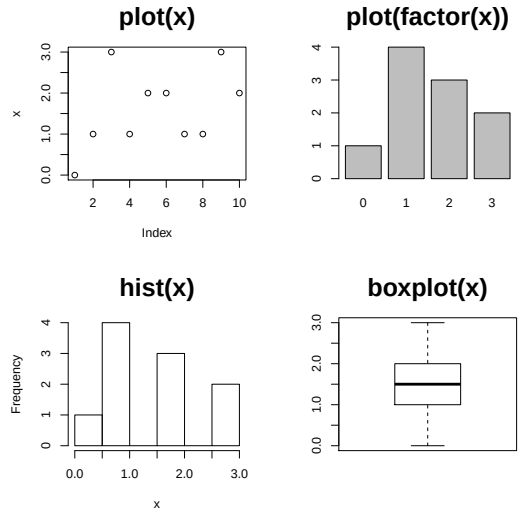
Order does not matter for named arguments!
Partial keyword matching is possible (“alternative" or “alt” or “a") (partial matching is possible)
For a readable code, the use of explicit argument names is highly recommended!

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
- Similarly, the `plot(x,y)` produces a scatter plot, when `x` is a numeric variable and a bar plot of category frequencies, when `x` is a factor variable

Some simple plots:



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()`
- Notice that objects in the global environment may mask objects in packages and attached data frames

More on factors: the `cut` Function

- The `cut` function converts a numerical variable into groups (a factor variable) according to a set of break points
- The intervals are left-open, right-closed by default (`right=FALSE` changes that)
- . . . and that the lowest endpoint is *not* included by default (set `include.lowest=TRUE` if it bothers you)

Example

```
> age <- c(35,20,21,50,46,23,30)
> agegr<-cut(age, c(20,30,40,50))
> table(agegr)
agegr      # the 20-year old is not included!
(20,30] (30,40] (40,50]
      3       1       2
> agegr<-cut(age, c(20,30,40,50),right=FALSE)
> table(agegr)
agegr    # the 50-year old is not included!
[20,30) [30,40) [40,50)
      3       2       1
> agegr<-cut(age, c(20,30,40,50),right=FALSE,
+                            include.lowest=TRUE)
> table(agegr)
agegr
[20,30) [30,40) [40,50]
      3       2       2
```


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 format (YYYY-MM-DD) you need to supply a format specification

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

- You can calculate differences between `Date` objects. The result is an object of class `"difftime"`. To get the number of days between two dates, use

```
> as.numeric(as.Date("2017-6-1")-
               as.Date("1959-3-11"),"days")
[1] 17607
```

Creating your own functions

A very simple example:

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

The function `logit` requires one argument *p* and produces the logit of *p*. Try `logit(0.5)`, or `logit(0.25)`,...

More complex (but still simple):

```
simpsum <- function(x, dec=5) {
m <- mean(x, na.rm=TRUE)
s  <- sd(x, na.rm=TRUE)
 round(c(mean=m,sd=s), dec) }
```

The function `simpsum` requires one argument *x*, but the second argument `dec` (no of decimal points in the output) has a default value 5. Try `simpsum(a)`, or `simpsum(a,dec=2)`.

Statistical Practice in Epidemiology

Poisson and Logistic Regression

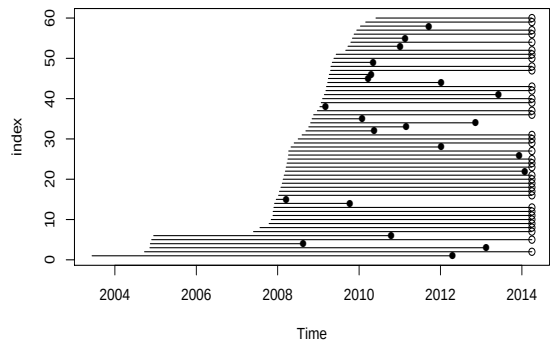
Janne Pitkäniemi (EL)

Points to be covered

- Incidence rates, rate ratios and rate differences from *follow-up studies* can be computed by fitting *Poisson regression models*.
- Odds ratios can be computed from binary data by fitting *Logistic regression models*.
- Odds-ratios can be estimated from case-control studies.
- Both models are special instances of *Generalized linear models*.
- There are various ways to do these tasks in R.

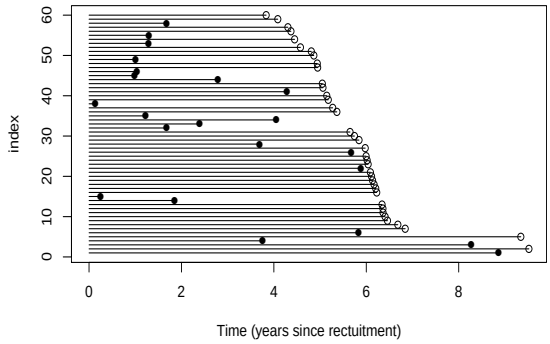
The Estonian Biobank cohort: survival among the elderly

Follow-up of 60 random individuals aged 75-103 at recruitment, until death (•) or censoring (o) in April 2014 (linkage with the Estonian Causes of Death Registry).



The Estonian Biobank cohort: survival among the elderly

Follow-up time for 60 random individuals aged 75-103 at recruitment (time-scale: time in study).



Events, dates and risk time

- Mortality as the outcome:
  - `d:` indicator for **status** at exit:
    - **1**: death observed
    - **0**: censored alive
- Dates:

  `doe` = date of **E**ntry to follow-up,
  `dox` = date of e**X**it, end of follow-up.
- Follow-up time (years) computed as:

  `y = (dox - doe)/365.25`

Crude overall rate computed in two ways

Total no. cases, person-years & rate (/1000 y):

```
> D <- sum( d ); Y <- sum(y) ; R <- D/(Y/1000)
>   round( c(D=D, Y=Y, R=R), 2)
     D        Y        R
884.00 11678.24    75.70
```

Poisson regression model with only intercept ("1").

```
> m1 <- glm( d ~ 1, family=poisson, offset=log(y))
> coef(m1)
( Intercept )
   -2.581025

> exp( coef(m1) )*1000
( Intercept )
    75.69636
```

Why do we get the same results?

Constant hazard — Poisson model

Let $Y \sim exp(\lambda)$, then $f(y;\lambda) = \lambda e^{-\lambda y} I(y > 0)$
Constant rate: $\lambda(y) = \frac{f(y;\lambda)}{S(y;\lambda)} = \lambda$
Observed data $\{(y_i, \delta_i); i = 1, ..., n\}$.
The likelihood $L(\lambda) = \prod_{i=1}^{n} \lambda^{\delta_i} e^{-\lambda y_i}$ and

$log(L) = \sum_{i=1}^{n} [\delta_i log(\lambda) - \lambda y_i]$
Solving the *score equations*: $\frac{\partial \log L(\lambda)}{\partial \lambda} = \sum \left[\frac{\delta_i}{\lambda} - y_i\right]$
$= \frac{D}{\lambda} - Y = 0$ and $D - \lambda Y = 0$
$\rightarrow$ **maximum likelihood estimator** (MLE) of λ:

$$\widehat{\lambda} = \frac{D}{Y} = \frac{\text{number of cases}}{\text{total person-time}} = \text{ empirical rate!}$$

offset term — Poisson model

- Previous model without offset: Intercept 6.784=log(884)
- We should use an offset if we suspect that the underlying **population sizes (person-years) differ** for each of the observed counts – For example varying person-years by tratment group, sex,age,...
- We need a term in the model that "scales" the likelihood, but does not depend on model parameters (include a **term with reg. coef. fixed to 1**) – offset term is log(y)

$$log(\tfrac{\mu}{y}) = \beta_0 + \beta_1 x_1$$
$$log(\mu) = 1 \times log(y) + \beta_0 + \beta_1 x_1$$

Comparing rates: The Thorotrast Study

- Cohort of seriously ill patients in Denmark on whom angiography of brain was performed.
- Exposure: `contrast` medium used in angiography,
  1. `thor` = thorotrast (with ^{232}Th), used 1935-50
  2. `ctrl` = other medium (?), used 1946-63
- Outcome of interest: death

  `doe` = date of **E**ntry to follow-up,
  `dox` = date of e**X**it, end of follow-up.

- `data(thoro)` in the `Epi` package.

Comparing rates: thorotrast vs. control

Tabulating cases, person-years & rates by group

```
> stat.table( contrast,
+             list ( N = count(),
+                    D = sum(d),
+                    Y = sum(y),
+                 rate = ratio(d,y,1000) ) )
```

contrast	N	D	Y	rate
ctrl	1236	797.00	30517.56	26.12
thor	807	748.00	19243.85	38.87

Rate ratio, RR $= 38.89/26.12 = 1.49$,
Std. error of log-RR, SE $= \sqrt{1/748 + 1/797} = 0.051$,
Error factor, EF $= \exp(1.96 \times 0.051) = 1.105$,
95% confidence interval for RR:
$(1.49/1.105, 1.49 \times 1.105) = (1.35, 1.64)$.

Rate ratio estimation with Poisson regression

- Include `contrast` as the explanatory variable (factor).
- Insert person years in units that you want rates in

```
> m2 <- glm( d ~ contrast, offset=log(y/1000),
+                 family = poisson )
> round( summary(m2)$coef, 4)[, 1:2]

               Estimate Std. Error
(Intercept)      3.2626     0.0354
contrast thor    0.3977     0.0509
```

- Rate ratio and CI?
  Call function `ci.exp()` in `Epi`

```
> round( ci.exp( m2 ), 3 )

             exp(Est.)  2.5% 97.5%
(Intercept)     26.116 24.364 27.994
contrast thor    1.488  1.347  1.644
```


Rates in groups with Poisson regression

- Include `contrast` as the explanatory variable (factor).
- Remove the intercept (−1)
- Insert person-years in units that you want rates in

```
> m3 <- glm( d ~ contrast - 1,
                offset=log(y/1000),
+               family = poisson )
> round( summary(m3)$coef, 4)[, 1:2]

             Estimate Std. Error
contrast ctrl  3.2626     0.0354
contrast thor  3.6602     0.0366

> round( ci.exp( m3 ), 3 )

             exp(Est.)  2.5% 97.5%
contrast ctrl   26.116 24.364 27.994
contrast thor   38.870 36.181 41.757
```

Rates in groups with Poisson regression

- You can have it all in one go:

```
> CM <- rbind( c(1,0), c(0,1), c(-1,1) )
> rownames(CM) <- c("Ctrl","Thoro","Th vs.Ct")
> colnames(CM) <- names( coef(m3) )
> CM
         contrast ctrl contrast thor
Ctrl                1              0
Thoro               0              1
Th vs. Ct          -1              1

> round( ci.exp( m3, ctr.mat=CM ),3 )

         exp(Est.)  2.5% 97.5%
Ctrl        26.116 24.364 27.994
Thoro       38.870 36.181 41.757
Th vs. Ct    1.488  1.347  1.644
```


Rate ratio estimation with Poisson regression

- Response may also be specified as individual *rates*:
  `d/y`
  `weights=` instead of `offset=` are needed.

```
> m4<-glm( d/(y/1000)~contrast, weights=y/1000,
+          family=poisson)
> round( ci.exp(m4), 3 )

             exp(Est.)  2.5% 97.5%
(Intercept)     26.116 24.365 27.994
contrast thor    1.488  1.347  1.644
```


Rate difference estimation with Poisson regression

- The approach with `d/y` enables additive rate models too:

```
> m5 <-glm(d/(y/1000) ~contrast,weights=y/1000,
+          family=poisson(link="identity") )
> round( ci.exp(m5,Exp=F), 3 )

             Estimate  2.5% 97.5%
(Intercept)    26.116 24.303 27.929
contrast thor  12.753  9.430 16.077
```

Rates difference

- As before you can have it all:

```
> m6 <- glm( d/(y/1000) ~ contrast -1,
+  family = poisson(link="identity"),
+  weights = y/1000)
> round(ci.exp(m6, ctr.mat=CM, Exp=F ), 3)


          Estimate  2.5% 97.5%
Ctrl        26.116 24.303 27.929
Thoro       38.870 36.084 41.655
Th vs. Ct   12.753  9.430 16.077


> round( ci.exp( m3, ctr.mat=CM), 3 )


          exp(Est.)  2.5% 97.5%
Ctrl         26.116 24.364 27.994
Thoro        38.870 36.181 41.757
Th vs. Ct     1.488  1.347  1.644
```


Binary data: Treatment success Y/N

85 diabetes-patients with foot-wounds:

- Dalterapin (Dal)
- Placebo (Pl)

Treatment/Placebo given to diabetes patients, the design is propective because outcome is measured better/worse. Is the probability of outcome more than 15% – yes, then use the risk difference or risk ratio (RR)

	Treatment group	
	Dalterapin	Placebo
Better	29	20
Worse	14	22
Total	43	42

$$\hat{p}_{\text{Dal}} = \frac{29}{43} = 67\% \qquad \hat{p}_{\text{Pl}} = \frac{20}{42} = 47\%$$

The difference between the probabilities is the fraction of the patients that benefit from the treatment: $p_{\text{Dal}} - p_{\text{Pl}}$

```
> library(Epi)
> dlt <- rbind( c(29,14), c(20,22) )
> colnames( dlt ) <- c("Better","Worse")
> rownames( dlt ) <- c("Dal","Pl")
> kable(twoby2( dlt ),"latex")
```

2 by 2 table analysis :
/ ... /

	Better	Worse	P(Better)	95% conf.	interval
Dal	29	14	0.6744	0.5226	0.7967
Pl	20	22	0.4762	0.3316	0.6249

		95% conf.	interval
Relative Risk:	1.4163	0.9694	2.0692
Sample Odds Ratio:	2.2786	0.9456	5.4907
Conditional MLE Odds Ratio:	2.2560	0.8675	6.0405
Probability difference :	0.1982	−0.0110	0.3850

Exact P−value: 0.0808

Logistic regression for binary data

For grouped binary data, the response is a two-column matrix with columns (successes,failures).

```
trt <- factor(c("Dal","Pl"))
trt <- relevel( trt , 2 )
b1 <- glm( dlt ~ trt, family=binomial )
round( ci.exp( b1 ), 4 )
```

```
             exp(Est.)    2.5% 97.5%
(Intercept)     0.9091 0.4962 1.6657
trtDal          2.2786 0.9456 5.4907
```

- The default parameters in logistic regression are **odds** (the intercept: $20/22 = 0.9090$) and the **odds-ratio** ($(29/14)/(20/22) = 2.28$).
- This is not what you want, because odds ratio is biased estimate of the risk ratio.(recall if p>10% $\frac{p}{1-p} \not\approx p$)

Logistic regression for binary data - Risk ratio (Relative risk)

```
> library(Epi)
> library(xtable)
> dlt <- rbind( c(29,14), c(20,22) )
> diab<-expand.grid(dlt)
> colnames(diab)[1]<-"d"
> diab$out <- c("Better","Better","Worse","Worse")
> diab$trt <- as.factor(c("Dal","Pl","Dal","Pl"))
> diab$totals<-rep(rowSums(dlt),2)
> diab$trt<-relevel( diab$trt, 2 )
> print(xtable(diab,digits=c(0,0,0,0,0)),include.rownames = F)
```

d	out	trt	totals
29	Better	Dal	43
20	Better	Pl	42
14	Worse	Dal	43
22	Worse	Pl	42

Logistic regression for binary data - risk ratio

```
> library(Epi)
> library(xtable)
> b2 <- glm(d/totals~trt,
+           weights=totals,
+           family=binomial(link="log"),
+           data=diab[c(1,2),])
> xtable(round( ci.exp( b2 ), digits=6 ))
```

	exp(Est.)	2.5%	97.5%
(Intercept)	0.48	0.35	0.65
trtDal	1.42	0.97	2.07

Diabetics with Dalterapin treatment have 1.4 times the risk of getting better than those treated with placebo

Logistic regression in case-control studies

- Model for disease occurrence in the target population:

$$\ln\left[\frac{p}{1-p}\right] = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

- Sampling fractions: P(inclusion in study | control) $= s_{\text{ctr}}$
  P(inclusion in study | case) $= s_{\text{case}}$
- Model for observed case-control data:

$$\ln[\text{odds (case | incl.) }] = \ln\left[\frac{p}{1-p}\right] + \ln\left[\frac{s_{\text{cas}}}{s_{\text{ctr}}}\right]$$

$$= \left(\ln\left[\frac{s_{\text{cas}}}{s_{\text{ctr}}}\right] + \beta_0\right) + \beta_1 x_1 + \beta_2 x_2$$

Logistic regression in case-control studies

Analysis of $P(\textit{case}|\textit{inclusion})$ — *i.e.* binary observations:

$$Y = \begin{cases} 1 & \sim \text{ case} \\ 0 & \sim \text{ control} \end{cases}$$

$$\ln[\text{odds (case | incl.) }] = \left(\ln\left[\frac{s_{\text{cas}}}{s_{\text{ctr}}}\right] + \beta_0\right) + \beta_1 x_1 + \beta_2 x_2$$

- Effect of covariates is estimated correctly.
- Intercept is meaningless
  depends on s_{cas} and s_{ctr} that are often unknown.

Case-control study: Food-poisoning outbreak

- An outbreak of acute gastrointestinal illness (AGI) occurred in a psychiatric hospital in Dublin in 1996.
- Out of all 423 patients and staff members, 65 were affected during 27 to 31 August, 1996.
- 65 cases and 62 randomly selected control subjects were interviewed.
- Exposure of interest: chocolate mousse cake.
- 47 cases and 5 controls reported having eaten the cake.

Ref: `http://www.eurosurveillance.org/ViewArticle.aspx?ArticleId=188` – here original numbers somewhat modified.

Outbreak: crude summary of data

- Target population information
  - N = 423, size of the whole study population
  - D = 65, no. of cases of AGI
  - B = 358, no. of non-cases
- Case-control data
  - C = 62, no. of controls, random sample from 358 non-cases
  - f = 62/358 = 0.173, sampling fraction of non-cases
  - D1 = 47 cases exposed to chocolate mousse
  - D0 = 18 unexposed cases
  - C1 = 5 controls exposed to chocolate mousse
  - C0 = 57 unexposed controls

Outbreak: results of analysis

Overall incidence proportion (IP) of AGI in the population

```
> D <- 65; N <- 423; IP <- D/N
> round(IP, 3)
```

```
[1] 0.154
```

Analysis of case-control data

```
> D1 <- 47; D0 <- D - D1;
> C <- 62 ; C1 <- 5; C0 <- C - C1
```

Case-control ratios by exposure (not as useful as the following!)

```
> round( c( D1/C1, D0/C0 ), 2)
```

```
[1] 9.40 0.32
```

Exposure odds in cases and controls

```
> round( c( D1/D0, C1/C0 ), 2)
```

```
[1] 2.61 0.09
```


Outbreak: results of analysis

Estimation of the incidence odds ratio (IOR) = exposure odds ratio

```
> IOR <- (D1/D0)/(C1/C0)
> SE.logIOR <- sqrt(1/D1 + 1/D0 + 1/C1 + 1/C0 )
> CI.IOR <- IOR * exp( c(-1,1)*1.96*SE.logIOR )
> round( c(IOR, SE.logIOR, CI.IOR ), 2)
```

```
[1] 29.77  0.54 10.28 86.21
```

Same with glm model

```
> count<-c(D1,D0,C1,C0)
> cc<-c(1,1,0,0)
> exposed<-c(1,0,1,0)
> mousse<-data.frame(cbind(cc,exposed,count))
> ci.exp(glm(cc~exposed,weights=count,family="binomial",data=mousse))
```

```
             exp(Est.)       2.5%      97.5%
(Intercept)  0.3157895  0.1858913  0.5364586
exposed     29.7666667 10.2778305 86.2102603
```

Conclusion: What did we learn?

- Poisson regression models.
- In Poisson models the response can be either:
  - case indicator `d` with `offset = log(y)`, or
  - rate `d/y` with `weights = y`.
- Both may be fitted on either grouped data, or individual records.
- Binary date can be modeled with odds.
- Case-control studies:
  Odds-ratios can be computed by logistic regression models, but **Intercept** from model is **meaningless**.

Linear and generalized linear models

Friday 15 June, 2018

Esa Läärä

Statistical Practice in Epidemiology with **R**
14 to 20 June, 2018
International Agency for Research on Cancer, Lyon, France

Outline

- Simple linear regression.
- Fitting a model and extracting results.
- Predictions and diagnostics.
- Categorical factors and contrast matrices.
- Main effects and interactions.
- Generalized linear models.
- Modelling curved effects.

Variables in generalized linear models

- The **outcome** or **response** variable must be numeric.
- Main types of response variables are
  - Metric or continuous (a measurement with units)
  - Binary (two values coded 0/1)
  - Failure (does the subject fail at end of follow-up)
  - Count (aggregated failure data, number of cases)
- **Explanatory** variables or **regressors** can be
  - Numeric or quantitative variables
  - Categorical factors, represented by class indicators or contrast matrices.

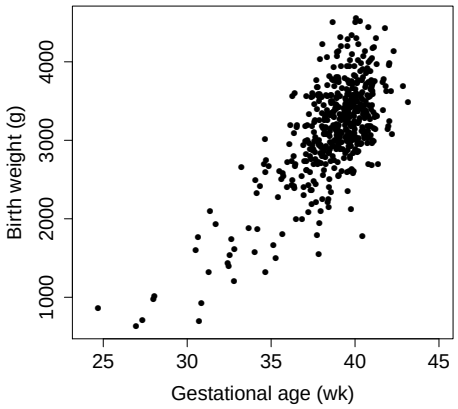
The `births` data in `Epi`

`id`:	Identity number for mother and baby.
`bweight`:	Birth weight of baby.
`lowbw`:	Indicator for birth weight less than 2500 g.
`gestwks`:	Gestation period in weeks.
`preterm`:	Indicator for gestation period less than 37 weeks.
`matage`:	Maternal age.
`hyp`:	Indicator for maternal hypertension (0 = no, 1 = yes).
`sex`:	Sex of baby (1 = male, 2 = female).

Declaring and transforming some variables as factors:

```
> library(Epi) ; data(births)
> births <- transform(births,
+    hyp = factor(hyp, labels=c("N", "H")),
+    sex = factor(sex, labels=c("M", "F")),
+    gest4 = cut(gestwks,breaks=c(20, 35, 37, 39, 45), right=FALSE)  )
> births <- subset(births, !is.na(gestwks))
```


Birth weight and gestational age




```
> with(births, plot(bweight ~ gestwks, xlim = c(24,45), pch = 16, cex.axis=1.5,
+  xlab= "Gestational age (wk)", ylab= "Birth weight (g)" ) )
```


Metric response, numeric explanatory variable

Roughly linear relationship btw `bweight` and `gestwks`

→ Simple **linear regression model** fitted.

```
> m <- lm(bweight ~ gestwks, data=births)
```

- `lm()` is the function that fits linear regression models, assuming **Gaussian** distribution for **error** terms.
- `bweight ~ gestwks` is the **model formula**
- `m` is a **model object** belonging to **class** "`lm`".

`> coef(m)` – Printing the estimated regression coefficients

```
(Intercept)     gestwks
    -4489.1       197.0
```

Interpretation of **intercept** and **slope**?

Model object and extractor functions

Model object = **list** of different elements, each being separately accessible. – See `str(m)` for the full list.

Functions that extract results from the fitted model object

- `summary(m)` – lots of output
- `coef(m)` – beta-hats only (see above)
- `ci.lin(m)[,c(1,5,6)]` – $\widehat{\beta}_j$s plus confidence limits

```
             Estimate    2.5%   97.5%
(Intercept)  -4489.1 -5157.3 -3821.0
gestwks        197.0   179.7   214.2
```

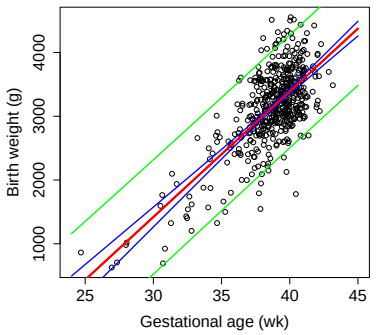
  This function is in `Epi` package
- `anova(m)` – Analysis of Variance Table

Other extractor functions, for example

- `fitted(m), resid(m), vcov(m), ...`
- `predict(m, newdata = ..., interval=...)`
  - Predicted responses for desired combinations of new values of the regressors – `newdata`
  - Argument `interval` specifies whether **confidence** intervals for the *mean* response or **prediction** intervals for *individual* responses are returned.
- `plot(m)` – produces various diagnostic plots based on residuals (raw or standardized)

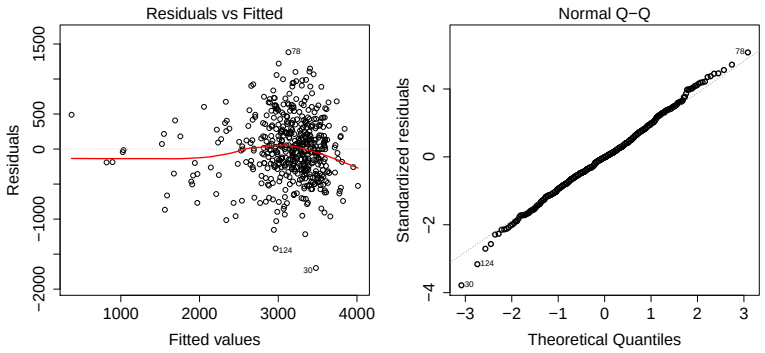
Many of these are special **methods** for certain **generic functions**, aimed at acting on objects of class "`lm`".

Fitted values, confidence & prediction intervals




```
> nd <- data.frame( gestwks = seq(24, 45, by = 0.25 ) )
> pr.c1 <- predict( m, newdata=nd, interval="conf" )
> pr.p1 <- predict( m, newdata=nd, interval="pred" )
> with(births, plot(bweight ~ gestwks, xlim = c(24,45), cex.axis=1.5, cex.lab =
> matlines( nd$gestwks, pr.c1, lty=1, lwd=c(3,2,2), col=c('red','blue','blue'))
> matlines( nd$gestwks, pr.p1, lty=1, lwd=c(3,2,2), col=c('red','green','green')
```

A couple of diagnostic plots




```
> par(mfrow=c(1,2))
> plot(m, 1:2, cex.lab = 1.5, cex.axis=1.5, cex.caption=1.5, lwd=2)
```

- Some deviation from linearity?
- Reasonable agreement with Gaussian error assumption?

Factor as an explanatory variable

- How `bweight` depends on maternal hypertension?

```
> mh <- lm( bweight ~ hyp, data=births)

             Estimate   2.5%  97.5%
(Intercept)   3198.9 3140.2 3257.6
hypH          -430.7 -585.4 -275.9
```

- Removal of intercept → mean `bweights` by `hyp`:

```
> mh2 <- lm( bweight ~ -1 + hyp, data = births)
> coef(mh2)
  hypN   hypH
3198.9 2768.2
```

- Interpretation: `-430.7 = 2768.2 - 3198.9` = difference between level 2 *vs.* reference level 1 of `hyp`

Additive model with both gestwks and hyp

- Joint effect of `hyp` and `gestwks` under additivity is modelled e.g. by updating a simpler model:

```
> mhg <- update(mh, . ~ . + gestwks)

             Estimate    2.5%   97.5%
(Intercept)  -4285.0 -4969.7 -3600.3
hypH           -143.7  -259.0   -28.4
gestwks         192.2   174.7   209.8
```

- The effect of `hyp`: `H` *vs.* `N` is attenuated (from -430.7 to -143.7).
- This suggests that much of the effect of hypertension on birth weight is mediated through a shorter gestation period among hypertensive mothers.

Model with interaction of `hyp` and `gestwks`

- `mhgi <- lm(bweight ~ hyp + gestwks + hyp:gestwks, data = births)`
- Or with shorter formula: `bweight ~ hyp * gestwks`

```
               Estimate    2.5%   97.5%
(Intercept)    -3960.8 -4758.0 -3163.6
hypH           -1332.7 -2841.0   175.7
gestwks          183.9   163.5   204.4
hypH:gestwks      31.4    -8.3    71.1
```

- Estimated slope: 183.9 g/wk in reference group `N` and $183.9 + 31.4 =$ 215.3 g/wk in hypertensive mothers.

$\Leftrightarrow$ For each additional week the difference in mean `bweight` between `H` and `N` group increases by 31.4 g.

- *Interpretation of* `Intercept` *and "main effect"* `hypH`*?*

Model with interaction (cont'd)

More interpretable parametrization obtained if `gestwks` is **centered** at some reference value, using e.g. the **insulate** operator `I()` for explicit transformation of an original term.

- `mi2 <- lm(bweight ~ hyp*I(gestwks-40), ...)`

```
                     Estimate   2.5%  97.5%
(Intercept)            3395.6 3347.5 3443.7
hypH                    -77.3 -219.8   65.3
I(gestwks - 40)         183.9  163.5  204.4
hypH:I(gestwks - 40)     31.4   -8.3   71.1
```

- Main effect of `hyp` $= -77.3$ is the difference between `H` and `N` at `gestwks` $= 40$.
- `Intercept` $=$ 3395.6 is the estimated mean `bweight` at the reference value 40 of `gestwks` in `group N`.

Factors and contrasts in R

- A categorical explanatory variable or **factor** with L **levels** will be represented by $L - 1$ linearly independent columns in the **model matrix** of a linear model.
- These columns can be defined in various ways implying alternative **parametrizations** for the effect of the factor.
- Parametrization is defined by given type of **contrasts**.
- Default: **treatment** contrasts, in which 1st class is the **reference**, and regression coefficient β_k for class k is interpreted as $\beta_k = \mu_k - \mu_1$
- Own parametrization may be tailored by function `C()`, with the pertinent **contrast matrix** as argument.
- Or, use `ci.lin(mod, ctr.mat = CM)` after fitting.

Two factors: additive effects

- Factor X has 3 levels, Z has 2 levels – Model:

$$\mu = \alpha + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \gamma_1 Z_1 + \gamma_2 Z_2$$

- X_1 (reference), X_2, X_3 are the indicators for X,
- Z_1 (reference), Z_2 are the indicators for Z.
- Omitting X_1 and Z_1 the model for mean is:

$$\mu = \alpha + \beta_2 X_2 + \beta_3 X_3 + \gamma_2 Z_2$$

with predicted means μ_{jk} $(j = 1, 2, 3; k = 1, 2)$:

		$Z = 1$	$Z = 2$
	1	$\mu_{11} = \alpha$	$\mu_{11} = \alpha + \gamma_2$
X	2	$\mu_{21} = \alpha + \beta_2$	$\mu_{22} = \alpha + \beta_2 + \gamma_2$
	3	$\mu_{31} = \alpha + \beta_3$	$\mu_{32} = \alpha + \beta_3 + \gamma_2$

Two factors with interaction

- Effect of Z differs at different levels of X:

		$Z = 1$	$Z = 2$
	1	$\mu_{11} = \alpha$	$\mu_{12} = \alpha + \gamma_2$
X	2	$\mu_{21} = \alpha + \beta_2$	$\mu_{22} = \alpha + \beta_2 + \gamma_2 + \delta_{22}$
	3	$\mu_{31} = \alpha + \beta_3$	$\mu_{32} = \alpha + \beta_3 + \gamma_2 + \delta_{32}$

- How much the effect of Z (level 2 *vs.* 1) changes when the level of X is changed from 1 to 3:

$$\begin{aligned}\delta_{32} &= (\mu_{32} - \mu_{31}) - (\mu_{12} - \mu_{11}) \\ &= (\mu_{32} - \mu_{12}) - (\mu_{31} - \mu_{11}),\end{aligned}$$

$=$ how much the effect of X (level 3 *vs.* 1) changes when the level of Z is changed from 1 to 2.

- See the exercise: interaction of `hyp` and `gest4`.

Contrasts in R

- All contrasts can be implemented by supplying a suitable **contrast function** giving the **contrast matrix** e.g:

```
> contr.cum(3)          > contr.sum(3)
1 0 0                    1    1   0
2 1 0                    2    0   1
3 1 1                    3   -1  -1
```

- In model formula factor name `faktori` can be replaced by expression like `C(faktori, contr.cum)`.
- Function `ci.lin()` has an option for calculating CI's for linear functions of the parameters of a fitted model `mall` when supplied by a relevant contrast matrix

```
> ci.lin(mall, ctr.mat = CM)[ , c(1,5,6)]
```

$\rightarrow$ No need to specify contrasts in model formula!

From linear to generalized linear models

- An alternative way of fitting our 1st Gaussian model:

```
> m <- glm(bweight ~ gestwks, family=gaussian, data=bi
```

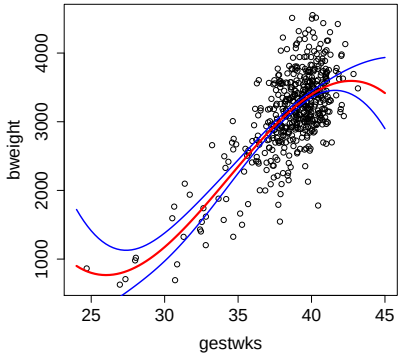
- Function `glm()` fits **generalized linear models** (GLM).
- Requires specification of the
  - **family** – i.e. the assumed "error" distribution for Y_is,
  - **link** function – a transformation of the expected Y_i.
- Covers common models for other types of response variables and distributions, too, e.g. **logistic** regression for binary responses and **Poisson** regression for counts.
- Fitting: method of **maximum likelihood**.
- Many extractor functions for a `glm` object similar to those for an `lm` object.

More about numeric regressors

What if dependence of Y on X is non-linear?

- **Categorize** the values of X into a factor.
  - Continuous effects violently discretized by often arbitrary cutpoints. – Inefficient.
- Fit a low-degree (e.g. 2 to 4) **polynomial** of X.
  - Tail behaviour may be problematic.
- Use **fractional polynomials**.
  - Invariance problems. Only useful if $X = 0$ is well-defined.
- Use a **spline** model: smooth function $s(X; \beta)$.
  - More flexible models that act locally.
  - Effect of X reported by graphing $\widehat{s}(X; \beta)$ & its CI
  - See Martyn's lecture

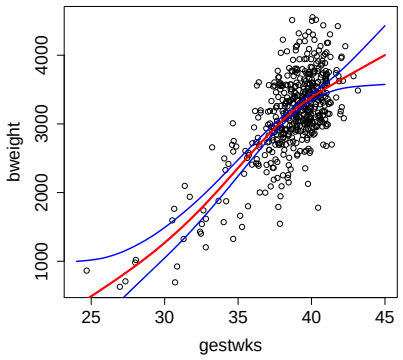
Mean `bweigth` as 3rd order polynomial of `gestwks`




```
> mp3 <- update( m, . ~ . - gestwks + poly(gestwks, 3) )
```

- The model is linear in parameters with 4 terms & 4 df.
- Otherwise good, but the tails do not behave well.

Penalized spline model with cross-validation




```
> library(mgcv)
> mpen <- gam( bweight ~ s(gestwks), data = births)
```

- Looks quite nice.
- Model degrees of freedom ≈ 4.2;
  almost 4, as in the 3rd degree polynomial model

What was covered

- A wide range of models from simple linear regression to splines.
- R functions fitting linear and generalized models: `lm()` and `glm()`.
- Parametrization of categorical explanatory factors; contrast matrices.
- Extracting results and predictions: `ci.lin()`, `fitted()`, `predict()`, `...` .
- Model diagnostics: `resid()`, `plot.lm()`, `...` .

More Advanced Graphics in R

Martyn Plummer

International Agency for Research on Cancer
Lyon, France

SPE 2018, Lyon

Outline

Graphics Systems in R

R has several different graphics systems:

- Base graphics (the `graphics` package)
- Lattice graphics (the `lattice` package)
- Grid graphics (the `grid` package)
- Grammar of graphics (the `ggplot2` package)

Why so many? Which one to use?

Base Graphics

- The oldest graphics system in R.
- Based on S graphics (Becker, Chambers and Wilks, *The New S Language, 1988*)
- Implemented in the base package `graphics`
  - Loaded automatically so always available
- Ink on paper model; once something is drawn "the ink is dry" and it cannot be erased or modified.

Lattice Graphics

- A high-level data visualization system with an emphasis on multivariate data
- An implementation of Trellis graphics, first described by William Cleveland in the book *Visualizing Data*, 1993.
- Implemented in the base package `lattice`.
- More fully described by the `lattice` package author Deepayan Sarkar in the book *Lattice: Multivariate Data Visualization with R*, 2008.

Grammar of Graphics

- Originally described by Leland Wilkinson in the book *The Grammar of Graphics*, 1999 and implemented in the statistical software nViZn (part of SPSS)
- Statistical graphics, like natural languages, can be broken down into components that must be combined according to certain rules.
- Provides a *pattern language* for graphics:
  - geometries, statistics, scales, coordinate systems, aesthetics, themes, ...
- Implemented in R in the CRAN package `ggplot2`
- Described more fully by the `ggplot2` package author Hadley Wickham in the book *ggplot2: Elegant Graphics for Data Analysis*, 2009.

Grid Graphics

- A complete rewrite of the graphics system of R, independent of base graphics.
- Programming with graphics:
  - Grid graphics commands create graphical objects (Grobs)
  - Printing a Grob displays it on a graphics device
  - Functions can act on grobs to modify or combine them
- Implemented in the base package `grid`, and extended by CRAN packages `gridExtra`, `gridDebug`, ...
- Described by the package author Paul Murrell in the book *R Graphics (2nd edition)*, 2011.

Putting It All Together

- Base graphics are the default, and are used almost exclusively in this course
- `lattice` and `ggplot2` are alternate, high-level graphics packages
- `grid` provides alternate low-level graphics functions.
  - A *domain-specific language* for graphics within R
  - Underlies both `lattice` and `ggplot`
  - Experts only
- All graphics packages take time to learn...

Graphics Devices

Graphics devices are used by all graphics systems (base, lattice, ggplot2, grid).

- Plotting commands will draw on the current *graphics device*
- This default graphics device is a window on your screen:
  On Windows windows()
  On Unix/Linux x11()
  On Mac OS X quartz()
  It normally opens up automatically when you need it.
- You can have several graphics devices open at the same time (but only one is current)

Graphics Device in RStudio

RStudio has its own graphics device RStudioGD built into the graphical user interface

- You can see the contents in a temporary, larger window by clicking the zoom button.
- You can write the contents directly to a file with the export menu
- Sometimes small size of the RStudioGD causes problems. Open up a new device calling `RStudioGD()`. This will appear in its own window, free from the GUI.

Writing Graphs to Files

There are also non-interactive graphics devices that write to a file instead of the screen.

`pdf` produces Portable Document Format files

`win.metafile` produces Windows metafiles that can be included in Microsoft Office documents (windows only)

`postscript` produces postscript files

`png, bmp, jpeg` all produce bitmap graphics files

- Turn off a graphics device with `dev.off()`. Particularly important for non-interactive devices.
- Plots may look different in different devices

Types of Plotting Functions

- High level
  - Create a new page of plots with reasonable default appearance.
- Low level
  - Draw elements of a plot on an existing page:
    - Draw title, subtitle, axes, legend . . .
    - Add points, lines, text, math expressions . . .
- Interactive
  - Querying mouse position (`locator`), highlighting points (`identify`)

Basic x-y Plots

- The `plot` function with one or two numeric arguments
- Scatterplot or line plot (or both) depending on `type` argument: `"l"` for lines, `"p"` for points (the default), `"b"` for both, plus quite a few more
- Also: formula interface, `plot(y~x)`, with arguments similar to the modeling functions like `lm`

Customizing Plots

- Most plotting functions take optional parameters to change the appearance of the plot
  - e.g., `xlab, ylab` to add informative axis labels
- Most of these parameters can be supplied to the `par()` function, which changes the default behaviour of subsequent plotting functions
- Look them up via `help(par)`! Here are some of the more commonly used:
  - Point and line characteristics: `pch, col, lty, lwd`
  - Multiframe layout: `mfrow, mfcol`
  - Axes: `xlim, ylim, xaxt, yaxt, log`

Adding to Plots

- `title()` add a title above the plot
- `points()`, `lines()` adds points and (poly-)lines
- `text()` text strings at given coordinates
- `abline()` line given by coefficients (*a* and *b*) or by fitted linear model
- `axis()` adds an axis to one edge of the plot region. Allows some options not otherwise available.

Approach to Customization

- Start with default plots
- Modify parameters (using `par()` settings or plotting arguments)
- Add more graphics elements. Notice that there are graphics parameters that turn things *off*, e.g. `plot(x, y, xaxt="n")` so that you can add completely customized axes with the `axis` function.
- Put all your plotting commands in a script or inside a function so you can start again

Demo 1

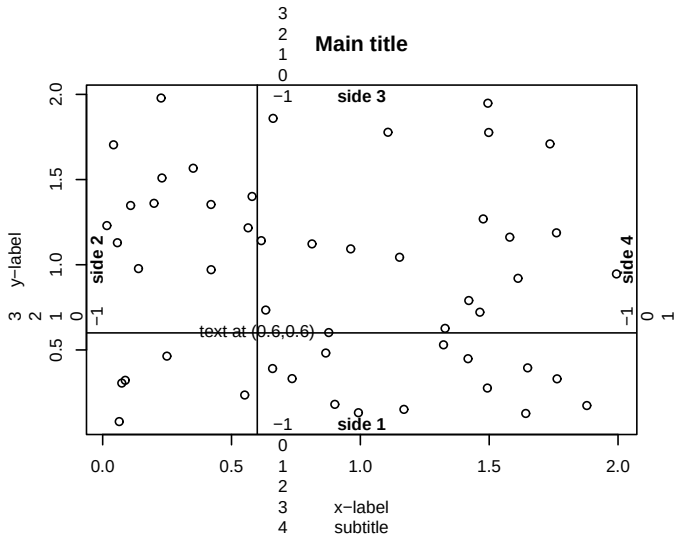
```
library(ISwR)
par(mfrow=c(2,2))
matplot(intake)
matplot(t(intake))
matplot(t(intake),type="b")
matplot(t(intake),type="b",pch=1:11,col="black",
        lty="solid", xaxt="n")
axis(1,at=1:2,labels=names(intake))
```


Margins

- R sometimes seems to leave too much empty space around plots (especially in multi-frame layouts).
- There is a good reason for it: You might want to put something there (titles, axes).
- This is controlled by the `mar` parameter. By default, it is `c(5,4,4,2)+0.1`
  - The units are *lines of text*, so depend on the setting of `pointsize` and `cex`
  - The sides are indexed in clockwise order, starting at the bottom (1=bottom, 2=left, 3=top, 4=right)
- The `mtext` function is designed to write in the margins of the plot
- There is also an *outer margin* settable via the `oma` parameter. Useful for adding overall titles etc. to multiframe plots

Demo 2

```
x <- runif(50,0,2)
y <- runif(50,0,2)
plot(x, y, main="Main title", sub="subtitle",
     xlab="x-label", ylab="y-label")
text(0.6,0.6,"text at (0.6,0.6)")
abline(h=.6,v=.6)
for (side in 1:4)
   mtext(-1:4,side=side,at=.7,line=-1:4)
mtext(paste("side",1:4), side=1:4, line=-1,font=2)
```

The `lattice` package provides functions that produce similar plots to base graphics (with a different "look and feel")

base	lattice
plot	xyplot
hist	histogram
boxplot	bwplot
barplot	barchart
heatmap, contour	levelplot
dotchart	dotplot

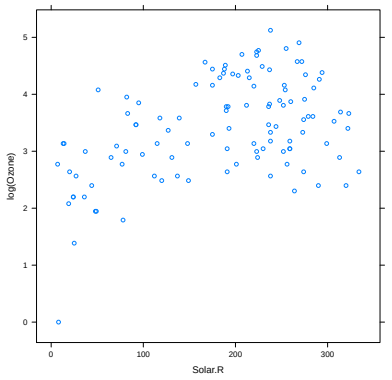
Lattice graphics can also be used to explore *multi-dimensional data*

Panels

- Plotting functions in `lattice` consistently use a formula interface, e.g `y~x` to plot *y* against *x*
- The formula allows conditioning variables, e.g. `y~x|g1*g2*...`
- Conditioning variables create an array of *panels*,
  - One panel for each value of the conditioning variables
  - Continuous conditioning variables are divided into *shingles* (slightly overlapping ranges, named after the roof covering)
  - All panels have the same scales on the *x* and *y* axes.

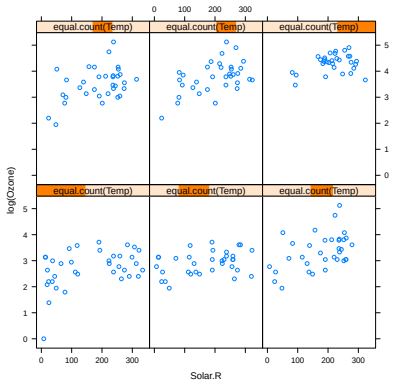
Ozone Concentration by Solar Radiation

```
xyplot(log(Ozone)~Solar.R, data=airquality)
```



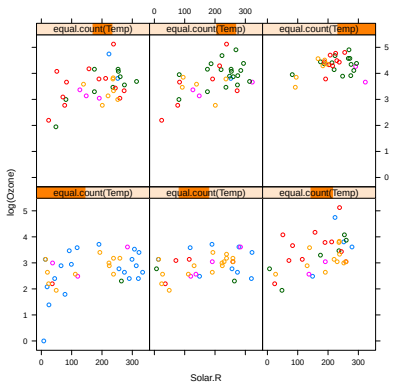
Conditioned on Temperature

```
xyplot(log(Ozone)~Solar.R | equal.count(Temp),
data=airquality)
```



Coloured by Month

```
xyplot(log(Ozone)~Solar.R | equal.count(Temp),
group=Month, data=airquality)
```



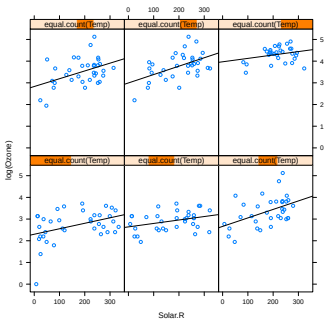
Customizing Panels

- What goes inside each panel of a Lattice plot is controlled by a *panel function*
- There are many standard functions: `panel.xyplot`, `panel.lmline`, etc.
- You can write your own panel functions, most often by combining standard ones

```
mypanel <- function(x,y,...){
   panel.xyplot(x,y,...) #Scatter plot
   panel.lmline(x,y,type="l") #Regression line
}
```


With Custom Panel

```
xyplot(log(Ozone)~Solar.R | equal.count(Temp),
panel=mypanel, data=airquality)
```



Each panel shows a scatter plot (`panel.xyplot`) and a regression line (`panel.lmline`)

A Few Words on Grid Graphics

- Experts only, but . . .
- Recall that `lattice` and `ggplot2` both use `grid`
- The key concepts you need are *grobs* and *viewports*

Grobs: Graphical Objects

- Grobs are created by plotting functions in `grid`, `lattice`, `ggplot2`
- Grobs are only displayed when they are printed
- Grobs can be modified or combined before being displayed
- The `ggplot2` package uses the `+` operator to combine grobs representing different elements of the plot

Viewports

- The plotting region is divided into viewports
- Grobs are displayed inside a viewport
- The panels in lattice graphics are examples of viewports, but in general
  - Viewports can be different sizes (inches, centimetres, lines of text, or relative units)
  - Each viewport may have its own coordinate systems

Statistical Practice in Epidemiology 2018

Survival analysis with competing risks

Janne Pitkäniemi (EL)

Points to be covered

1. Survival or time to event data & censoring.
2. Competing risks: event-specific cumulative incidences & hazards.
3. Kaplan–Meier and Aalen–Johansen estimators.
4. Regression modelling of hazards: Cox model.
5. Packages `survival`, `mstate`, `(cmprisk)`.
6. Functions `Surv()`, `survfit()`, `plot.survfit()`, `coxph()`.

Survival time – time to event

Time spent (`lex.dur`) in a given **state** (`lex.Cst`) from its beginning till a certain *endpoint* or *outcome* **event** (`lex.Xst`) or *transition* occurs, changing the state to another.

Examples of such times and outcome events:

- lifetime: birth → death,
- duration of marriage: wedding → divorce,
- healthy exposure time:
  start of exposure → onset of disease,
- clinical survival time:
  diagnosis of a disease → death.

Ex. Survival of 338 oral cancer patients

Important variables:

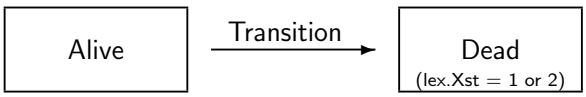
- `time` = duration of patientship from diagnosis (**entry**) till death (`death`) or censoring (`Alive`), (`lex.Cst` is (`Alive`))
- `event` = indicator for the outcome and its observation at the end of follow-up (**exit**):
  0 = censoring,
  1 = death from oral cancer

Special features:

- Two possible endpoints
- Censoring – incomplete observation of the survival time.

Set-up of classical survival analysis

- **Two-state model**: only one type of event changes the initial state.
- Major applications: analysis of lifetimes since birth and of survival times since diagnosis of a disease until death from any cause.



- **Censoring**: Death and final lifetime not observed for some subjects due to emigration or closing the follow-up while they are still alive

Distribution concepts: hazard function

The **hazard rate** or **intensity** function $\lambda(t)$

$$\lambda(t) = P(t < T \leq t + \Delta | T > t)/\Delta, \ \ forsmall\Delta$$

$\approx$ the conditional probability that the event occurs in a short interval $(t, t + \Delta]$, given that it does not occur before t, divided by interval length.

In other words, during a short interval

$$\text{risk of event} \approx \text{hazard} \times \text{interval length}$$

Distribution concepts: survival and cumulative hazard functions

Survival function

$$S(t) = P(T > t),$$

= probability of avoiding the event at least up to t
(the event occurs only after t).
The **cumulative hazard** (or integrated intensity):

$$\Lambda(t) = \int_0^t \lambda(u)du$$

Connections between the functions:

$$S(t) \quad = \quad \exp\{-\Lambda(t)\}$$

Observed data on survival times

For individuals $i = 1, \ldots, n$ let
T_i = time to outcome event,
U_i = time to censoring.

Censoring is assumed **noninformative**, *i.e.*
independent from occurrence of events.

We observe

$y_i = \min\{T_i, U_i\}$, *i.e.* the exit time, and

$\delta_i = 1_{\{T_i < U_i\}}$, indicator (1/0) for the outcome event occurring first, before censoring.

Censoring must properly be taken into account in the statistical analysis.

Approaches for analysing survival time

- **Parametric model** (like Weibull, gamma, etc.) on hazard rate $\lambda(t)$ $\rightarrow$ Likelihood:

$$L = \prod_{i=1}^{n} \lambda(y_i)^{\delta_i} S(y_i)$$

- **Piecewise constant rate** model on $\lambda(t)$
– see Bendix's lecture on time-splitting (Poisson likelihood).
- **Non-parametric** methods, like
Kaplan–Meier (KM) estimator of survival curve $S(t)$ and
Cox proportional hazards model on $\lambda(t)$.

R package survival

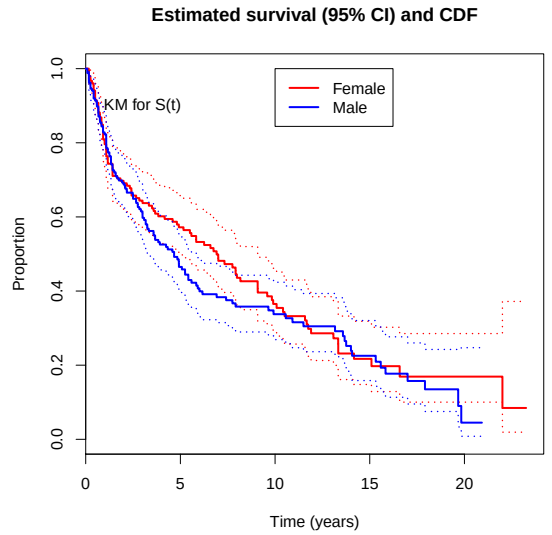
Tools for analysis with one outcome event.

- `Surv(time, event) -> sobj`
  creates a **survival object** `sobj` assuming that all start at 0, containing pairs (y_i, δ_i),
- `Surv(entry, exit, event) -> sobj2`
  creates a survival object from `entry` and `exit` times,
- `survfit(sobj ~ x) -> sfo`
  creates a **survfit** object `sfo` containing KM or other non-parametric estimates (also from a fitted Cox model),
- `plot(sfo)`
  plot method for survival curves and related graphs,
- `coxph(sobj ~ x1 + x2)`
  fits a Cox model with covariates `x1` and `x2`.
- `survreg()` – parametric survival models.

Ex. Oral cancer data (cont'd)

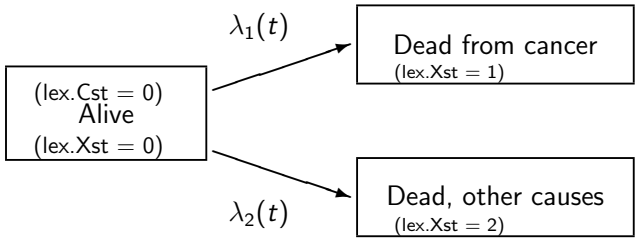
```
> orca$suob <- Surv(orca$time, 1*(orca$event > 0) )
> orca$suob[1:7]   #  + indicates censored observation
[1] 5.081+ 0.419  7.915  2.480  2.500  0.167  5.925+
> km1 <- survfit( suob ~ 1, data = orca)
> km1              #  brief  summary
Call: survfit(formula = suob ~ 1, data = orca)

      n  events  median 0.95LCL 0.95UCL
 338.00  229.00    5.42    4.33    6.92
> summary(km1)     #  detailed KM-estimate
Call: survfit(formula = suob ~ 1, data = orca)

   time n.risk n.event survival std.err lower 95% CI upper 95% CI
  0.085    338       2   0.9941 0.00417       0.9859        1.000
  0.162    336       2   0.9882 0.00588       0.9767        1.000
  0.167    334       4   0.9763 0.00827       0.9603        0.993
  0.170    330       2   0.9704 0.00922       0.9525        0.989
  0.246    328       1   0.9675 0.00965       0.9487        0.987
  0.249    327       1   0.9645 0.01007       0.9450        0.984
  0.252    326       3   0.9556 0.01120       0.9339        0.978
  0.329    323       1   0.9527 0.01155       0.9303        0.976
  0.334    322       1   0.9497 0.01189       0.9267        0.973
  0.413    321       1   0.9467 0.01221       0.9231        0.971
  0.419    320       6   0.9290 0.01397       0.9020        0.957
```


Oral cancer: Kaplan-Meier estimates



Competing risks model: causes of death

- Often the interest is focused on the risk or hazard of dying from one specific cause.
- That cause may eventually not be realized, because a **competing cause** of death hits first.



- Generalizes to several competing causes.

Competing events & competing risks

In many epidemiological and clinical contexts there are competing events that may occur before the target event and remove the person from the population at risk for the event, *e.g.*

- *target event*: occurrence of endometrial cancer, *competing events*: hysterectomy or death.
- *target event*: relapse of a disease (ending the state of remission), *competing event*: death while still in remission.
- *target event*: divorce, *competing event*: death of either spouse.

Event-specific quantities

Cumulative incidence function (CIF) or

$$F_c(t) = P(T \leq t \text{ and } C = c), \quad c = 1, 2,$$

From these one can recover

- $F(t) = \sum_c F_c(t)$, CDF of event-free survival time T, *i.e.* cumulative risk of any event by t.
- $S(t) = 1 - F(t)$, **event-free survival function**, *i.e.* probability of avoiding all events by t, but $S(t) \neq F_1(t) + F_2(t)$

Event-specific quantities (cont'd)

Event- or **cause-specific hazard function**

$$\lambda_c(t) = \lim_{\Delta \to 0} \frac{P(t < T \leq t + \Delta \text{ and } C = c \mid T > t)}{\Delta}$$
$$= \frac{f_c(t)}{1 - F(t)}$$

CIF = risk of event c over risk period $[0, t]$ in the presence of competing risks, also obtained

$$F_c(t) = \int_0^t \lambda_c(v) S(v) dv, \quad c = 1, 2,$$

More on the technical definitions of relevant quantities:
http://bendixcarstensen.com/AdvCoh/papers/fundamentals.pdf

Warning of "net risk" and "cause-specific survival"

- The "**net risk**" of outcome c by time t, assuming hypothetical elimination of competing risks, is often defined as
$$F_1^*(t) = 1 - S_1^*(t) = 1 - \exp\{-\Lambda_1(t)\} \neq S(t)$$
- In clinical survival studies, function $S_1^*(t)$ is often called "**cause-specific survival**", or "**net survival**"
- Yet, these *-functions, $F_1^*(t)$ and $S_1^*(t)$, lack proper probability interpretation when competing risks exist.
- Hence, their use should be viewed critically (Andersen & Keiding, *Stat Med*, 2012)

Analysis with competing events

Let U_i = censoring time, T_i = time to first event, and C_i = variable for event 1 or 2. We observe

- $y_i = \min\{T_i, U_i\}$, *i.e.* the exit time, and
- $\delta_{ic} = 1_{\{T_i < U_i \ \& \ C_i = c\}}$, indicator (1/0) for event c being first observed, $c = 1, 2$.

Non-parametric estimation of CIF

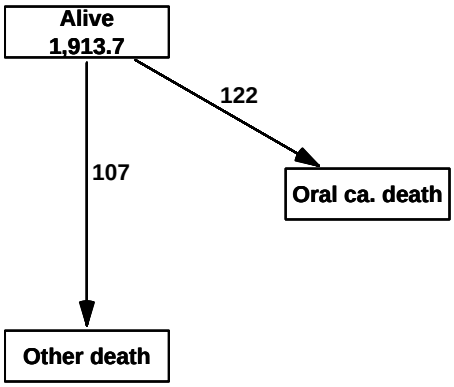
- Let $t_1 < t_2 < \cdots < t_K$ be the K distinct time points at which any outcome event was observed,
Let also $\widetilde{S}(t)$ be KM estimator for overall $S(t)$.
- **Aalen-Johansen estimator** (AJ) for the cumulative incidence function $F(t)$ should be used

R tools for competing risks analysis

- `survfit( Surv(...,type="mstate") )` in Survival-package can be fitted for any transition of a multistate model and to obtain A-J estimates.
- Package `cmprsk` – `cuminc(ftime, fstatus, ...)` computes CIF-estimates, and can be compared in more than two samples. `plot.cuminc()` plots them.
- Package `Epi` – `Lexis` tools for multistate analyses Will be advertised by Bendix!

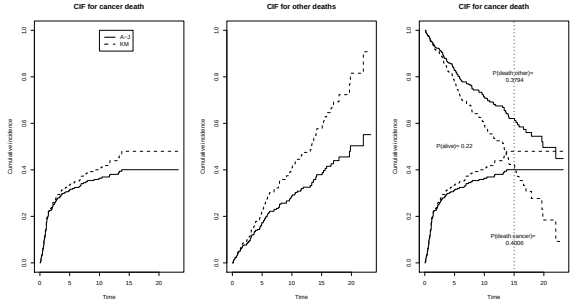
Box diagram for transitions

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



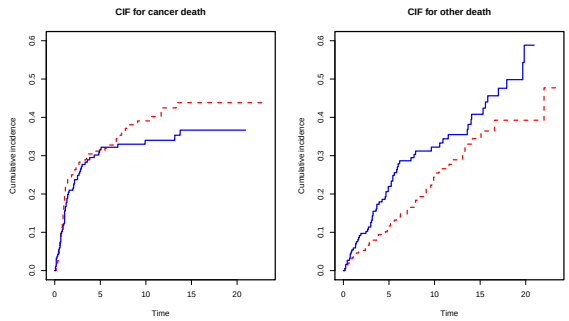
Ex. Survival from oral cancer

- AJ-estimates of CIFs (solid) for both causes.
- Naive KM-estimates of CIF (dashed) $>$ AJ-estimates
- CIF curves may also be stacked (right).



NB. The sum of the naive KM-estimates of CIF exceeds 100% at 13 years!

Ex. CIFs by cause in men and women



CIF for cancer higher in women (chance?) but for other causes higher in men (no surprise).

Regression models for time-to-event data

Regression models for hazards can be defined *e.g.* for

(a) hazards, multiplicatively:

$$\lambda_i(t) = \lambda_0(t;\alpha) r(\eta_i), \quad \text{where}$$

$\lambda_0(t;\alpha)$ = baseline hazard and
$r(\eta_i)$ = relative rate function, typically $\exp(\eta_i)$

(b) hazards, additively:

$$\lambda_i(t) = \lambda_0(t;\alpha) + \eta_i.$$

Relative hazards model or Cox model

In model (b), the baseline hazard $\lambda_0(t,\alpha)$ may be given a parametric form (*e.g.* Weibull) or a piecewise constant rate (exponential) structure.

Often a parameter-free form $\lambda_0(t)$ is assumed. Then

$$\lambda_i(t) = \lambda_0(t)\exp(\eta_1),$$

specifies the **Cox model** or the **semiparametric proportional hazards model**.

$\eta_i = \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$ not depending on time.

Generalizations: **time-dependent** covariates $x_{ij}(t)$

PH model: interpretation of parameters

Present the model explicitly in terms of x's and β's.

$$\lambda_i(t) = \lambda_0(t)\exp(\beta_1 x_{i1} + \cdots + \beta_p x_{ip})$$

Consider two individuals, i and i', having the same values of all other covariates except the j^{th} one.

The ratio of hazards is constant:

$$\frac{\lambda_i(t)}{\lambda_{i'}(t)} = \frac{\exp(\eta_i)}{\exp(\eta_{i'})} = \exp\{\beta_j(x_{ij} - x_{i'j})\}.$$

Thus $e^{\beta_j} = \text{HR}_j =$ **hazard ratio** or relative rate associated with a unit change in covariate X_j.

Ex. Total mortality of oral ca. patients

Fitting Cox models with sex and sex + age.

```
> cm0 <- coxph( suob ~ sex, data = orca)
> summary( cm0)
        coef exp(coef) se(coef)    z Pr(>|z|)
sexMale 0.126     1.134    0.134 0.94     0.35
        exp(coef) exp(-coef) lower .95 upper .95
sexMale      1.13      0.882     0.872      1.47

> cm1 <- coxph( suob ~ sex + age, data = orca)
> summary(cm1)
        exp(coef) exp(-coef) lower .95 upper .95
sexMale      1.49      0.669      1.14      1.96
age          1.04      0.960      1.03      1.05
```

The M/F contrast visible only after age-adjustment.

Predictions from the Cox model

- Individual survival *times* cannot be predicted but ind'l survival *curves* can. PH model implies:

$$S_i(t) = [S_0(t)]^{\exp(\beta_1 x_{i1} + \ldots + \beta_p x_{ip})}$$

- Having estimated β by partial likelihood, the baseline $S_0(t)$ is estimated by Breslow method
- From these, a survival curve for an individual with given covariate values is predicted.
- In R: `pred <- survfit(mod, newdata=...)` and `plot(pred)`, where `mod` is the fitted `coxph` object, and `newdata` specifies the covariate values. `newdata` is always needed for predictions.

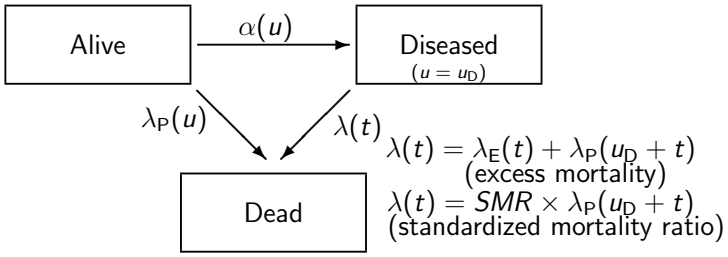
Modelling with competing risks

Main options, providing answers to different questions.

(a) Cox model for event-specific hazards $\lambda_c(t) = f_c(t)/[1 - F(t)]$, when *e.g.* the interest is in the biological effect of the prognostic factors on the fatality of the very disease that often leads to the relevant outcome.

(b) **Fine–Gray model** for the hazard of the subdistribution $\gamma_c(t) = f_c(t)/[1 - F_c(t)]$ when we want to assess the impact of the factors on the overall cumulative incidence of event c.
– Function `crr()` in package `cmprsk`.

Competing risks model: excess hazard of death



where

- $\lambda_P(u)$ is the hazard of dying from any cause among disease-free members
- $\lambda_E(t)$ is the excess hazard of dying from the disease among diseased cohort members

Rectal cancer

Ex. rectal cancers in females in Finland 2008-2012. Calculate observed mortality, excess mortality and relative mortality.

```
> library(popEpi) # R-package for population-based cancer analys
> library(Epi)
> library(survival)
> data("sire")
> head(sire)
```

```
   sex    bi_date    dg_date    ex_date status   dg_age
1:   1 1952-05-27 1994-02-03 2012-12-31      0 41.68877
2:   1 1959-04-04 1996-09-20 2012-12-31      0 37.46378
3:   1 1958-06-15 1994-05-30 2012-12-31      0 35.95616
4:   1 1957-05-10 1997-09-04 2012-12-31      0 40.32055
5:   1 1957-01-20 1996-09-24 2012-12-31      0 39.67745
6:   1 1962-05-25 1997-05-17 2012-12-31      0 34.97808
```

Rectal cancer

```
> data(sire)
>  ## split data
>  fotcut <- c(0,3/12,6/12,1,2,3,4,5)
>  lex.split <- lexpand(sire, birth = bi_date, entry = dg_date,
+                      exit = ex_date,
+                      status=status %in% 1:2,
+                      breaks = list(fot=fotcut),
+                      pophaz=popmort,# population mortality
+                      pp=F, # weights for survival estimation
+                      aggre = list(fot) )
> head(lex.split)
```

	fot	pyrs	at.risk	d.exp	from0to0	from0to1
1:	0.00	1946.997	8227	71.43614	105	717
2:	0.25	1779.831	7405	61.05649	103	431
3:	0.50	3215.778	6871	105.11004	190	633
4:	1.00	5459.795	6048	174.61314	340	791
5:	2.00	4501.971	4917	145.38757	294	492
6:	3.00	3825.438	4131	128.43103	281	322

Rectal cancer – mortality models

Modeling mortality by splitted follow-up time since cancer diagnosis (fot)

Estimate excess mortality $\lambda_E(t)$ (link function d.exp)

```
> excess.mort <- relpois_ag(formula = from0to1 ~ -1 + fot,
+ data = lex.split,
+ d.exp = d.exp,
+ offset = log(pyrs))
```

Estimate relative mortality (offset=log(d.exp))

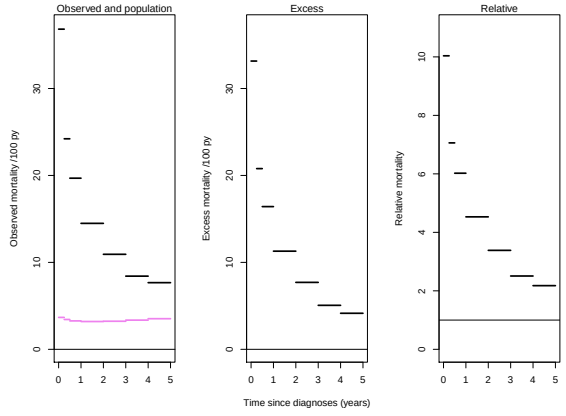
```
> relative.mort <- glm(formula = from0to1 ~ -1 + as.factor(fot),
+ family=poisson(link="log"),
+ data=lex.split,
+ offset=log(d.exp))
```

Estimate observed mortality (offset=log(pyrs))

```
> obs.mort <- glm(formula = from0to1 ~ -1 + as.factor(fot),
+ family=poisson(link="log"),
+ data=lex.split,
+ offset=log(pyrs))
```


Rectal cancer mortality

Observed, expected, excess and relative mortality



Representation of follow-up

Bendix Carstensen Steno Diabetes Center Copenhagen
Gentofte, Denmark
http://BendixCarstensen.com

IARC, Lyon,

June 2018

http://BendixCarstensen.com/SPE

Representation of follow-up

Bendix Carstensen

Representation of follow-up

IARC, Lyon,

June 2018

http://BendixCarstensen.com/SPE time-split

Follow-up and rates

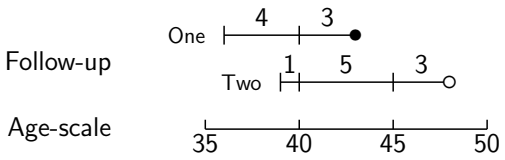
- In follow-up studies we estimate rates from:
  - D — events, deaths
  - Y — person-years
  - $\hat{\lambda} = D/Y$ rates
  - . . . empirical counterpart of intensity — **estimate**
- Rates differ between persons.
- Rates differ within persons:
  - By age
  - By calendar time
  - By disease duration
  - . . .
- Multiple timescales.
- Multiple states (little boxes — later)

Examples: 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.



— assuming a constant rate λ throughout.

Representation of follow-up data

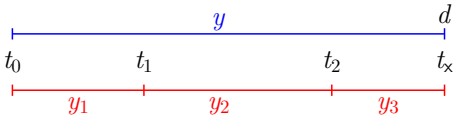
A cohort or follow-up study records:
Events and **Risk time**.

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

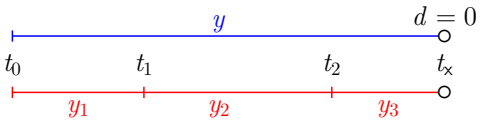
Follow-up **data** for each individual must therefore have (at least) three variables:

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

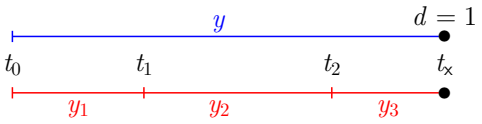
Specific for each **type** of outcome.



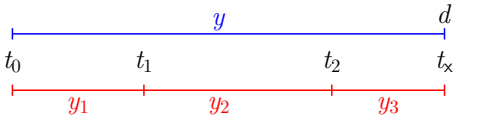
Probability	log-Likelihood
$\mathrm{P}(d \text{ at } t_x \mid \text{entry } t_0)$	$d\log(\lambda) - \lambda y$
$= \mathrm{P}(\text{surv } t_0 \to t_1 \mid \text{entry } t_0)$	$= 0\log(\lambda) - \lambda y_1$
$\times \mathrm{P}(\text{surv } t_1 \to t_2 \mid \text{entry } t_1)$	$+ 0\log(\lambda) - \lambda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



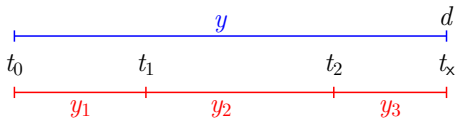
Probability	log-Likelihood
$\mathrm{P}(\text{surv } t_0 \to t_x \mid \text{entry } t_0)$	$0 \log(\lambda) - \lambda y$
$= \mathrm{P}(\text{surv } t_0 \to t_1 \mid \text{entry } t_0)$	$= 0 \log(\lambda) - \lambda y_1$
$\times \mathrm{P}(\text{surv } t_1 \to t_2 \mid \text{entry } t_1)$	$+ 0 \log(\lambda)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



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



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
$\mathrm{P}(d$ at $t_x \mid$entry $t_0)$	$d\log(\lambda) - \lambda y$
$= \mathrm{P}(\text{surv } t_0 \to t_1 \mid \text{entry } t_0)$	$= 0\log(\lambda_1) - \lambda_1 y_1$
$\times \mathrm{P}(\text{surv } t_1 \to t_2 \mid \text{entry } t_1)$	$+ 0\log(\lambda_2) - \lambda_2 y_2$
$\times \mathrm{P}(d$ at $t_x \mid$entry $t_2)$	$+ d\log(\lambda_3) - \lambda_3 y_3$

— allows different rates (λ_i) in each interval

Dividing time into bands:

If we want to compute D and Y in intervals on some timescale we must decide on:

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
- . . . do not confuse **duration** and **timescale** !

Follow-up data in `Epi` — `Lexis` objects

```
> thoro[1:6,1:8]
```

	id	sex	birthdat	contrast	injecdat	volume	exitdat	exitstat
1	1	2	1916.609	1	1938.791	22	1976.787	1
2	2	2	1927.843	1	1943.906	80	1966.030	1
3	3	1	1902.778	1	1935.629	10	1959.719	1
4	4	1	1918.359	1	1936.396	10	1977.307	1
5	5	1	1902.931	1	1937.387	10	1945.387	1
6	6	2	1903.714	1	1937.316	20	1944.738	1

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 = thoro )
```

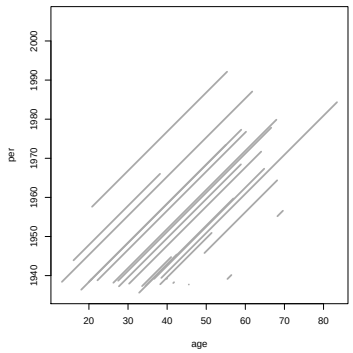
entry is defined on **three** timescales,
but exit is only needed on **one** timescale:
Follow-up time is the same on all timescales:

exitdat - injecdat

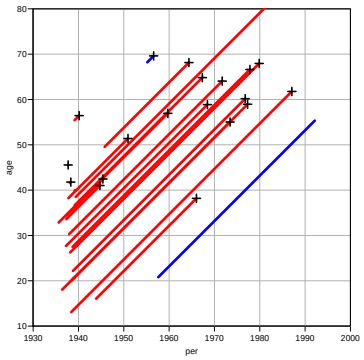
One element of entry and exit must have same name (per).

The looks of a Lexis object

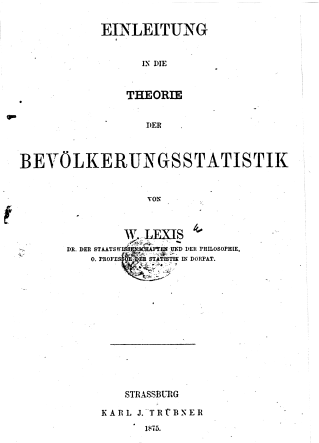
```
> thL[1:4,1:9]
    age     per tfi lex.dur lex.Cst lex.Xst lex.id
1 22.18 1938.79   0   37.99       0       1      1
2 49.54 1945.77   0   18.59       0       1      2
3 68.20 1955.18   0    1.40       0       1      3
4 20.80 1957.61   0   34.52       0       0      4
...

> summary( thL )
Transitions:
     To
From    0    1 Records:  Events:  Risk time:  Persons:
   0 504 1964      2468     1964    51934.08      2468
```


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




```
> 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 )
```



EINLEITUNG

IN DIE

THEORIE

DER

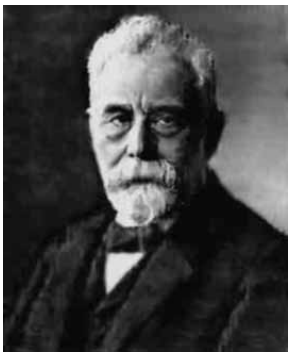
BEVÖLKERUNGSSTATISTIK

VON

W. LEXIS

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

STRASSBURG
KARL J. TRÜBNER
1875.

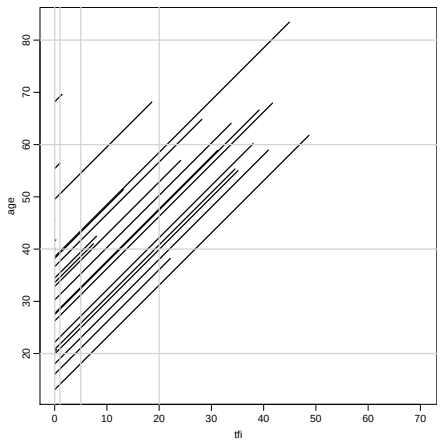


Splitting follow-up time

```
> 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 inj
1       1 22.2 1938.8  0.0     1.0       0       0    1   2    1916.6        1   1
2       1 23.2 1939.8  1.0     4.0       0       0    1   2    1916.6        1   1
3       1 27.2 1943.8  5.0    12.8       0       0    1   2    1916.6        1   1
4       1 40.0 1956.6 17.8     2.2       0       0    1   2    1916.6        1   1
5       1 42.2 1958.8 20.0    17.8       0       0    1   2    1916.6        1   1
6       1 60.0 1976.6 37.8     0.2       0       1    1   2    1916.6        1   1
7       2 49.5 1945.8  0.0     1.0       0       0  640   2    1896.2        1   1
8       2 50.5 1946.8  1.0     4.0       0       0  640   2    1896.2        1   1
9       2 54.5 1950.8  5.0     5.5       0       0  640   2    1896.2        1   1
10      2 60.0 1956.2 10.5     8.1       0       1  640   2    1896.2        1   1
11      3 68.2 1955.2  0.0     1.0       0       0 3425   1    1887.0        2   1
12      3 69.2 1956.2  1.0     0.4       0       1 3425   1    1887.0        2   1
13      4 20.8 1957.6  0.0     1.0       0       0 4017   2    1936.8        2   1
14      4 21.8 1958.6  1.0     4.0       0       0 4017   2    1936.8        2   1
15      4 25.8 1962.6  5.0    14.2       0       0 4017   2    1936.8        2   1
16      4 40.0 1976.8 19.2     0.8       0       0 4017   2    1936.8        2   1
17      4 40.8 1977.6 20.0    14.5       0       0 4017   2    1936.8        2   1
```


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

```
 age  tfi lex.dur lex.Cst lex.Xst
22.2  0.0     1.0       0       0
23.2  1.0     4.0       0       0
27.2  5.0    12.8       0       0
40.0 17.8     2.2       0       0
42.2 20.0    17.8       0       0
60.0 37.8     0.2       0       1
```


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 the likelihood.
- Each term looks like a contribution from a Possion variate (albeit with values only 0 or 1)
- 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 latter is where we will need splines.

The Poisson likelihood for split data

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

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

- Assuming that the death indicator ($d_{pi} \in \{0, 1\}$) is Poisson, a model with with offset $\log(y_{pi})$ will give the same result.
- If we assume that rates are constant we get the simple expression with (D, Y)
- . . . 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?

```
> spl1 <- splitLexis( thL , breaks=seq(0,100,20)  , time.scale="age" )
> spl2 <- splitLexis( spl1, breaks=c(0,1,5,20,100), time.scale="tfi" )
> options( digits=5 )
> spl2[1:10,1:11]
```

	lex.id	age	per	tfi	lex.dur	lex.Cst	lex.Xst	id	sex	birthdat	contrast
1	1	22.182	1938.8	0.000	1.00000	0	0	1	2	1916.6	1
2	1	23.182	1939.8	1.000	4.00000	0	0	1	2	1916.6	1
3	1	27.182	1943.8	5.000	12.81793	0	0	1	2	1916.6	1
4	1	40.000	1956.6	17.818	2.18207	0	0	1	2	1916.6	1
5	1	42.182	1958.8	20.000	17.81793	0	0	1	2	1916.6	1
6	1	60.000	1976.6	37.818	0.17796	0	1	1	2	1916.6	1
7	2	16.063	1943.9	0.000	1.00000	0	0	2	2	1927.8	1
8	2	17.063	1944.9	1.000	2.93703	0	0	2	2	1927.8	1
9	2	20.000	1947.8	3.937	1.06297	0	0	2	2	1927.8	1
10	2	21.063	1948.9	5.000	15.00000	0	0	2	2	1927.8	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	928.00	20094.74	4.62
2	1036.00	31839.35	3.25
Total	1964.00	51934.08	3.78

Fitting a simple model

contrast	D	Y	Rate
1	928.00	20094.74	4.62
2	1036.00	31839.35	3.25

```
> m0 <- glm( (lex.Xst==1) ~ 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      4.62 4.33  4.93
factor(contrast)2      3.25 3.06  3.46
```


SMR

Bendix Carstensen

Representation of follow-up

IARC, Lyon,

June 2018

http://BendixCarstensen.com/SPE

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)
- Hospital registers (disease rates)

Cohort rates vs. population rates: RSR

- **Additive:** $\lambda(a) = \delta(a) + \lambda_P(a)$
- Note that the survival (since $a = a_0$, say) is:

$$\begin{aligned} S(a) &= \exp\Big(-\int_{a_0}^{a} \delta(a) + \lambda_P(a)\,\mathrm{d}a\Big) \\ &= \exp\Big(-\int_{a_0}^{a} \delta(a)\,\mathrm{d}a\Big) \times S_P(a) \\ \Rightarrow \quad r(a) = S(a)/S_P(a) &= \exp\Big(-\int_{a_0}^{a} \delta(a)\,\mathrm{d}a\Big) \end{aligned}$$

- Additive model for rates $\Leftrightarrow$ Relative survival model.

Cohort rates vs. population rates: SMR

- **Multiplicative:** $\lambda(a) = \theta\lambda_P(a)$
- Cohort rates proportional to reference rates:
  $\lambda(a) = \theta \times \lambda_P(a)$ — θ the same in all age-bands.
- D_a deaths during Y_a person-years an age-band a gives the likelihood:

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

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

- $\lambda_P(a)\, Y_a = E_a$ is the "expected" number of cases in age a, so the log-likelihood contribution from age a is:

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

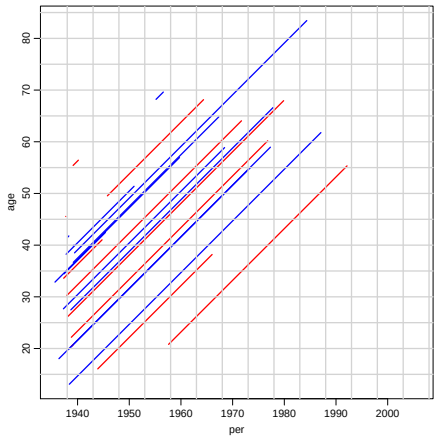
- **Note:** $\lambda_P(a)$ is known for all values of a.
- The log-likelihood is similar to the log-likelihood for a rate, except that person-years Y is replaced by expected numbers, E, so:

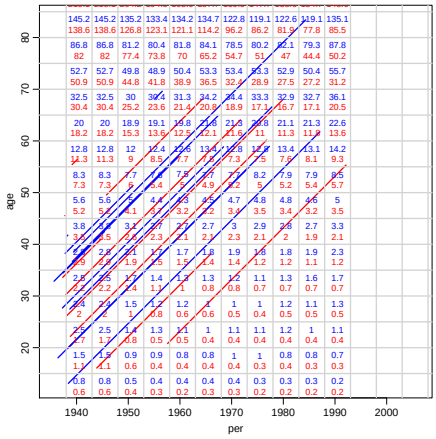
$$\hat{\theta} = \frac{D}{\lambda_P\, Y} = \frac{D}{E} = \frac{\text{Observed}}{\text{Expected}} = \text{SMR}$$

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

Modelling the SMR in practise

- 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, $e_i = y_i \times \lambda_P$ (`lex.dur * rate`)
- λ_P is the population rate corresponding to the age, period and sex of the follow-up period y_i.





Split the data to fit with population data

```
> tha  <- splitLexis(thL, time.scale="age", breaks=seq(0,90,5) )
> thap <- splitLexis(tha, time.scale="per", breaks=seq(1938,2038,5) )
> dim( thap )
```

```
[1] 23094    21
```

Create variables to fit with the population data

```
> thap$agr <- timeBand( thap, "age", "left" )
> thap$cal <- timeBand( thap, "per", "left" )
> round( thap[1:5,c("lex.id","age","agr","per","cal","lex.dur","lex.Xst","sex")],
```

```
  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
```



```
> data( gmortDK )
> gmortDK[1:6,1:6]
```

```
  agr per sex   risk    dt     rt
1   0  38   1 996019 14079 14.135
2   5  38   1 802334   726  0.905
3  10  38   1 753017   600  0.797
4  15  38   1 773393  1167  1.509
5  20  38   1 813882  2031  2.495
6  25  38   1 789990  1862  2.357
```

```
> gmortDK$cal <- gmortDK$per+1900
> #
> thapx <- merge( thap, gmortDK[,c("agr","cal","sex","rt")] )
> #
> 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           923.00 20072.53  222.01    4.16
 2          1036.00 31839.35  473.88    2.19

 Total      1959.00 51911.87  695.89    2.82
 ----------------------------------------------
```



```
 ----------------------------------------------
 contrast         D        Y       E     SMR
 ----------------------------------------------
 1           923.00 20072.53  222.01    4.16
 2          1036.00 31839.35  473.88    2.19

 Total      1959.00 51911.87  695.89    2.82
 ----------------------------------------------
> m.SMR <- glm( lex.Xst ~ factor(contrast) - 1,
+               offset = log(E),
+               family = poisson,
+                 data = thapx )
> round( ci.exp( m.SMR ), 2 )

                   exp(Est.) 2.5% 97.5%
factor(contrast)1       4.16 3.90  4.43
factor(contrast)2       2.19 2.06  2.32
```

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

Nested case-control studies and case-cohort studies

Monday, 18 June, 2018

Esa Läärä & Martyn Plummer

Statistical Practice in Epidemiology with R
International Agency for Research on Cancer
Lyon, France, 14 to 20 June, 2018

Points to be covered

- Outcome-dependent sampling designs a.k.a. **case-control** studies *vs.* **full cohort** design.
- **Nested case-control** study (NCC): sampling of controls from risk-sets during follow-up of study population.
- **Matching** in selection of control subjects in NCC.
- R tools for NCC: function `ccwc()` in `Epi` for sampling controls, and `clogit()` in `survival` for model fitting.
- **Case-cohort** study (CC): sampling a subcohort from the whole cohort as it is at the start of follow-up.
- R tools for CC model fitting: function `cch()` in `survival`

Example: Smoking and cervix cancer

Study population, measurements, follow-up, and sampling design

- Joint cohort of $N \approx$ 500 000 women from 3 Nordic biobanks.
- Follow-up: From variable entry times since 1970s till 2000.
- For each of 200 cases, 3 controls were sampled; matched for biobank, age (± 2 y), and time of entry (± 2 mo).
- Frozen sera of cases and controls analyzed for cotinine *etc.*

Main result: Adjusted OR = 1.5 (95% CI 1.1 to 2.3) for high (>242.6 ng/ml) *vs.* low (<3.0 ng/ml) cotinine levels.

Simen Kapeu *et al.* (2009) *Am J Epidemiol*

Example: USF1 gene and CVD

Study population, measurements, follow-up, and sampling design

- Two FINRISK cohorts, total $N \approx 14000$ M & F, 25-64 y.
- Baseline health exam, questionnaire & blood specimens at recruitment in the 1990s – Follow-up until the end of 2003.
- Subcohort of 786 subjects sampled.
- 528 incident cases of CVD; 72 of them in the subcohort.
- Frozen blood from cases and subchort members genotyped.

Main result: Female carriers of a high risk haplotype had a 2-fold hazard of getting CVD [95% CI: 1.2 to 3.5]

Komulainen *et al.* (2006) *PLoS Genetics*

Full cohort design & its simple analysis

- **Full cohort design**: Data on exposure variables obtained for all subjects in a large study population.
- Summary data for crude comparison:

	Exposed	Unexposed	Total
Cases	D_1	D_0	D
Non-cases	B_1	B_0	B
Group size at start	N_1	N_0	N
Follow-up times	Y_1	Y_0	Y

- Crude estimation of **hazard ratio** $\rho = \lambda_1/\lambda_0$: **incidence rate ratio** IR, with standard error of log(IR):

$$\widehat{\rho} = \mathrm{IR} = \frac{D_1/Y_1}{D_0/Y_0} \qquad \mathrm{SE}[\log(\mathrm{IR})] = \sqrt{\frac{1}{D_1} + \frac{1}{D_0}}.$$

- More refined analyses: Poisson or Cox regression.

Problems with full cohort design

Obtaining exposure and covariate data

- Slow and expensive in a big cohort.
- Easier with questionnaire and register data,
- Extremely costly and laborious for *e.g.*
  - measurements from biological specimens, like genotyping, antibody assays, *etc.*
  - dietary diaries & other manual records

Can we obtain equally valid estimates of hazard ratios etc. with nearly as good precision by some other strategies?

Yes – we can!

Estimation of hazard ratio

The incidence rate ratio can be expressed:

$$\text{IR} = \frac{D_1/D_0}{Y_1/Y_0} = \frac{\text{cases: exposed / unexposed}}{\text{person-times: exposed / unexposed}}$$

$$= \frac{\textit{exp're odds} \text{ in cases}}{\textit{exp're odds} \text{ in p-times}} = \textbf{exposure odds ratio} \text{ (EOR)}$$

= Exposure distribution in cases *vs.* that in cohort!

Implication for more efficient design:

- *Numerator*: Collect exposure data on all cases.
- *Denominator*: Estimate the ratio of person-times Y_1/Y_0 of the exposure groups in the cohort by **sampling** "control" subjects, on whom exposure is measured.

Case-control designs

General principle: Sampling of subjects from a given study population is *outcome-dependent*.

Data on risk factors are collected separately from

(I) **Case group**: All (or high % of) the D subjects in the study population (total N) encountering the outcome event during the follow-up.

(II) **Control group**:

- Random **sample** (simple or stratified) of C subjects ($C << N$) from the population.
- Eligible controls must be bf risk (alive, under follow-up & free of outcome) at given time(s).

Study population in a case-control study?

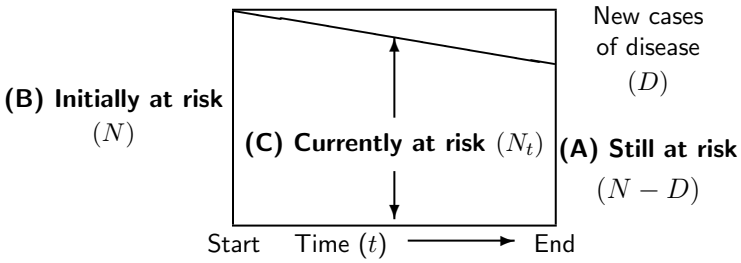
Ideally: The study population comprises subjects who *would be* included as cases, *if they got* the outcome in the study

- *Cohort-based studies*: **cohort** or **closed** population of well-identified subjects under intensive follow-up for outcomes (*e.g.* biobank cohorts).
- *Register-based studies*: **open** or **dynamic** population in a region covered by a disease register.
- *Hospital-based* studies: dynamic **catchment** population of cases – may be hard to identify (*e.g.* hospitals in US).

In general, the role of control subjects is to represent the distribution of person-times by exposure variables in the underlying population from which the cases emerge.

Sampling of controls – alternative frames

Illustrated in a simple longitudinal setting:
Follow-up of a cohort over a fixed risk period & no censoring.



Rodrigues, L. & Kirkwood, B.R. (1990). Case-control designs of common diseases . . . *Int J Epidemiol* **19**: 205-13.

Sampling schemes or designs for controls

(A) Exclusive or **traditional, "case-noncase" sampling**

- Controls chosen from those $N - D$ subjects still at risk (healthy) *at the end* of the risk period (follow-up).

(B) Inclusive sampling or **case-cohort design** (CC)

- The control group – *subcohort* – is a random sample of the whole cohort (N) *at start*.

(C) Concurrent sampling or **density sampling**

- Controls drawn *during the follow-up*
- **Risk-set** or **time-matched sampling:**
  A set of controls is sampled from the *risk set at each time t of diagnosis* of a new case

  a.k.a. **nested case-control design** (NCC)

Nested case-control – two meanings

- In some epidemiologic books, the term "nested case-control study" (NCC) covers jointly all variants of sampling: **(A)**, **(B)**, and **(C)**, from a cohort.

  Rothman *et al.* (2008): *Modern Epidemology, 3rd Ed.*
  Dos Santos Silva (1999): *Cancer Epidemiology.* Ch 8-9

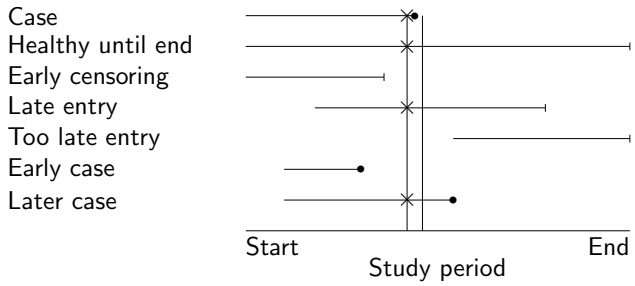
- In biostatistical texts NCC typically refers only to the variant of concurrent or density sampling **(C)**, in which *risk-set* or *time-matched* sampling is employed.

  Borgan & Samuelsen (2003) in *Norsk Epidemiologi*
  Langholz (2005) in *Encyclopedia of Biostatistics.*

- We shall follow the biostatisticians!

NCC: Risk-set sampling with staggered entry

Sampling frame to select controls for a given case:
Members ($\times$) of the **risk set** at t_k, *i.e.* the population at risk at the time of diagnosis t_k of case k.



Sampled risk set contains the case and the control subjects randomly sampled from the non-cases in the risk set at t_k.

Use of different sampling schemes

(A) Exclusive sampling, or "textbook" case-control design

- Almost exclusively(!) used in studies of epidemics.
- (Studies on birth defects with *prevalent* cases.)

(B) Inclusive sampling or case-cohort design

- Good esp. for multiple outcomes, if measurements of risk factors from stored material remain stable.

(C) Concurrent or density sampling
(without or with time-matching, *i.e.* NCC)

- The only logical design in an open population.
- Most popular in chronic diseases (Knol *et al.* 2008).

Designs **(B)** and **(C)** allow valid estimation of hazard ratios ρ without any "rare disease" assumption.

Case-control studies: Textbooks vs. real life

- Many epi texts focus on the traditional design: **exclusive sampling** of controls, ignoring other designs.
- Claim: "*Odds ratio is the only estimable parameter.*"
- Yet, over 60% of published case-control studies apply **concurrent sampling** or **density sampling** of controls from an **open** or **dynamic** population.
- Thus, the parameter most often estimated is the **hazard ratio** (HR) or **rate ratio** ρ.
- Still, 90% of authors really estimating HR, reported as having estimated an OR (*e.g.* Simen Kapeu *et al.*)

Knol *et al.* (2008). What do case-control studies estimate? *Am J Epidemiol* **168**: 1073-81.

Exposure odds ratio – estimate of what?

- Crude summary of case-control data

	exposed	unexposed	total
cases	D_1	D_0	D
controls	C_1	C_0	C

- Depending on study base & sampling strategy, the empirical **exposure odds ratio** (EOR)

$$\text{EOR} = \frac{D_1/D_0}{C_1/C_0} = \frac{\text{cases: exposed / unexposed}}{\text{controls: exposed / unexposed}}$$

  is a consistent estimator of
  (a) hazard ratio, (b) risk ratio, (c) risk odds ratio,
  (d) prevalence ratio, or (e) prevalence odds ratio

- **NB.** In case-cohort studies with variable follow-up times C_1/C_0 is substituted by $\widehat{Y}_1/\widehat{Y}_0$, from estimated p-years.

Precision and efficiency

With exclusive **(A)** or concurrent **(C)** sampling of controls (unmatched), estimated variance of log(EOR) is

$$\begin{aligned} \widehat{\text{var}}[\log(\text{EOR})] &= \frac{1}{D_1} + \frac{1}{D_0} + \frac{1}{C_1} + \frac{1}{C_0} \\ &= \text{cohort variance} + \text{sampling variance} \end{aligned}$$

- Depends basically on the numbers of cases, when there are ≥ 4 controls per case.
- Is not much bigger than $1/D_1 + 1/D_0 =$ variance in a full cohort study with same numbers of cases.

$\Rightarrow$ Usually < 5 controls per case is enough.

$\Rightarrow$ *These designs are very cost-efficient!*

Estimation in concurrent or density sampling

- To put it simply: Exposure odds C_1/C_0 among controls = consistent estimator of exposure odds Y_1/Y_0 of person-times.
- Therefore, crude EOR $= (D_1/D_0)/(C_1/C_0)$ = consistent estimator of hazard ratio $\rho = \lambda_1/\lambda_0$.
- Variance of log(EOR) estimated as above.
- Yet, with a closed population or cohort, stability of exposure distribution may be unrealistic.
- Solution: **Time-matched** sampling of controls from **risk sets**, *i.e.* NCC, & matched EOR to estimate HR.

Prentice & Breslow (1978), Greenland & Thomas (1982).

Matching in case-control studies

= **Stratified sampling** of controls, *e.g.* from the same region, sex, and age group as a given case

- **Frequency matching** or **group matching:** For cases in a specific stratum (*e.g.* same sex and 5-year age-group), a set of controls from a similar subgroup.
- **Individual matching** (1:1 or 1:m matching): For each case, choose 1 or more (rarely > 5) closely similar controls (*e.g.* same sex, age within ± 1 year.
- **NCC**: Sampling from risk-sets implies time-matching at least. Additional matching for other factors possible.
- **CC**: Subcohort selection involves no matching with cases.

Virtues of matching

- Increases *efficiency*, if the matching factors are both
  (i) strong *risk factors* of the disease, and
  (ii) *correlated* with the main exposure.
  – Major reason for matching.
- *Confounding* due to poorly quantified factors (sibship, neighbourhood, *etc.*) may be removed by close matching – only if properly analyzed.
- Biobank studies: Matching for storage time, freeze-thaw cycle & analytic batch improves **comparability of measurements** from frozen specimens
  → Match on the time of baseline measurements within the case's risk set.

Warnings for overmatching

Matching a case with a control subject is a different issue than matching an unexposed subject to an exposed one in a cohort study – much trickier!

- Matching on an *intermediate* variable between exposure and outcome. ⇒ *Bias*!
- Matching on a *surrogate* or *correlate* of exposure, which is not a true risk factor.
  ⇒ *Loss of efficiency*.

→ **Counter-matching:** Choose a control which is not similar to the case w.r.t a correlate of exposure.

⇒ Increases efficiency!

- Requires appropriate weighting in the analysis.

Sampling matched controls for NCC using R

- Suppose key follow-up items are recorded for all subjects in a cohort, in which a NCC study is planned.
- Function `ccwc()` in package `Epi` can be used for risk-set sampling of controls. – Arguments:

`entry` :	Time of entry to follow-up
`exit` :	Time of exit from follow-up
`fail` :	Status on exit (1 for case, 0 for censored)
`origin` :	Origin of analysis time scale (e.g. time of birth)
`controls` :	Number of controls to be selected for each case
`match` :	List of matching factors
`data` :	Cohort data frame containing input variables

- Creates a data frame for a NCC study, containing the desired number of matched controls for each case.

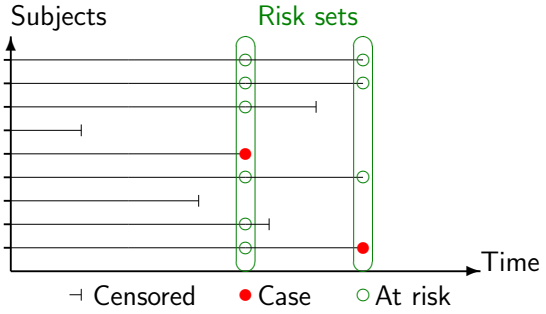
Analysis of matched studies

- Close matching induces a new parameter for each matched case-control set or stratum.
  $\Rightarrow$ **unconditional logistic regression** breaks down.
- Matching on well-defined variables (like age, sex) – include these factors as covariates.
- Matching on "soft" variables (like sibship) can be dealt with **conditional logistic regression**.
- Same method in matched designs **(A)**, exclusive, and **(C)**, concurrent, but interpretation of β_js differs:

  (A) β_j = log of risk odds ratio (ROR),
  (C) β_j = log of hazard ratio (HR).

Full cohort design: Follow-up & risk sets

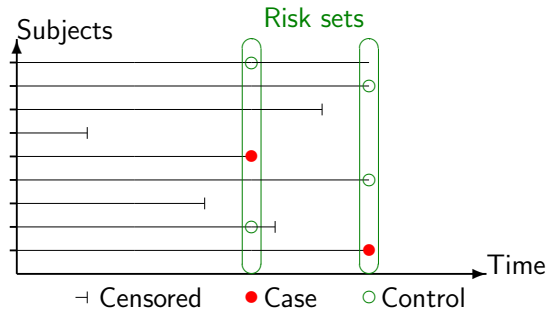
Each member of the cohort provides exposure data for all cases, as long as this member is at risk, *i.e.*
(i) alive, (ii) not censored & (iii) free from outcome.



Times of new cases define the **risk-sets**.

Nested case-control (NCC) design

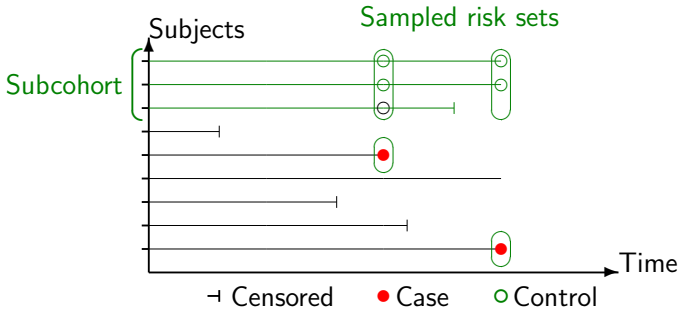
Whenever a new case occurs, a set of controls (here 2/case) are sampled from its risk set.



NB. A control once selected for some case can be selected as a control for another case, and can later on become a case, too.

Case-cohort (CC) design

Subcohort: Sample of the whole cohort randomly selected at the outset. – Serves as reference group for all cases.



NB. A subcohort member can become a case, too.

Modelling in NCC and other matched studies

Cox proportional hazards model:

$$\lambda_i(t, x_i; \beta) = \lambda_0(t) \exp(x_{i1}\beta_1 + \cdots + x_{ip}\beta_p),$$

Estimation: partial likelihood $L^P = \prod_k L_k^P$:

$$L_k^P = \exp(\eta_{i_k}) / \sum_{i \in \widetilde{R}(t_k)} \exp(\eta_i),$$

where $\widetilde{R}(t_k)$ = **sampled risk set** at observed event time t_k, containing the case + sampled controls $(t_1 < \cdots < t_D)$

$\Rightarrow$ Fit stratified Cox model, with $\widetilde{R}(t_k)$'s as the strata.

$\Leftrightarrow$ **Conditional logistic regression**
– function `clogit()` in `survival`, wrapper of `coxph()`.

Modelling case-cohort data

Cox's PH model $\lambda_i(t) = \lambda_0(t)\exp(\eta_i)$ again, but ...

- Analysis of survival data relies on the theoretical principle that you *can't know the future*.
- Case-cohort sampling breaks this principle: cases are sampled based on what *is known* to be happening to them during follow-up.
- The union of cases and subcohort is a mixture
  1. random sample of the population, and
  2. "high risk" subjects who are *certain* to become cases.

$\Rightarrow$ Ordinary Cox partial likelihood is wrong.

- Overrepresentation of cases must be corrected for, by (I) **weighting**, or (II) **late entry method**.

Correction method I – weighting

The method of **weighted partial likelihood** borrows some basics ideas from survey sampling.

- Sampled risk sets
  $\widetilde{R}(t_k) = \{\text{cases}\} \cup \{\text{subcohort members}\}$ at risk at t_k.
- Weights:
  - $w = 1$ for all cases (within and out of subcohort),
  - $w = N_{\text{non-cases}}/n_{\text{non-cases}}$ = inverse of sampling-fraction f for selecting a non-case to the subcohort.
- Function `coxph()` with option `weights = w` would provide consistent estimation of β parameters.
- However, the SEs must be corrected!
- R solution: Function `cch()` – a wrapper of `coxph()` – in package `survival`, with `method = "LinYing"`.

Comparison of NCC and CC designs

- Statistical efficiency

  Broadly similar in NCC and CC with about same amounts of cases and controls.
- Statistical modelling and valid inference

  Straightforward for both designs with appropriate software, now widely available for CC, too
- Analysis of outcome rates on several time scales?

NCC: Only the time scale used in risk set definition can be the time variable t in the baseline hazard of PH model.

CC: Different choices for the basic time in PH model possible, because subcohort members are not time-matched to cases.

Comparison of designs (cont'd)

- Missing data

  NCC: With close 1:1 matching, a case-control pair is lost, if either of the two has data missing on key exposure(s).

  CC: Missingness of few data items is less serious.

- Quality and comparability of biological measurements

  NCC: Allows each case and its controls to be matched also for analytic batch, storage time, freeze-thaw cycle, $\rightarrow$ better comparability.

  CC: Measurements for subcohort performed at different times than for cases $\rightarrow$ differential quality & misclassification.

- Possibility for studying many diseases with same controls

  NCC: Complicated, but possible if matching is not too refined.

  CC: Easy, as no subcohort member is "tied" with any case.

Conclusion

- "Case-controlling" is very cost-effective.
- Case-cohort design is useful especially when several outcomes are of interest, given that the measurements on stored materials remain stable during the study.
- Nested case-control design is better suited *e.g.* for studies involving biomarkers that can be infuenced by analytic batch, long-term storage, and freeze-thaw cycles.
- Matching helps in improving effciency and in reducing bias – but only if properly done.
- Handy R tools are available for all designs.

Some topics on causal inference

Krista Fischer

Estonian Genome Center, University of Tartu, Estonia

Statistical Practice in Epidemiology, Lyon 2018

Statistical associations vs causal effects in epidemiology

Does the exposure (smoking level, obesity, etc) have a causal effect on the outcome (cancer diagnosis, mortality, etc)?

is not the same question as

Is the exposure associated with the outcome?

Conventional statistical analysis will answer the second one, but not necessarily the first.

What is a causal effect?

There is more than just one way to define it.
A causal effect may be defined:

- At the individual level:
  Would my cancer risk be different if I were a (non-)smoker?
- At the population level:
  Would the population cancer incidence be different if the prevalence of smoking were different?
- At the *exposed subpopulation level*:
  Would the cancer incidence in smokers be different if they were nonsmokers?

None of these questions is "mathematical" enough to provide a mathematically correct definition of causal effect

Causal effects and counterfactuals

- Defining the causal effect of an observed exposure always involves some counterfactual (what-if) thinking.
- The individual causal effect can be defined as the difference
  $$Y(X = 1) - Y(X = 0)$$
  . where $Y(1) = Y(X = 1)$ and $Y(0) = Y(X = 0)$ are defined as individual's potential (counterfactual) outcomes if this individual's exposure level X were set to 1 or 0, respectively.
- Sometimes people (e.g J. Pearl) use the "do" notation to distinguish counterfactual variables from the observed ones: $Y(do(X = 1))$ and $Y(do(X = 0))$.

The "naïve" association analysis

- With a binary exposure X, compare average outcomes in exposed and unexposed populations:
  $$E(Y|X = 1) - E(Y|X = 0)$$
  Is cancer incidence different in smokers and nonsmokers?
- But mostly:
  $$E(Y|X = 1) \neq E(Y(1))$$
  Cancer risk in smokers is not the same as the potential cancer risk in the population if everyone were smoking
- Similarly:
  $$E(Y|X = 0) \neq E(Y(0))$$
- In most cases there is always some unobserved confounding present and therefore the naïve analysis does not provide causal effect estimates.

Counterfactual outcomes in different settings

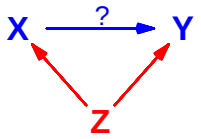
- Randomized trials: probably the easiest setting to imagine $Y(X)$ for different X
- "Actionable" exposures: smoking level, vegetable consumption, . . . – potential interventions may alter exposure levels in future.
- Non-actionable exposures: e.g genotypes. It is difficult to ask *"What if I had different genes?"*. Still useful concept to formalize genetic effects (heritability, attributable risk).
- Combinations: With X– a behavioral intervention level, Z–smoking level and Y–a disease outcome, one could formalize the effect of intervention on outcome by using $Y(X, Z(X))$

Classical/generalized regression estimates vs causal effects?

- In the presence of confounding, regression analysis provides a biased estimate for the true causal effect
- To reduce such bias, one needs to collect data on most important confounders and adjust for them
- However, too much adjustment may actually introduce more biases
- Causal graphs (Directed Acyclic Graphs, DAGs) may be extremly helpful in identifying the optimal set of adjustment variables

Adjustment for confounders I

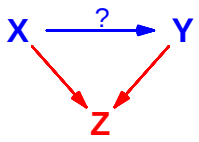
"Classical" confounding: situation where third factors Z influence both, X and Y



For instance, one can assume: $X = Z + U$ and $Y = Z + V$, where U and V are independent of Z.
X and Y are independent, conditional on Z, but marginally dependent.
One should adjust the analysis for Z, by fitting a regression model for Y with covariates X and Z. There is a causal effect between X and Y, if the effect of X is present in such model.

Adjustment may sometimes make things worse

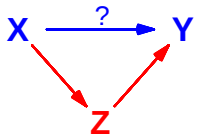
Example: the effect of X and Y on Z:



A simple model may hold: $Z = X + Y + U$,
where U is independent of X and Y.
Hence $Y = Z - X - U$.
We see the association between X and Y only when the "effect" of Z has been taken into account. But this is not the causal effect of X on Y.
One should NOT adjust the analysis for Z!

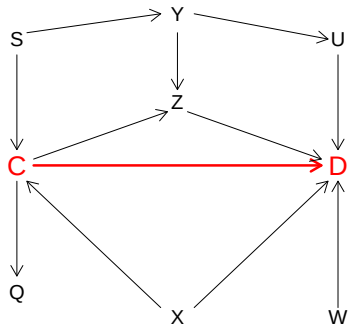
More possibilities: mediation

Example: the effect of X on Y is (partly) mediated by Z:



$Y = X + Z + U$,
If you are interested in the total effect of X on Y – don't adjust for Z!
If you are interested in the direct effect of X on Y – adjust for Z.
(Only if the Z-Y association is unconfounded)

Actually there might be a complicated system of causal effects:

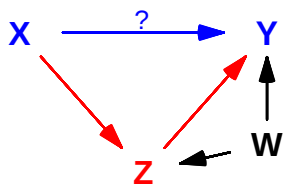


C-smoking; D-cancer
Q, S, U, W, X, Y, Z - other factors that influence cancer risks and/or smoking (genes, social background, nutrition, environment, personality, . . .)

To check for confounding,

1. Sketch a causal graph
2. Remove all arrows corresponding to the causal effect of interest (thus, create a graph where the causal null-hypothesis would hold).
3. Remove all nodes (and corresponding edges) except those contained in the exposure (C) and outcome (D) variables and their (direct or indirect) ancestors.
4. Connect by an undirected edge every pair of nodes that both share a common child and are not already connected by a directed edge.
   - If now C and D are still associated, we say that the $C - D$ association is confounded
   - Identify the set of nodes that need to be deleted to separate C and D – inferences conditional on these variables give unconfounded estimates of the causal effects.

Example: mediation with confounding



Follow the algorithm to show that one should adjust the analysis for W. If W is an unobserved confounder, no valid causal inference is possible in general. However, the total effect of X on Y is estimable.

Instrumental variables estimation and Mendelian randomization

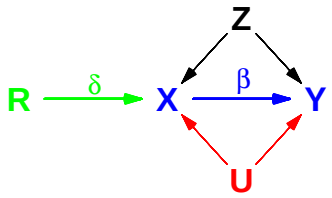
"Mendelian randomization" – genes as Instrumental Variables

- Most of the exposures of interest in chronic disease epidemiology cannot be randomized.
- Sometimes, however, nature will randomize for us: there is a SNP (Single nucleotide polymorphism, a DNA marker) that affects the exposure of interest, but not directly the outcome.
- Example: a SNP that is associated with the enzyme involved in alcohol metabolism, genetic lactose intolerance, etc.

However, the crucial assumption that the SNP cannot affect outcome in any other way than throughout the exposure, cannot be tested statistically!

General instrumental variables estimation

A causal graph with exposure X, outcome Y, confounder U and an *instrument* Z:

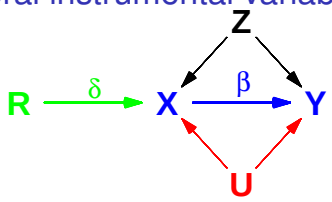


Simple regression will yield a biased estimate of the causal effect of X on Y, as the graph implies:

$$Y = \alpha_y + \beta X + \gamma U + \epsilon,\ \mathrm{E}(\epsilon|X,U) = 0$$

so $\mathrm{E}(Y|X) = \alpha_y + \beta X + \gamma \mathrm{E}(U|X)$.
Thus the coefficient of X will also depend on γ and the association between X and U.

General instrumental variables estimation



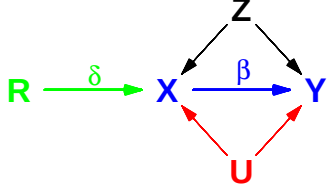
$$Y = \alpha_y + \beta X + \gamma U + \epsilon,\ \mathrm{E}(\epsilon|X,U) = 0$$

How can Z help?
If $\mathrm{E}(X|Z) = \alpha_x + \delta Z,$ we get

$$\mathrm{E}(Y|Z) = \alpha_y + \beta E(X|Z) + \gamma E(U|Z) = \alpha_y + \beta(\alpha_x + \delta Z) = \alpha_y^* + \beta\delta Z.$$

As δ and $\beta\delta$ are estimable, also β becomes estimable.

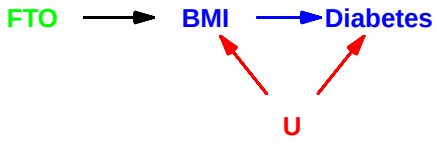
General instrumental variables estimation



1. Regress X on Z, obtain an estimate $\hat{\delta}$
2. Regress Y on Z, obtain an estimate $\hat{\delta\beta}$
3. Obtain $\hat{\beta} = \frac{\hat{\delta\beta}}{\hat{\delta}}$
4. Valid, if Z is not associated with U and does not have any effect on Y (other than mediated by X)
5. Standard error estimation is more tricky – use for instance `library(sem)`, function `tsls()`.

Mendelian randomization example

FTO genotype, BMI and Blood Glucose level (related to Type 2 Diabetes risk; Estonian Biobank, n=3635, aged 45+)



- Average difference in Blood Glucose level (Glc, mmol/L) per BMI unit is estimated as 0.085 (SE=0.005)
- Average BMI difference per FTO risk allele is estimated as 0.50 (SE=0.09)
- Average difference in Glc level per FTO risk allele is estimated as 0.13 (SE=0.04)
- Instrumental variable estimate of the mean Glc difference per BMI unit is 0.209 (se=0.078)

IV estimation in R (using `library(sem)`):

```
> summary(tsls(Glc~bmi, ~fto,data=fen),digits=2)

 2SLS Estimates

Model Formula: Glc ~ bmi

Instruments: ~fto

Residuals:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-6.3700 -1.0100 -0.0943  0.0000  0.8170 13.2000

            Estimate Std. Error t value Pr(>|t|)
(Intercept)   -1.210      2.106    -0.6    0.566
bmi            0.209      0.078     2.7    0.008 **
```


IV estimation: can untestable assumptions be tested?

```
> summary(lm(Glc~bmi+fto,data=fen))
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.985  0.106   18.75   <2e-16 ***
bmi         0.088  0.004   23.36   <2e-16 ***
fto         0.049  0.030    1.66    0.097 .

For Type 2 Diabetes:
> summary(glm(t2d~bmi+fto,data=fen,family=binomial))
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -7.515    0.187  -40.18   <2e-16 ***
bmi          0.185    0.006   31.66   <2e-16 ***
fto          0.095    0.047    2.01    0.044 *
```

Does FTO have a direct effect on Glc or T2D?
A significant FTO effect would not be a proof here (nor does non-significance prove the opposite)! (WHY?)

Can we test pleiotropy?

A naïve approach would be to fit a linear regression model for Y, with both X and G as covariates.
But in this case we estimate:

$$\mathrm{E}(Y|X,G) = \text{const} + \beta_{pl}G + \beta X + \gamma \mathrm{E}(U|X,G).$$

It is possible to show that U is not independent of neither X nor G – therefore, the coefficient of G in the resulting model would be nonzero even if $\beta_{pl} = 0$.
Therefore there is no formal test for pleiotropy possible in the case of one genetic instrument – only biological arguments could help to decide, whether assumptions are likelt to be fulfilled
In the case of *multiple genetic instruments* and *meta-analysis*, sometimes the approach of *Egger regression* can be used (Bowden et al, 2015). But even that is not an assumption-free method!

Summary

- There is no unique definition of “the causal effect”
- The validity of any causal effect estimates depends on the validity of the underlying assumptions.
- Adjustment for other available variables may remove (some) confounding, but it may also create more confounding. Do not adjust for variables that may themselves be affected by the outcome.
- Instrumental variables approaches can be helpful, but beware of assumptions!

Some references

- A webpage by Miguel Hernan and Jamie Robins: http://www.hsph.harvard.edu/miguel-hernan/causal-inference-book/
- An excellent overview of Mendelian randomization: Sheehan, N., Didelez, V., Burton, P., Tobin, M., Mendelian Randomization and Causal Inference in Observational Epidemiology, PLoS Med. 2008 August; 5(8).
- A way to correct for pleiotropy bias: Bowden J, Davey Smith G, Burgess S, Mendelian randomization with invalid instruments: effect estimation and bias detection through Egger regression. Int J Epidemiol. 2015 Apr;44(2):512-25.
- . . . and how to interpret the findings (warning against overuse): Burgess, S., Thompson, S.G., Interpreting findings from Mendelian randomization using the MR-Egger method, Eur J Epidemiol (2017).

Multistate models

Bendix Carstensen Steno Diabetes Center
Gentofte, Denmark
http://BendixCarstensen.com

IARC, Lyon,

June 2018

http://BendixCarstensen.com/SPE

Multistate models

Bendix Carstensen, Martyn Plummer

Multistate models

IARC, Lyon,

June 2018

http://BendixCarstensen.com/SPE ms-Markov

Common assumptions in survival analysis

1. Subjects are **either** "healthy" **or** "diseased", with no intermediate state.
2. The disease is **irreversible**, or requires intervention to be cured.
3. The time of disease incidence is known **exactly**.
4. The disease is **accurately** diagnosed.

These assumptions are true for death and many chronic diseases.

Is the disease a dichotomy?

A disease may be preceded by a sub-clinical phase before it shows symptoms.

AIDS	Decline in CD4 count
Cancer	Pre-cancerous lesions
Type 2 Diabetes	Impaired glucose tolerance

Or a disease may be classified into degrees of severity (mild, moderate, severe).

A model for cervical cancer

Invasive squamous cell cancer of the cervix is preceded by cervical intraepithelial neoplasia (CIN)

The purpose of a screening programme is to detect and treat CIN.

Aim of the modeling the transition rates between states, is to be able predict how population moves between states

Probabilities of state occupancy can be calculated.

When does the disease occur?

You may need a clinical visit to diagnose the disease:

- examination by physician, or
- laboratory test on blood sample, or
- examination of biopsy by pathologist

We do not know what happens between consecutive visits (interval censoring).

Informative observation process?

Is the **reason** for the visit dependent on the **evolution** of disease?

Ignoring this may cause bias, like informative censoring.

Different reasons for follow-up visits:

- Fixed intervals (OK)
- Random intervals (OK)
- Doctor's care (OK)
- Self selection (**Not** OK — visits are likely to be close to event times)

Markov models for multistate diseases

The natural generalization of Poisson regression to multiple disease states:

- Probability of transition between states depends **only** on current state
- — this is the **Markov** property
- $\Rightarrow$ transition rates are constant over time
- (time-fixed) covariates may influence transition rates
- the formal Markov property is **very** restrictive
- In clinical litterature "Markov model" is often used about any type of multistate model

Compnents of a multistate (Markov) model

- Define the disease states.
- Define which transitions between states are allowed.
- Select covariates influencing transition rates (may be different between transitions)
- Constrain some covariate effects to be the same, or zero.
- Not a trivial task — do we want *e.g.*
  - cause of death
  - disease status at death

Likelihood for multistate model

- The likelihood of the model depends on the probability of being in state j at time t_1, given that you were in state i at time t_0.
- Assume transition rates constant in small time intervals
- $\Rightarrow$ each interval contributes terms to the likelihood:
  - one for each person at risk of a transition in the interval
  - . . . for each possible transition
  - each term has the form of a Poisson likelihood contribution
  - the total likelihood for each time interval is a product of terms over persons and (possible) transitions
- Total likelihood is product of terms for all intervals
- — components **not** independent, but the total likelihood is a product; hence of the same form as the likelihood of independent Poisson variates

Purpose of multistate modeling

- Separation of intensities of interest (model definition)
- Evaluation of covariate effects on these
- — biological interpretability of covariate effects
- Use a fitted model to compute:
- state occupancy probabilities: $\mathrm{P}\{\text{in state } X \text{ at time } t\}$
- time spent in a given state

Special multistate models

- If all transition rates depend on only one time scale
- — but possibly different (time-fixed) covariates
- $\Rightarrow$ easy to compute state probabilities
- For this reason the most commonly available models
- but not the most realistic models.
- Realistically transition rates depend on:
- multiple time scales
- time since entry to certain states.

Multistate models with Lexis

Bendix Carstensen

Multistate models

IARC, Lyon,

June 2018

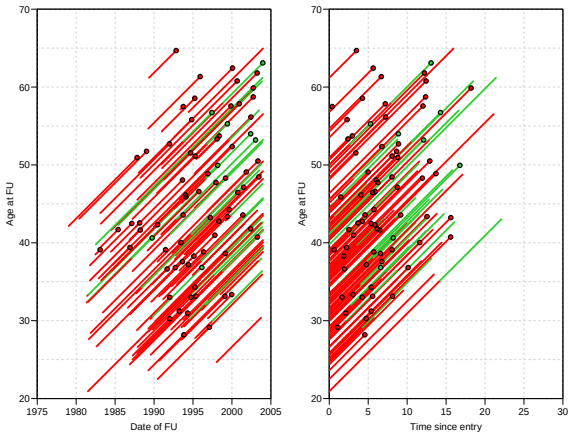
http://BendixCarstensen.com/SPE ms-Lexis

Example: Renal failure data from Steno

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

- 96 patients entering at nephrotic range albuminuria (NRA), i.e. U-alb>300mg/day.
- Is remission from this condition (i.e return to U-alb<300mg/day) predictive of the prognosis?
- Endpoint of interest: Death or end stage renal disease (ESRD), i.e. dialysis or kidney transplant.

		Remission	
	Total	Yes	No
No. patients	125	32	93
No. events	77	8	69
Follow-up time (years)	1084.7	259.9	824.8
Cox-model:			
Timescale:	Time since nephrotic range albuminuria (NRA)		
Entry:	2.5 years of GFR-measurements after NRA		
Outcome:	ESRD or Death		
Estimates:	RR	95% c.i.	p
Fixed covariates:			
Sex (F vs. M):	0.92	(0.53,1.57)	0.740
Age at NRA (per 10 years):	1.42	(1.08,1.87)	0.011
Time-dependent covariate:			
Obtained remission:	0.28	(0.13,0.59)	0.001



Features of the analysis

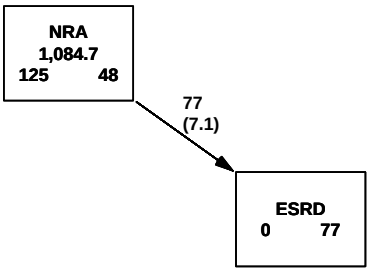
- Remission is included as a time-dependent variable.
- Age at entry is included as a fixed variable.

```
renal[1:5,]
id       dob       doe       dor       dox event
17 1967.944 1996.013        NA 1997.094     2
26 1959.306 1989.535 1989.814 1996.136     1
27 1962.014 1987.846        NA 1993.239     3
33 1950.747 1995.243 1995.717 2003.993     0
42 1961.296 1987.884 1996.650 2003.955     0
```

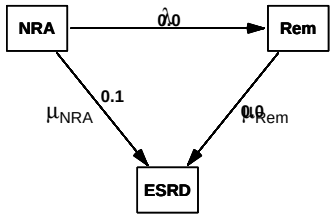
Note patient 26, 33 and 42 obtain remission.


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

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

```
> boxes( Lr, boxpos=list(x=c(25,75),
+                        y=c(75,25)),
+            scale.R=100, show.BE=TRUE )
```



Illness-death model



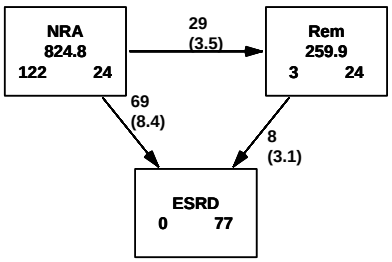
λ: remission rate.
μ_{NRA}: mortality/ESRD rate **before** remission.
μ_{rem}: mortality/ESRD rate **after** remission.

Cutting follow-up at remission: `cutLexis`

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

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

Showing states and FU: boxes.Lexis

```
> boxes( Lc, boxpos=list(x=c(15,85,50),
+                        y=c(85,85,20)),
            scale.R=100, show.BE=TRUE )
```

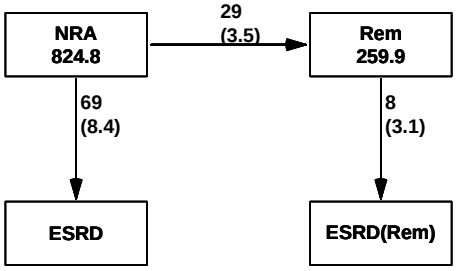


Splitting states: cutLexis

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

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

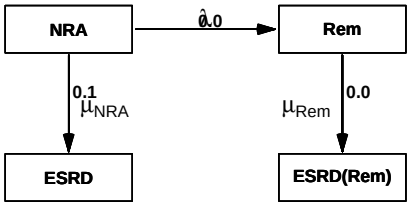

Showing states and FU: boxes.Lexis

```
> boxes( Lc, boxpos=list(x=c(15,85,15,85),
+                        y=c(85,85,20,20)), scale.R=100 )
```



Likelihood for a general MS-model

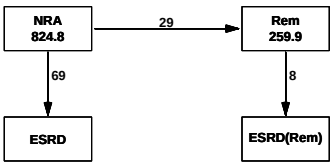
- Product of likelihoods for each transition
  — each one as for a survival model
- **Risk time** is the risk time in the "From" state
- **Events** are transitions to the "To" state
- All other transitions out of "From" are treated as **censorings**
- Possible to fit models separately for each transition



Cox-analysis with remission as time-dependent covariate:

- Ignores λ, the remission rate.
- Assumes μ_{NRA} and μ_{rem} use the same timescale.

Model for all transitions



Cox-model:

- Different timescales for transitions possible
- . . . only one per transition
- No explicit representation of estimated rates.

Poisson-model:

- Timescales can be different
- Multiple timescales can be accomodated simultaneously
- Explicit representation of all transition rates

Calculus of probabilities

$$\begin{aligned}&\mathrm{P}\{\text{Remission } \textbf{before} \text{ time } t\}\\ &= \int_0^t \lambda(u)\exp\left(-\int_0^u \lambda(s) + \mu_{\mathsf{NRA}}\,\mathrm{d}s\right)\,\mathrm{d}u\end{aligned}$$

$$\begin{aligned}&\mathrm{P}\{\text{Being in remission } \textbf{at} \text{ time } t\}\\ &= \int_0^t \lambda(u)\exp\left(-\int_0^u \lambda(s) + \mu_{\mathsf{NRA}}(s)\,\mathrm{d}s\right) \times\\ &\qquad\qquad \exp\left(-\int_u^t \mu_{\mathsf{rem}}(s)\,\mathrm{d}s\right)\,\mathrm{d}u\end{aligned}$$

Note μ_{rem} could also depend on u, time since obtained remission.

Sketch of programming, assuming that λ (`lambda`), μ_{NRA} (`mu.nra`) and μ_{rem} (`mu.rem`) are known for each age (stored in vectors)

```
c.rem      <- cumsum( lambda )
c.mort.nra <- cumsum( mu.nra )
c.mort.rem <- cumsum( mu.rem )
pr1  <- cumsum( lambda * exp( -( c.rem + c.mort.nra ) ) )

intgr(t,s) <- function(t,s){
           lambda[s] * exp( -( c.rem[s] + c.mort.nra[s] ) ) *
                     exp( -( c.mort.rem[t]-c.mort.rem[s] ) ) }
for( t in 1:100 ) p2[t] <- sum( intgr(t,1:t) )
```

If μ_{rem} depends on time of remission, then `c.mort.rem` should have an extra argument.

Calculation of integrals

The possibility of computing the state-occupancy probabilities relies on:

- Availablity of closed-form formulae for the probailities in terms of the transition rates
- Transition rates are assumed to be continuous functions of time
- Transition rates can be calulated at any point of time...
- This will allow simple calulation of the integrals from the closed-form expressions.

Semi-Markov models

- **if** we only have one time scale, which is common for **all** transitions
- — in practical terms: transition intensities only depend on state and the current time.
- then we can construct transition matrices for each tiny time interval

$$P_{ij}(t, t+h) = \mathrm{P}\left\{\text{state } j \text{ at } t+h \mid \text{state } i \text{ at } t\right\}$$

- Simple matrix multiplication then gives the matrix of transition probabilities between states between any two timepoints.

Prediction in multistate models: simLexis

Bendix Carstensen

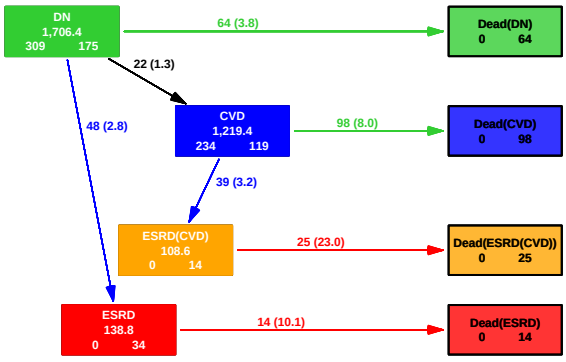
Multistate models

IARC, Lyon,

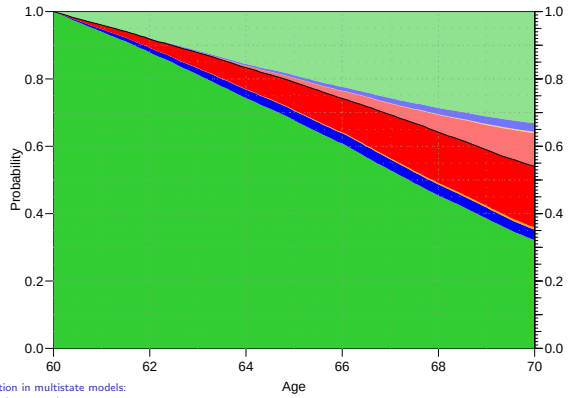
June 2018

http://BendixCarstensen.com/SPE simLexis

A more complicated multistate model



A more complicated multistate model

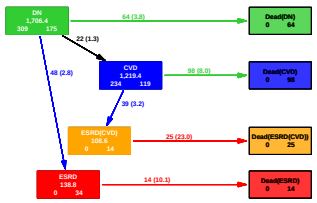


State probabilities

How do we get from rates to probabilities:

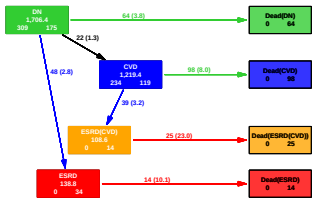
- 1: Analytical calculations:
  - immensely complicated formulae
  - computationally fast (once implemented)
  - difficult to generalize
- 2: Simulation of persons' histories
  - conceptually simple
  - computationally not quite simple
  - easy to generalize
  - hard to get confidence intervals (bootstrap)

Simulation in a multistate model



- Simulate a "survival time" for each transition **out** of a state.
- The smallest of these is the transition time.
- Choose the corresponding transition type as transition.

Transition object are `glms`




```
Tr <- list( "DN" = list( "Dead(DN)"  = E1d,
                         "CVD"       = E1c,
                         "ESRD"      = E1e ),
            "CVD" = list( "Dead(CVD)" = E1d,
                          "ESRD(CVD)" = E1e ),
           "ESRD" = list( "Dead(ESRD)"= E1n ),
      "ESRD(CVD)" = list( "Dead(ESRD(CVD))"= E1n ) )
```


`simLexis`

Input required:

- A `Lexis` object representing the initial state of the persons to be simulated.
  (`lex.dur` and `lex.Xst` will be ignored.)
- A transition object with the estimated Poisson models collected in a list of lists.

Output produced:

- A `Lexis` object with simulated event histories for may persons
- Use `nState` to count how many persons in each state at different times

Using `simLexis`

Put one record a new `Lexis` object (`init`, say). representing a person with the desired covariates.

Must have same structure as the one used for estimation:

```
init <- subset( S5, FALSE,
                    select=c(timeScales(S5),"lex.Cst",
                         "dm.type","sex","hba1c",
                         "sys.bt","tchol","alb",
                         "smoke","bmi","gfr","hmgb",
                         "ins.kg") )
init[1,"sex"] <- "M"
init[1,"age"] <- 60
...

sim1 <- simLexis( Tr1, init,
                  time.pts=seq(0,25,0.2),
                  N=500 ) )
```

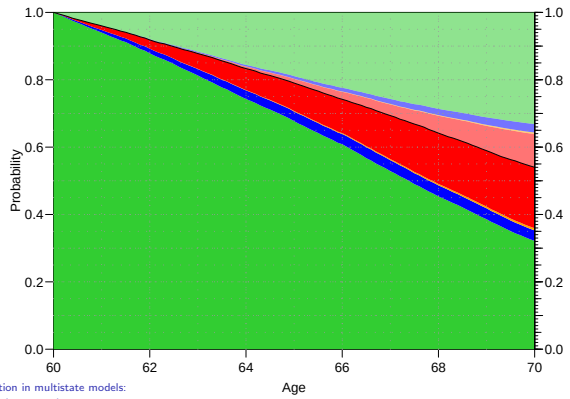
Output from simLexis

```
> summary( sim1 )

Transitions:
     To
From          DN CVD ES(CVD)   ES Dead(CVD) Dead(ES(CVD)) Dead(ES) Dead(DN)
  DN         212  81       0  145         0             0        0       62
  CVD          0  50       7    0        24             0        0        0
  ESRD(CVD)    0   0       3    0         0             4        0        0
  ESRD         0   0       0   70         0             0       75        0
  Sum        212 131      10  215        24             4       75       62

Transitions:
     To
From          Records:  Events: Risk time:  Persons:
  DN               500       288    9245.95       500
  CVD               81        31     667.90        81
  ESRD(CVD)          7         4      45.72         7
  ESRD             145        75     891.11       145
  Sum              733       398   10850.67       500
```


Using a simulated Lexis object — pState

```
nw1 <- pState( nState( sim1,
                       at = seq(0,15,0.1),
                       from = 60,
                       time.scale = "age" ),
               perm = c(1:4,7:5,8) ) )
head( pState )
when         DN     CVD ES(CVD)     ES Dead(ES) Dead(ES(CVD)) Dead(CVD) Dead(DN)
  60   1.0000 1.0000  1.0000 1.0000   1.0000        1.0000    1.0000        1
  60.1 0.9983 0.9986  0.9986 0.9997   0.9997        0.9997    0.9997        1
  60.2 0.9954 0.9964  0.9964 0.9990   0.9990        0.9990    0.9990        1
  60.3 0.9933 0.9947  0.9947 0.9981   0.9981        0.9981    0.9982        1
  60.4 0.9912 0.9929  0.9929 0.9973   0.9973        0.9973    0.9974        1
  60.5 0.9894 0.9913  0.9913 0.9964   0.9964        0.9964    0.9965        1

plot( pState )
```


Simulated probabilities



How many persons should you simulate?

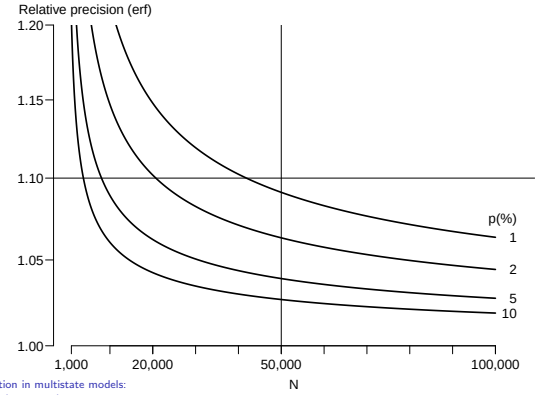
- All probabilities have the same denominator — the initial number of persons in the simulation, N, say.
- Thus, any probability will be of the form $p = x/N$
- For small probabilities we have that:

$$\text{s.e.}\big(\log(\hat{p})\big) = (1-p)/\sqrt{Np(1-p)}$$

- So c.i. of the form $p \overset{\times}{\div} \text{erf}$ where:

$$\text{erf} = \exp\big(1.96 \times (1-p)/\sqrt{Np(1-p)}\big)$$

Precision of simulated probabilities



Multistate model overview

- Clarify what the relevant states are
- Allows proper estimation of transition rates
- — and relationships between them
- Separate model for each transition (arrow)
- The usual survival methodology to compute probabilities breaks down
- Simulation allows estimation of cumulative probabilities:
  - Estimate transition rates (as usual)
  - Simulate probabilities (**not** as usual)

Your turn: "Renal complications"