

Multistate models:

Occurrence rates, cumulative risks, competing risks,
state probabilities with multiple states and time scales in
Register Research with R and `Epi::Lexis`

Bendix Carstensen Steno Diabetes Center Copenhagen
Herlev, Denmark
<http://BendixCarstensen.com>
Dorte Vistisen Steno Diabetes Center Copenhagen

SDCG, Nuuk, 1-3 March 2022

R installed

We will be using the Epi package

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

and sometimes also functions from the `tidyverse` package — but beware some conflicts exist

R is a calculator—use the console

```
> 3 + 2
```

```
[1] 5
```

```
> x <- 3 * 7
```

```
> x
```

```
[1] 21
```

```
> 5 / 12 -> z
```

```
> z
```

```
[1] 0.4166667
```

R operates on **objects** — vectors, data frames, models etc.

Everything is a function

Fit a regression model

```
> m1 <- lm(y ~ x, data = dd)
```

uses the variables `y` and `x` from the data frame `dd` and saves the result in `m1`, `m1` is an object of **class** `lm`

But it prints nothing.

You get the result printed by typing the name of the object:

```
> m1
```

Call:

```
lm(formula = y ~ x, data = dd)
```

Coefficients:

```
(Intercept)          x  
-2.6360         0.3888
```

A summary method exists for some objects

```
> summary(m1)
```

```
Call:
```

```
lm(formula = y ~ x, data = dd)
```

```
Residuals:
```

Min	1Q	Median	3Q	Max
-0.83379	-0.31168	-0.06909	0.30041	0.87754

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-2.63598	0.21710	-12.14	3.05e-16
x	0.38882	0.02847	13.66	< 2e-16

```
Residual standard error: 0.4394 on 48 degrees of freedom
```

```
Multiple R-squared: 0.7954, Adjusted R-squared: 0.7911
```

```
F-statistic: 186.6 on 1 and 48 DF, p-value: < 2.2e-16
```

A str method exists for all objects

```
> str(m1)
```

```
List of 12
```

```
$ coefficients : Named num [1:2] -2.636 0.389
..- attr(*, "names")= chr [1:2] "(Intercept)" "x"
$ residuals    : Named num [1:50] 0.597 0.878 0.499 0.112 0.142 ...
..- attr(*, "names")= chr [1:50] "1" "2" "3" "4" ...
$ effects      : Named num [1:50] -1.4519 6.0016 0.49 -0.0472 0.1142 ...
..- attr(*, "names")= chr [1:50] "(Intercept)" "x" "" "" ...
$ rank         : int 2
$ fitted.values: Named num [1:50] -0.5893 0.0796 0.5955 -0.5328 0.4572 ...
..- attr(*, "names")= chr [1:50] "1" "2" "3" "4" ...
$ assign       : int [1:2] 0 1
$ qr          :List of 5
..$ qr        : num [1:50, 1:2] -7.071 0.141 0.141 0.141 0.141 ...
.. ..- attr(*, "dimnames")=List of 2
.. .. ..$ : chr [1:50] "1" "2" "3" "4" ...
.. .. ..$ : chr [1:2] "(Intercept)" "x"
.. ..- attr(*, "assign")= int [1:2] 0 1
..$ qraux: num [1:2] 1.14 1
..$ pivot: int [1:2] 1 2
$ tol    : num 1e-07
```

When something goes wrong

... you want to see how your data looks — is it as you expected?

Use these functions to see what you have:

```
> summary(dd)
> str(dd)
> dim(dd)
> length(dd)
> names(dd)
> class(dd)
> mode(dd)
```

You can create your own functions

Calculate logit (log-odds) from probability:

```
> p2l <- function(pr)
+   {
+     odds <- log(pr / (1 - pr))
+     odds
+   }
```

function name is `p2l` (probability 2 logit)

argument name is `pr`

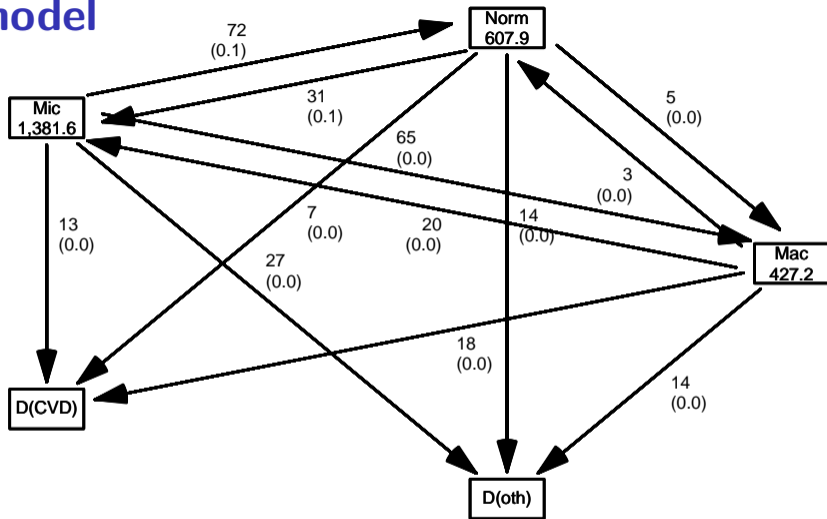
function value is the value of the last expression in the **function body** (which is what is between the `{}`s):

```
> p2l(p = 0.37)
[1] -0.5322168
```

argument value is `0.37`

Your turn — practicals chapter 1!

A multistate model



A multistate model

- ▶ Not really a model
- ▶ What is the data:
 - ▶ Sequence of transitions: (when, from, to)
... same as:
 - ▶ sequence of: (state time, next state)
- ▶ What are the target parameters:
 - ▶ Rates (the arrows)
 - ▶ State probabilities (of being in a state at a given time)
 - ▶ Survival probability
 - ▶ Sojourn times (how long time do you spend in a state)
 - ▶ Probability of ever visiting a state

What is a statistical model

- ▶ Specification of a statistical machinery that could have generated data
- ▶ ... so when we have a statistical model we can simulate a data set
- ▶ The basis for the likelihood of data is the statistical model
⇒ Estimation of parameters in the model
- ▶ Parameter estimates needed for prediction of rates (hazards)

Survival data

Persons enter the study at some date.

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

Observation:

Actual time span to death (“event”)

or

Some time alive (“at least this long”)

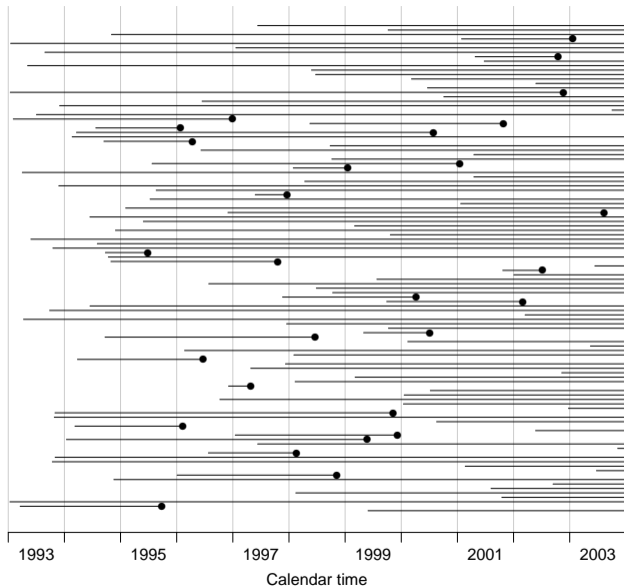
Examples of time-to-event measurements

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

Each line a person

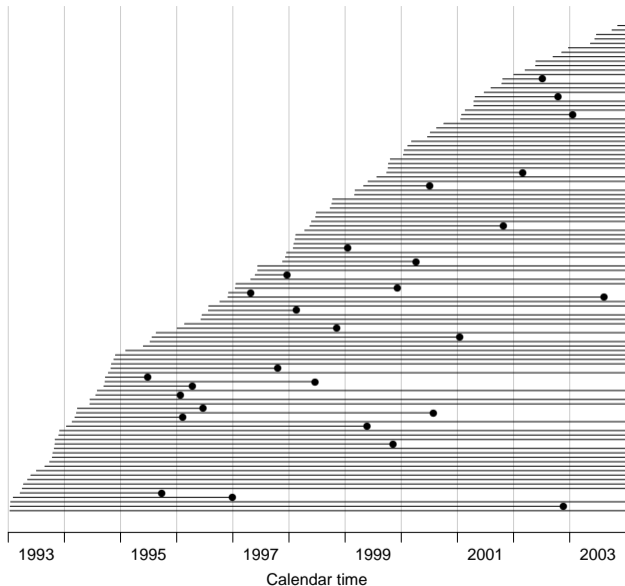
Each blob a death

Study ended at 31
Dec. 2003

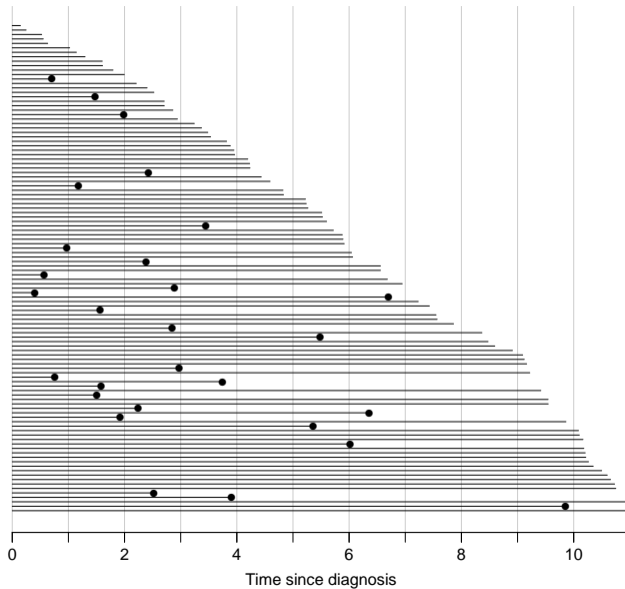


Ordered by date of entry

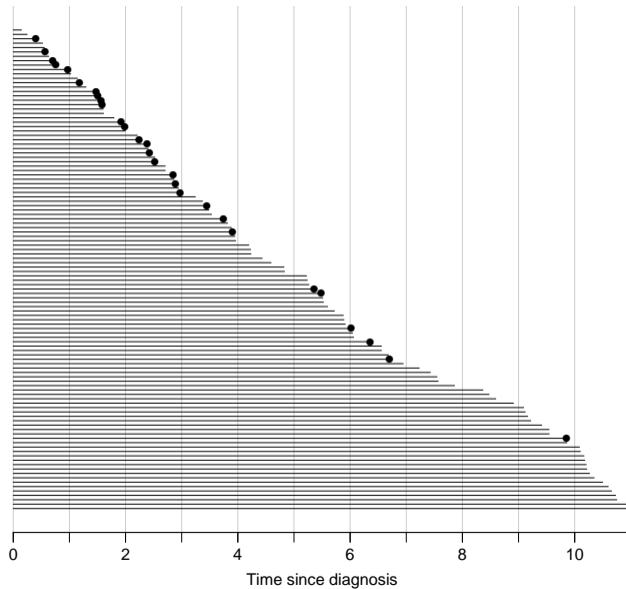
Most likely the order in your database.



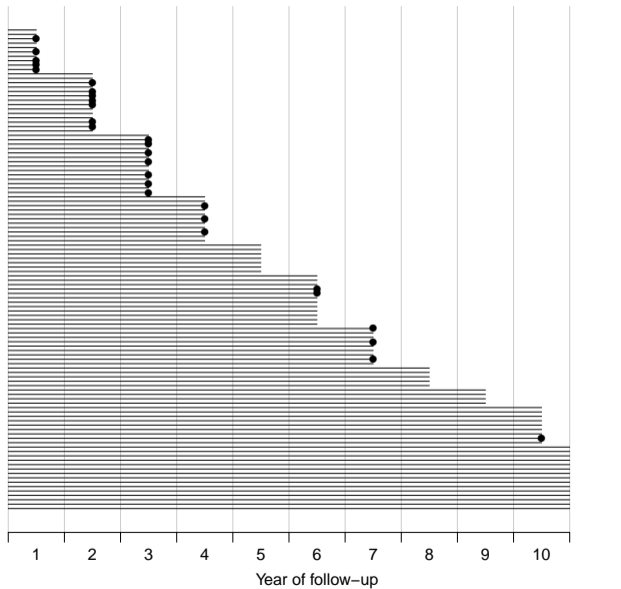
Timescale changed
to
“Time since
diagnosis”.



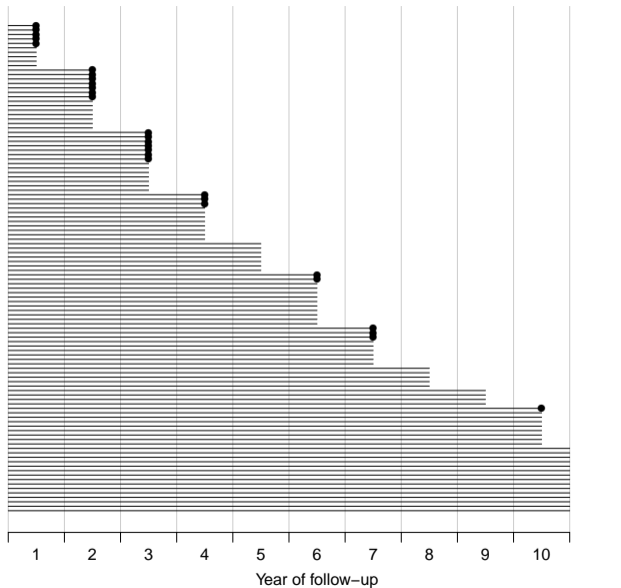
Patients ordered
by survival time.



Survival times
grouped into
bands of survival.



Patients ordered
by survival status
within each band.



Survival after Cervix cancer

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

Life-table estimator of death probability: $D/(N - L/2)$

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

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

Survival after Cervix cancer

Year	Stage I			Stage II		
	<i>N</i>	<i>D</i>	<i>L</i>	<i>N</i>	<i>D</i>	<i>L</i>
1	110	5	5	234	24	3
2	100	7	7	207	27	11
3	86	7	7	169	31	9

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

Estimated risk in year 2 for Stage I women is $7/96.5 = 0.0725$

Estimated risk in year 3 for Stage I women is $7/82.5 = 0.0848$

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

Estimated 2 year survival is $0.9535 \times (1 - 0.0725) = 0.8843$

Estimated 3 year survival is $0.8843 \times (1 - 0.0848) = 0.8093$

This is the life-table estimator.

- ▶ No need to use 1 year intervals
- ▶ Very small intervals will leave at most 1 censoring or 1 death in each
- ▶ Interval with 1 death and n_t persons at risk:
 $P\{\text{Death}\} = 1/n_t$
- ▶ corresponding death probability $(n_t - 1)/n_t$
- ▶ if you multiply these over times with event:

$$S(t) = \prod_{t \text{ with event}} (n_t - 1)/n_t$$

... you have the **Kaplan-Meier estimator**

- ▶ looks complicated but just a question of book keeping

Prerequisites

```
> library(Epi)
> library(popEpi)
> # popEpi::splitMulti returns a data.frame rather than a data.table
> options("popEpi.datatable" = FALSE)
```


The lung data set

```
> library(survival)
> data(lung)
> lung$sex <- factor(lung$sex,
+                   levels = 1:2,
+                   labels = c("M", "W"))
> lung$time <- lung$time / (365.25/12)
> head(lung)
```

	inst	time	status	age	sex	ph.ecog	ph.karno	pat.karno	meal.cal	wt.loss
1	3	10.053388	2	74	M	1	90	100	1175	NA
2	3	14.948665	2	68	M	0	90	90	1225	15
3	3	33.182752	1	56	M	0	90	90	NA	15
4	5	6.899384	2	57	M	1	90	60	1150	11
5	1	29.010267	2	60	M	0	100	90	NA	0
6	12	33.577002	1	74	M	1	50	80	513	0

Survival function

- ▶ Use `survfit` to construct the Kaplan-Meier estimator of overall survival:

```
> ?Surv  
> ?survfit
```

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

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

```
      n  events  median 0.95LCL 0.95UCL  
228.00 165.00  10.18    9.36   11.93
```

```
> # summary(km) # very long output
```

We can plot the survival curve—this is the default plot for a `survfit` object:

```
> plot(km)
```

What is the median survival? What does it mean? Explore if survival patterns between men and women are different:

```
> kms <- survfit(Surv(time, status == 2) ~ sex, data = lung)
> kms
```

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

	n	events	median	0.95LCL	0.95UCL
sex=M	138	112	8.87	6.97	10.2
sex=W	90	53	14.00	11.43	18.1

We see that men have worse survival than women, but they are also a bit older (`age` is age at diagnosis of lung cancer):

```
> with(lung, tapply(age, sex, mean))
```

```
      M      W  
63.34058 61.07778
```

Formally there is a significant difference in survival between men and women

```
> survdiff(Surv(time, status==2) ~ sex, data = lung)
```

Call:

```
survdiff(formula = Surv(time, status == 2) ~ sex, data = lung)
```

	N	Observed	Expected	(O-E) ² /E	(O-E) ² /V
sex=M	138	112	91.6	4.55	10.3
sex=W	90	53	73.4	5.68	10.3

```
Chisq= 10.3 on 1 degrees of freedom, p= 0.001
```

Rates and rate-ratios

- ▶ Occurrence **rate**:

$$\lambda(t) = \lim_{h \rightarrow 0} P \{ \text{event in } (t, t + h] \mid \text{alive at } t \} / h$$

—measured in probability per time: time^{-1}

- ▶ observation in a survival study: (exit status, time alive)
- ▶ empirical rate $(d, y) = (\text{deaths}, \text{time})$
- ▶ the Cox model is a model for rates as function of time (t) and covariates (x_1, x_2) :

$$\lambda(t, x) = \lambda_0(t) \exp(\beta_1 x_1 + \beta_2 x_2)$$

—mortality depends on the person's sex and age, say.

- ▶ Data looks like data for a K-M analysis **plus** covariate values

Rates and rate-ratios: Simple Cox model

Now explore how sex and age (at diagnosis) influence the mortality—note that in a Cox-model we are addressing the mortality rate and not the survival:

```
> c0 <- coxph(Surv(time, status == 2) ~ sex, data = lung)
> c1 <- coxph(Surv(time, status == 2) ~ sex + age, data = lung)
> summary(c1)
> ci.exp(c0)
> ci.exp(c1)
```

What variables from `lung` are we using?

```
> c0 <- coxph(Surv(time, status == 2) ~ sex, data = lung)
> c1 <- coxph(Surv(time, status == 2) ~ sex + age, data = lung)
> summary(c1)
```

Call:

```
coxph(formula = Surv(time, status == 2) ~ sex + age, data = lung)
```

n= 228, number of events= 165

	coef	exp(coef)	se(coef)	z	Pr(> z)
sexW	-0.513219	0.598566	0.167458	-3.065	0.00218
age	0.017045	1.017191	0.009223	1.848	0.06459

	exp(coef)	exp(-coef)	lower .95	upper .95
sexW	0.5986	1.6707	0.4311	0.8311
age	1.0172	0.9831	0.9990	1.0357

Concordance= 0.603 (se = 0.025)

Likelihood ratio test= 14.12 on 2 df, p=9e-04

Wald test = 13.47 on 2 df, p=0.001

Score (logrank) test = 13.72 on 2 df, p=0.001

```

> ci.exp(c0)

      exp(Est.)      2.5%      97.5%
sexW 0.5880028 0.4237178 0.8159848

> ci.exp(c1)

      exp(Est.)      2.5%      97.5%
sexW 0.598566 0.4310936 0.8310985
age  1.017191 0.9989686 1.0357467

```

What do these estimates mean?

$$\lambda(t, x) = \lambda_0(t) \exp(\beta_1 x_1 + \beta_2 x_2)$$

Where is β_1 ? Where is β_2 ? Where is $\lambda_0(t)$?

What is the mortality RR for a 10 year age difference?

If mortality is assumed constant ($\lambda(t) = \lambda$), then the likelihood for the Cox-model is equivalent to a Poisson likelihood, which can be fitted using the `poisreg` family from the `Epi` package:

```
> ?poisreg
```

```
> p1 <- glm(cbind(status == 2, time) ~ sex + age,  
+          family = poisreg,  
+          data = lung)  
> ci.exp(p1) # Poisson
```

	exp(Est.)	2.5%	97.5%
(Intercept)	0.03255152	0.01029228	0.1029511
sexW	0.61820515	0.44555636	0.8577537
age	1.01574132	0.99777446	1.0340317

```
> ci.exp(c1) # Cox
```

	exp(Est.)	2.5%	97.5%
sexW	0.598566	0.4310936	0.8310985
age	1.017191	0.9989686	1.0357467

Sex and age effects are quite close between the Poisson and the Cox models.

Poisson model has an intercept term, the estimate of the (assumed) constant underlying mortality.

The risk time part of the response (second argument in the `cbind`) was entered in units of months (remember we rescaled in the beginning?), the `(Intercept)` (taken from the `ci.exp`) is a rate per 1 person-month.

What age and sex does the `(Intercept)` refer to?

```
> ci.exp(p1) # Poisson
```

	exp(Est.)	2.5%	97.5%
<code>(Intercept)</code>	0.03255152	0.01029228	0.1029511
<code>sexW</code>	0.61820515	0.44555636	0.8577537
<code>age</code>	1.01574132	0.99777446	1.0340317

poisreg and poisson

poisreg: `cbind(d,y) ~ ...`

```
> p1 <- glm(cbind(status == 2, time) ~ sex + age,  
+          family = poisreg,  
+          data = lung)
```

poisson: `d ~ ... + offset(log(y))`

```
> px <- glm(status == 2 ~ sex + age + offset(log(time)),  
+          family = poisson,  
+          data = lung)  
> ## or:  
> px <- glm(status == 2 ~ sex + age,  
+          offset = log(time),  
+          family = poisson,  
+          data = lung)
```

Representation of follow-up: Lexis object

```
> L1 <- Lexis(exit = list(tfl = time),
+             exit.status = factor(status,
+                                   levels = 1:2,
+                                   labels = c("Alive", "Dead")),
+             data = lung)
```

NOTE: entry.status has been set to "Alive" for all.

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

```
> head(L1)
```

	tfl	lex.dur	lex.Cst	lex.Xst	lex.id	inst	time	status	age	sex	ph.ecog	ph.ka
1	0	10.053388	Alive	Dead	1	3	10.053388	2	74	M	1	
2	0	14.948665	Alive	Dead	2	3	14.948665	2	68	M	0	
3	0	33.182752	Alive	Alive	3	3	33.182752	1	56	M	0	
4	0	6.899384	Alive	Dead	4	5	6.899384	2	57	M	1	
5	0	29.010267	Alive	Dead	5	1	29.010267	2	60	M	0	
6	0	33.577002	Alive	Alive	6	12	33.577002	1	74	M	1	

	meal.cal	wt.loss
1	1175	NA
2	1225	15
3	NA	15

New variables in a Lexis object

`tfl`: time from lung cancer **at the time of entry**, therefore it is 0 for all persons; the entry time is 0 from the entry time. But it defines a **timescale**.

`lex.dur`: the **length** of time a person is in state `lex.Cst`, here measured in months, because `time` is.

`lex.Cst`: Current `s`tate, the state in which the `lex.dur` time is spent.

`lex.Xst`: eXit `s`tate, the state to which the person moves after the `lex.dur` time in `lex.Cst`.

`lex.id`: an id of each record in the source dataset. Can be explicitly set by `id=`.

Lexis object: Overview of follow-up

Overkill?

The point is that the machinery generalizes to multistate data.

```
> summary(L1)
```

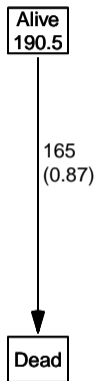
Transitions:

To

From	Alive	Dead	Records:	Events:	Risk time:	Persons:
Alive	63	165	228	165	2286.42	228

What is the average follow-up time for persons?

```
> boxes(L1, boxpos = TRUE, scale.Y = 12, digits.R = 2)
```



Explain the numbers in the graph.

Cox model using the Lexis-specific variables:

```
> cl <- coxph(Surv(tfl,  
+             tfl + lex.dur,  
+             lex.Xst == "Dead") ~ sex + age,  
+             data = L1)
```

Surv(from-time, to-time, event indicator)

Using the Lexis features:

```
> cL <- coxph.Lexis(L1, tfl ~ sex + age)
```

```
survival::coxph analysis of Lexis object L1:  
Rates for the transition Alive->Dead  
Baseline timescale: tfl
```

```
> round(cbind(ci.exp(cL),  
+            ci.exp(cl)), 3)
```

	exp(Est.)	2.5%	97.5%	exp(Est.)	2.5%	97.5%
sexW	0.599	0.431	0.831	0.599	0.431	0.831
age	1.017	0.999	1.036	1.017	0.999	1.036

The crude Poisson model:

```
> pc <- glm(cbind(lex.Xst == "Dead", lex.dur) ~ sex + age,  
+          family = poisreg,  
+          data = L1)
```

or even simpler, by using the **Lexis** features:

```
> pL <- glm.Lexis(L1, ~ sex + age)
```

```
stats::glm Poisson analysis of Lexis object L1 with log link:  
Rates for the transition: Alive->Dead
```

```
> round(cbind(ci.exp(pL),  
+            ci.exp(pc)), 3)
```

	exp(Est.)	2.5%	97.5%	exp(Est.)	2.5%	97.5%
(Intercept)	0.033	0.010	0.103	0.033	0.010	0.103
sexW	0.618	0.446	0.858	0.618	0.446	0.858
age	1.016	0.998	1.034	1.016	0.998	1.034

Poisson and Cox model

The crude Poisson model is a Cox-model with the (quite brutal) assumption that baseline rate is constant over time.

But results are similar:

```
> round(cbind(ci.exp(cL),  
+            ci.exp(pL)[-1,]), 3)
```

	exp(Est.)	2.5%	97.5%	exp(Est.)	2.5%	97.5%
sexW	0.599	0.431	0.831	0.618	0.446	0.858
age	1.017	0.999	1.036	1.016	0.998	1.034

Likelihood and records

Suppose a person is alive from t_e (entry) to t_x (exit) and that the person's status at t_x is d , where $d = 0$ means alive and $d = 1$ means dead. If we choose, say, two time points, t_1, t_2 between t_e and t_x , standard use of conditional probability (formally, repeated use of Bayes' formula) gives

$$\begin{aligned} P \{d \text{ at } t_x \mid \text{entry at } t_e\} &= P \{\text{survive } (t_e, t_1] \mid \text{alive at } t_e\} \times \\ &\quad P \{\text{survive } (t_1, t_2] \mid \text{alive at } t_1\} \times \\ &\quad P \{\text{survive } (t_2, t_3] \mid \text{alive at } t_2\} \times \\ &\quad P \{d \text{ at } t_x \mid \text{alive at } t_3\} \end{aligned}$$

Rates and likelihood

For a start assume that the mortality is constant over time

$$\lambda(t) = \lambda:$$

$$P \{ \text{death during } (t, t + h] \} \approx \lambda h \quad (1)$$

$$\Rightarrow P \{ \text{survive } (t, t + h] \} \approx 1 - \lambda h$$

where the approximation gets better the smaller h is.

Dividing follow-up time

- ▶ Survival for a time span: $y = t_x - t_e$
- ▶ Subdivided in N intervals, each of length $h = y/N$
- ▶ Survival probability for the entire span from t_e to t_x is the **product** of probabilities of surviving each of the small intervals, conditional on being alive at the beginning each interval:

$$P \{ \text{survive } t_e \text{ to } t_x \} \approx (1 - \lambda h)^N = \left(1 - \frac{\lambda y}{N} \right)^N$$

Dividing follow-up time

- ▶ From mathematics it is known that $(1 + x/n)^n \rightarrow \exp(x)$ as $n \rightarrow \infty$ (some define $\exp(x)$ this way).
- ▶ So if we divide the time span y in small pieces we will have that $N \rightarrow \infty$:

$$P \{\text{survive } t_e \text{ to } t_x\} \approx \left(1 - \frac{\lambda y}{N}\right)^N \rightarrow \exp(-\lambda y), \quad N \rightarrow \infty \quad (2)$$

- ▶ The contribution to the likelihood from a person observed for a time span of length y is $\exp(-\lambda y)$, and the contribution to the log-likelihood is therefore $-\lambda y$.

Dividing follow-up time

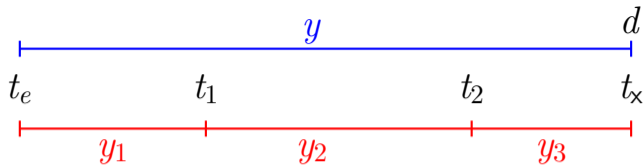
- ▶ A person dying at the end of the last interval, the contribution to the likelihood from the last interval will be
- ▶ the probability surviving till just before the end of the interval,
- ▶ **multiplied** by
- ▶ the probability of dying in the last tiny instant (of length ϵ) of the interval
- ▶ The probability of dying in this tiny instant is $\lambda\epsilon$
- ▶ log-likelihood contribution from this last instant is $\log(\lambda\epsilon) = \log(\lambda) + \log(\epsilon)$.

Total likelihood

The total likelihood for one person is the product of all these terms from the follow-up intervals (i) for the person; and the log-likelihood (ℓ) is therefore:

$$\begin{aligned}\ell(\lambda) &= -\lambda \sum_i y_i + \sum_i d_i \log(\lambda) + \sum_i d_i \log(\epsilon) \\ &= \sum_i (d_i \log(\lambda) - \lambda y_i) + \sum_i d_i \log(\epsilon)\end{aligned}$$

The last term does not depend on λ , so can be ignored



Probability

$$P(d \text{ at } t_x | \text{entry } t_e)$$

$$= P(\text{surv } t_e \rightarrow t_1 | \text{entry } t_e)$$

$$\times P(\text{surv } t_1 \rightarrow t_2 | \text{entry } t_1)$$

$$\times P(d \text{ at } t_x | \text{entry } t_2)$$

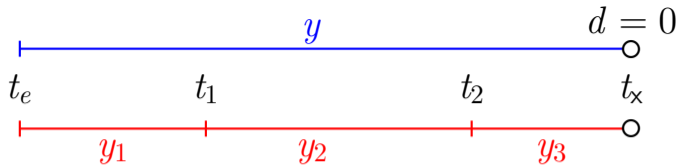
log-Likelihood

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

$$= 0 \log(\lambda) - \lambda y_1$$

$$+ 0 \log(\lambda) - \lambda y_2$$

$$+ d \log(\lambda) - \lambda y_3$$



Probability

$$P(\text{surv } t_e \rightarrow t_x | \text{entry } t_e)$$

$$= P(\text{surv } t_e \rightarrow t_1 | \text{entry } t_e)$$

$$\times P(\text{surv } t_1 \rightarrow t_2 | \text{entry } t_1)$$

$$\times P(\text{surv } t_2 \rightarrow t_x | \text{entry } t_2)$$

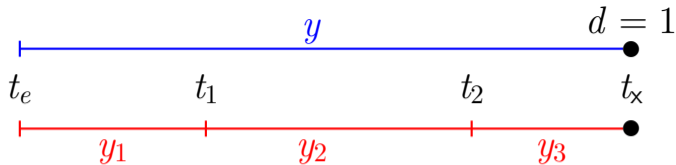
log-Likelihood

$$0 \log(\lambda) - \lambda y$$

$$= 0 \log(\lambda) - \lambda y_1$$

$$+ 0 \log(\lambda) - \lambda y_2$$

$$+ 0 \log(\lambda) - \lambda y_3$$



Probability

$$P(\text{event at } t_x | \text{entry } t_e)$$

$$= P(\text{surv } t_e \rightarrow t_1 | \text{entry } t_e)$$

$$\times P(\text{surv } t_1 \rightarrow t_2 | \text{entry } t_1)$$

$$\times P(\text{event at } t_x | \text{entry } t_2)$$

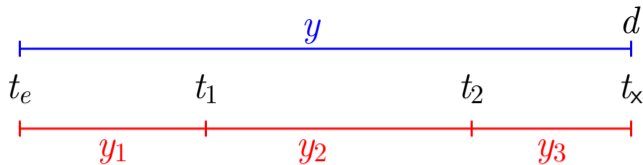
log-Likelihood

$$1 \log(\lambda) - \lambda y$$

$$= 0 \log(\lambda) - \lambda y_1$$

$$+ 0 \log(\lambda) - \lambda y_2$$

$$+ 1 \log(\lambda) - \lambda y_3$$



Probability

$$P(d \text{ at } t_x | \text{entry } t_e)$$

$$= P(\text{surv } t_e \rightarrow t_1 | \text{entry } t_e)$$

$$\times P(\text{surv } t_1 \rightarrow t_2 | \text{entry } t_1)$$

$$\times P(d \text{ at } t_x | \text{entry } t_2)$$

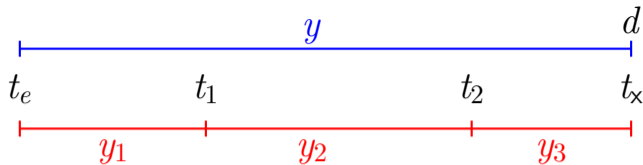
log-Likelihood

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

$$= 0 \log(\lambda) - \lambda y_1$$

$$+ 0 \log(\lambda) - \lambda y_2$$

$$+ d \log(\lambda) - \lambda y_3$$



Probability

$$P(d \text{ at } t_x | \text{entry } t_e)$$

$$= P(\text{surv } t_e \rightarrow t_1 | \text{entry } t_e)$$

$$\times P(\text{surv } t_1 \rightarrow t_2 | \text{entry } t_1)$$

$$\times P(d \text{ at } t_x | \text{entry } t_2)$$

log-Likelihood

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

$$= 0 \log(\lambda_1) - \lambda_1 y_1$$

$$+ 0 \log(\lambda_2) - \lambda_2 y_2$$

$$+ d \log(\lambda_3) - \lambda_3 y_3$$

— allows different rates (λ_i) in each interval

Baseline hazard: splitting time

```
> S1 <- splitMulti(L1, tfl = 0:36)
> summary(L1)
```

Transitions:

	To					
From	Alive	Dead	Records:	Events:	Risk time:	Persons:
Alive	63	165	228	165	2286.42	228

```
> summary(S1)
```

Transitions:

	To					
From	Alive	Dead	Records:	Events:	Risk time:	Persons:
Alive	2234	165	2399	165	2286.42	228

What happened to no. records?

What happened to amount of risk time?

What happened to no. events?

```

> wh <- names(L1)[1:10] # names of variables in some order
> subset(L1, lex.id == 10)[,wh]

      tfl  lex.dur lex.Cst lex.Xst lex.id inst      time status age sex
10     0 5.453799   Alive   Dead    10     7 5.453799      2  61   M

> subset(S1, lex.id == 10)[,wh]

```

```

      tfl  lex.dur lex.Cst lex.Xst lex.id inst      time status age sex
163     0 1.0000000   Alive   Alive    10     7 5.453799      2  61   M
164     1 1.0000000   Alive   Alive    10     7 5.453799      2  61   M
165     2 1.0000000   Alive   Alive    10     7 5.453799      2  61   M
166     3 1.0000000   Alive   Alive    10     7 5.453799      2  61   M
167     4 1.0000000   Alive   Alive    10     7 5.453799      2  61   M
168     5 0.4537988   Alive   Dead    10     7 5.453799      2  61   M

```

In `S1` each record now represents a small interval of follow-up for a person, so each person has many records.

Natural splines for baseline hazard

```
> ps <- glm(cbind(lex.Xst == "Dead", lex.dur)
+           ~ Ns(tfl, knots = seq(0, 36, 12)) + sex + age,
+           family = poisreg,
+           data = S1)
```

or even simpler:

```
> ps <- glm.Lexis(S1, ~ Ns(tfl, knots = seq(0, 36, 12)) + sex + age)
```

```
stats::glm Poisson analysis of Lexis object S1 with log link:
Rates for the transition: Alive->Dead
```

```
> ci.exp(ps)
```

	exp(Est.)	2.5%	97.5%
(Intercept)	0.0189837	0.005700814	0.06321569
Ns(tfl, knots = seq(0, 36, 12))1	2.4038681	0.809442081	7.13896863
Ns(tfl, knots = seq(0, 36, 12))2	4.1500822	0.436273089	39.47798357
Ns(tfl, knots = seq(0, 36, 12))3	0.8398973	0.043928614	16.05849662
sexW	0.5987171	0.431232662	0.83124998
age	1.0165872	0.998377104	1.03512945

Comparing with estimates from the Cox-model and from the model with constant baseline:

```
> round(cbind(ci.exp(c1),  
+             ci.exp(ps, subset = c("sex", "age")),  
+             ci.exp(pc, subset = c("sex", "age"))), 3)
```

	exp(Est.)	2.5%	97.5%	exp(Est.)	2.5%	97.5%	exp(Est.)	2.5%	97.5%
sexW	0.599	0.431	0.831	0.599	0.431	0.831	0.618	0.446	0.858
age	1.017	0.999	1.036	1.017	0.998	1.035	1.016	0.998	1.034

But where **is** the baseline hazard?

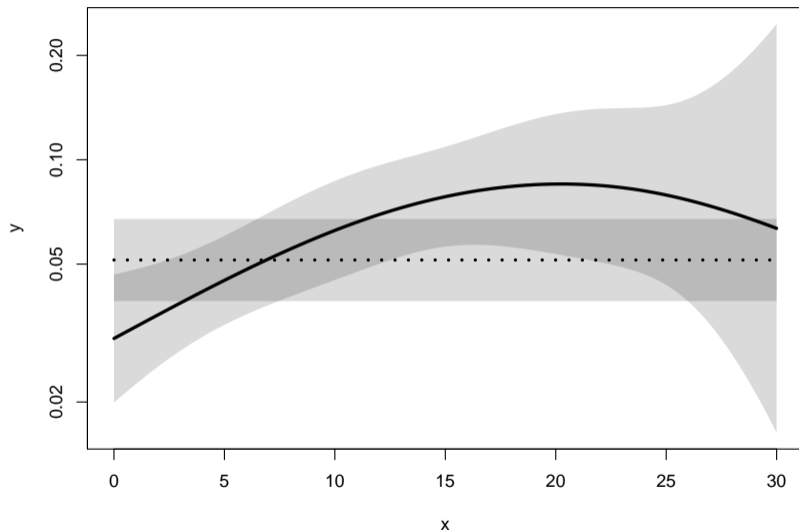
ps is a model for the hazard so we can predict the value of it at defined values for the covariates in the model:

```
> prf <- data.frame(tfl = seq(0, 30, 0.2),  
+                   sex = "W",  
+                   age = 60)
```

We can over-plot with the predicted rates from the model where mortality rates are constant, the only change is the model (**pc** instead of **ps**):

```
> matshade(prf$tfl, ci.pred(ps, prf),  
+          plot = TRUE, log = "y", lwd = 3)  
> matshade(prf$tfl, ci.pred(pc, prf), lty = 3, lwd = 3)
```

Here is the baseline hazard!



Survival function and hazard function

$$S(t) = \exp\left(-\int_0^t \lambda(u) \, du\right)$$

Simple, but the CI for $S(t)$ not so simple. . .

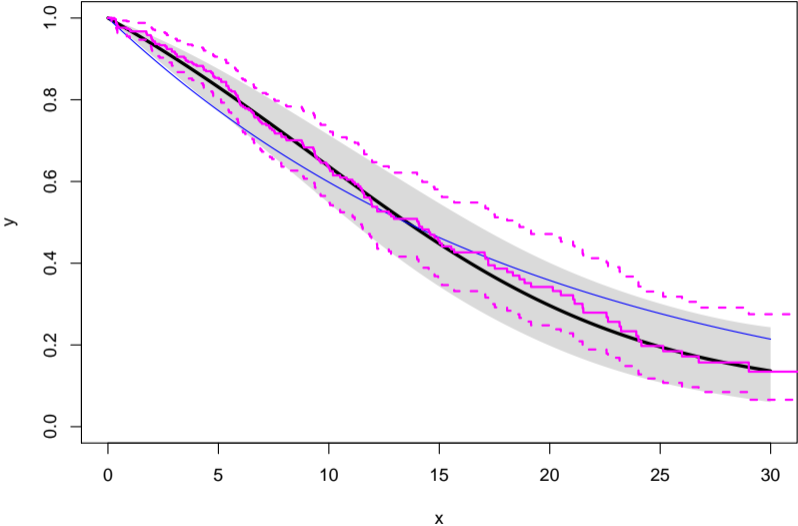
Implemented in the `ci.surv` function

Arguments: 1:model, 2:prediction data frame, 3:equidistance

Prediction data frame must correspond to a sequence of equidistant time points:

```
> matshade(prf$tfl, ci.surv(ps, prf, intl = 0.2),  
+          plot = TRUE, ylim = 0:1, lwd = 3)  
> lines(prf$tfl, ci.surv(pc, prf, intl = 0.2)[,1], col="blue")  
> lines(survfit(c1, newdata = data.frame(sex = "W", age = 60)),  
+       lwd = 2, lty = 1, col="magenta")
```

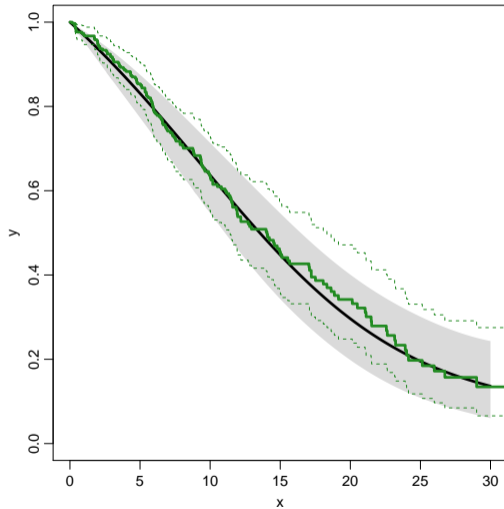
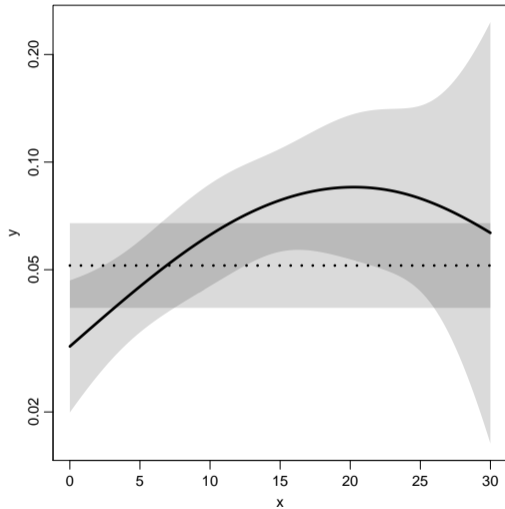
Survival functions



Hazard and survival functions

```
> par(mfrow = c(1,2), mar=c(3,3,1,1), mgp=c(3,1,0)/1.6)
> #
> # hazard scale
> matshade(prf$tfl, ci.pred(ps, prf),
+          plot = TRUE, log = "y", lwd = 3)
> matshade(prf$tfl, ci.pred(pc, prf), lty = 3, lwd = 3)
> #
> # survival
> matshade(prf$tfl, ci.surv(ps, prf, intl = 0.2),
+          plot = TRUE, ylim = 0:1, lwd = 3)
> lines(survfit(c1, newdata = data.frame(sex = "W", age = 60)),
+       col = "forestgreen", lwd = 3, conf.int = FALSE)
> lines(survfit(c1, newdata = data.frame(sex = "W", age = 60)),
+       col = "forestgreen", lwd = 1, lty = 1)
```

Hazard and survival functions

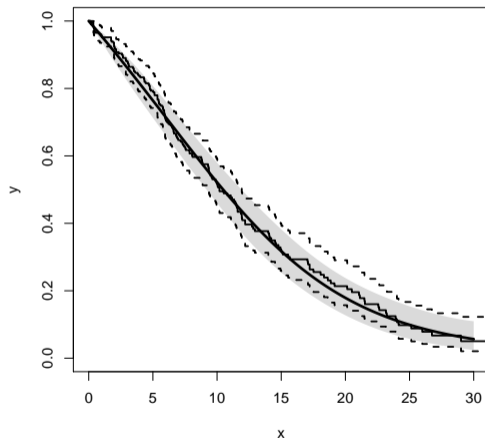
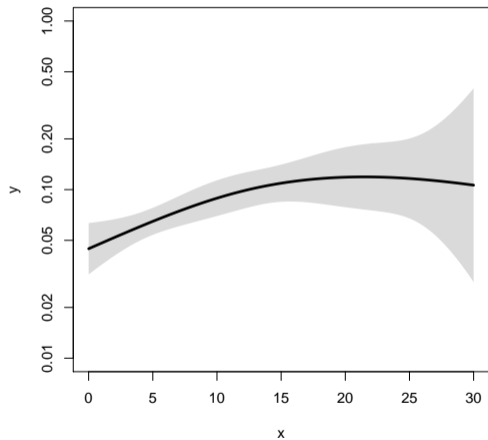


K-M estimator and smooth Poisson model

Kaplan-Meier estimator and compared to survival from corresponding Poisson-model, which is one with time (`tfl`) as the only covariate:

```
> par(mfrow=c(1,2))
> pk <- glm(cbind(lex.Xst == "Dead",
+               lex.dur) ~ Ns(tfl, knots = seq(0, 36, 12)),
+         family = poisreg,
+         data = S1)
> # hazard
> matshade(prf$tfl, ci.pred(pk, prf),
+          plot = TRUE, log = "y", lwd = 3, ylim = c(0.01,1))
> # survival from smooth model
> matshade(prf$tfl, ci.surv(pk, prf, intl = 0.2) ,
+          plot = TRUE, lwd = 3, ylim = 0:1)
> # K-M estimator
> lines(km, lwd = 2)
```


K-M estimator and smooth Poisson model

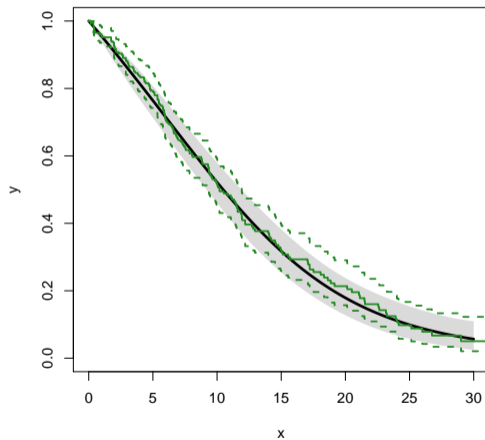
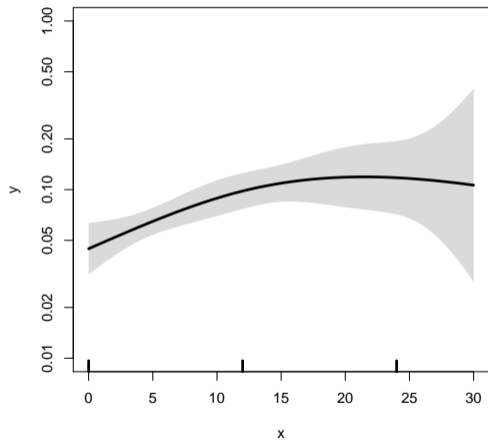


K-M estimator and smooth Poisson model

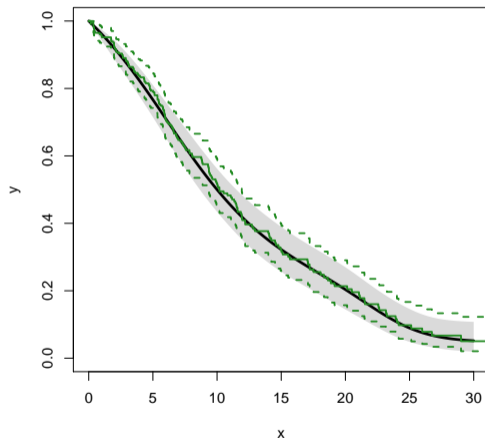
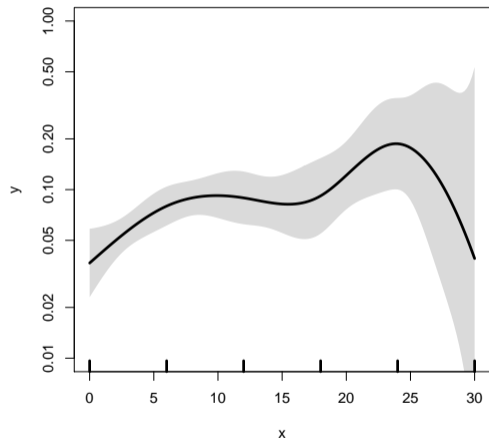
We can explore how the tightness of the knots in the smooth model influence the underlying hazard and the resulting survival function:

```
> zz <- function(dk) # distance between knots
+ {
+   par(mfrow=c(1,2))
+   kn <- seq(0, 36, dk)
+   pk <- glm(cbind(lex.Xst == "Dead",
+                   lex.dur) ~ Ns(tfl, knots = kn),
+             family = poisreg,
+             data = S1)
+   matshade(prf$tfl, ci.pred(pk, prf),
+            plot = TRUE, log = "y", lwd = 3, ylim = c(0.01,1))
+   rug(kn, lwd=3)
+
+   matshade(prf$tfl, ci.surv(pk, prf, intl = 0.2) ,
+            plot = TRUE, lwd = 3, ylim = 0:1)
+   lines(km, lwd = 2, col = "forestgreen")
+ }
> zz(12)
```

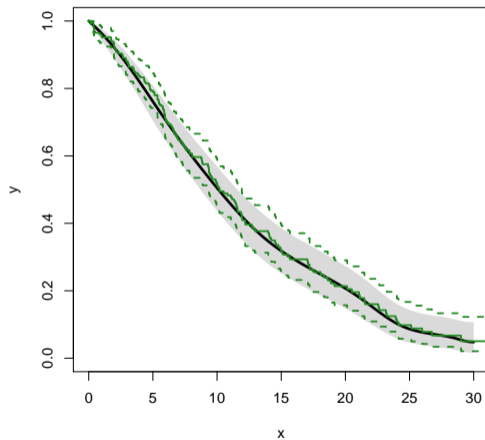
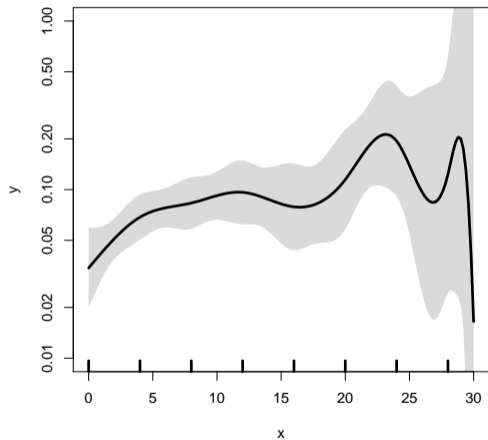
K-M estimator and smooth Poisson model



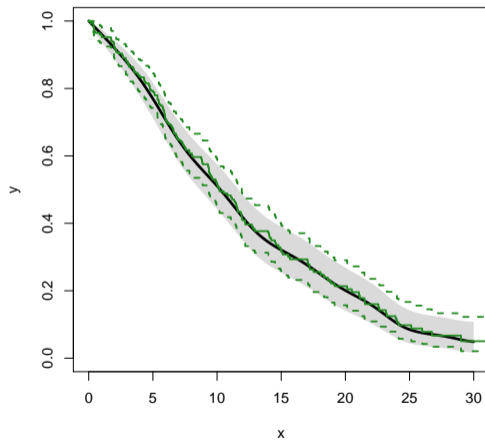
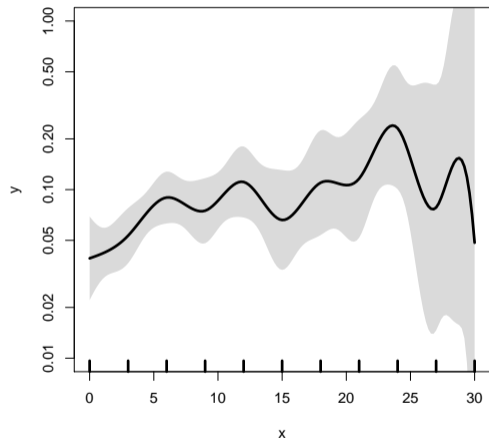
K-M estimator and smooth Poisson model



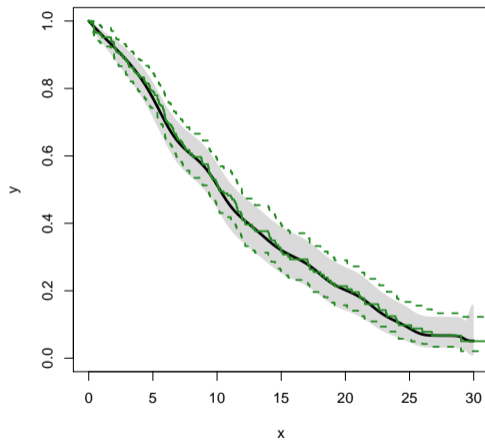
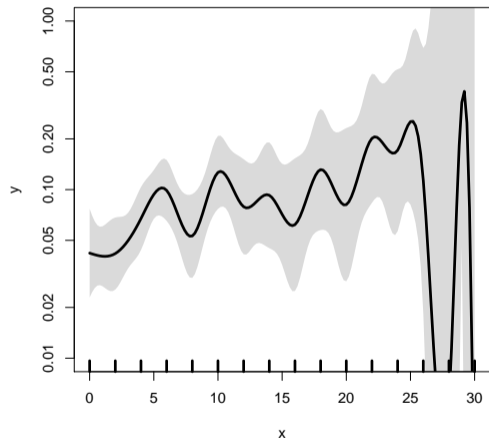
K-M estimator and smooth Poisson model



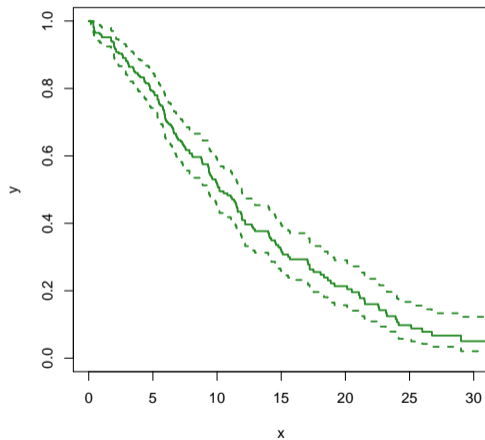
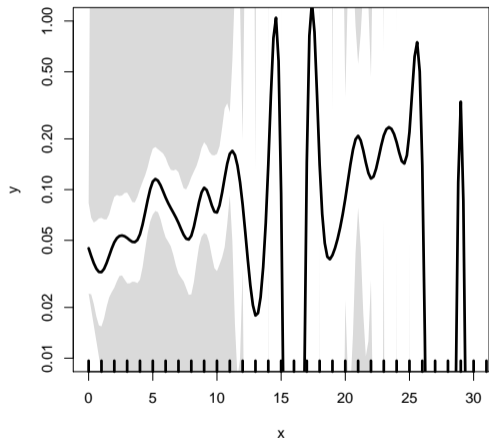
K-M estimator and smooth Poisson model



K-M estimator and smooth Poisson model



K-M estimator and smooth Poisson model



Survival analysis summary

- ▶ 1 to 1 correspondence between hazard function and survival function
- ▶ K-M and Cox use a very detailed baseline hazard (omits it)
- ▶ Smooth parametric hazard function more credible:
 - ▶ Define `Lexis` object
 - ▶ Split along time
 - ▶ Fit Poisson model
 - ▶ Prediction data frame
 - ▶ `ci.pred` to get baseline rates
 - ▶ `ci.surv` to get baseline survival

```
> data(lung)
> lung$sex <- factor(lung$sex, labels=c("M", "F"))
> Lx <- Lexis(exit = list(tfe=time),
+           exit.status = factor(status,labels = c("Alive", "Dead")),
+           data = lung)
> sL <- splitMulti(Lx, tfe=seq(0, 1200, 10))
```

Smooth parametric hazard function

```
> m0 <- glm.Lexis(sL, ~ Ns(tfe, knots = seq(0, 1000, 200)) + sex + age)
```

Prediction data frame

```
> nd <- data.frame(tfe = seq(0, 900, 20) + 10, sex = "M", age = 65)
```

Predictions

```
> rate <- ci.pred(m0, nd) * 365.25 # per year, not per day
> surv <- ci.surv(m0, nd, int = 20)
```

Plot the rates

```
> matshade(nd$tfe, rate, log = "y", plot = TRUE)
```

Plot the survival function

```
> matshade(nd$tfe - 10, surv, ylim = c(0, 1), plot = TRUE)
```

```
> library(survival)
> library(Epi)
> library(popEpi)
> # popEpi::splitMulti returns a data.frame rather than a data.table
> options("popEpi.datatable" = FALSE)
> library(tidyverse)
> clear()
```

```
> data(DMlate)
> # str(DMlate)
> set.seed(1952)
> DMlate <- DMlate[sample(1:nrow(DMlate), 2000),]
> str(DMlate)
```

```
'data.frame': 2000 obs. of 7 variables:
 $ sex : Factor w/ 2 levels "M","F": 2 1 2 1 1 1 1 1 1 1 ...
 $ dobth: num 1964 1944 1957 1952 1952 ...
 $ dodm : num 2003 2006 2008 2007 2003 ...
 $ dodth: num NA NA NA NA NA NA NA NA NA NA ...
 $ dooad: num NA 2006 NA 2007 2006 ...
 $ doins: num NA NA NA 2008 NA ...
 $ dox : num 2010 2010 2010 2010 2010 ...
```

Lexis object from DM to Death

```
> Ldm <- Lexis(entry = list(per = dodm,  
+                          age = dodm - dobth,  
+                          tfd = 0),  
+             exit = list(per = dox),  
+             exit.status = factor(!is.na(dodth),  
+                                 labels = c("DM", "Dead")),  
+             data = DMlate)
```

NOTE: entry.status has been set to "DM" for all.

NOTE: Dropping 1 rows with duration of follow up < tol

```
> summary(Ldm)
```

Transitions:

To

From	DM	Dead	Records:	Events:	Risk time:	Persons:
DM	1521	478	1999	478	10742.34	1999

Cut follow-up at the date of OAD

```
> Cdm <- cutLexis(Ldm,  
+               cut = Ldm$doodad,  
+               timescale = "per",  
+               new.state = "OAD")  
> summary(Cdm)
```

Transitions:

	To						
From	DM	OAD	Dead	Records:	Events:	Risk time:	Persons:
DM	685	634	226	1545	860	5414.3	1545
OAD	0	836	252	1088	252	5328.1	1088
Sum	685	1470	478	2633	1112	10742.3	1999

Cut follow-up at the date of OAD, dooad

```
> subset(Ldm, lex.id %in% c(2:3,20))[,c(1:7,12)]
```

	per	age	tfd	lex.dur	lex.Cst	lex.Xst	lex.id	doad
235221	2005.6	61.517	0	4.3532	DM	DM	2	2005.8
230872	2007.9	51.097	0	2.1109	DM	DM	3	NA
114618	2006.0	73.183	0	3.7919	DM	Dead	20	2007.0

```
> subset(Cdm, lex.id %in% c(2:3,20))[,c(1:7,12)]
```

	per	age	tfd	lex.dur	lex.Cst	lex.Xst	lex.id	doad
2	2005.6	61.517	0.00000	0.13415	DM	OAD	2	2005.8
2001	2005.8	61.651	0.13415	4.21903	OAD	OAD	2	2005.8
3	2007.9	51.097	0.00000	2.11088	DM	DM	3	NA
20	2006.0	73.183	0.00000	1.01848	DM	OAD	20	2007.0
2019	2007.0	74.201	1.01848	2.77344	OAD	Dead	20	2007.0

Restrict to those alive in DM

```
> Adm <- subset(Cdm, lex.Cst == "DM")
> summary(Adm)
```

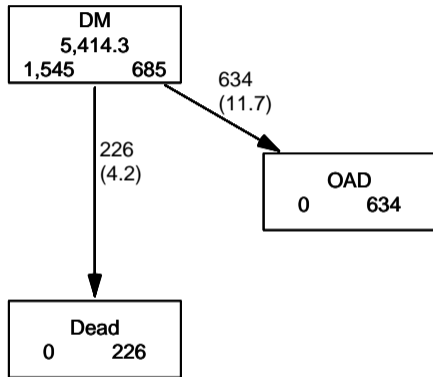
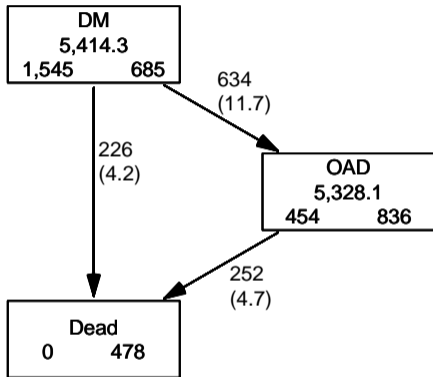
Transitions:

To

From	DM	OAD	Dead	Records:	Events:	Risk time:	Persons:	
	DM	685	634	226	1545	860	5414.3	1545

```
> par(mfrow=c(1,2))
> boxes(Cdm, boxpos = TRUE, scale.R = 100, show.BE = TRUE)
> boxes(Adm, boxpos = TRUE, scale.R = 100, show.BE = TRUE)
```

Transitions in Cdm and Adm



Survival function?

$$S(t) = \exp \left(- \int_0^t \lambda(u) + \mu(u) \, du \right)$$

$$S(t) = \exp \left(- \int_0^t \lambda(u) \, du \right)$$

$$S(t) = \exp \left(- \int_0^t \mu(u) \, du \right)$$

Survival function?

- ▶ Regarding either Dead or OAD as censorings — or neither?
- ▶ **Simple survival**: what is the probability of being in each of the states Alive and Dead
—depends on **one** rate, Alive \rightarrow Dead
- ▶ **Competing risks**: what is the probability of being in each of the states DM, OAD and Dead
—depends on **two** rates, DM \rightarrow OAD and DM \rightarrow Dead

Survival function and Cumulative risk function

`survfit` does the trick; the requirements are:

1. (start, stop, event) arguments to `Surv`
2. the third argument to the `Surv` function is a factor
3. an `id` argument is given, pointing to an id variable that links together records belonging to the same person.
4. the initial state (DM) must be the first level of the factor `lex.Xst`

Survival function and Cumulative risk function

```
> levels(Adm$lex.Xst)
[1] "DM"    "OAD"    "Dead"

> m3 <- survfit(Surv(tfd, tfd + lex.dur, lex.Xst) ~ 1,
+               id = lex.id,
+               data = Adm)
> # names(m3)
> m3$states

[1] "(s0)" "OAD"    "Dead"

> head(cbind(time = m3$time, m3$pstate))

      time
[1,] 0.0027379 0.99871 0.0012945 0.00000000
[2,] 0.0054757 0.99288 0.0064725 0.00064725
[3,] 0.0082136 0.98900 0.0090615 0.00194175
[4,] 0.0109514 0.98770 0.0097087 0.00258900
[5,] 0.0136893 0.98382 0.0135922 0.00258900
[6,] 0.0164271 0.98058 0.0168285 0.00258900
```

Survival function and cumulative risks—formulae

$$S(t) = \exp\left(-\int_0^t \lambda(u) + \mu(u) \, du\right)$$

$$R_{\text{Dead}}(t) = \int_0^t \mu(u) S(u) \, du$$

$$\begin{aligned} R_{\text{OAD}}(t) &= \int_0^t \lambda(u) S(u) \, du \\ &= \int_0^t \lambda(u) \exp\left(-\int_0^u \lambda(s) + \mu(s) \, ds\right) \, du \end{aligned}$$

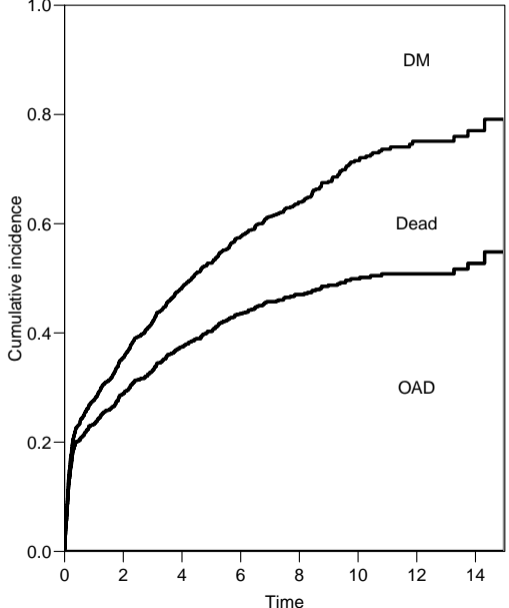
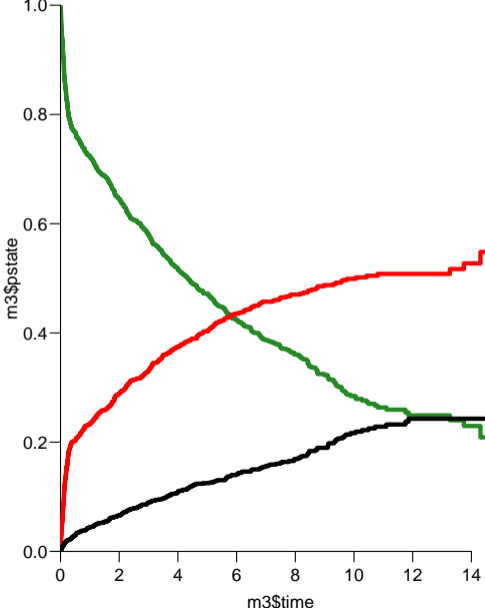
$$S(t) + R_{\text{OAD}}(t) + R_{\text{Dead}}(t) = 1, \quad \forall t$$

Survival function and cumulative risks

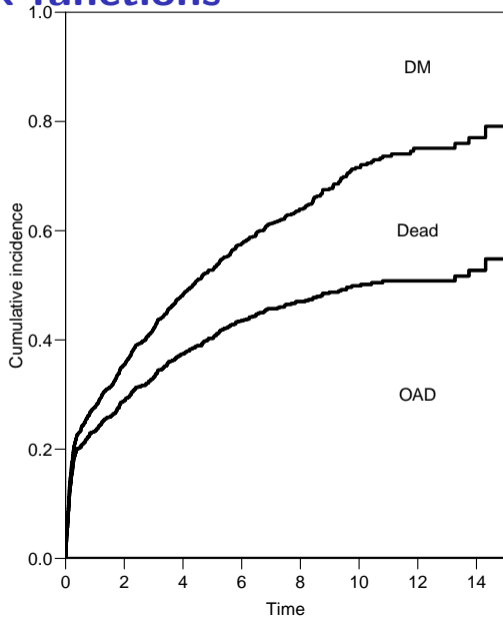
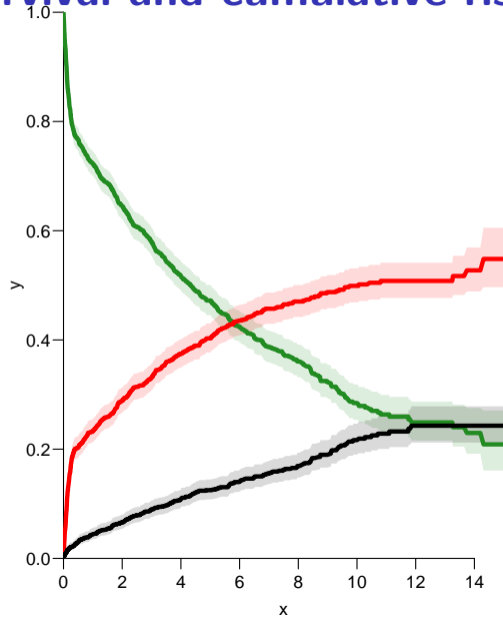
```
> par( mfrow=c(1,2) )
> matplot(m3$time, m3$pstate,
+         type="s", lty=1, lwd=4,
+         col=c("ForestGreen", "red", "black"),
+         xlim=c(0,15), xaxs="i",
+         ylim=c(0,1), yaxs="i" )
> stackedCIF(m3, lwd=3, xlim=c(0,15), xaxs="i", yaxs="i" )
> text(rep(12,3), c(0.9,0.3,0.6), levels(Cdm))
> box(bty="o")

> par( mfrow = c(1,2) )
> matshade(m3$time, cbind(m3$pstate,
+                         m3$lower,
+                         m3$upper)[,c(1,4,7,2,5,8,3,6,9)]),
+         plot = TRUE, lty = 1, lwd = 4,
+         col = c("ForestGreen", "red", "black"),
+         xlim=c(0,15), xaxs="i",
+         ylim = c(0,1), yaxs = "i")
> stackedCIF(m3, lwd=3, xlim=c(0,15), xaxs="i", yaxs="i" )
> text(rep(12,3), c(0.9,0.3,0.6), levels(Cdm))
> box(bty="o")
```

Survival and cumulative risk functions



Survival and cumulative risk functions



Survival function and cumulative risks—don't

$$S(t) = \exp\left(-\int_0^t \lambda(u) + \mu(u) \, du\right)$$

$$R_{\text{Dead}}(t) = \int_0^t \mu(u) S(u) \, du$$

$$R_{\text{OAD}}(t) = \int_0^t \lambda(u) S(u) \, du$$

$$= \int_0^t \lambda(u) \exp\left(-\int_0^u \lambda(s) + \mu(s) \, ds\right) \, du$$

$$\neq \int_0^t \lambda(u) \exp\left(-\int_0^u \lambda(s) \, ds\right) \, du$$

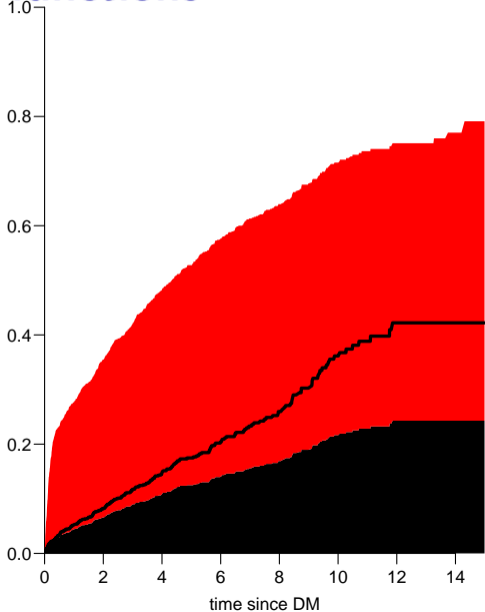
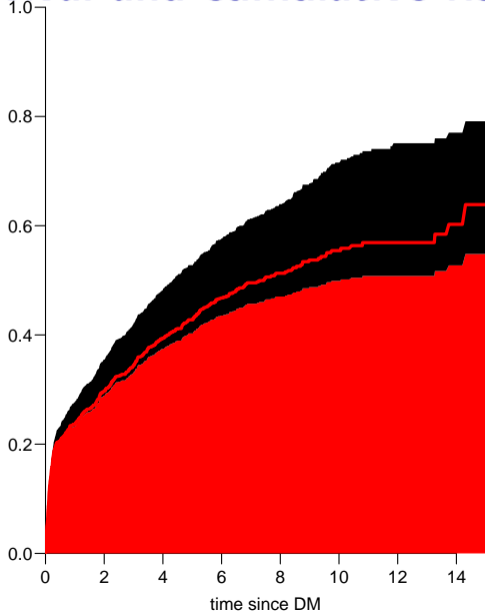
$$= 1 - \exp\left(-\int_0^t \lambda(s) \, ds\right) \text{ — nice formula, but wrong!}$$

Probability of OAD **assuming** Dead does not exist **and** rate of OAD unchanged!

Survival function and cumulative risks—don't

```
> m2 <- survfit(Surv(tfd,
+                 tfd + lex.dur,
+                 lex.Xst == "OAD" ) ~ 1,
+                 data = Adm)
> M2 <- survfit(Surv(tfd,
+                 tfd + lex.dur,
+                 lex.Xst == "Dead") ~ 1,
+                 data = Adm)
> par(mfrow = c(1,2))
> mat2pol(m3$pstate, c(2,3,1), x = m3$time,
+         col = c("red", "black", "transparent"),
+         xlim=c(0,15), xaxs="i",
+         yaxs = "i", xlab = "time since DM", ylab = "" )
> lines(m2$time, 1 - m2$surv, lwd = 3, col = "red" )
> mat2pol(m3$pstate, c(3,2,1), x = m3$time, yaxs = "i",
+         col = c("black","red","transparent"),
+         xlim=c(0,15), xaxs="i",
+         yaxs = "i", xlab = "time since DM", ylab = "" )
> lines(M2$time, 1 - M2$surv, lwd = 3, col = "black" )
```

Survival and cumulative risk functions



Cause-specific rates

- ▶ There is nothing wrong with modeling the cause-specific event-rates, the problem lies in how you transform them into probabilities.
- ▶ The relevant model for a competing risks situation normally consists of separate models for each of the cause-specific rates.
- ▶ ... not for technical or statistical reasons, but for **substantial** reasons:
it is unlikely that rates of different types of event (OAD initiation and death, say) depend on time in the same way.

Cause-specific rates

```
> Sdm <- splitMulti(Adm, tfd = seq(0, 20, 0.1))  
> summary(Adm)
```

Transitions:

	To						
From	DM	OAD	Dead	Records:	Events:	Risk time:	Persons:
DM	685	634	226	1545	860	5414.3	1545

```
> summary(Sdm)
```

Transitions:

	To						
From	DM	OAD	Dead	Records:	Events:	Risk time:	Persons:
DM	54064	634	226	54924	860	5414.3	1545

Cause-specific rates

```
> round(cbind(  
+ with(subset(Sdm, lex.Xst == "OAD" ), quantile(tfd + lex.dur, 0:5/5)),  
+ with(subset(Sdm, lex.Xst == "Dead"), quantile(tfd + lex.dur, 0:5/5))), 2)
```

	[,1]	[,2]
0%	0.00	0.01
20%	0.09	0.51
40%	0.24	1.73
60%	1.27	3.58
80%	3.37	6.20
100%	14.31	11.86

```
> okn <- c(0, 0.5, 3, 10)  
> dkn <- c(0, 2.0, 5, 9)  
> OAD.glm <- glm.Lexis(Sdm, ~ Ns(tfd, knots = okn), to = "OAD" )
```

```
stats::glm Poisson analysis of Lexis object Sdm with log link:  
Rates for the transition: DM->OAD
```

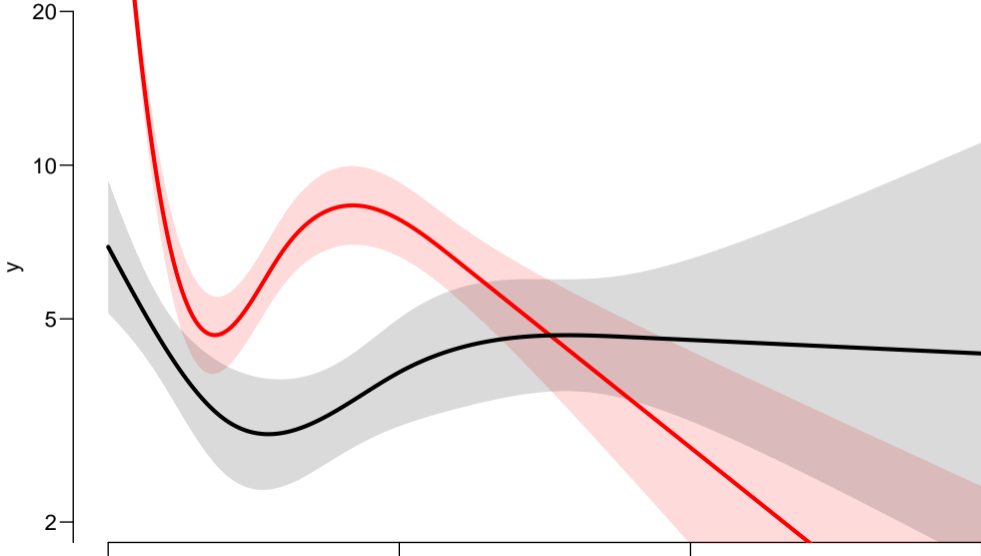
```
> Dead.glm <- glm.Lexis(Sdm, ~ Ns(tfd, knots = dkn), to = "Dead")
```

```
stats::glm Poisson analysis of Lexis object Sdm with log link:  
Rates for the transition: DM->Dead
```

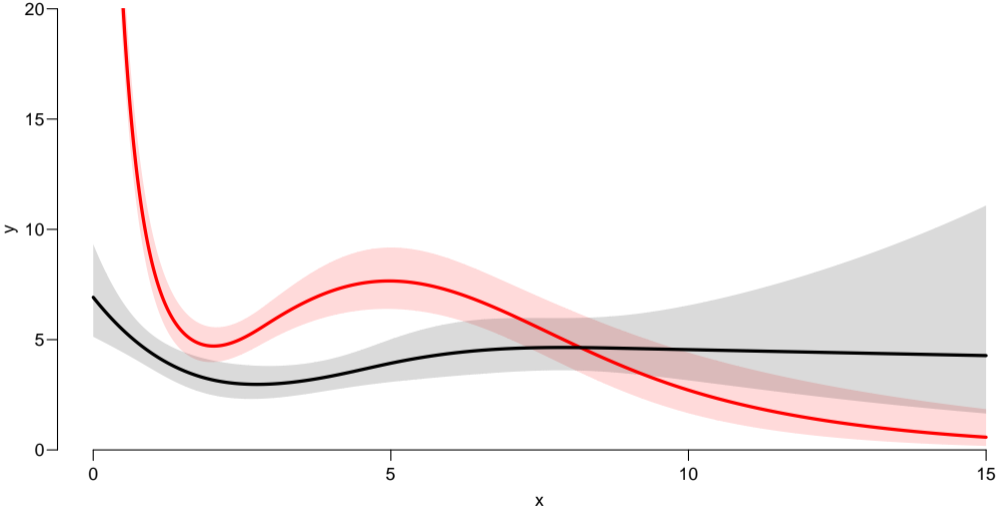
Cause-specific rates

```
> int <- 0.01
> nd <- data.frame(tfd = seq(0, 15, int))
> l.glm <- ci.pred( OAD.glm, nd)
> m.glm <- ci.pred(Dead.glm, nd)
> matshade(nd$tfd,
+          cbind(l.glm, m.glm) * 100,
+          plot = TRUE,
+          yaxs="i", ylim = c(0, 20),
+          # log = "y", ylim = c(2, 20),
+          col = rep(c("red","black"), 2), lwd = 3)
```

Survival and cumulative risk functions



Survival and cumulative risk functions



Integrals with R

- ▶ Integrals look scary to many people, but they are really just areas under curves.
- ▶ The key is to understand how a curve is represented in R.
- ▶ A curve of the function $\mu(t)$ is a set of two vectors: one vector of ts and one vector $y = \mu(t)s$.
- ▶ When we have a model such as the `glm` above that estimates the mortality as a function of time (`tfd`), we can get the mortality as a function of time by first choosing the timepoints, say from 0 to 15 years in steps of 0.01 year (≈ 4 days), using `ci.pred`
- ▶ Then use the formulae with all the integrals to get the state probabilities.

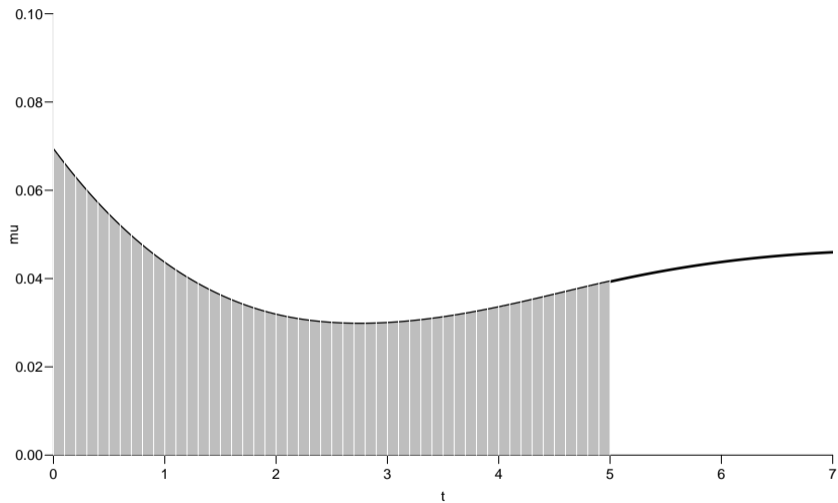
Integrals with R

```
> t <- seq(0, 15, 0.01)
> nd <- data.frame(tfd = t)
> mu <- ci.pred(Dead.glm, nd)[,1]
> head(cbind(t, mu))
```

```
      t      mu
1 0.00 0.069190
2 0.01 0.068853
3 0.02 0.068517
4 0.03 0.068183
5 0.04 0.067851
6 0.05 0.067520
```

```
> plot(t, mu, type="l", lwd = 3,
+       xlim = c(0, 7), xaxs = "i",
+       ylim = c(0, 0.1), yaxs = "i")
> polygon(t[c(1:501,501:1)], c(mu[1:501], rep(0, 501)),
+         col = "gray", border = "transparent")
> abline(v=0:50/10, col="white")
```

Integrals with R



Numerical integration with R

```
> mid <- function(x) x[-1] - diff(x) / 2  
> (x <- c(1:5, 7, 10))
```

```
[1] 1 2 3 4 5 7 10
```

```
> mid(x)
```

```
[1] 1.5 2.5 3.5 4.5 6.0 8.5
```

`mid(x)` is a vector that is 1 shorter than the vector `x`, just as `diff(x)` is.

So if we want the integral over the period 0 to 5 years, we want the sum over the first 500 intervals, corresponding to the first 501 interval endpoints:

```
> cbind(diff(t), mid(mu))[1:5,]
```

```
      [,1]      [,2]  
2 0.01 0.069022  
3 0.01 0.068685  
4 0.01 0.068350  
5 0.01 0.068017  
6 0.01 0.067686
```

Numerical integration with R

In practice we will want the integral **function** of μ , so for every t we want $M(t) = \int_0^t \mu(s) d(s)$. This is easily accomplished by the function `cumsum`:

```
> Mu <- c(0, cumsum(diff(t) * mid(mu)))  
> head(cbind(t, Mu))
```

	t	Mu
	0.00	0.00000000
2	0.01	0.00069022
3	0.02	0.00137707
4	0.03	0.00206057
5	0.04	0.00274074
6	0.05	0.00341760

Note the first value which is the integral from 0 to 0, so by definition 0.

Cumulative risks from parametric models

If we have estimates of λ and μ as functions of time, we can derive the cumulative risks.

In practice this will be by numerical integration; compute the rates at closely spaced intervals and evaluate the integrals as sums. This is easy.

but what is not so easy is to come up with confidence intervals for the cumulative risks.

Simulation of cumulative risks: `ci.Crisk`

1. generate a random vector from the multivariate normal distribution with mean equal to the parameters of the model, and variance-covariance equal to the estimated variance-covariance of the parameter estimates
2. use this to generate a simulated set of rates $(\lambda(t), \mu(t))$, evaluated at closely spaced times
3. use these in numerical integration to derive state probabilities at these times
4. repeat 1000 times, say, to obtain 1000 sets of state probabilities at these times
5. use these to derive confidence intervals for the state probabilities as the 2.5 and 97.5 percentiles of the state probabilities at each time

Cumulative risks from parametric models

```
> cR <- ci.Crisk(mods = list(OAD = OAD.glm,  
+                           Dead = Dead.glm),  
+               nd = nd)
```

NOTE: Times are assumed to be in the column `tfd` at equal distances of 0.01

```
> str(cR)
```

List of 4

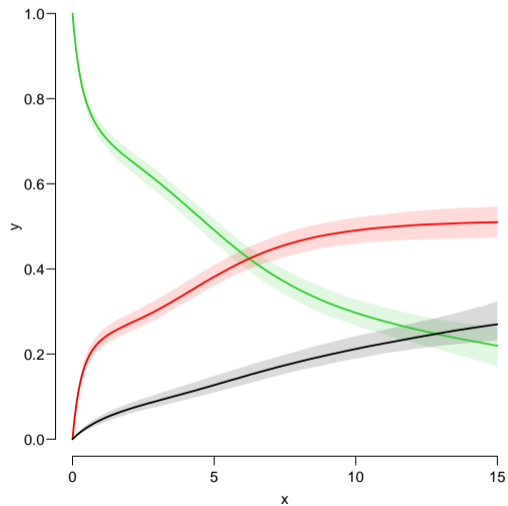
```
$ Crisk: num [1:1501, 1:3, 1:3] 1 0.992 0.984 0.976 0.969 ...  
..- attr(*, "dimnames")=List of 3  
.. ..$ tfd : chr [1:1501] "0" "0.01" "0.02" "0.03" ...  
.. ..$ cause: chr [1:3] "Surv" "OAD" "Dead"  
.. ..$      : chr [1:3] "50%" "2.5%" "97.5%"  
$ Srisk: num [1:1501, 1:2, 1:3] 0 0.000692 0.001375 0.002049 0.002715 ...  
..- attr(*, "dimnames")=List of 3  
.. ..$ tfd : chr [1:1501] "0" "0.01" "0.02" "0.03" ...  
.. ..$ cause: chr [1:2] "Dead" "Dead+OAD"  
.. ..$      : chr [1:3] "50%" "2.5%" "97.5%"  
$ Stime: num [1:1501, 1:3, 1:3] 0 0.00996 0.01984 0.02964 0.03937 ...  
..- attr(*, "dimnames")=List of 3  
.. ..$ tfd : chr [1:1501] "0" "0.01" "0.02" "0.03" ...
```

Cumulative risks from parametric models

So now plot the cumulative *risks* of being in each of the states (the **Crisk** component):

```
> matshade(cR$time, cbind(cR$Crisk[,1,],  
+                         cR$Crisk[,2,],  
+                         cR$Crisk[,3,]), plot = TRUE,  
+          lwd = 2, col = c("limegreen", "red", "black"))
```

Survival and cumulative risk functions



Stacked probabilities: (matrix 2 polygons)

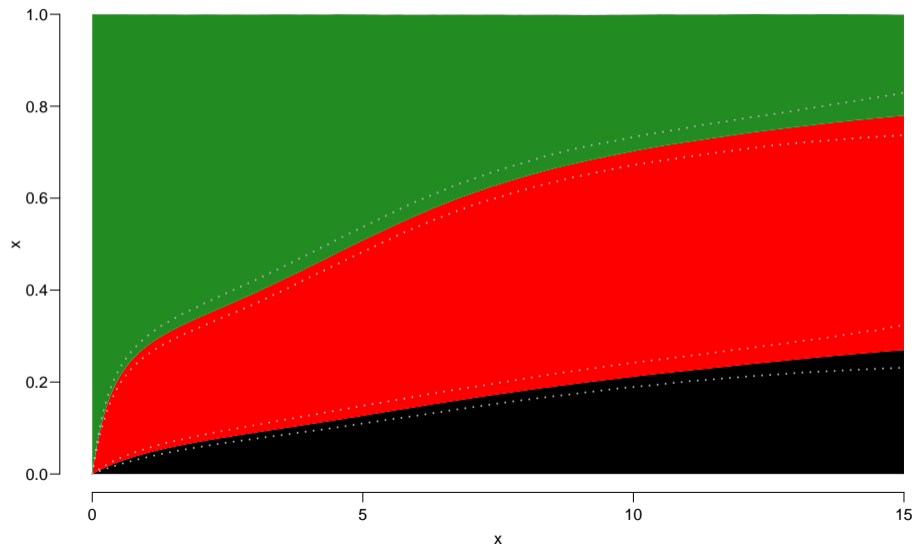
```
> mat2pol(cR$Crisk[,3:1,1], col = c("forestgreen","red","black")[3:1])
```

1st argument to `mat2pol` must be a 2-dimensional matrix, with rows representing the x -axis of the plot, and columns states.

The component `Srisk` has the confidence limits of the stacked probabilities:

```
> mat2pol(cR$Crisk[,3:1,1], col = c("forestgreen","red","black")[3:1])
> matlines(as.numeric(dimnames(cR$Srisk)[["tfd"]]),
+         cbind(cR$Srisk[, "Dead"      ,2:3],
+             cR$Srisk[, "Dead+OAD",2:3]),
+         lty = 3, lwd = 2, col = gray(0.7))
```

Survival and cumulative risk functions



Expected life time: using simulated objects

The areas between the lines (up to say 10 years) are **expected sojourn times**, that is:

- ▶ expected years alive without OAD
- ▶ expected years lost to death without OAD
- ▶ expected years after OAD, including years dead after OAD

Not all of these are of direct relevance; actually only the first may be so.

They are available (with simulation-based confidence intervals) in the component of `cR`, `Stime` (Sojourn time).

Expected life time: using simulated objects

A relevant quantity would be the expected time alive without OAD during the first 5, 10 and 15 years:

```
> str(cR$Stime)
```

```
num [1:1501, 1:3, 1:3] 0 0.00996 0.01984 0.02964 0.03937 ...  
- attr(*, "dimnames")=List of 3  
..$ tfd : chr [1:1501] "0" "0.01" "0.02" "0.03" ...  
..$ cause: chr [1:3] "Surv" "OAD" "Dead"  
..$      : chr [1:3] "50%" "2.5%" "97.5%"
```

```
> round(cR$Stime[c("5", "10", "15"), "Surv", ], 1)
```

```
tfd 50% 2.5% 97.5%  
5 3.2 3.1 3.3  
10 5.1 4.9 5.3  
15 6.4 6.0 6.7
```

Background: Steno 2 trial

- ▶ Clinical trial for diabetes pts. with kidney disease (micro-albuminuria)
- ▶ 80 pts. randomised to either of
 - ▶ Conventional treatment
 - ▶ Intensified multifactorial treatment
- ▶ 1993–2001
- ▶ follow-up till 2018

Steno 2 trial: goal

- ▶ Is there a treatment effect on:
 - ▶ CVD mortality
 - ▶ non-CVD mortality
 - ▶ Albuminuria state
- ▶ Rate-ratios
- ▶ Life times
- ▶ Changes in clinical parameters

```
> data(steno2)
> steno2 <- cal.yr(steno2)
> steno2 <- transform(steno2,
+                       doEnd = pmin(doDth, doEnd, na.rm = TRUE))
> str(steno2)
```

```
'data.frame': 160 obs. of 14 variables:
 $ id      : num  1 2 3 4 5 6 7 8 9 10 ...
 $ allo    : Factor w/ 2 levels "Int","Conv": 1 1 2 2 2 2 2 1 1 1 ...
 $ sex     : Factor w/ 2 levels "F","M": 2 2 2 2 2 2 2 1 2 2 2 ...
 $ baseCVD : num  0 0 0 0 0 1 0 0 0 0 ...
 $ deathCVD: num  0 0 0 0 1 0 0 0 1 0 ...
 $ doBth   : 'cal.yr' num  1932 1947 1943 1945 1936 ...
 $ doDM    : 'cal.yr' num  1991 1982 1983 1977 1986 ...
 $ doBase  : 'cal.yr' num  1993 1993 1993 1993 1993 ...
 $ doCVD1  : 'cal.yr' num  2014 2009 2002 1995 1994 ...
 $ doCVD2  : 'cal.yr' num  NA 2009 NA 1997 1995 ...
 $ doCVD3  : 'cal.yr' num  NA 2010 NA 2003 1998 ...
 $ doESRD  : 'cal.yr' num  NaN NaN NaN NaN 1998 ...
 $ doEnd   : 'cal.yr' num  2015 2015 2002 2003 1998 ...
 $ doDth   : 'cal.yr' num  NA NA 2002 2003 1998 ...
```

A Lexis object

```
> L2 <- Lexis(entry = list(per = doBase,  
+                          age = doBase - doBth,  
+                          tfi = 0),  
+            exit = list(per = doEnd),  
+            exit.status = factor(deathCVD + !is.na(doDth),  
+                                labels=c("Mic", "D(oth)", "D(CVD)")),  
+            id = id,  
+            data = steno2)
```

NOTE: `entry.status` has been set to "Mic" for all.

Explain the coding of `exit.status`.

A Lexis object

```
> summary(L2, t = TRUE)
```

```
Transitions:
```

```
      To  
From  Mic D(oth) D(CVD) Records: Events: Risk time: Persons:  
  Mic  67    55     38     160     93    2416.59     160
```

```
Timescales:
```

```
per age tfi  
"" "" ""
```

How many persons are there in the cohort?

How many deaths are there in the cohort?

How much follow-up time is there in the cohort?

How many states are there in the model (so far)?

Albuminuria status

```
> data(st2alb) ; head(st2alb, 3)
```

```
  id      doTr state
1  1 1993-06-12  Mic
2  1 1995-05-13 Norm
3  1 2000-01-26  Mic
```

```
> cut2 <- rename(cal.yr(st2alb),
+               lex.id = id,
+               cut = doTr,
+               new.state = state)
> with(cut2, addmargins(table(table(lex.id))))
```

```
  1  2  3  4  5 Sum
4 25 40 46 41 156
```

What does this table mean?

Albuminuria status as states

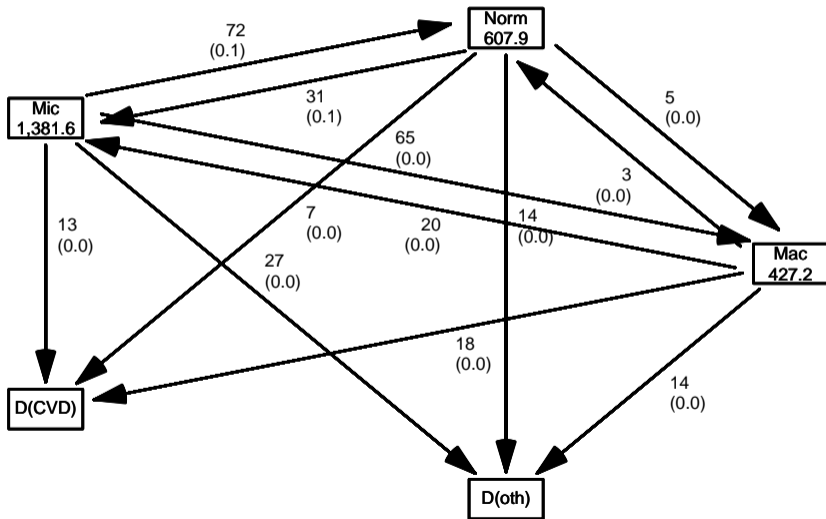
```
> L2$per <- as.numeric(L2$per)
> cut2$cut <- as.numeric(cut2$cut)
> L3 <- rcutLexis(L2, cut2, time = "per")
> summary(L3)
```

Transitions:

	To									
From	Mic	Norm	Mac	D(oth)	D(CVD)	Records:	Events:	Risk time:	Persons:	
Mic	299	72	65	27	13	476	177	1381.57	160	
Norm	31	90	5	14	7	147	57	607.86	69	
Mac	20	3	44	14	18	99	55	427.16	64	
Sum	350	165	114	55	38	722	289	2416.59	160	

```
> boxes(L3, boxpos = TRUE, cex = 0.8)
```

What's wrong with this



What's in jump

```
> (jump <-  
+ subset(L3, (lex.Cst == "Norm" & lex.Xst == "Mac") |  
+           (lex.Xst == "Norm" & lex.Cst == "Mac"))[,  
+           c("lex.id", "per", "lex.dur", "lex.Cst", "lex.Xst")])
```

	lex.id	per	lex.dur	lex.Cst	lex.Xst
291	70	1999.487	2.6748802	Mac	Norm
353	86	2001.759	12.8158795	Norm	Mac
506	130	2000.910	1.8781656	Mac	Norm
511	131	1997.756	4.2354552	Norm	Mac
525	136	1997.214	0.4709103	Mac	Norm
526	136	1997.685	4.2436687	Norm	Mac
654	171	1996.390	5.3388090	Norm	Mac
676	175	2004.585	9.8836413	Norm	Mac

—and what will you do about it?

How to fix things

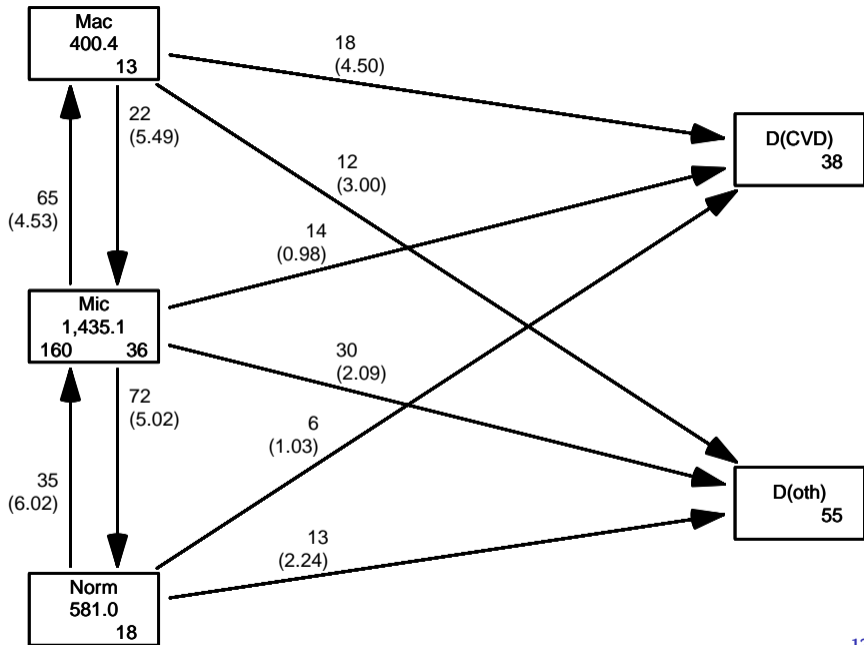
```
> set.seed(1952)
> xcut <- transform(jump,
+                   cut = per + lex.dur * runif(per, 0.1, 0.9),
+                   new.state = "Mic")
> xcut <- select(xcut, c(lex.id, cut, new.state))
> L4 <- rcutLexis(L3, xcut)
> L4 <- Relevel(L4, c("Norm", "Mic", "Mac", "D(CVD)", "D(oth)"))
> summary(L4)
```

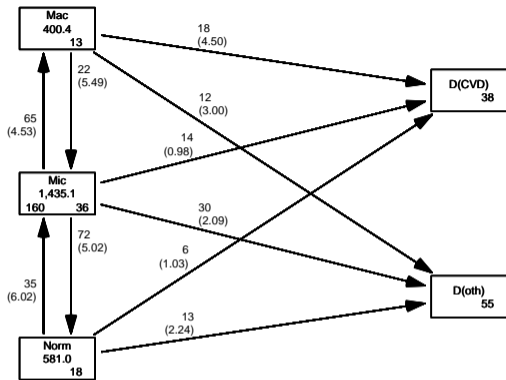
Transitions:

	To								
From	Norm	Mic	Mac	D(CVD)	D(oth)	Records:	Events:	Risk time:	Persons:
Norm	90	35	0	6	13	144	54	581.04	66
Mic	72	312	65	14	30	493	181	1435.14	160
Mac	0	22	41	18	12	93	52	400.41	60
Sum	162	369	106	38	55	730	287	2416.59	160

Plot the boxes

```
> boxes(L4, boxpos = list(x = c(20, 20, 20, 80, 80),
+                           y = c(10, 50, 90, 75, 25)),
+       show.BE = "nz",
+       scale.R = 100, digits.R = 2,
+       cex = 0.9, pos.arr = 0.3)
```





Explain all the numbers in the graph.

Describe the overall effect of albuminuria on the two mortality rates.

Modeling transition rates

- ▶ A model with a smooth effect of timescales on the rates require follow-up in small bits
- ▶ Achieved by `splitLexis` (or `splitMulti` from `popEpi`)
- ▶ Compare the `Lexis` objects

```
> S4 <- splitMulti(L4, tfi = seq(0, 25, 1/2))
> summary(L4)
```

Transitions:

	To									
From	Norm	Mic	Mac	D(CVD)	D(oth)	Records:	Events:	Risk time:	Persons:	
Norm	90	35	0	6	13	144	54	581.04	66	
Mic	72	312	65	14	30	493	181	1435.14	160	
Mac	0	22	41	18	12	93	52	400.41	60	
Sum	162	369	106	38	55	730	287	2416.59	160	

```
> summary(S4)
```

Transitions:

	To									
From	Norm	Mic	Mac	D(CVD)	D(oth)	Records:	Events:	Risk time:	Persons:	
Norm	1252	35	0	6	13	1306	54	581.04	66	
Mic	72	3101	65	14	30	3282	181	1435.14	160	
Mac	0	22	844	18	12	896	52	400.41	60	
Sum	1324	3158	909	38	55	5484	287	2416.59	160	

How the split works:

```
> subset(L4, lex.id == 96)[,1:7]
```

	per	age	tfi	lex.dur	lex.Cst	lex.Xst	lex.id
417	1993.650	51.53183	0.0000000	0.4544832	Mic	Norm	96
418	1994.104	51.98631	0.4544832	2.5790554	Norm	Norm	96
419	1996.683	54.56537	3.0335387	1.9028063	Norm	Norm	96
420	1998.586	56.46817	4.9363450	2.8966461	Norm	D(CVD)	96

```
> subset(S4, lex.id == 96)[c(1:5,NA,17:19),1:7]
```

	lex.id	per	age	tfi	lex.dur	lex.Cst	lex.Xst
3138	96	1993.650	51.53183	0.0000000	0.45448323	Mic	Norm
3139	96	1994.104	51.98631	0.4544832	0.04551677	Norm	Norm
3140	96	1994.150	52.03183	0.5000000	0.50000000	Norm	Norm
3141	96	1994.650	52.53183	1.0000000	0.50000000	Norm	Norm
3142	96	1995.150	53.03183	1.5000000	0.50000000	Norm	Norm
NA	NA	NA	NA	NA	NA	<NA>	<NA>
3154	96	2000.150	58.03183	6.5000000	0.50000000	Norm	Norm
3155	96	2000.650	58.53183	7.0000000	0.50000000	Norm	Norm
3156	96	2001.150	59.03183	7.5000000	0.33299110	Norm	D(CVD)

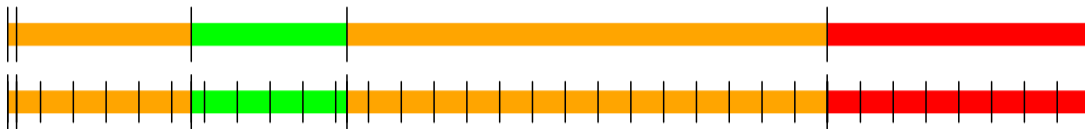
```
> subset(L4, lex.id == 159)[,1:7]
```

	per	age	tfi	lex.dur	lex.Cst	lex.Xst	lex.id
646	1994.025	67.49624	0.0000000	0.1341547	Mic	Mic	159
647	1994.159	67.63039	0.1341547	2.6639288	Mic	Norm	159
648	1996.823	70.29432	2.7980835	2.3737166	Norm	Mic	159
649	1999.196	72.66804	5.1718001	7.3210130	Mic	Mac	159
650	2006.517	79.98905	12.4928131	3.9479808	Mac	D(CVD)	159

```
> subset(S4, lex.id == 159)[c(1:2,NA,6:7,NA,12:13,NA,27:28,NA,36:37),1:7]
```

	lex.id	per	age	tfi	lex.dur	lex.Cst	lex.Xst
4853	159	1994.025	67.49624	0.0000000	0.1341547	Mic	Mic
4854	159	1994.159	67.63039	0.1341547	0.3658453	Mic	Mic
NA	NA	NA	NA	NA	NA	<NA>	<NA>
4858	159	1996.025	69.49624	2.0000000	0.5000000	Mic	Mic
4859	159	1996.525	69.99624	2.5000000	0.2980835	Mic	Norm
NA.1	NA	NA	NA	NA	NA	<NA>	<NA>
4864	159	1998.525	71.99624	4.5000000	0.5000000	Norm	Norm
4865	159	1999.025	72.49624	5.0000000	0.1718001	Norm	Mic
NA.2	NA	NA	NA	NA	NA	<NA>	<NA>
4879	159	2005.525	78.99624	11.5000000	0.5000000	Mic	Mic
4880	159	2006.025	79.49624	12.0000000	0.4928131	Mic	Mac
NA.3	NA	NA	NA	NA	NA	<NA>	<NA>
4888	159	2009.525	82.99624	15.5000000	0.5000000	Mac	Mac
4889	159	2010.025	83.49624	16.0000000	0.4407940	Mac	D(CVD)

How the split works



Same amount of follow-up

Same transitions

More intervals (5, resp. 37)

Different value of time scales between intervals

Purpose of the split

- ▶ Assumption of constant rate in each interval
- ▶ All intervals are (shorter than) 0.5 years
- ▶ Magnitude of the rates depend on covariates:
 - ▶ fixed covariates
 - ▶ time scales
 - ▶ randomly varying covariates (not now)
- ▶ value of covariates differ between intervals
- ▶ each record contributes one term to the (log-)likelihood for a specific rate
from a given origin state (`lex.Cst`)
to a given destination state (`lex.Cst`).
- ▶ —looks as the likelihood for a single Poisson observation

Modeling the rate: Mic \rightarrow D(CVD)

```
> mr <- glm(cbind(lex.Xst == "D(CVD)" & lex.Cst != lex.Xst,
+               lex.dur)
+          ~ Ns(tfi, knots = seq( 0, 20, 5)) +
+            Ns(age, knots = seq(50, 80, 10)),
+          family = poisreg,
+          data = subset(S4, lex.Cst == "Mic"))
```

... the same as:

```
> mp <- glm((lex.Xst == "D(CVD)" & lex.Cst != lex.Xst)
+          ~ Ns(tfi, knots = seq( 0, 20, 5)) +
+            Ns(age, knots = seq(50, 80, 10)),
+          offset = log(lex.dur),
+          family = poisson,
+          data = subset(S4, lex.Cst == "Mic"))
> summary(coef(mr) - coef(mp))
```

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
	-1.315e-12	-2.389e-13	-2.343e-14	-1.540e-13	7.050e-15	6.466e-13

Modeling the rate: Mic \rightarrow D(CVD)

A convenient wrapper for `Lexis` objects:

```
> mL <- glm.Lexis(S4,  
+               ~ Ns(tfi, knots = seq( 0, 20, 5)) +  
+               Ns(age, knots = seq(50, 80, 10)),  
+               from = "Mic",  
+               to = "D(CVD)")
```

```
stats::glm Poisson analysis of Lexis object S4 with log link:  
Rates for the transition: Mic->D(CVD)
```

```
> summary(coef(mr) - coef(mL))
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0	0	0	0	0	0

`glm.Lexis` by default models all transitions to absorbing states, from states preceding these

```
> mX <- glm.Lexis(S4,  
+               ~ Ns(tfi, knots = seq( 0, 20, 5)) +  
+               Ns(age, knots = seq(50, 80, 10)) +  
+               lex.Cst)
```

```
stats::glm Poisson analysis of Lexis object S4 with log link:  
Rates for transitions: Norm->D(CVD), Mic->D(CVD), Mac->D(CVD), Norm->D(oth), Mic->
```

Describe the model(s) in `mX`:

- ▶ What rates are modeled ?
- ▶ How are they modeled (assumptions about shapes) ?
- ▶ What are the differences between the rates modeled?
- ▶ What would you rather do?

```
> mox <- glm.Lexis(S4, ~ Ns(tfi, knots = seq( 0, 20, 5)) +  
+                      Ns(age, knots = seq(50, 80, 10)) +  
+                      lex.Cst / allo,  
+                      to = "D(oth)")
```

stats::glm Poisson analysis of Lexis object S4 with log link:
Rates for transitions: Norm->D(oth), Mic->D(oth), Mac->D(oth)

```
> mCx <- glm.Lexis(S4, ~ Ns(tfi, knots = seq( 0, 20, 5)) +  
+                      Ns(age, knots = seq(50, 80, 10)) +  
+                      lex.Cst / allo,  
+                      to = "D(CVD)")
```

stats::glm Poisson analysis of Lexis object S4 with log link:
Rates for transitions: Norm->D(CVD), Mic->D(CVD), Mac->D(CVD)

```
> det <- glm.Lexis(S4, ~ Ns(tfi, knots = seq( 0, 20, 5)) +
+                      Ns(age, knots = seq(50, 80, 10)) +
+                      lex.Cst / allo,
+                      from = c("Norm", "Mic"),
+                      to = c("Mic", "Mac"))
```

stats::glm Poisson analysis of Lexis object S4 with log link:
Rates for transitions: Norm->Mic, Mic->Mac

```
> imp <- glm.Lexis(S4, ~ Ns(tfi, knots = seq( 0, 20, 5)) +
+                      Ns(age, knots = seq(50, 80, 10)) +
+                      lex.Cst / allo,
+                      from = c("Mac", "Mic"),
+                      to = c("Mic", "Norm"))
```

stats::glm Poisson analysis of Lexis object S4 with log link:
Rates for transitions: Mac->Mic, Mic->Norm

Specification of the model

```
> Tr <- list(Norm = list("Mic" = det,  
+                       "D(oth)" = mox,  
+                       "D(CVD)" = mCx),  
+           Mic = list("Mac" = det,  
+                     "Norm" = imp,  
+                     "D(oth)" = mox,  
+                     "D(CVD)" = mCx),  
+           Mac = list("Mic" = imp,  
+                     "D(oth)" = mox,  
+                     "D(CVD)" = mCx))  
> lapply(Tr, names)  
  
$Norm  
[1] "Mic"      "D(oth)" "D(CVD)"  
  
$Mic  
[1] "Mac"      "Norm"   "D(oth)" "D(CVD)"  
  
$Mac  
[1] "Mic"      "D(oth)" "D(CVD)"
```


Spcification of the prediction population

```
> ini <- L2[,c("per", "age", "tfi")]
> ini <- rbind(transform(ini, lex.Cst = factor("Mic"), allo = factor("Int")),
+             transform(ini, lex.Cst = factor("Mic"), allo = factor("Conv")))
> ini$lex.Cst <- factor(ini$lex.Cst, levels = levels(L4))
> str(ini)
```

Classes 'Lexis' and 'data.frame': 320 obs. of 5 variables:

```
$ per      : num  1993 1993 1993 1993 1993 ...
$ age      : 'cal.yr' num  61.1 46.6 49.9 48.5 57.3 ...
$ tfi      : num   0 0 0 0 0 0 0 0 0 0 0 ...
$ lex.Cst: Factor w/ 5 levels "Norm","Mic","Mac",...: 2 2 2 2 2 2 2 2 2 2 ...
$ allo     : Factor w/ 2 levels "Int","Conv": 1 1 1 1 1 1 1 1 1 1 ...
- attr(*, "breaks")=List of 3
  ..$ per: NULL
  ..$ age: NULL
  ..$ tfi: NULL
- attr(*, "time.scales")= chr  "per" "age" "tfi"
- attr(*, "time.since")= chr  "" "" ""
```

Simulating a cohort

```
> set.seed(1952)
> system.time(
+ Sorg <- simLexis(Tr = Tr, # models for each transition
+                 init = ini, # cohort of straters
+                 N = 10, # how many copies of each person in ini
+                 t.range = 20, # how long should we simulate before censoring
+                 n.int = 100))# how many intervals for evaluating rates
```

```
user system elapsed
19.460  9.104  17.566
```

Simulated cohort

```
> summary(Sorg)
```

```
Transitions:
```

```
  To
```

From	Norm	Mic	Mac	D(CVD)	D(oth)	Records:	Events:	Risk time:	Persons:
Norm	387	655	0	114	266	1422	1035	11582.90	1300
Mic	1422	646	1302	281	574	4225	3579	26869.84	3200
Mac	0	370	308	383	241	1302	994	7831.63	1206
Sum	1809	1671	1610	778	1081	6949	5608	46284.38	3200

Who is where when?

```
> Nst <- nState(Sorg,
+             at = seq(0, 20, 0.2),
+             from = 0,
+             time.scale = "tfi")
> str(Nst)

'table' int [1:101, 1:5] 0 88 167 233 295 341 389 443 500 542 ...
- attr(*, "dimnames")=List of 2
 ..$ when : chr [1:101] "0" "0.2" "0.4" "0.6" ...
 ..$ State: chr [1:5] "Norm" "Mic" "Mac" "D(CVD)" ...
```

```
> head(Nst)
```

	State				
when	Norm	Mic	Mac	D(CVD)	D(oth)
0	0	3200	0	0	0
0.2	88	3077	33	2	0
0.4	167	2966	62	5	0
0.6	233	2863	98	6	0
0.8	295	2772	120	13	0
1	341	2693	148	17	1

Who is where when?

```
> Nint <- nState(subset(Sorg, allo == "Int"),
+               at = seq(0, 20, 0.1),
+               from = 0,
+               time.scale = "tfi")
> Nconv<- nState(subset(Sorg, allo == "Conv"),
+               at = seq(0, 20, 0.1),
+               from = 0,
+               time.scale = "tfi")
> head(Nint , 4)
```

	State					
when	Norm	Mic	Mac	D(CVD)	D(oth)	
0	0	1600	0	0	0	
0.1	24	1569	6	1	0	
0.2	55	1533	11	1	0	
0.3	76	1507	15	2	0	

```
> head(Nconv, 4)
```

	State					
when	Norm	Mic	Mac	D(CVD)	D(oth)	
0	0	1600	0	0	0	