Multistate models:

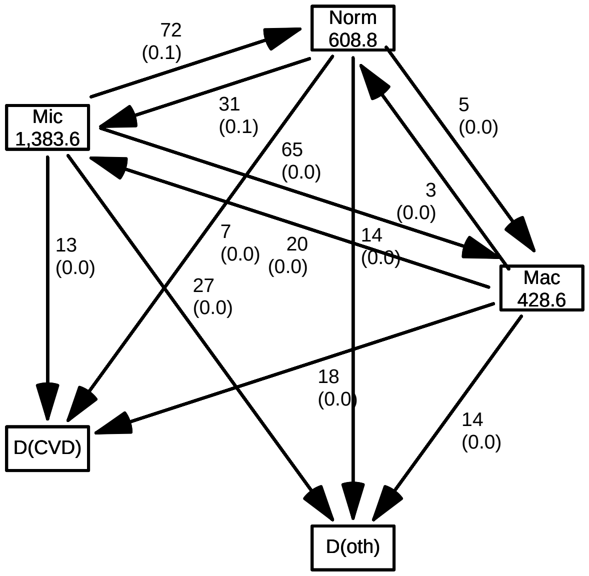
Occurrence rates, cumulative risks, competing risks, state probabilities with multiple states and time scales in Register Research with `R` and `Epi::Lexis`

Bendix Carstensen Steno Diabetes Center Copenhagen
Gentofte, Denmark
`http://BendixCarstensen.com`
Lars Diaz Steno Diabetes Center Copenhagen
Adam Hulman Steno Diabetes Center Midt

SDCC, October 2021

From `/home/bendix/teach/AdvCoh/courses/SDCA.2021/slides/slides.tex` Monday 4th October, 2021, 17:06

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 th elikelihood of data is the statistical model $\Rightarrow$ 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")

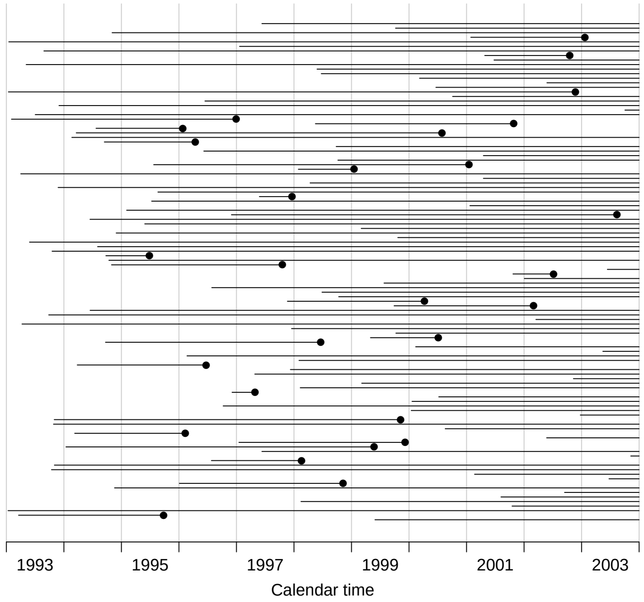
Examples of time-to-event measurements

- 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

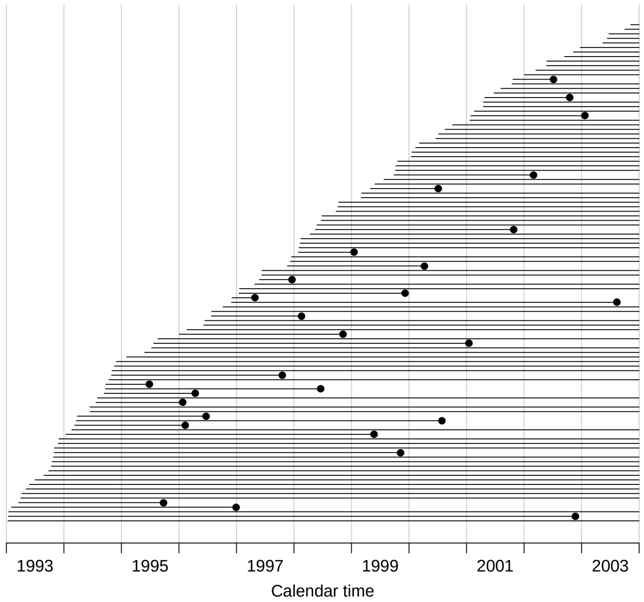
Each blob a death

Study ended at 31 Dec. 2003

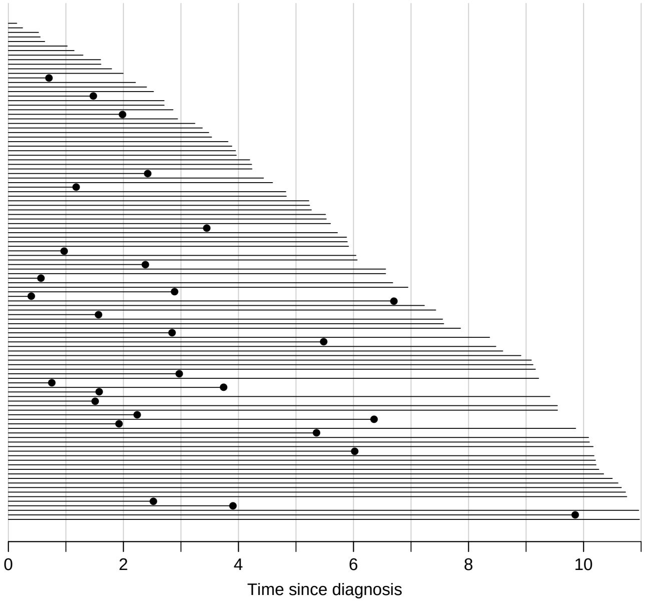


Ordered by date of entry

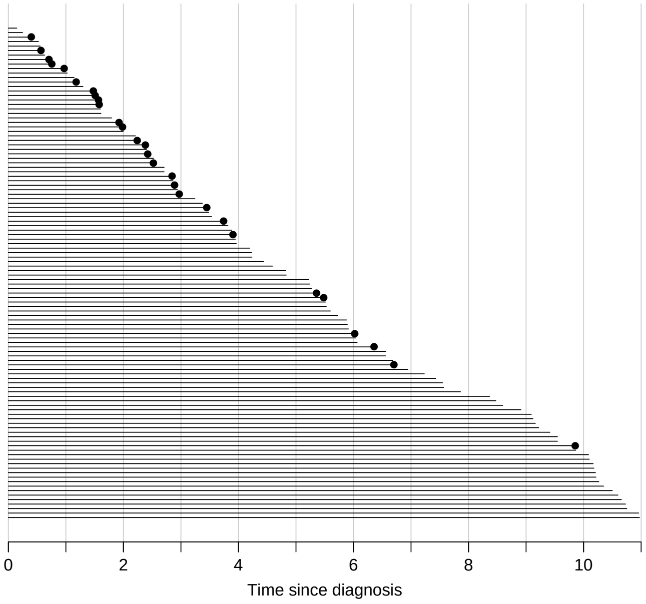
Most likely the order in your database.



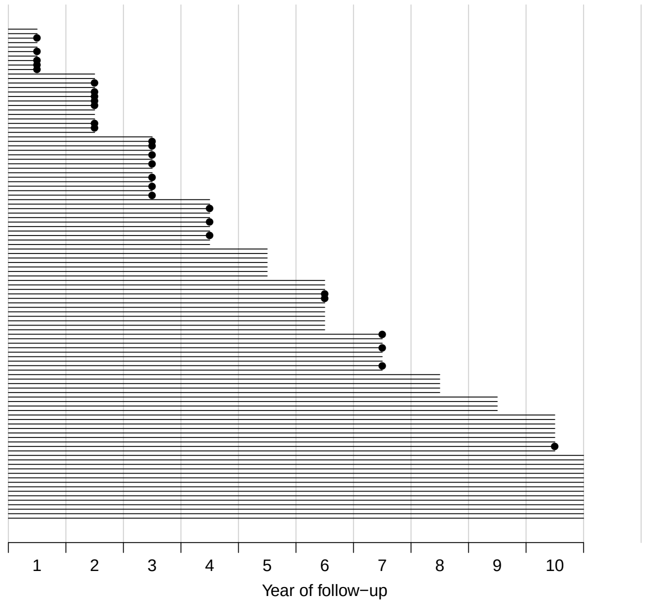
Timescale changed to
“Time since diagnosis”.



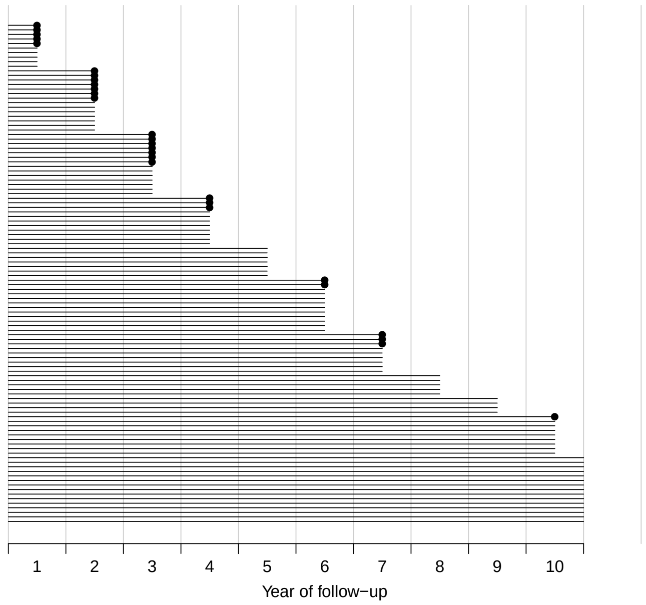
Patients ordered by survival time.



Survival times grouped into bands of survival.



Patients ordered by survival status within each band.



Survival after Cervix cancer

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

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

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

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: $\mathrm{P}\{\text{Death}\} = 1/n_t$
- 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 bút 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:**

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

  —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?

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

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%
(Intercept) 0.03255152 0.01029228 0.1029511
sexW        0.61820515 0.44555636 0.8577537
age         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 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.

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

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

lex.Xst: eXit state, 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=.

Overkill?

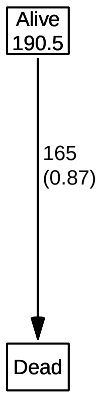
Lexis object: Overview of FU

```
> 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)
```

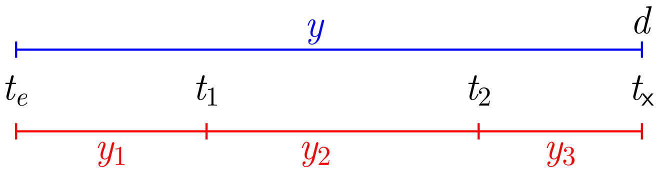
or even simpler, by using the Lexis features:

```
> 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
```

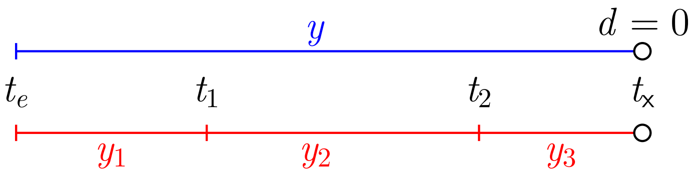
Likelihood and records

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

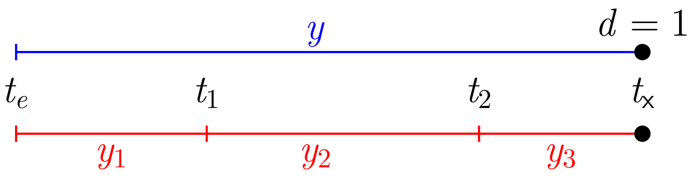
$$\begin{aligned}
\mathrm{P}\left\{d \text{ at } t_x \,|\, \text{entry at } t_e\right\} = {} & \mathrm{P}\left\{\text{survive } (t_e, t_1] \,|\, \text{alive at } t_e\right\} \times \\
& \mathrm{P}\left\{\text{survive } (t_1, t_2] \,|\, \text{alive at } t_1\right\} \times \\
& \mathrm{P}\left\{\text{survive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



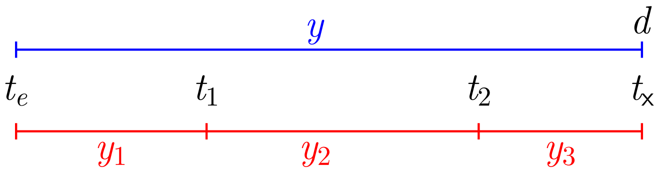
Probability	log-Likelihood
$\mathrm{P}(d \text{ at } t_x \vert \text{entry } t_e)$	$d\log(\lambda) - \lambda y$
$= \mathrm{P}(\text{surv } t_e \rightarrow t_1 \vert \text{entry } t_e)$	$= 0\log(\lambda) - \lambda y_1$
$\times\, \mathrm{P}(\text{surv } t_1 \rightarrow t_2 \vert \text{entry } t_1)$	$+\,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



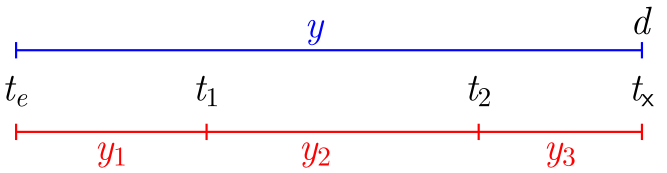
Probability	log-Likelihood
$\mathrm{P}(\text{surv } t_e \rightarrow t_x \vert \text{entry } t_e)$	$0\log(\lambda) - \lambda y$
$= \mathrm{P}(\text{surv } t_e \rightarrow t_1 \vert \text{entry } t_e)$	$= 0\log(\lambda) - \lambda y_1$
$\times\, \mathrm{P}(\text{surv } t_1 \rightarrow t_2 \vert \text{entry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



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



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



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

— allows different rates (λ_i) in each interval

Baseline hazard: splitting time

```
> Sl <- splitMulti(Ll, tfl = 0:36)
> summary(Ll)
```

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

```
> summary(Sl)
```

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

What happended to no. records?

What happended to amount of risk time?

What happended to no. events?

```
> wh <- names(Ll)[1:10] # names of variables in some order
> subset(Ll, 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(Sl, 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 Sl 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 = Sl)
```

or even simpler:

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

```
stats::glm Poisson analysis of Lexis object Sl 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(cl),
+             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?

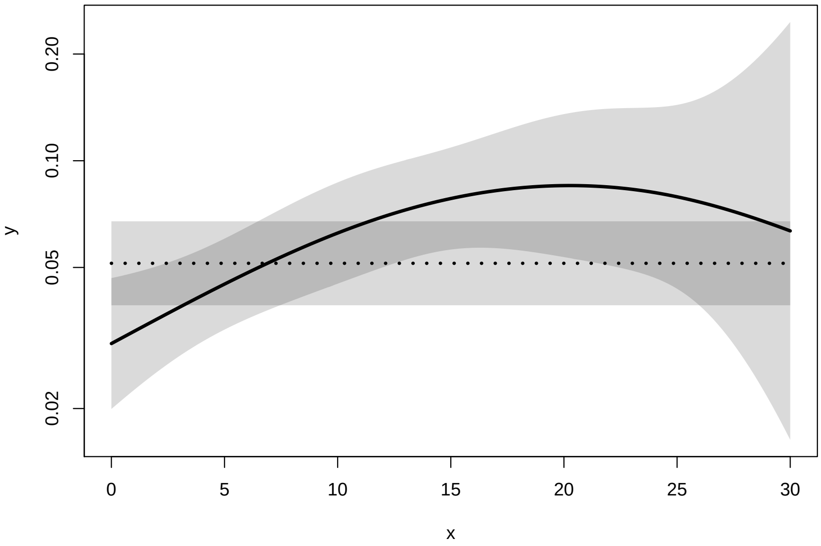
ps is a model for the hazard so we can predict the value of it at defined values for the covariates in the model:

```
> 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!



What are the units on the y-axis? Describe the mortality rates

Survival function and hazard function

$$S(t) = \exp(-\int_0^t \lambda(u)\,\mathrm{d}u)$$

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

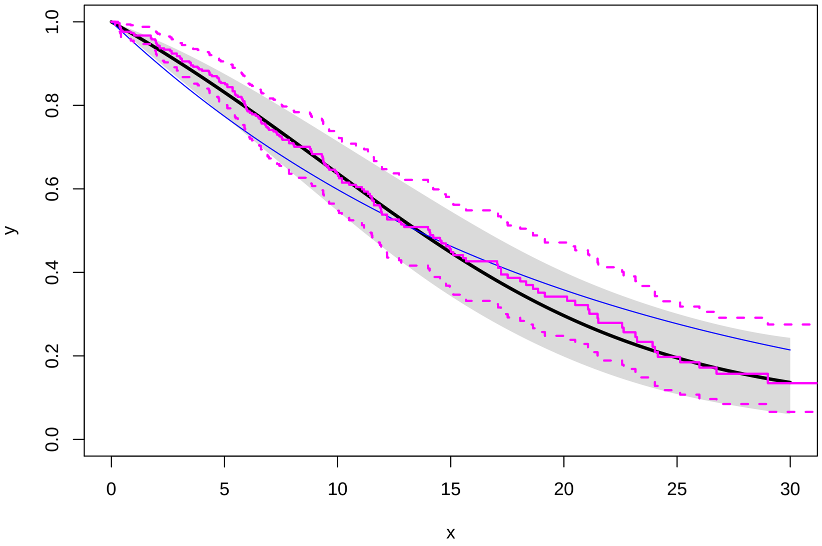
Implemented in the `ci.surv` function

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")
```

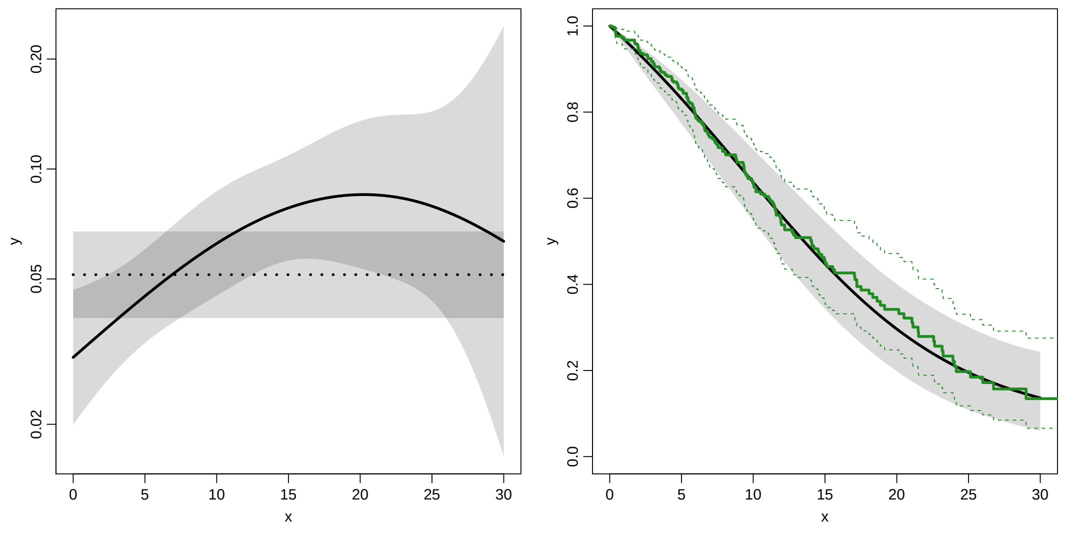
Survival functions



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

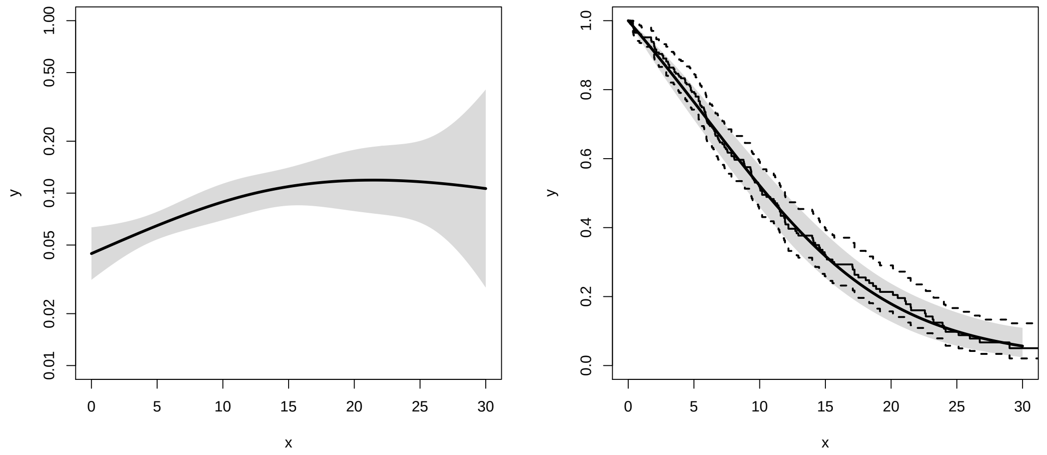


K-M estimator and smooth Poisson model

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

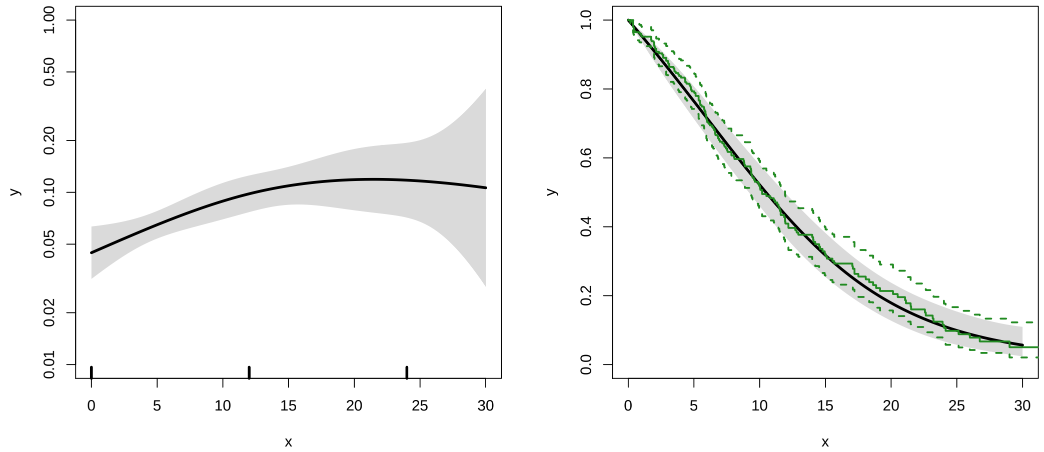


K-M estimator and smooth Poisson model

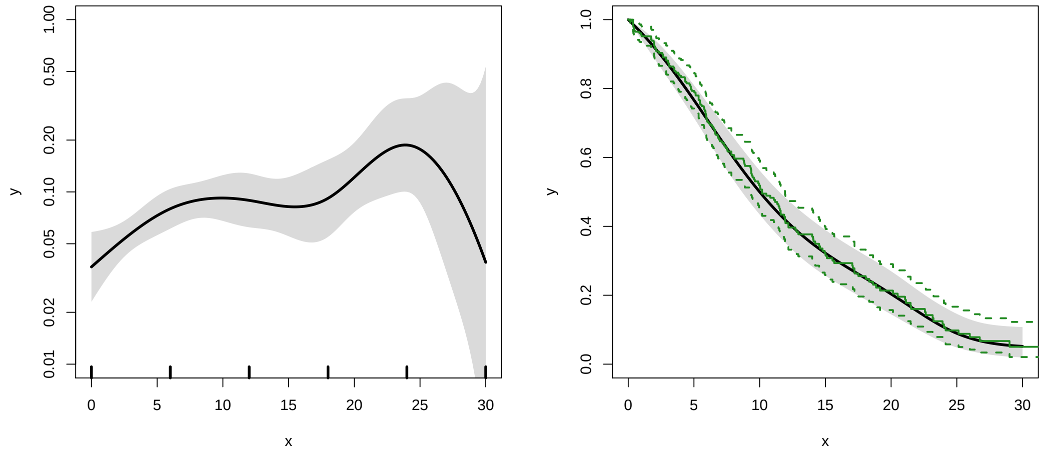
We can explore how the tightness of the knots in the smooth model influence the underlying hazard and the resulting survival function:

```
> 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)
```

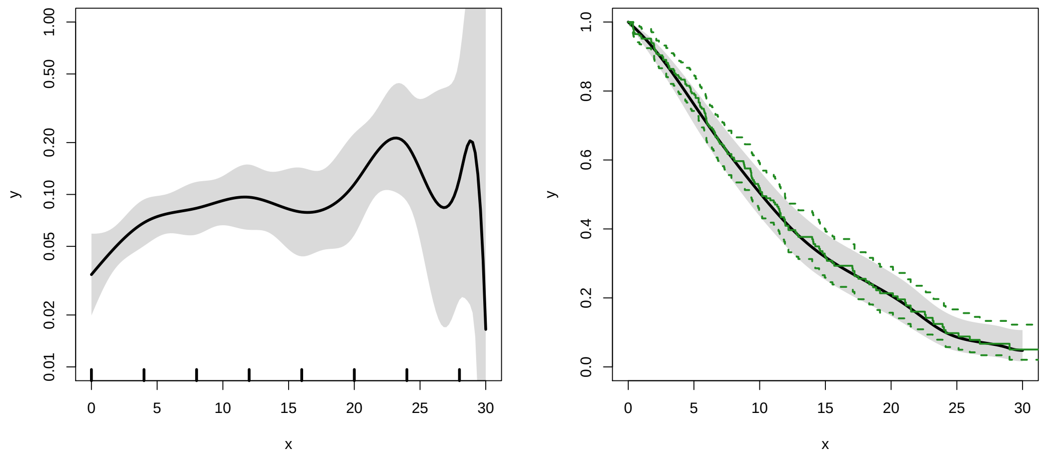
K-M estimator and smooth Poisson model



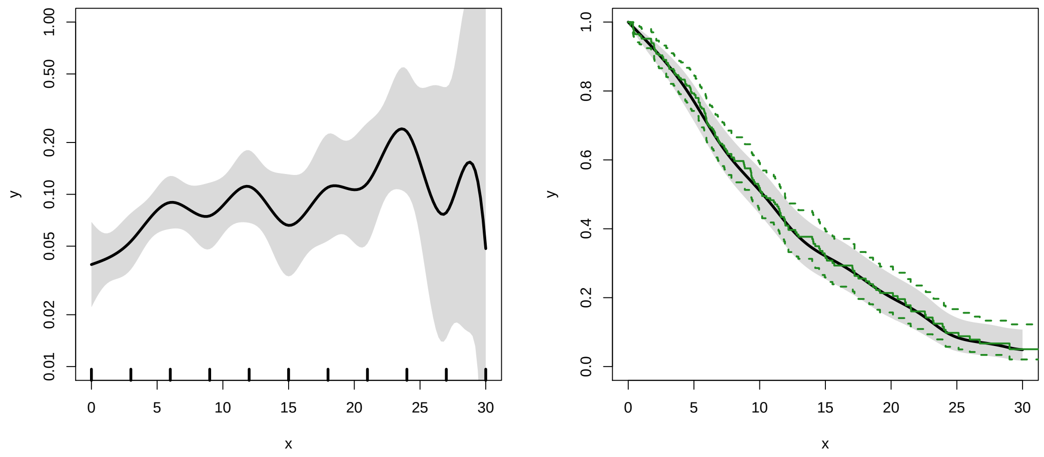
K-M estimator and smooth Poisson model



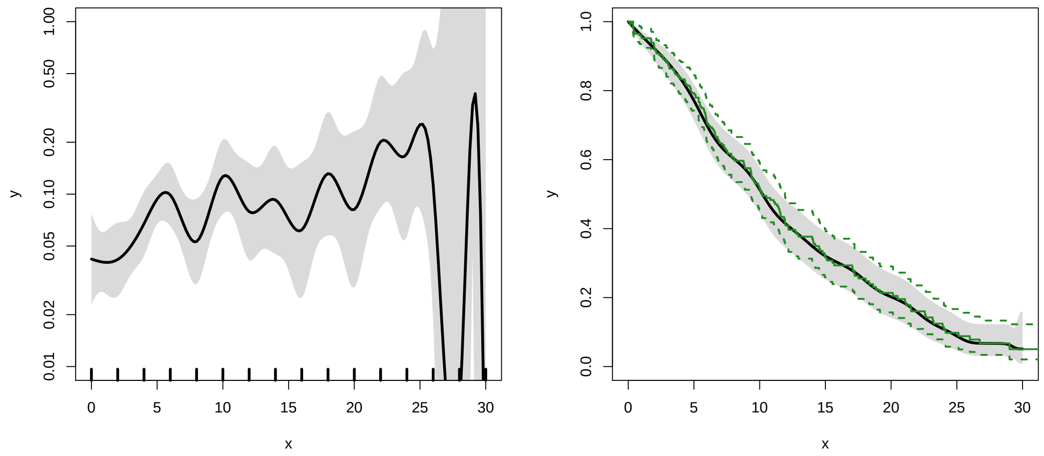
K-M estimator and smooth Poisson model



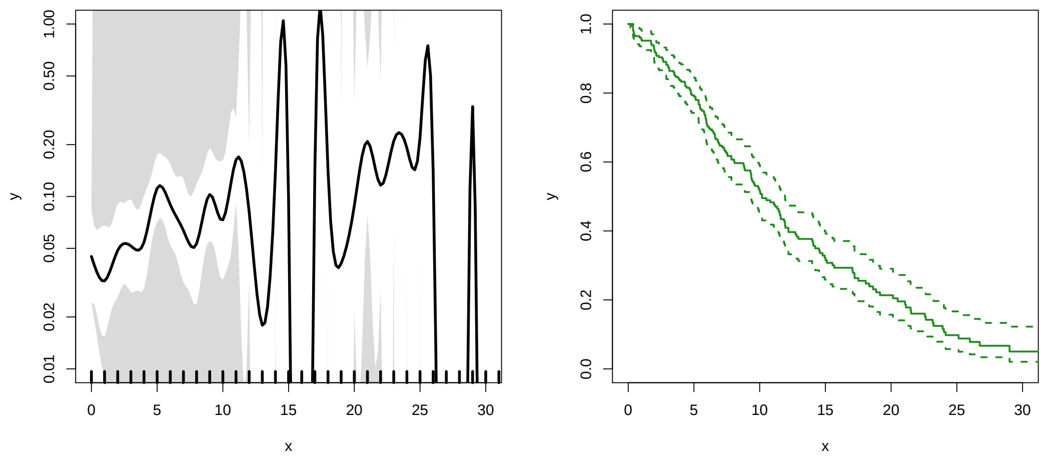
K-M estimator and smooth Poisson model



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, 30, 6) * 30) + 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 ...
```

```
> head(DMlate)
```

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$dooad,
+            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
```

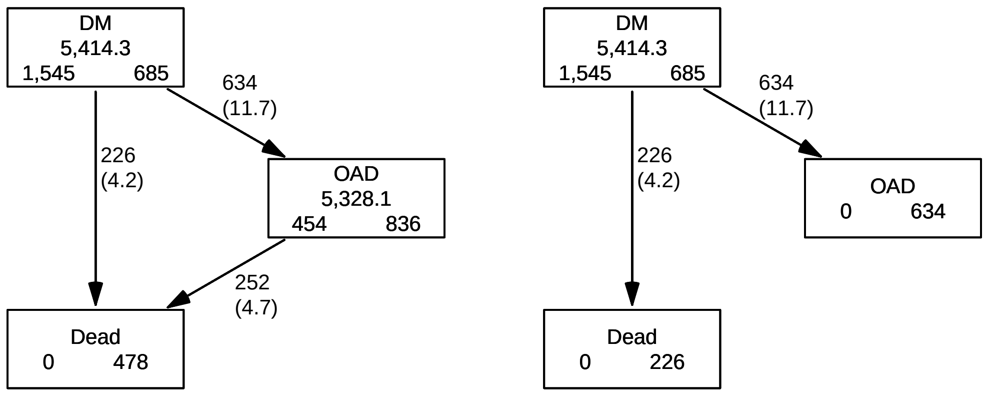
Cut follow-up at the date of OAD, dooad

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

          per    age tfd lex.dur lex.Cst lex.Xst lex.id  dooad
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  dooad
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?

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

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` → `Dead`
- **Competing risks**: what is the probability of being in each of the states `DM`, `OAD` and `Dead`
  —depends on **two** rates, `DM` → `OAD` and `DM` → `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

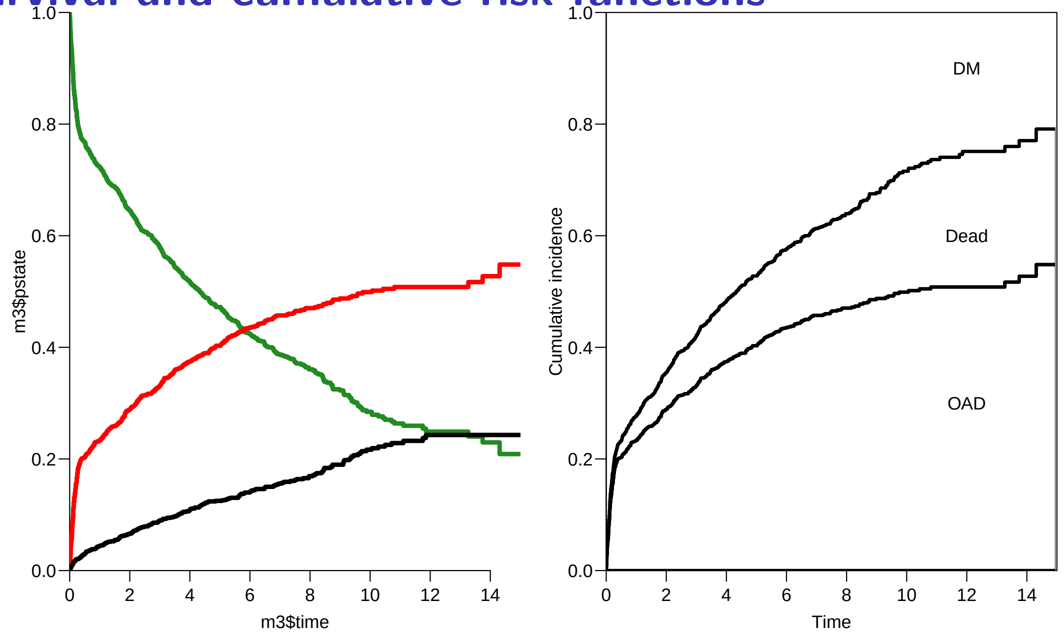
$$
\begin{aligned}
S(t) &= \exp\Big(-\int_0^t \lambda(u) + \mu(u)\,\mathrm{d}u\Big) \\
R_{\texttt{Dead}}(t) &= \int_0^t \mu(u) S(u)\,\mathrm{d}u \\
R_{\texttt{OAD}}(t) &= \int_0^t \lambda(u) S(u)\,\mathrm{d}u) \\
&= \int_0^t \lambda(u) \exp\Big(-\int_0^u \lambda(s) + \mu(s)\,\mathrm{d}s\Big)\,\mathrm{d}u
\end{aligned}
$$

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

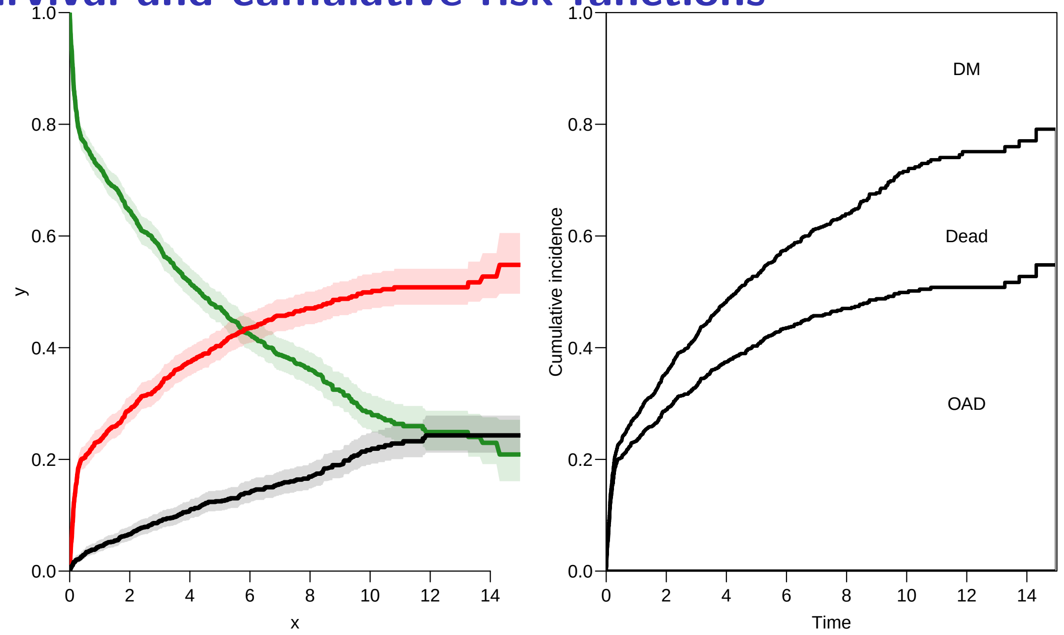
Survival function and cumulative risks

```
> 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

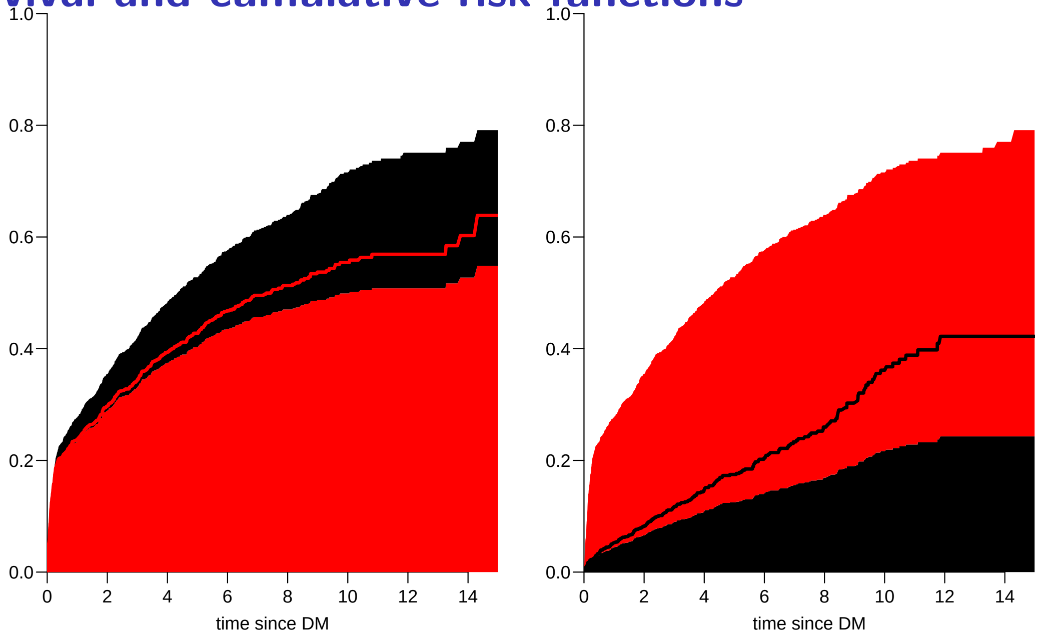
$$
\begin{aligned}
S(t) &= \exp\Big(-\int_0^t \lambda(u)+\mu(u)\,\mathrm{d}u\Big) \\
R_{\texttt{Dead}}(t) &= \int_0^t \mu(u)S(u)\,\mathrm{d}u \\
R_{\texttt{OAD}}(t) &= \int_0^t \lambda(u)S(u)\,\mathrm{d}u) \\
&= \int_0^t \lambda(u)\mathrm{exp}\Big(-\int_0^u \lambda(s)+\mu(s)\,\mathrm{d}s\Big)\,\mathrm{d}u \\
&\neq \int_0^t \lambda(u)\mathrm{exp}\Big(-\int_0^u \lambda(s)\,\mathrm{d}s\Big)\,\mathrm{d}u \\
&= 1-\exp\Big(-\int_0^t \lambda(s)\,\mathrm{d}s\Big) \text{ — nice formula, but wrong!}
\end{aligned}
$$

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
```

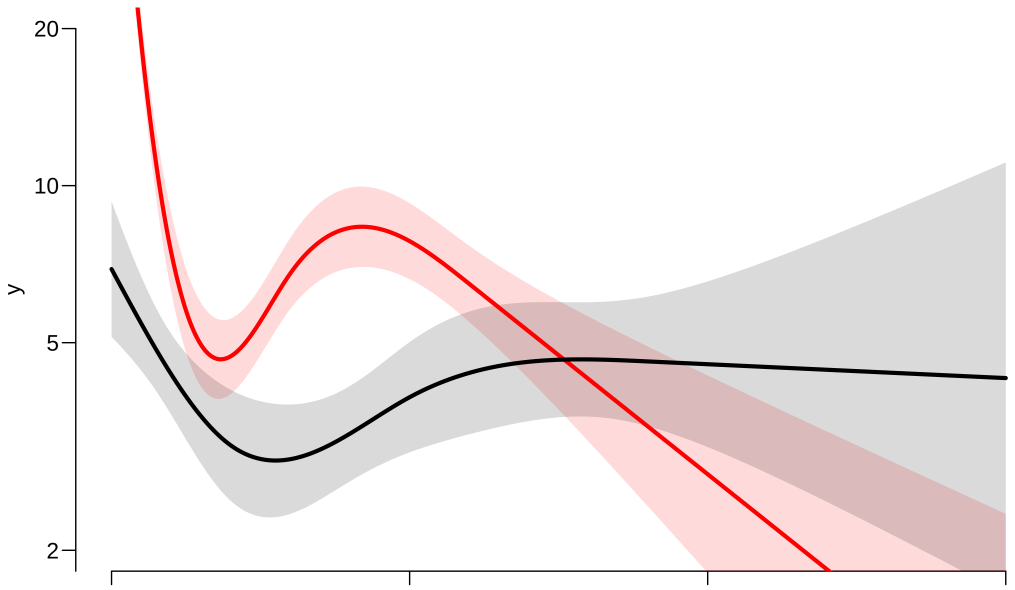
Cause-specific rates

```
> 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
```

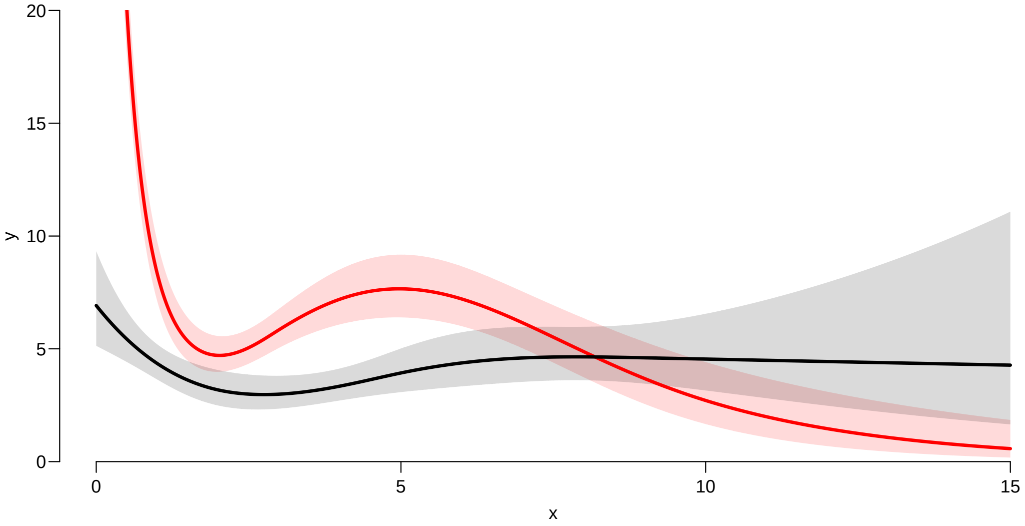
Cause-specific rates

```
> 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 funtion 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 formuale with all the integrals to get the state probabilities.

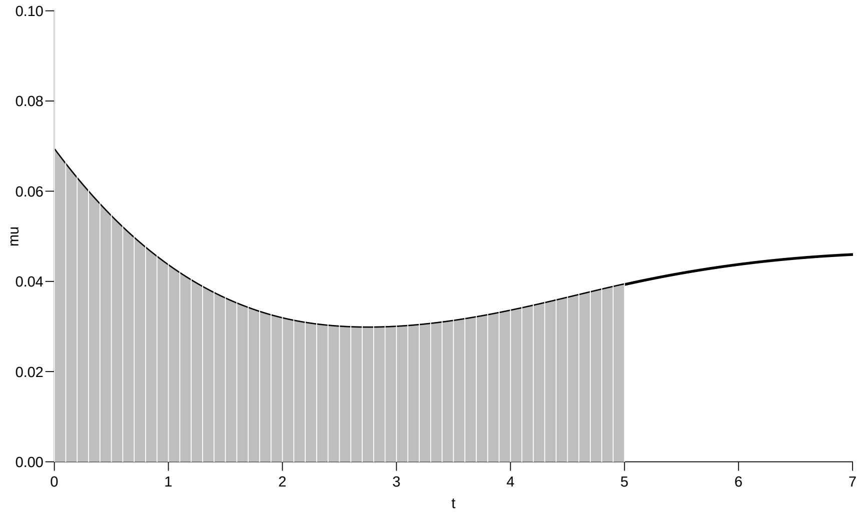
Integrals with R

```
> 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 $M(t) = \int_0^t \mu(s)\,\mathrm{d}(s)$. 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 a 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)
```

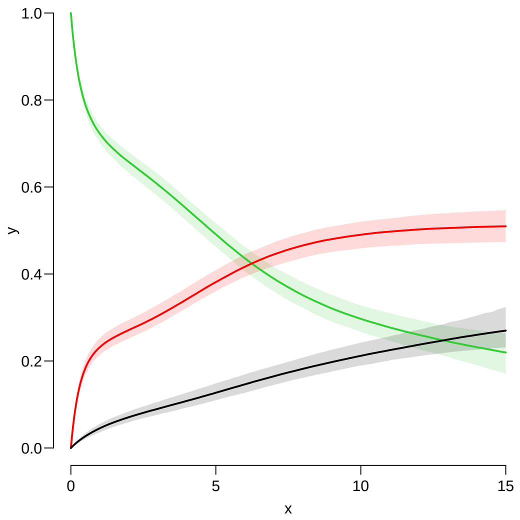
```
List of 4
 $ Crisk: num [1:1501, 1:3, 1:3] 1 0.992 0.984 0.976 0.969 ...
  ..- attr(*, "dimnames")=List of 3
  .. ..$ tfd  : chr [1:1501] "0" "0.01" "0.02" "0.03" ...
  .. ..$ cause: chr [1:3] "Surv" "OAD" "Dead"
  .. ..$      : chr [1:3] "50%" "2.5%" "97.5%"
 $ Srisk: num [1:1501, 1:2, 1:3] 0 0.000692 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)

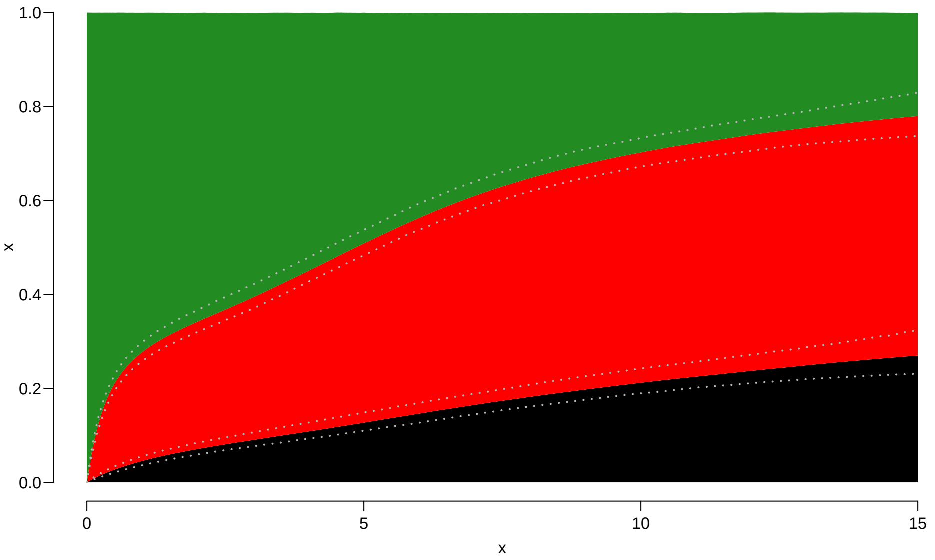
```
> mat2pol(cR$Crisk[,3:1,1], col = c("forestgreen","red","black")[3:1])
```

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
```