Multistate models

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

SPE, Tartu, Estonia,

June 2023

http://BendixCarstensen.com/SPE

From C:\Bendix\teach\SPE\git\lectures\multistate/multistate.tex Wednesday 7 June, 2023, 10:31

Multistate models

Bendix Carstensen, Martyn Plummer

Multistate models

SPE, Tartu, Estonia,

June 2023

Common assumptions in survival analysis

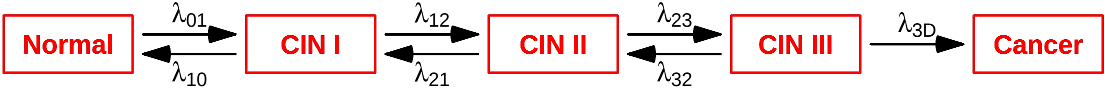
1. Subjects are **either** “healthy” **or** “diseased”, with no intermediate state.
2. The disease is **irreversible**, or requires intervention to be cured.
3. The time of disease incidence is known **exactly**.
4. The disease is **accurately** diagnosed.

These assumptions are true for death and many chronic diseases.

A question of definition:
– consider occurrence of **recording of** a given disease

A model for cervical cancer

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



Purpose of a screening programme is to detect and treat CIN — status of persons obtained at screening dates

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

- Transition rates between states
- Probability of state occupancy

Markov models for multistate processes

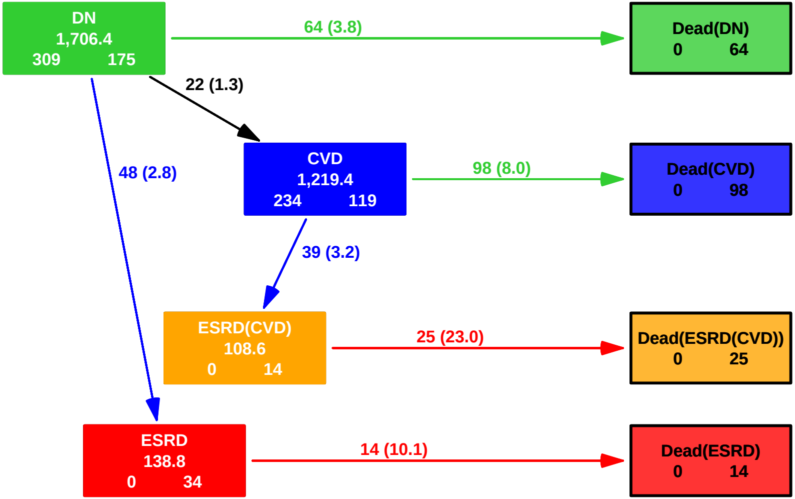
The natural generalization of Poisson regression to multiple disease states:

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

Components of a multistate (Markov) model

- Define the disease states
- Define which transitions between states are allowed
- Select covariates influencing transition rates (may be different between transitions)
- Not a trivial task — do we want *e.g.*
  - cause of death (CVD, Cancer, Other)
  - disease status at death (prev.CVD, prev.Can, neither)

A more complicated multistate model



Likelihood for a multistate model

- The likelihood of the model depends on the probability of being in state j at time t_1, given that you were in state i at time t_0.
- Assume transition rates constant in small time intervals
- $\Rightarrow$ each interval for a person contributes term(s) to the likelihood
- one term for each possible transition between states
- the total likelihood for person p in intervals i is a product of these terms, $d_{pi}\log(\lambda_{pi}) - \lambda_{pi}y_{pi}$
- $\Rightarrow$ each term has the form of the likelihood for a Poisson variate d with mean λy

Likelihood for a multistate model

- each term has the form of the likelihood for a Poisson variate d with mean λy
- terms are **not** independent, but the total likelihood is a product; hence of the same form as the likelihood from independent Poisson variates
- but observations from intervals from one person are neither Poisson nor independent

Realms of multistate modeling

- intensities — dimension time^{-1}
- state probabilities — dimensionless, time^{0}
  integral of intensities w.r.t. to time
- sojourn times — dimension time^{1}
  integral of state probabilities w.r.t. to time

Classes of multistate models

- Markov model: transition between states depends **only** on current state $\Rightarrow$ transition rates are constant
  time-homogeneous Markov model
- If transition rates depend on the **same timescale** only we have a time-**in**homogeneous Markov model
- If transition rates depend on the time since entry to the current state we have a **semi**-Markov model
- If transition rates depend on several timescales we have a general multistate model (there is no name for this)

Computing state probabilities from intensities in multistate models

- time-homogeneous Markov model:
  closed-form formulae exist
- time-**in**homogeneous Markov model:
  closed-form formulae exist (a bit more complicated)
- **semi**-Markov model:
  no closed form formulae exist
- general multistate model:
  no closed form formulae exist

No formulae means that any inference on state probabilities and sojourn times must be based on **simulation** from the model.

Multistate models with Lexis

Bendix Carstensen

Multistate models

SPE, Tartu, Estonia,

June 2023

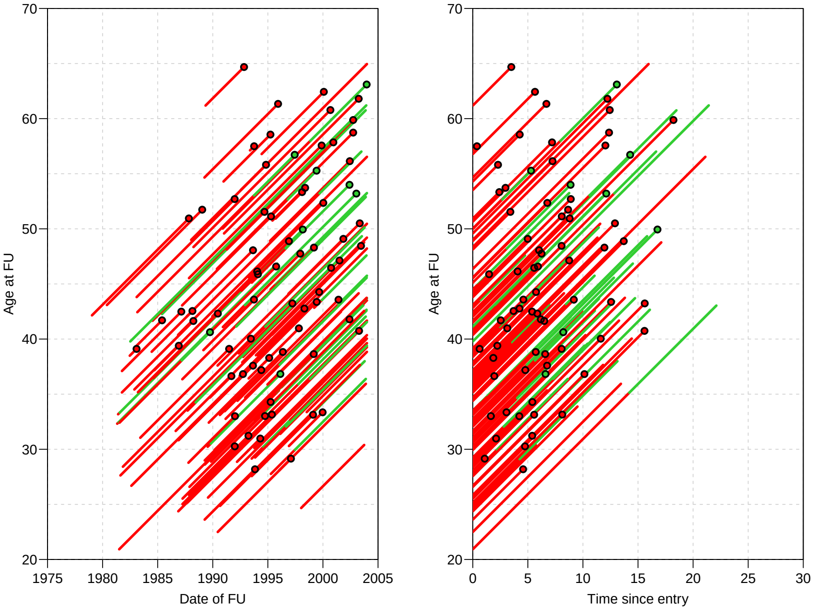
http://BendixCarstensen.com/SPE

Example: Renal failure data from Steno

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

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

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



70
60
50
40
30
20
Age at FU
1975
1980
1985
1990
1995
2000
2005
Date of FU
70
60
50
40
30
20
Age at FU
0
5
10
15
20
25
30
Time since entry

Features of the analysis

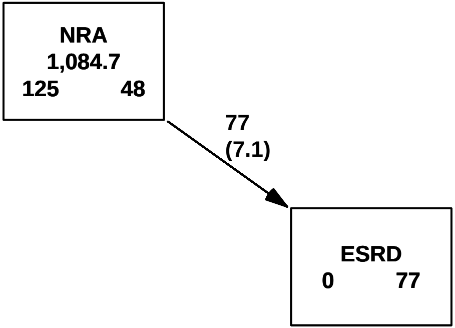
- Remission is included as a time-dependent variable.
- Age at entry is included as a fixed variable.

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

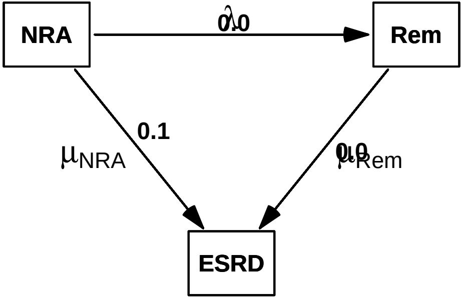
Note patient 26, 33 and 42 obtain remission.

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

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

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



Illness-death model



λ: remission rate.

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

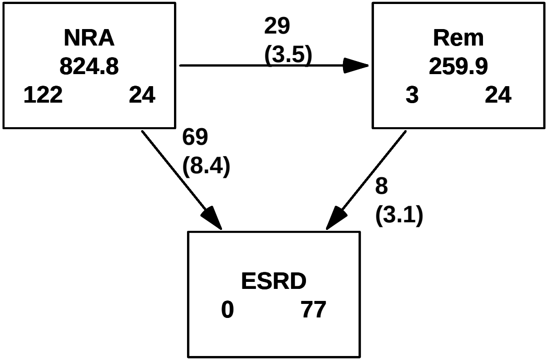
μ_{rem}: mortality/ESRD rate **after** remission.

Cutting follow-up at remission: cutLexis

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

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

Showing states and FU: boxes.Lexis

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

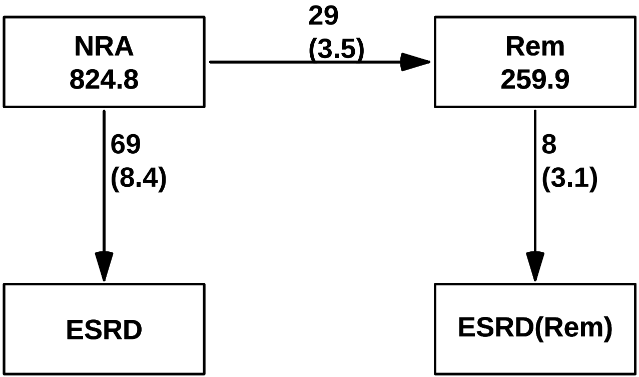


Cutting follow up at events: cutLexis

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

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

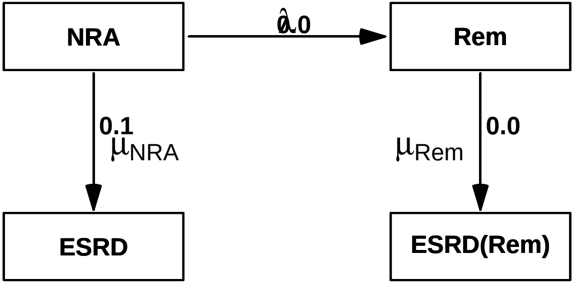
Showing states and FU: boxes.Lexis

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



Likelihood for a general MS-model

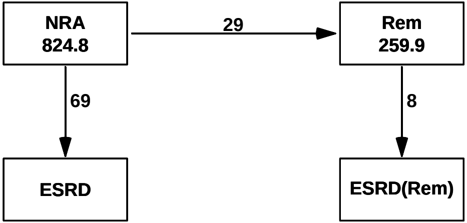
- Product of likelihoods for each transition
  — each one as for a survival model
- **Risk time** is the risk time in the "From" state
- **Events** are transitions to the "To" state
- All other transitions out of "From" are treated as **censorings**
- Possible to fit models separately for each transition



Cox-analysis with remission as time-dependent covariate:

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

Model for all transitions



Cox-model:

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

Poisson-model:

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

Calculating state probabilities

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

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

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

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

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

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

If μ_{rem} also depends on time since remission, then `c.mort.rem` should have an extra argument—technically very complicated

Prediction in multistate models: `simLexis` and renal failure

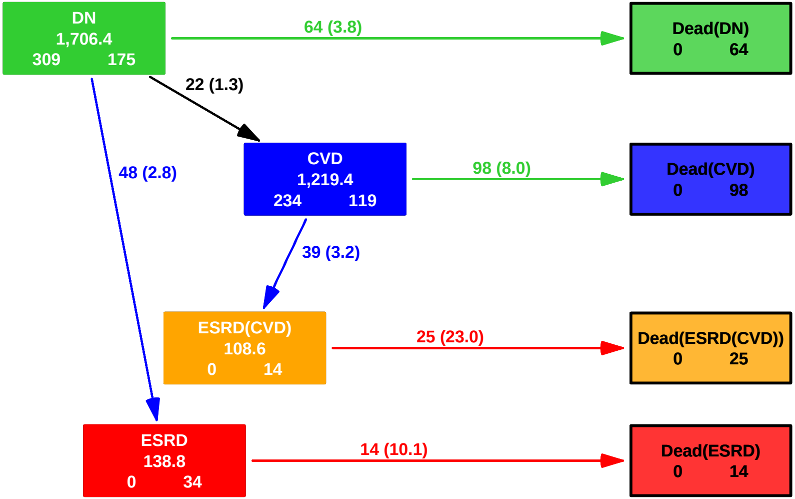
Bendix Carstensen

Multistate models

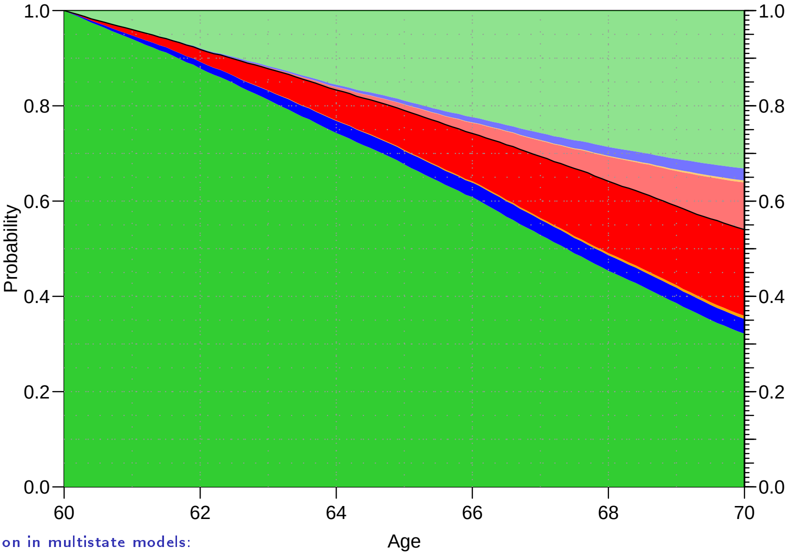
SPE, Tartu, Estonia,

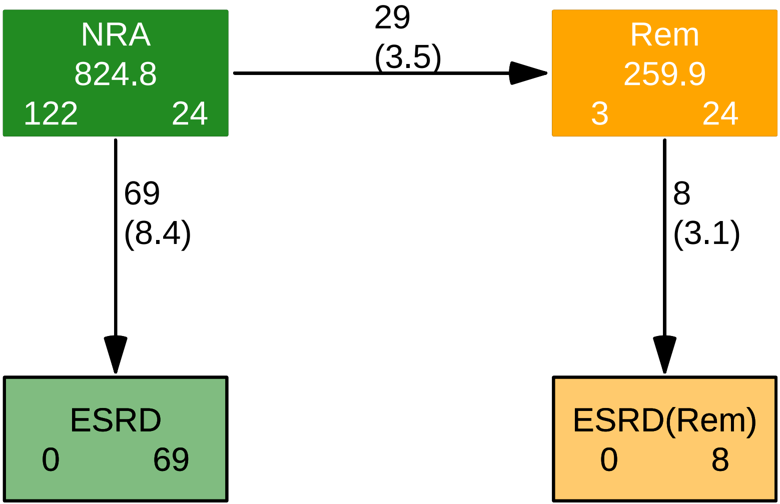
June 2023

A more complicated multistate model



A more complicated multistate model




NRA
824.8
122
24
29
(3.5)
Rem
259.9
3
24
69
(8.4)
8
(3.1)
ESRD
0
69
ESRD(Rem)
0
8

Modeling rates in a multistate model

Each transition modeled by a model for rates
(Cox-model, Poisson-model for split data, `glm` or `gam`)

Requires that follow-up is split in small intervals:

```
> sLc <- splitLexis(Lc, "tfi", breaks = seq(0, 30, 1/12))
> summary(sLc)
```

```
Transitions:
     To
From   NRA  Rem ESRD ESRD(Rem)  Records:  Events: Risk time:  Persons:
  NRA 9854   29   69         0      9952       98     824.77       122
  Rem    0 3139    0         8      3147        8     259.90        32
  Sum 9854 3168   69         8     13099      106    1084.67       125
```

Modeling rates in a multistate model

```
> # Rem-rate
> mr <- gam(cbind(lex.Xst == "Rem", lex.dur)
+           ~ s(tfi, k = 10) + sex,
+           family = poisreg,
+             data = subset(sLc, lex.Cst == "NRA"))
> # ESRD-rates
> mx <- gam(cbind(lex.Xst %in% c("ESRD", "ESRD(Ren)"), lex.dur)
+           ~ s(tfi, k = 10) + sex + I((doe - dob - 40) / 10) + I(lex.Cst == "Rem
+           family = poisreg,
+             data = subset(sLc, lex.Cst %in% c("NRA", "Rem")))
```

. . . using the Lexis properties

```
> # Remisson-rate
> mr <- gam.Lexis(sLc, from = "NRA", to = "Rem",
+                 formula = ~ s(tfi, k = 10) + sex)

mgcv::gam Poisson analysis of Lexis object sLc with log link:
Rates for the transition:
NRA->Rem

> # ESRD-rates
> mx <- gam.Lexis(sLc,
+                 formula = ~ s(tfi,k=10) + sex +
+                             I((doe - dob - 40) / 10) + I(lex.Cst == "Rem"))

mgcv::gam Poisson analysis of Lexis object sLc with log link:
Rates for transitions:
NRA->ESRD
Rem->ESRD(Rem)
```

Default is to model all transitions to absorbing states

State probabilities

How do we get from rates (Poisson-models) to probabilities:

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

Simulation of a survival time

- For a rate function $\lambda(t)$, $\Lambda(t) = \int_0^t \lambda(s)\,\mathrm{d}s$:

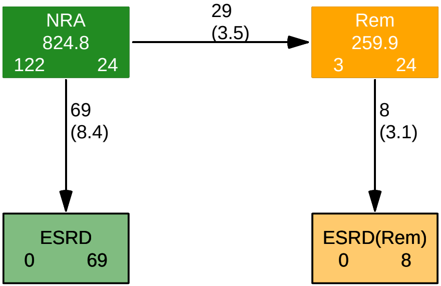
$$S(t) = \exp\bigl(-\Lambda(t)\bigr)$$

- Simulate a survival probability $u \in [0, 1]$:

$$u = S(t) \quad \Leftrightarrow \quad \Lambda(t) = -\log(u)$$

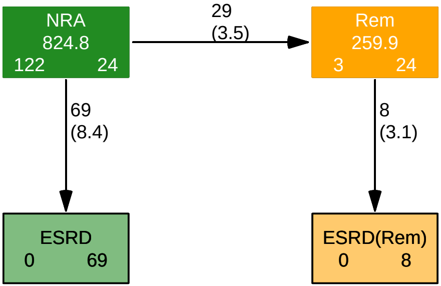
- Knowledge of $\Lambda(t)$ makes it easy to find a survival time — essentially just linear interpolation.

Simulation in a multistate model



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

Transition objects are glm/gam



```
> Tr <- list("NRA" = list("ESRD"      = mx,
+                         "Rem"       = mr),
+            "Rem" = list("ESRD(Rem)" = mx))
```

`simLexis`

Input required:

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

Output produced:

- A `Lexis` object with simulated event histories for may persons

Using simLexis I

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

Must have same structure as the one used for estimation — time scales must be initiated even if not used in models

```
> init <- sLc[NULL, c(timeScales(sLc), "lex.Cst")]
> init[1,"per"] <- 1994
> init[1,"age"] <- 40
> init[1,"tfi"] <- 0
> init[1,"lex.Cst"] <- "NRA"
> init[1,"sex"] <- "M"
> init[1,"dob"] <- 1954
> init[1,"doe"] <- 1994
> init
```

```
  per age tfi lex.Cst sex  dob  doe
 1994  40   0     NRA   M 1954 1994
```

Using simLexis II

```
> system.time(sim1 <- simLexis(Tr, init, N = 10000, t.range = 15))

   user  system elapsed 
  24.69    1.52   26.21 

> summary(sim1)

Transitions:
     To
From   NRA  Rem ESRD ESRD(Rem)  Records:  Events: Risk time:  Persons:
  NRA 969 1847 7184         0     10000     9031   72599.26     10000
  Rem   0 1224    0       623      1847      623   15654.67      1847
  Sum 969 3071 7184       623     11847     9654   88253.94     10000
```

This is a simulated cohort of 10,000 persons with NRA aged 40 in 1994.

Using a simulated Lexis object — pState I

```
> NN <- nState(sim1, at = seq(0, 15, 0.1),
+                   from = 0,
+             time.scale = "tfi")
> head(NN)
     State
when    NRA  Rem ESRD ESRD(Rem)
  0   10000    0    0         0
  0.1  9940   25   35         0
  0.2  9888   61   51         0
  0.3  9848   84   68         0
  0.4  9789  114   97         0
  0.5  9737  135  128         0
> nw1 <- pState(NN, perm = c(1, 2, 4, 3))
> head(nw1, 3)
```

Using a simulated Lexis object — pState II

```
      State
when        NRA       Rem ESRD(Rem) ESRD
  0      1.0000 1.0000    1.0000     1
  0.1 0.9940 0.9965    0.9965     1
  0.2 0.9888 0.9949    0.9949     1
```

```
> tail(nw1, 3)
```

```
       State
when         NRA          Rem ESRD(Rem) ESRD
  14.8 0.1020 0.22450000 0.2859000    1
  14.9 0.0992 0.22180000 0.2836000    1
  15    0.0000 0.06970925 0.1439466    1
```

Using a simulated Lexis object — pState III

```
> par(mar = c(3, 3, 0.5, 2), mgp = c(3, 1, 0) / 1.6, las = 1)
> plot(nw1, col = clr[c(2, 1, 4, 3)] )
> lines(as.numeric(rownames(nw1)), nw1[,2], lwd = 2)
> axis(side = 4, at = 0:5 / 5)
> axis(side = 4, at = 0:10 / 10, labels = NA)
> axis(side = 4, at = 0:20 / 20, labels = NA, tcl = -0.3)
> axis(side = 4, at = 0:100/100, labels = NA, tcl = -0.2)
```

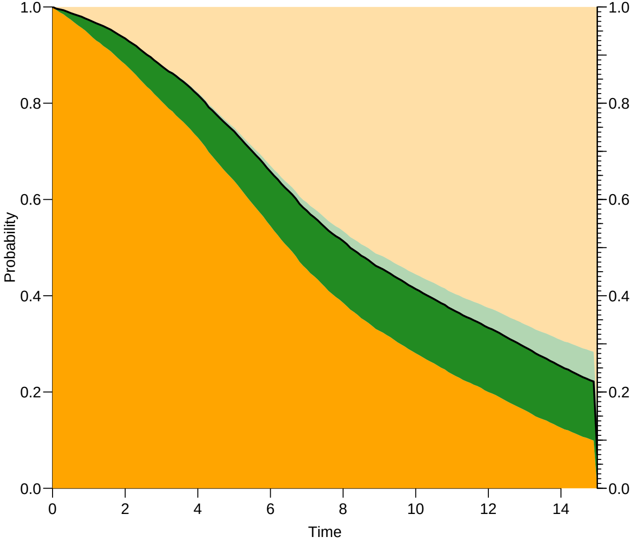
```
> nw2 <- pState( NN, perm = c(4,2,1,3) )
> head( nw2, 3 )

     State
when  ESRD(Rem)    Rem    NRA ESRD
  0           0 0.0000 1.0000    1
  0.1         0 0.0025 0.9965    1
  0.2         0 0.0061 0.9949    1

> tail( nw2, 3 )
```

Using a simulated Lexis object — pState IV

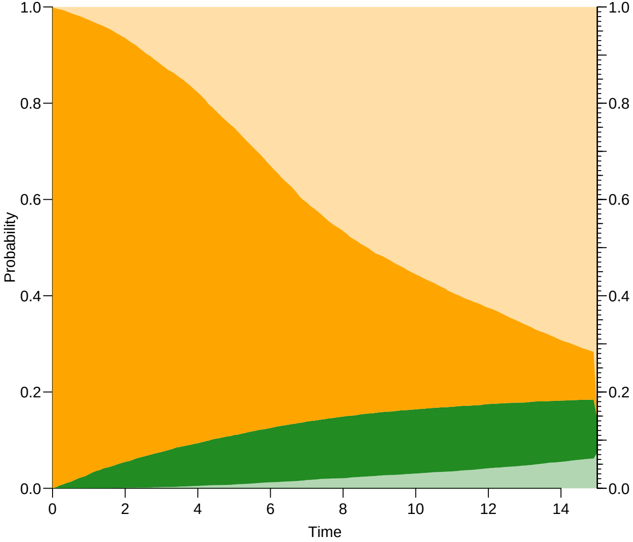
```
      State
when     ESRD(Rem)       Rem       NRA ESRD
  14.8 0.06140000 0.1839000 0.2859000    1
  14.9 0.06180000 0.1844000 0.2836000    1
  15   0.07423737 0.1439466 0.1439466    1
```

```
> par(mar = c(3, 3, 0.5, 2), mgp = c(3, 1, 0) / 1.6, las = 1)
> plot(nw2, col = clr[c(4, 1, 2, 3)])
> axis(side = 4, at = 0:5 / 5)
> axis(side = 4, at = 0:10 / 10, labels = NA)
> axis(side = 4, at = 0:20 / 20, labels = NA, tcl = -0.3)
> axis(side = 4, at = 0:100/100, labels = NA, tcl = -0.2)
```

Simulated probabilities



Simulated probabilities



How many persons should you simulate?

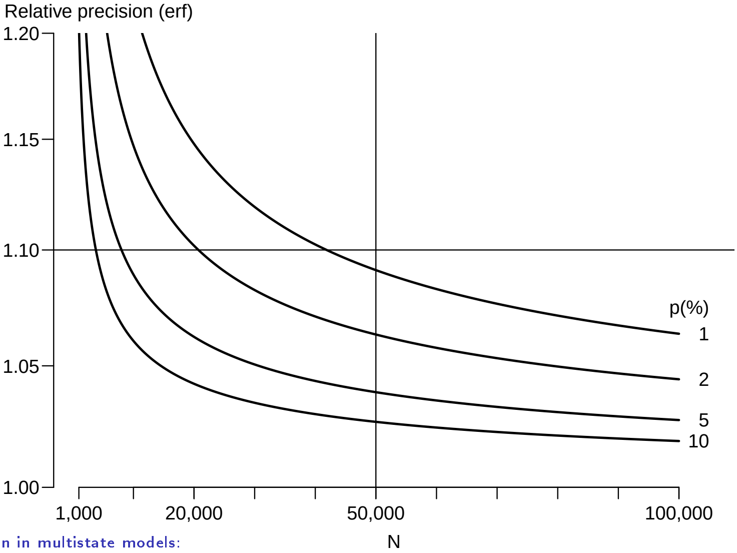
- All probabilities have the same denominator — the initial number of persons in the simulation, N, say.
- Thus, any probability will be of the form $p = x/N$
- For small probabilities we have that:

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

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

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

Precision of simulated probabilities



Multistate model overview

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