

Multistate models

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

University of Tartu,

June 2018

<http://BendixCarstensen.com/SPE>

Multistate models

Bendix Carstensen, Martyn Plummer

Multistate models

University of Tartu,

June 2018

<http://BendixCarstensen.com/SPE>

ms-Markov

Common assumptions in survival analysis

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

These assumptions are true for **death** and many **chronic diseases**.

Is the disease a dichotomy?

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

AIDS

Decline in CD4 count

Cancer

Pre-cancerous lesions

Type 2 Diabetes

Impaired glucose tolerance

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

A model for cervical cancer

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



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

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

Probabilities of **state** occupancy can be calculated.

When does the disease occur?

You may need a **clinical visit** to diagnose the disease:

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

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

Informative observation process?

Is the **reason** for the visit dependent on the **evolution** of disease?
Ignoring this may cause bias, like informative censoring.

Different reasons for follow-up visits:

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

Markov models for multistate diseases

The natural generalization of Poisson regression to multiple disease states:

- ▶ Probability of transition between states depends **only** on current state
- ▶ — this is the **Markov** property
- ▶ \Rightarrow transition rates are constant over time
- ▶ (time-fixed) covariates may influence transition rates
- ▶ the formal Markov property is **very** restrictive
- ▶ In clinical literature “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)
- ▶ Constrain some covariate effects to be the same, or zero.
- ▶ Not a trivial task — do we want e.g.
 - ▶ cause of death
 - ▶ disease status at death

Likelihood for multistate model

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

Purpose of multistate modeling

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

Special multistate models

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

Multistate models with Lexis

Bendix Carstensen

Multistate models

University of Tartu,

June 2018

<http://BendixCarstensen.com/SPE>

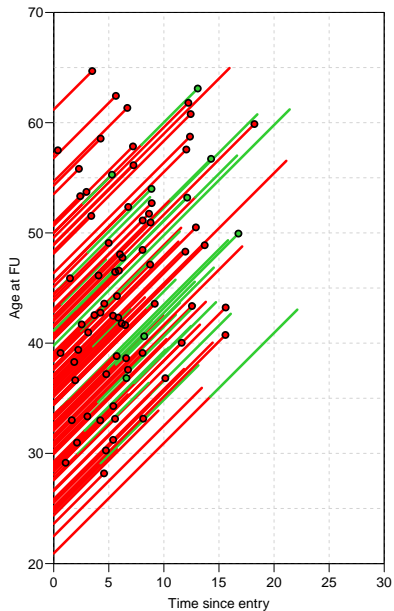
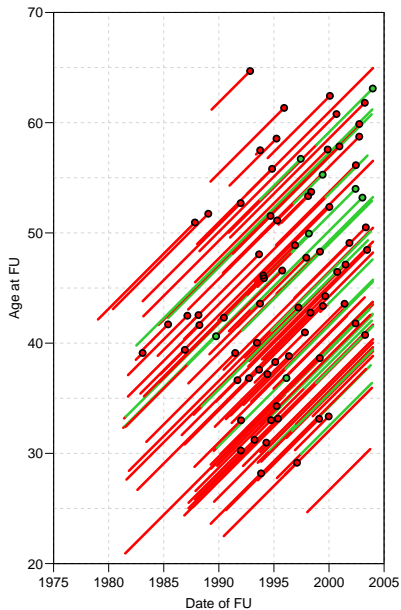
ms-Lexis

Example: Renal failure data from Steno

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

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

		Remission	
		Total	No
No. patients		125	93
No. events		77	69
Follow-up time (years)		1084.7	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	p
Fixed covariates:			
Sex (F vs. M):		0.92	0.740
Age at NRA (per 10 years):		1.42	0.011
Time-dependent covariate:			
Obtained remission:		0.28	0.001



Features of the analysis

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

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

Note patient 26, 33 and 42 obtain remission.

```

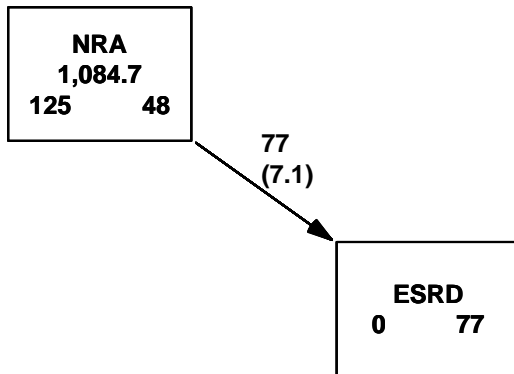
> Lr <- Lexis( entry = list( per=doe,
+                             age=doe-dob,
+                             tfi=0 ),
+             exit = list( per=dox ),
+             exit.status = event>0,
+             states = c("NRA","ESRD"),
+             data = renal )
> summary( Lr )

```

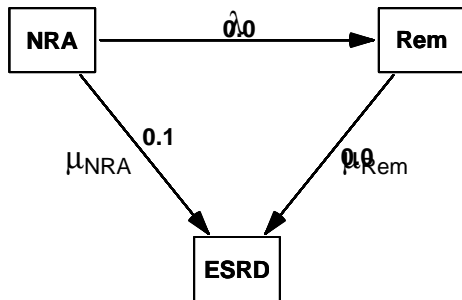
Transitions:

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

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



Illness-death model



λ : remission rate.

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

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

Cutting follow-up at remission: cutLexis

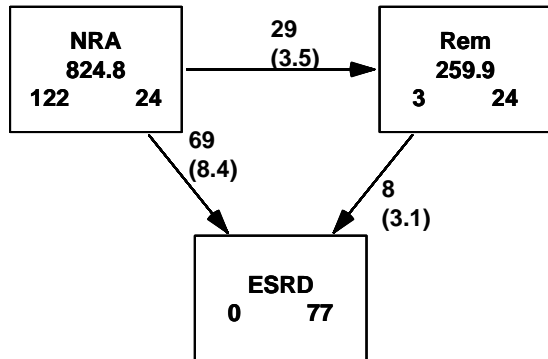
```
> Lc <- cutLexis( Lr, cut=Lr$dor,  
+               timescale="per",  
+               new.state="Rem",  
+               precursor.states="NRA" )  
> summary( Lc )
```

Transitions:

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

Showing states and FU: boxes.Lexis

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



Splitting states: cutLexis

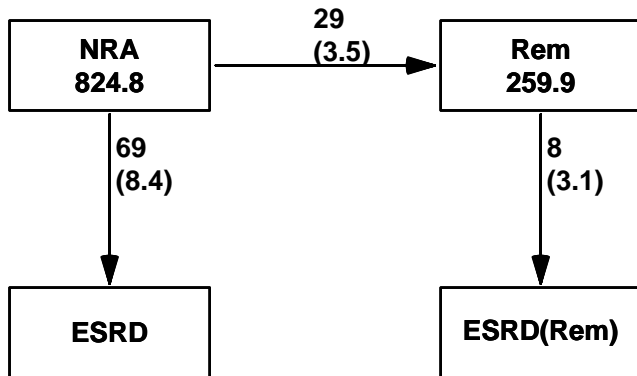
```
> Lc <- cutLexis( Lr, cut=Lr$dor,  
+               timescale="per",  
+               new.state="Rem",  
+               precursor.states="NRA",  
+               split.states=TRUE )  
> summary( Lc )
```

Transitions:

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

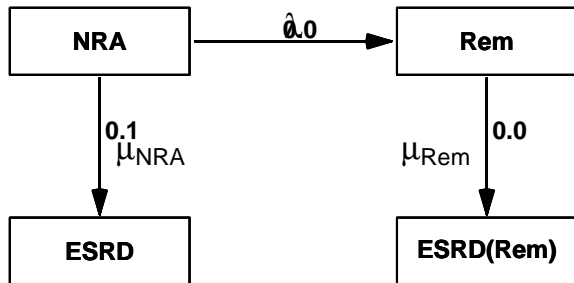
Showing states and FU: boxes.Lexis

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



Likelihood for a general MS-model

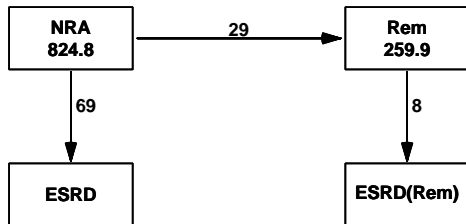
- ▶ Product of likelihoods for each transition
— each one as for a survival model
- ▶ **Risk time** is the risk time in the “From” state
- ▶ **Events** are transitions to the “To” state
- ▶ All other transitions out of “From” are treated as **censorings**
- ▶ Possible to fit models separately for each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



Cox-analysis with remission as time-dependent covariate:

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

Model for all transitions



Cox-model:

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

Poisson-model:

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

Calculus of probabilities

P {Remission **before** time t }

$$= \int_0^t \lambda(u) \exp \left(- \int_0^u \lambda(s) + \mu_{\text{NRA}} \, ds \right) \, du$$

P {Being in remission **at** time t }

$$= \int_0^t \lambda(u) \exp \left(- \int_0^u \lambda(s) + \mu_{\text{NRA}}(s) \, ds \right) \times \\ \exp \left(- \int_u^t \mu_{\text{rem}}(s) \, ds \right) \, du$$

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

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

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

integr(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( integr(t,1:t) )
```

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

Calculation of integrals

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

- ▶ Availability of closed-form formulae for the probabilities in terms of the transition rates
- ▶ Transition rates are assumed to be continuous functions of time
- ▶ Transition rates can be calculated at any point of time...
- ▶ This will allow simple calculation of the integrals from the closed-form expressions.

Semi-Markov models

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

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

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

Prediction in multistate models:

simLexis

Bendix Carstensen

Multistate models

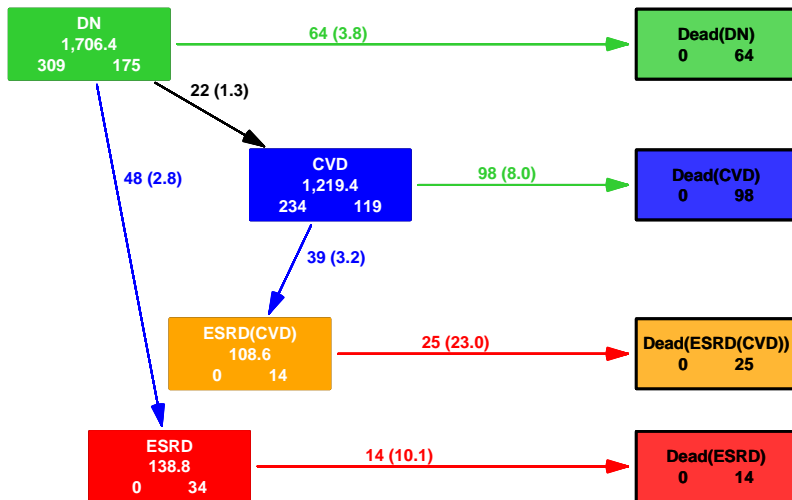
University of Tartu,

June 2018

<http://BendixCarstensen.com/SPE>

simLexis

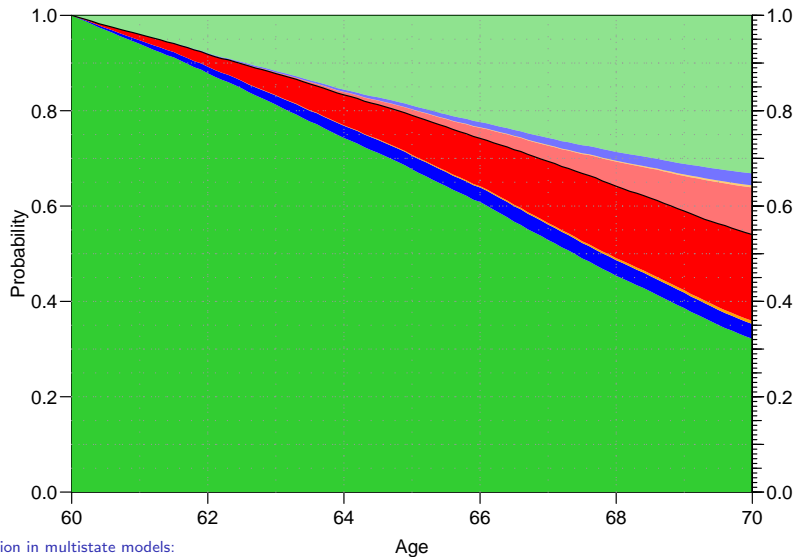
A more complicated multistate model



Prediction in multistate models:

simLexis (simLexis)

A more complicated multistate model



Prediction in multistate models:

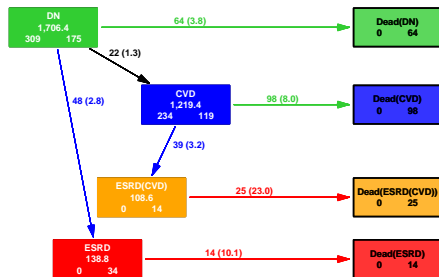
`simLexis (simLexis)`

State probabilities

How do we get from rates to probabilities:

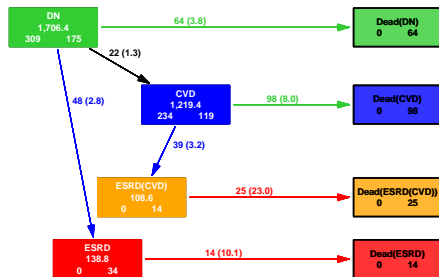
- ▶ 1: Analytical calculations:
 - ▶ immensely complicated formulae
 - ▶ computationally fast (once implemented)
 - ▶ difficult to generalize
- ▶ 2: Simulation of persons' histories
 - ▶ conceptually simple
 - ▶ computationally not quite simple
 - ▶ easy to generalize
 - ▶ hard to get confidence intervals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

Simulation in a multistate model



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

Transition object are glms



```

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

Prediction in multistate models:

simLexis (simLexis)

simLexis

Input required:

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

Output produced:

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

Using simLexis

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

Must have same structure as the one used for estimation:

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

init[1,"sex"] <- "M"
init[1,"age"] <- 60
...

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

Output from simLexis

```
> summary( sim1 )
```

Transitions:

To

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

Transitions:

To

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

Using a simulated Lexis object — pState

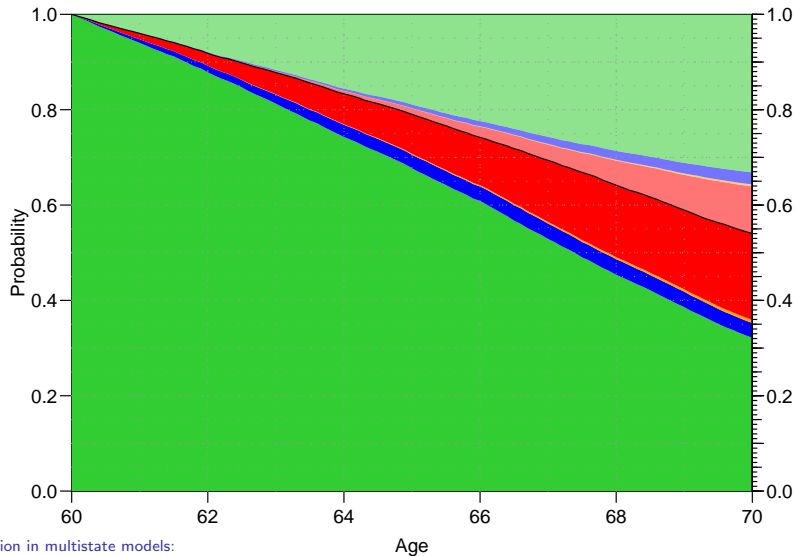
```
nw1 <- pState( nState( sim1,  
                      at = seq(0,15,0.1),  
                      from = 60,  
                      time.scale = "age" ),  
              perm = c(1:4,7:5,8) ) )
```

```
head( pState )
```

when	DN	CVD	ES(CVD)	ES	Dead(ES)	Dead(ES(CVD))	Dead(CVD)	Dead(DN)
60	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
60.1	0.9983	0.9986	0.9986	0.9997	0.9997	0.9997	0.9997	1
60.2	0.9954	0.9964	0.9964	0.9990	0.9990	0.9990	0.9990	1
60.3	0.9933	0.9947	0.9947	0.9981	0.9981	0.9981	0.9982	1
60.4	0.9912	0.9929	0.9929	0.9973	0.9973	0.9973	0.9974	1
60.5	0.9894	0.9913	0.9913	0.9964	0.9964	0.9964	0.9965	1

```
plot( pState )
```

Simulated probabilities



Prediction in multistate models:

`simLexis (simLexis)`

How many persons should you simulate?

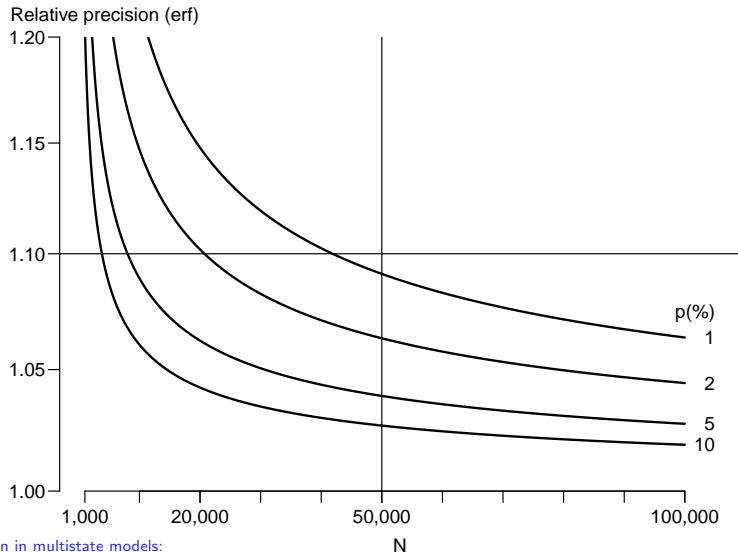
- ▶ All probabilities have the same denominator — the initial number of persons in the simulation, N , say.
- ▶ Thus, any probability will be of the form $p = x/N$
- ▶ For small probabilities we have that:

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

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

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

Precision of simulated probabilities



Prediction in multistate models:

`simLexis (simLexis)`

Multistate model overview

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

Your turn: “Renal complications”