Multistate models

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

SPE, Tartu, Estonia,

August 2019

http://BendixCarstensen.com/SPE

From /home/bendix/teach/SPE/git/SPE/lectures/multistate/multistate.tex Tuesday 27th August, 2019, 10:33

Multistate models

Bendix Carstensen, Martyn Plummer

Multistate models

SPE, Tartu, Estonia,

August 2019

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 model for cervical cancer

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

- Aim of the modeling the transition rates between states, is to be able predict how population moves between states:
- state occupancy probabilities
- visit probability
- length of stay (sojourn time)

Markov models for multistate diseases

Generalization of Poisson regression to multiple disease states:

- Transition rates between states depends **only** on current state (and possibly time since start) — the **Markov** property
- (time-fixed) covariates may influence transition rates
- the formal Markov property is **very** restrictive
- **semi**-Markov: rates depend on time since entry to current state
- In the clinical literature, the term "Markov model" is often used about any type of multistate model
- ... and the Markov property is handy in probability theory

Components of a multistate (Markov) model

- Define the (disease) states
- Define which transitions between states that occur
- 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

Components of multistate data

Times should be recorded as **dates**

- birth date
- entry date
- entry **state**
- exit date
- death date
- state entry date**s** — for all states
- ...some states may be revisited

From this each person's trajectory through states can be constructed

Likelihood for multistate model

- The likelihood of the observed data (sojourn times and transitions) depend on the (models for) the transition rates.
- Assume transition rates are constant in small time intervals
- $\Rightarrow$ each interval contributes terms to the log-likelihood:
  - one for each person (p) at risk in state s in the interval
  - ...for each possible transition $(s \rightarrow v)$
  - each term is a Poisson log-likelihood contribution:

$$d_{psv} \log(\lambda_{psv}) - \lambda_{psv} y_{ps}, \qquad \text{where:}$$

    - λ_{psv} rate for person p in state s going to state v
    - d_{psv} did person p in state s go to state v at end of interval
    - y_{ps} how long did person p spend in state s (how long is the interval)
- Total log-lik is sum of terms over persons and transitions

Practical multistate modeling

- Total log-lik is sum of terms over persons (p) and transitions ($s \rightarrow v$)
- — components **not** independent, but the total likelihood is a product; hence of the same form as the likelihood of independent Poisson variates
- practical analysis is just analysis of each transition rate separately
- as long as no two rates **out** of the **same** state are modeled we can use subsets of `Lexis` objects

Multistate models with `Lexis`

Bendix Carstensen

Multistate models

SPE, Tartu, Estonia,

August 2019

Example: Renal failure data from Steno

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

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

	Total	Remission: Yes	Remission: No
No. patients	125	32	93
No. events	77	8	69
Follow-up time (years)	1084.7	259.9	824.8

Cox-model:
Timescale: Time since nephrotic range albuminuria (NRA)
Entry: 2.5 years of GFR-measurements after NRA
Outcome: ESRD or Death

Estimates:	RR	95% c.i.	p
Fixed covariates:			
Sex (F vs. M):	0.92	(0.53,1.57)	0.740
Age at NRA (per 10 years):	1.42	(1.08,1.87)	0.011
Time-dependent covariate:			
Obtained remission:	0.28	(0.13,0.59)	0.001

Features of the analysis

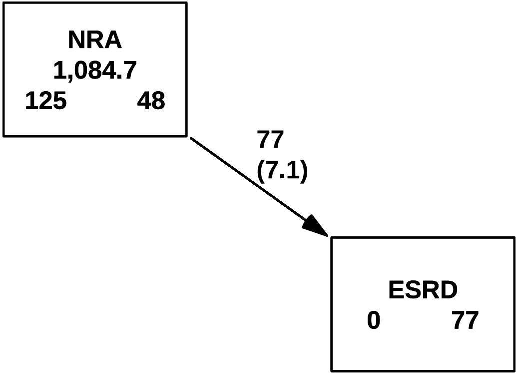
- Remission is included as a time-dependent variable.
- Age at entry is included as a fixed variable.

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

Note patient 26, 33 and 42 obtain remission.

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

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

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

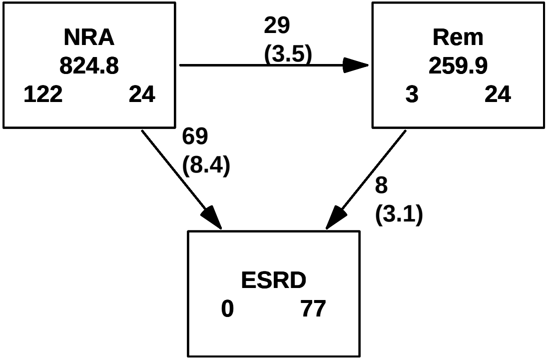


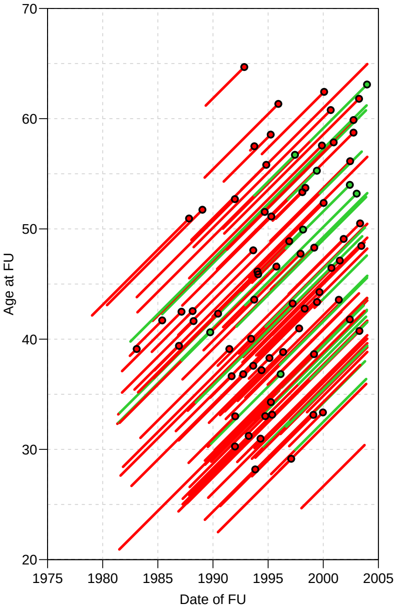
Cutting follow-up at remission: `cutLexis`

```
> 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
```

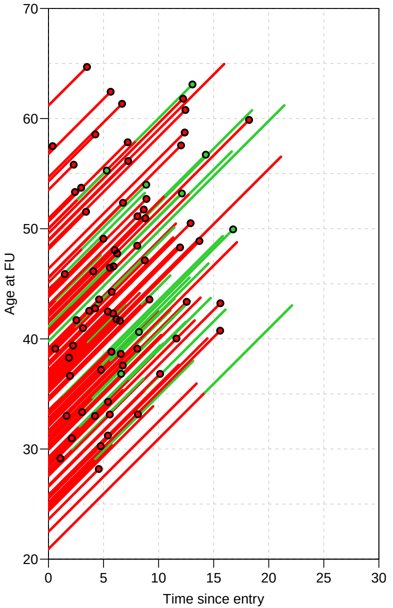
Showing states and FU: `boxes.Lexis`

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





70
60
50
40
30
20
Age at FU
1975
1980
1985
1990
1995
2000
2005
Date of FU



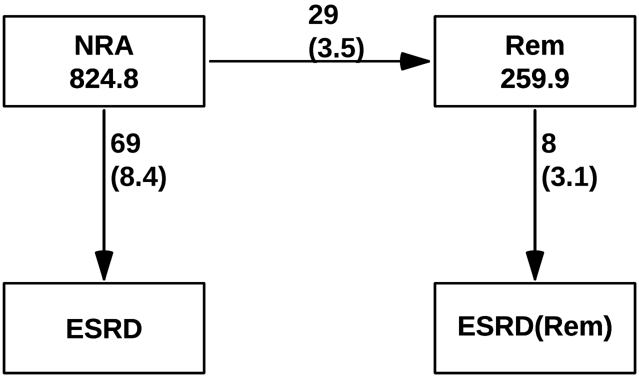
70
60
50
40
30
20
Age at FU
0
5
10
15
20
25
30
Time since entry

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

Prediction in multistate models: `simLexis` and renal failure

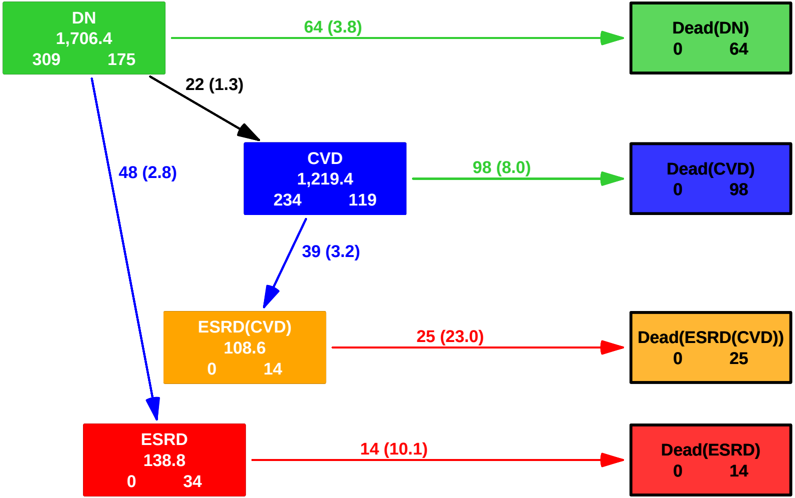
Bendix Carstensen

Multistate models

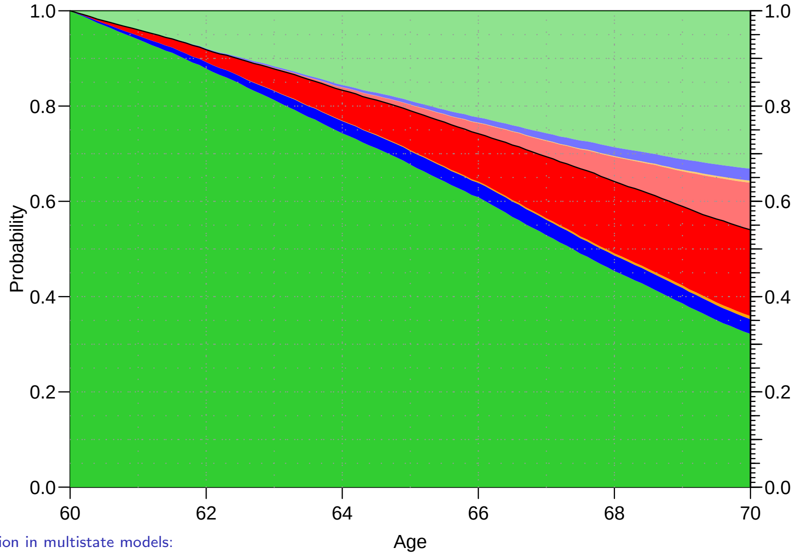
SPE, Tartu, Estonia,

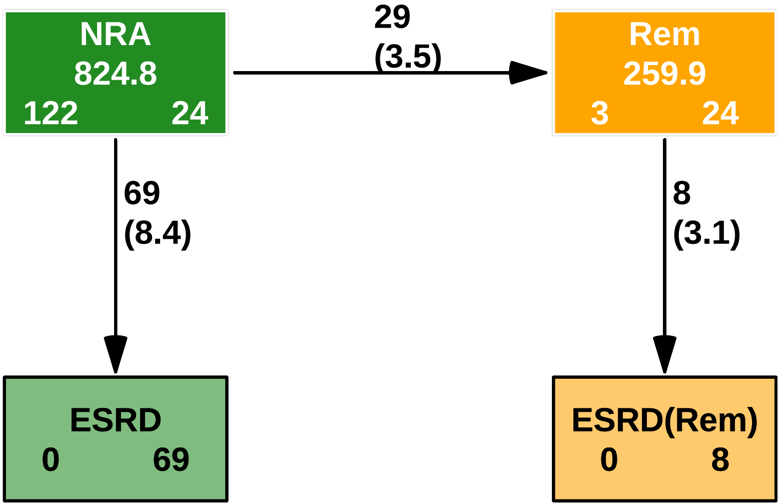
August 2019

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 in a multistate model

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

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

```
> # Rem-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

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 $\lambda(t)$, $\Lambda(t) = \int_0^t \lambda(s)\,\mathrm{d}s$:

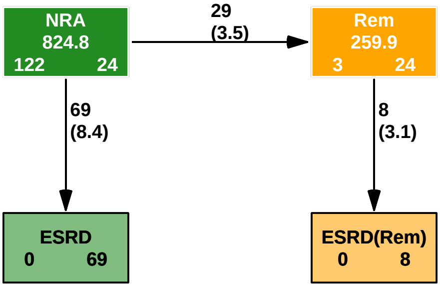
$$S(t) = \exp\big(-\Lambda(t)\big)$$

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

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

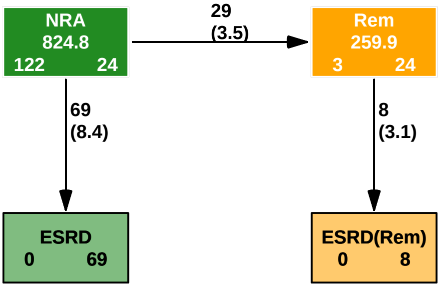
- Knowledge of $\Lambda(t)$ makes it easy to find a survival time — essentially just linear interpolation.

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 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` I

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 — 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
```

Using `simLexis` II

```
   per age tfi lex.Cst sex  dob  doe
1 1994  40   0     NRA   M 1954 1994

> system.time(
+ sim1 <- simLexis( Tr, init, N=10000 ) )

   user  system elapsed
 49.608  52.279  36.550

> summary(sim1)

Transitions:
     To
From   NRA  Rem ESRD ESRD(Rem)  Records:  Events: Risk time:  Persons:
  NRA 314 1870 7816         0     10000      9686   74864.10     10000
  Rem   0  940    0       930      1870       930   20593.77      1870
  Sum 314 2810 7816       930     11870     10616   95457.87     10000
```

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    39    21         0
  0.2  9891    60    49         0
  0.3  9834    83    83         0
  0.4  9793   100   107         0
  0.5  9731   134   135         0

> nw1 <- pState( NN, perm = c(1,2,4,3) )
> head( nw1, 3 )
```

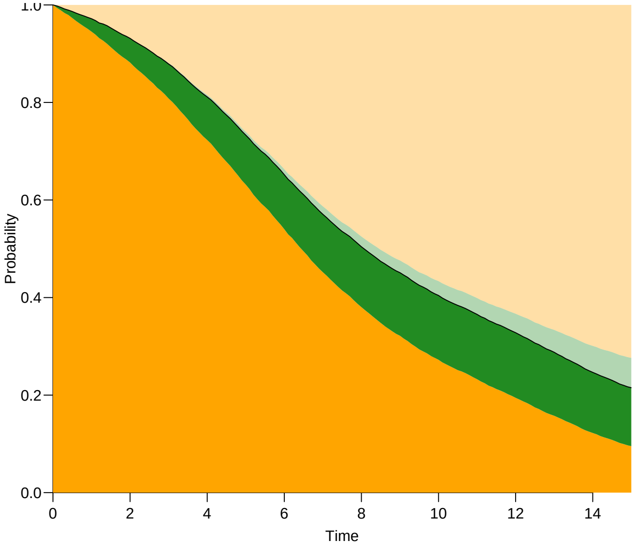
Using a simulated Lexis object — pState II

```
     State
when       NRA    Rem ESRD(Rem) ESRD
  0   1.0000 1.0000    1.0000    1
  0.1 0.9940 0.9979    0.9979    1
  0.2 0.9891 0.9951    0.9951    1

> tail( nw1, 3 )

      State
when        NRA    Rem ESRD(Rem) ESRD
  14.8 0.0996 0.2200    0.2802    1
  14.9 0.0971 0.2171    0.2779    1
  15   0.0953 0.2151    0.2763    1

> par( mar=c(3,3,0.1,0.1), 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] )
```

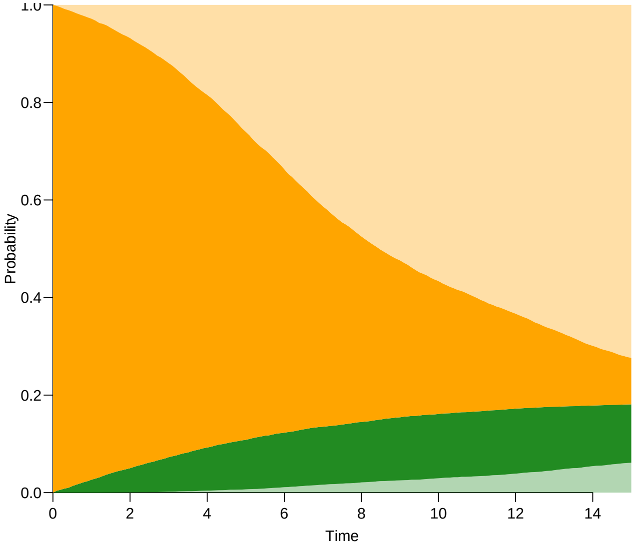
Using a simulated Lexis object — pState III

```
> 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 0.0039 0.9979    1
  0.2          0 0.0060 0.9951    1

> tail( nw2, 3 )

       State
when    ESRD(Rem)     Rem    NRA ESRD
  14.8     0.0602 0.1806 0.2802    1
  14.9     0.0608 0.1808 0.2779    1
  15       0.0612 0.1810 0.2763    1

> par( mar=c(3,3,0.1,0.1), mgp=c(3,1,0)/1.6, las=1 )
> plot( nw2, col=clr[c(4,1,2,3)] )
```

Simulated probabilities



Simulated probabilities



How many persons should you simulate?

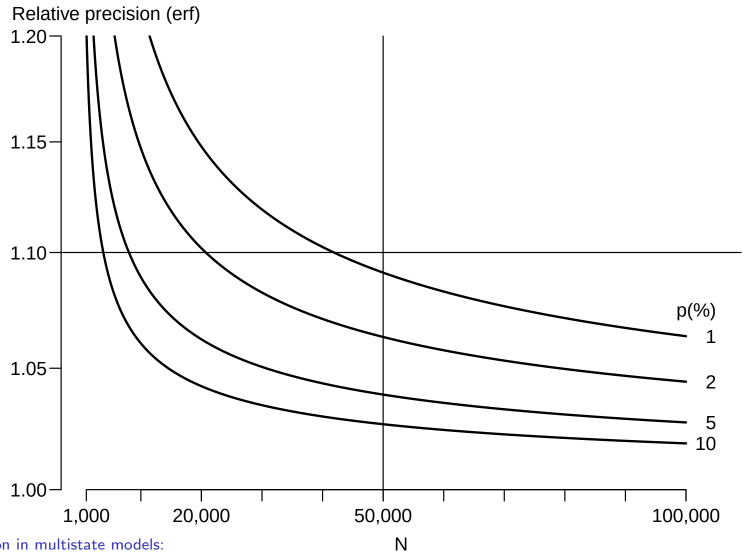
- All probabilities have the same denominator — the initial number of persons in the simulation, N, say.
- Thus, any probability will be of the form $p = x/N$
- For small probabilities we have that:

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

- 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

- 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)