Suvival workshop, part 1

Terry Therneau

Career

- 1975, BA St Olaf college
- 1976-79 Programmer, Mayo Clinic
- 1979-83 PhD student, Stanford
- 1983-85 Asst Professor, U of Rochester
- 1985-23 Mayo Clinic

Computing

- Languages: Fortran, Basic, Focal, APL, PL/1, C, awk, lex, yacc, (python)
- Assembler: IBM 11/30, PDP 11, VAX, IBM 360
- Statistical: BMDP, SAS, S, Splus, R, (minitab, SPSS, matlab)
- OS: DMS (11/30), DEC RSTS, DEC Tops20, JCL (cards), Wylbur, CMS, Unix (Bell, Berkeley, SUN, Linux)
- code: Panvalet, SCCS, rcs, cvs, svn, mercurial, git

Cox model

- 1977: use shared Fortran code
- 1978: create SAS proc coxregr, presented at SUGI 79, added to SAS Supplemental proceedures, meet Frank Harrell
- 1984: first S code, to investigate residuals
- 1987(?): survival becomes part of Splus, code on statlib
- ? move to R
- 9/2010 first commit to current Mercurial library

Cox model

$$\begin{aligned}\lambda(t;z) &= \exp(\beta_0(t) + \beta_1 x_1 + \beta_2 x_2 + \ldots) \\ &= e^{\beta_0(t)} e^{\eta} \\ &= \lambda_0(t) e^{\eta}\end{aligned}$$

- Lottery model
  - at each event time there is a drawing for the winner
  - each obs has $r_i = \exp(\eta)$ tickets
  - P(subject i wins) $= r_i / \sum_{atrisk} r_j$

Additive models

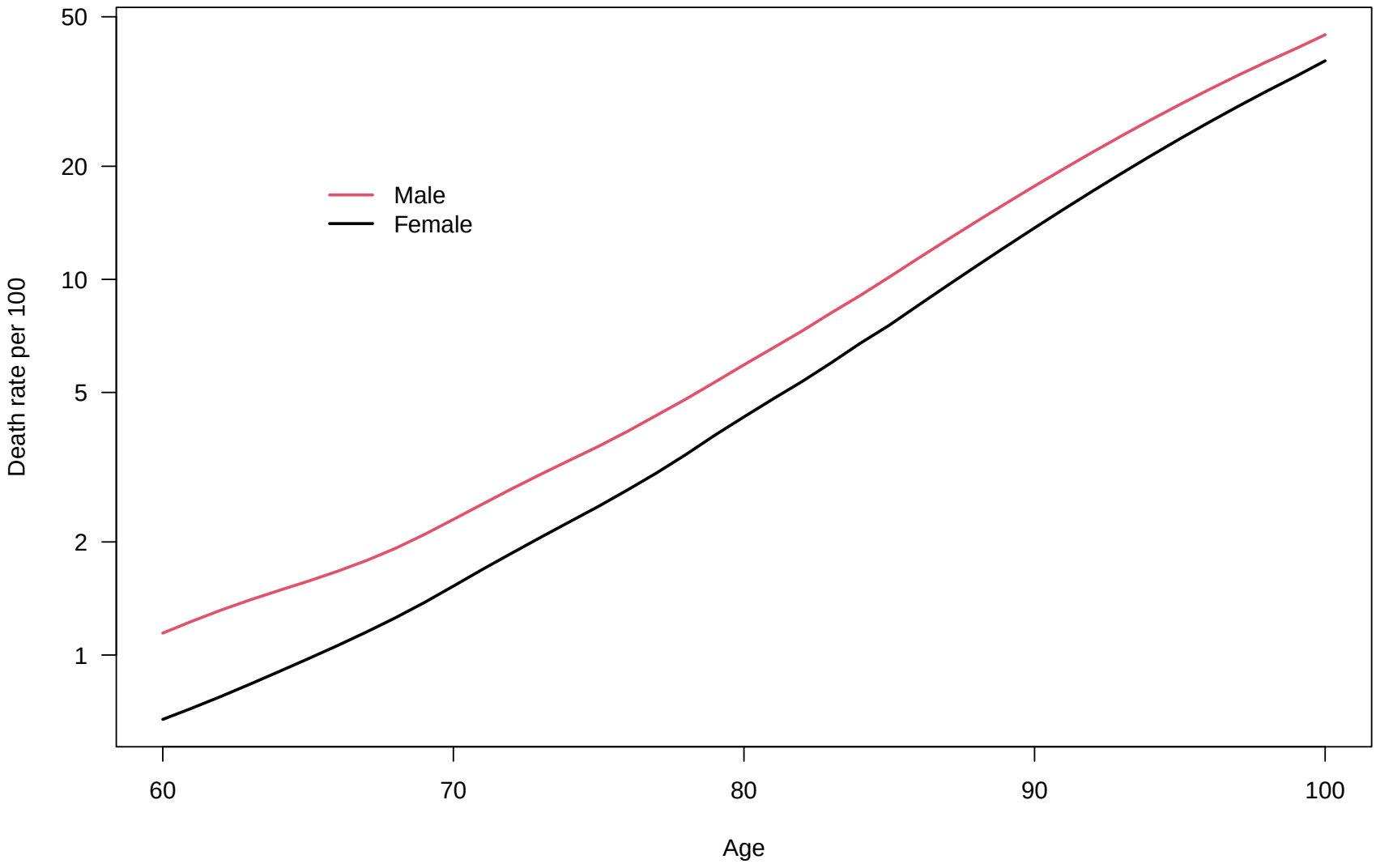
- The three most popular models in statistics
  - Linear: $E(y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots$
  - GLM: $E(y) = g\left(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots\right)$
  - Cox: $\lambda(t) = g\left(\beta_0(t) + \beta_1 x_1 + \beta_2 x_2 + \ldots\right)$
- Why? Simplicity.
  - If `x1= apoepos`, then β_1 is *THE* effect of APOE, independent of any other variables in the model.
  - Statisticians like this.
  - Investigators really like this (a single p-value)
- Generalized additive models will replace one of the βx terms with $s(x)$, but retain the separability.

Successful statistical models

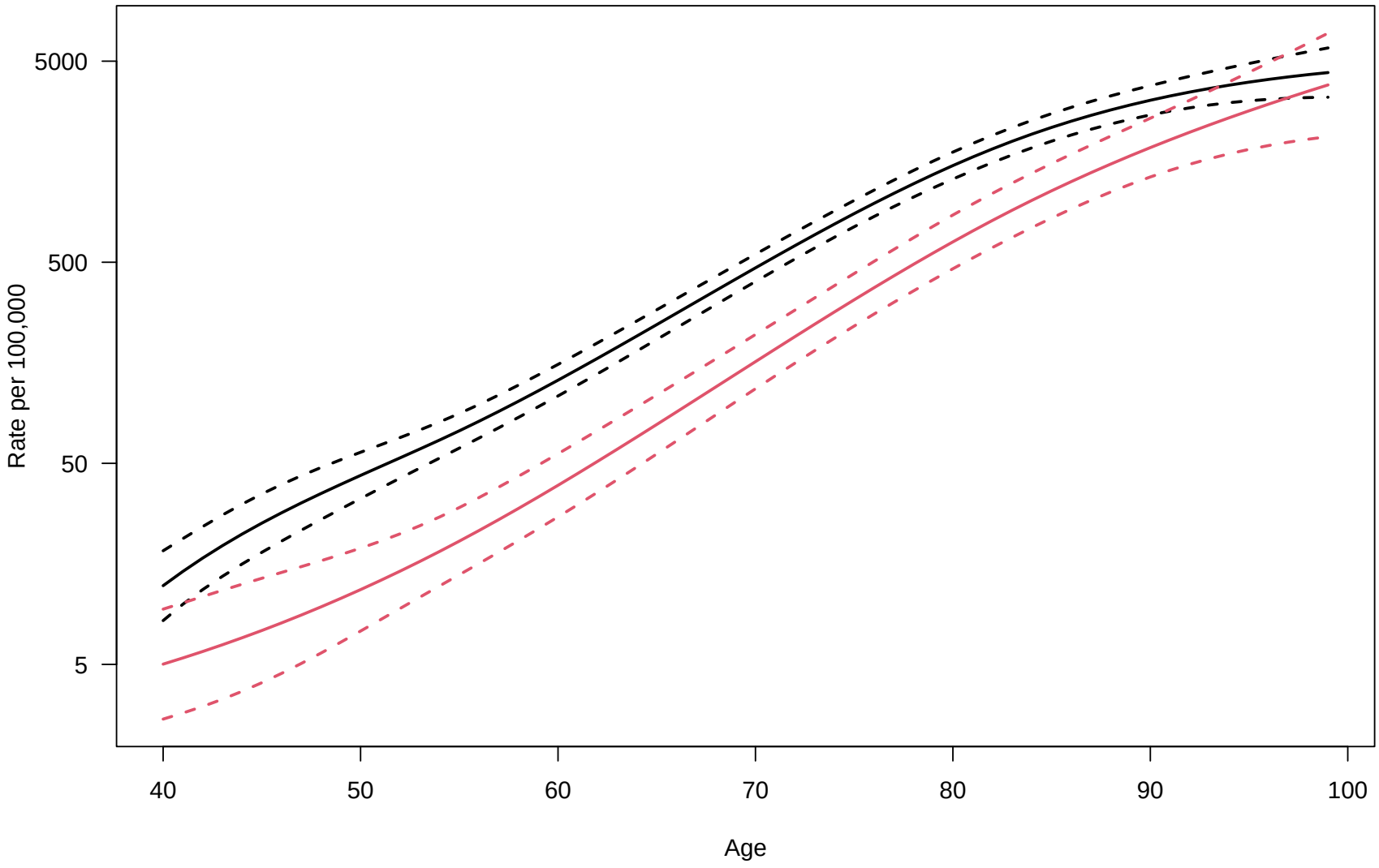
1. Simplicity: in the sense described above, leading to simple explanations for the effect of key predictors.
2. Statistical validity: the model must describe the data adequately. "All models are wrong. The practical question is whether a model is wrong enough to not be useful.'' George Box
3. Numerical stability: the code to fit a model does not require hand-holding or fiddling with tuning parameters: it just runs.
4. Speed

- The transform g gets chosen to fit criteria 3; if it helps with criteria 2 that is mostly luck. (It nearly always impedes interpretability).
- $\exp(\eta)$:
  - no negative values (dead coming back to life)
  - mulitiplicative hazards: sometimes okay, sometimes not

US Death Rates



Hip fracture rates



Assumptions

- Proportional hazards
  - Very strong assumption
  - Surprisingly often, it is 'close enough'
  - Always check it, however.
- Additivity
  - Strong assumption
  - Never perfectly true, maybe okay (but we love it so much)
  - Always check
  - adding '*' is not sufficient
- Linearity
  - Moderately strong, depending on the range of x
  - Use a spline, and look
  - IMHO, automatic df choices overfit
- No naked p values allowed!

PH failure

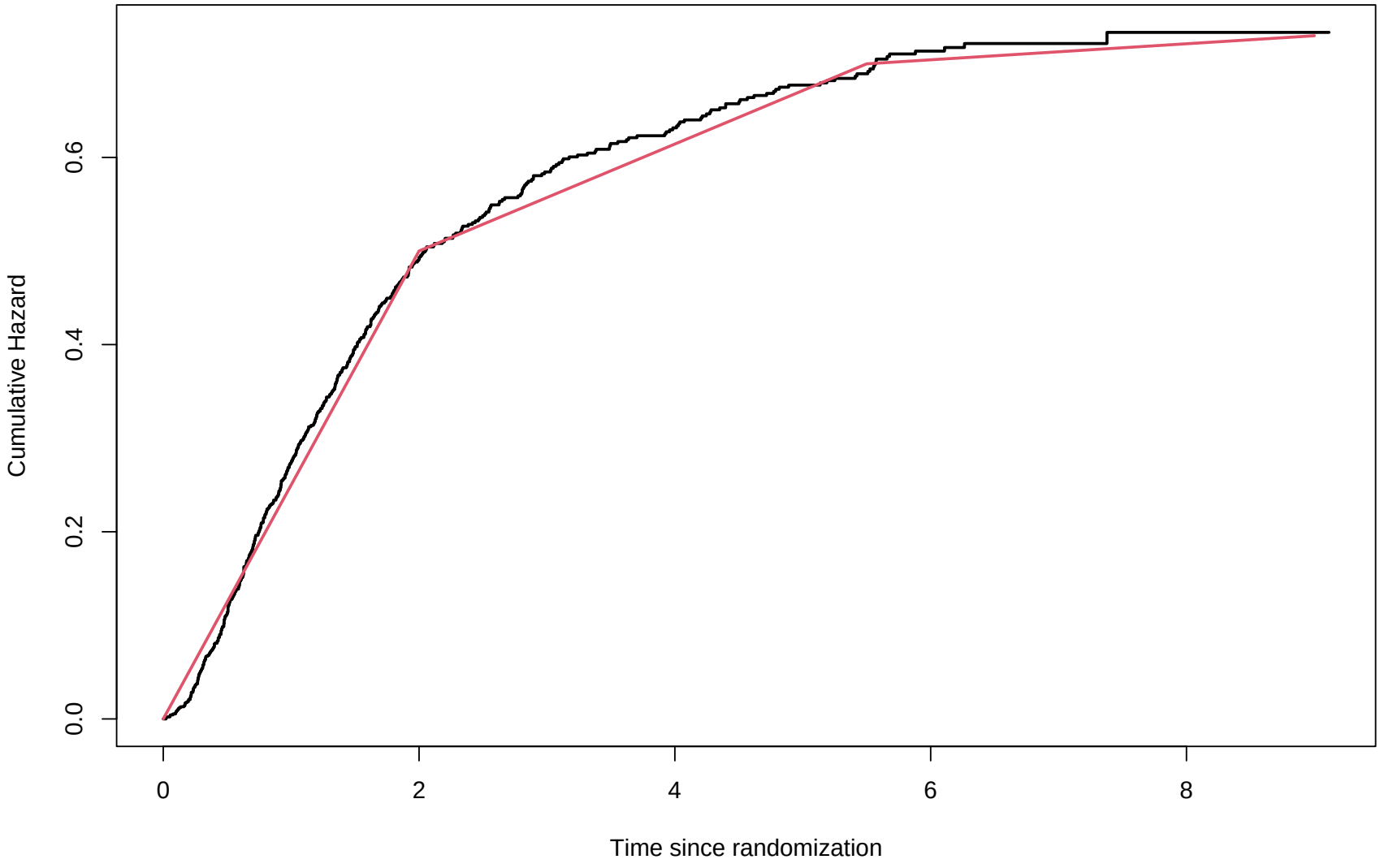
$$\lambda(t; x) = \lambda_1(t)e^{x\beta} + \lambda_2(t)e^{x\gamma} + \ldots$$

- $\lambda_1 =$ acute disease process
- $\lambda_2 =$ population mortality

Computation

- first derivative $= \sum(x_i - \bar{x}) = m'X$
- very quadratic
- simple starting estimate

Poisson approximation



```
cdata <- subset(colon, etype==1)
cdata$years <- cdata$time/365.35
csurv <- survfit(Surv(years, status) ~1, data=cdata)
plot(csurv, fun="cumhaz", conf.int=FALSE, lwd=2,
     xlab="Time since randomization", ylab="Cumulative Haza
lines(c(0, 2, 5.5, 9), c(0, .5, .7, .73), col=2, lwd=2)

cdata2 <- survSplit(Surv(years, status) ~., data=cdata, cut
    episode="interval")
cfit1 <- coxph(Surv(years, status) ~ rx + extent + node4, c
cfit2 <- glm(status ~ rx + extent + node4 + factor(interval
                  offset(log(time-tstart)), family=poisson,
round(summary(cfit1)$coef[,1:3], 2)

           coef exp(coef) se(coef)
rxLev     -0.03      0.97     0.11
rxLev+5FU -0.52      0.60     0.12
extent     0.54      1.71     0.11
node4      0.85      2.33     0.10

round(summary(cfit2)$coef[,1:2], 2)
```

Other models

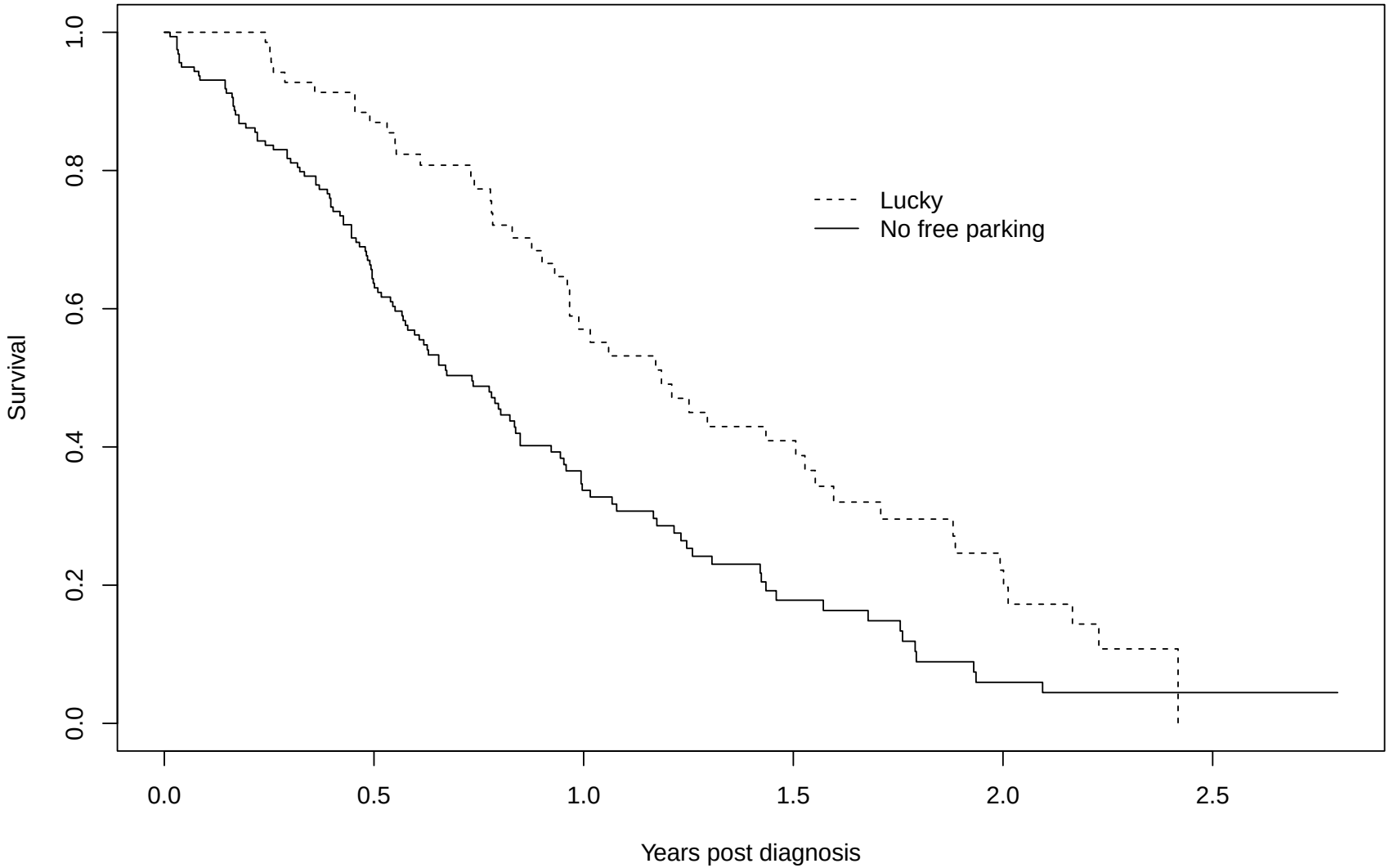
- Proportional odds
  - $P(y < k; x) = g(\beta_0(k) + \beta_1 x_1 + \beta_2 x_2 + \ldots)$
  - I am dubious
  - Essentially the same is assumed when a logistic regression fit is applied to population with different prevalence.
- Fine-Gray model
  - $p_k(t; x) = g(\beta_0(t) + \beta_1 x_1 + \beta_2 x_2 + \ldots)$
  - Rarely if ever true

Counting process notation

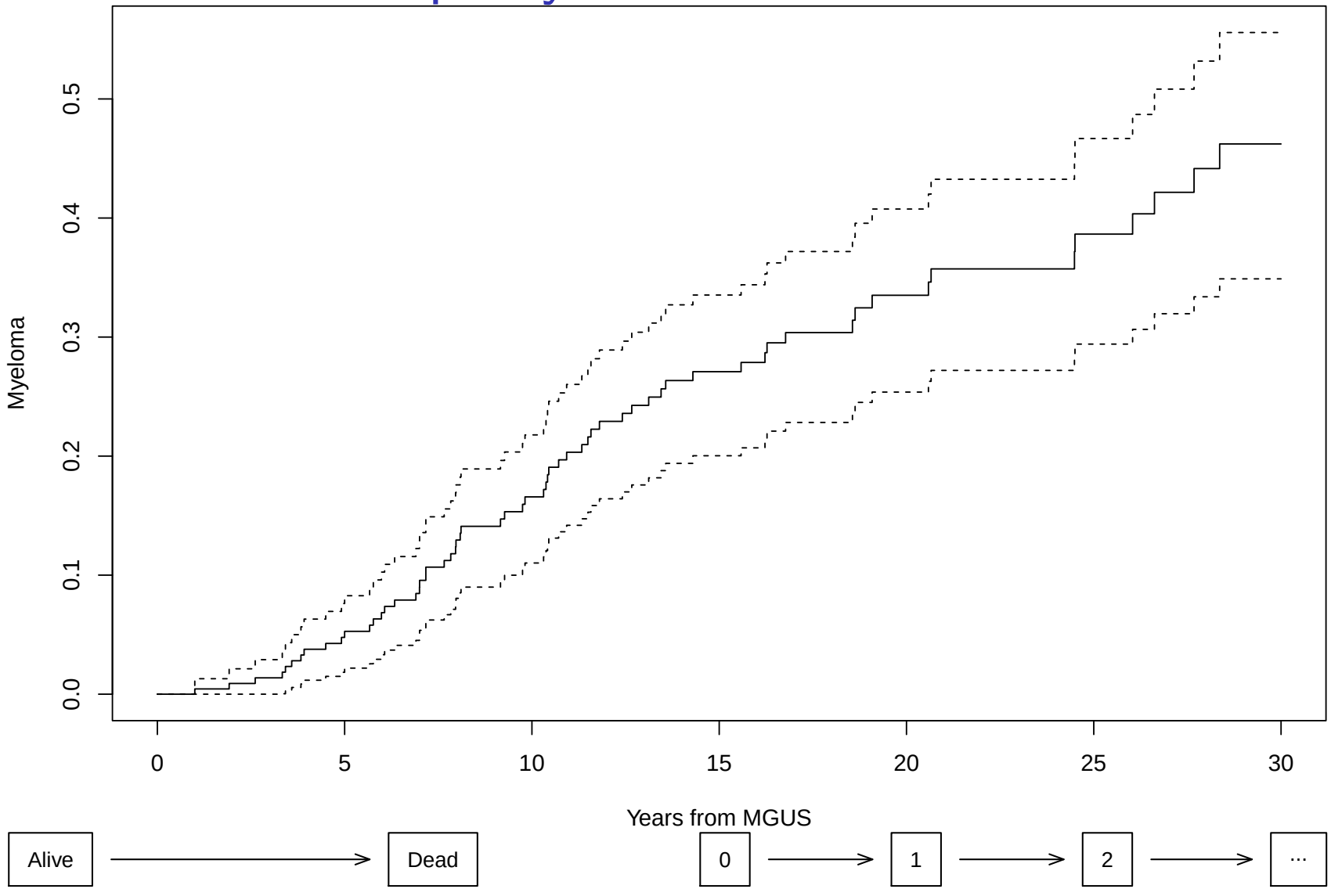
- $N_i(t)$ = number of events, up to time t, for subject i
- $N_{ijk}(t)$ = transtions from state j to state k
- $Y_{ij}(t) = 1$ if subject i is in state j and at risk
- $X(t)$ = covarates at time t
- Key: N is left continuous, Y and X are right continuous
- predictable process

Immortal time bias

- any of N, Y, or X depend on the future
- most common error is in X
  - responders vs non-responders
  - Redmond paper: total dose received, average dose received
  - many others
- Y, who is at risk
  - nested case-control, excluding future events from the risk set at time t
- N, what is an event
  - diabetes = two visits at least 6 months apart that satisfy criteria
  - incidence of diabetes defined as the first one


Survival
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0
1.5
2.0
2.5
Years post diagnosis
Lucky
No free parking

Monoclonal Gammopathy



Key Concepts

- Each arrow is a transition
  - Hazard rate
  - If Markov, each can be estimated independently
  - Looks like a Cox model
- Each box is a state
  - Estimation must be done all at once
  - $p_k(t)$ = prob(in state k at time t) depends on *all* the hazards
- Hazards can be done one at a time, absolute risk must be done all at once

Absolute risk

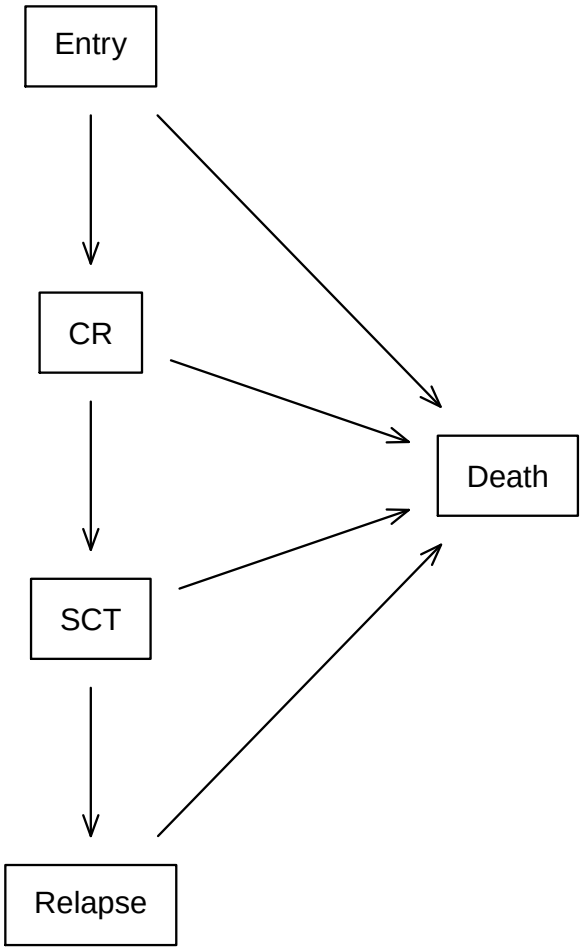
- $p(t)$ = probability in state
- E(N(t))$ = expected number of visits to each state
  - closely related to lifetime risk
- Sojourn time = E(time in each state)
  - restricted mean time in state (RMTS)
  - for alive/dead: restricted mean survival time (RMST)
- Duration in state = expected time per visit
- Estimands

Tools

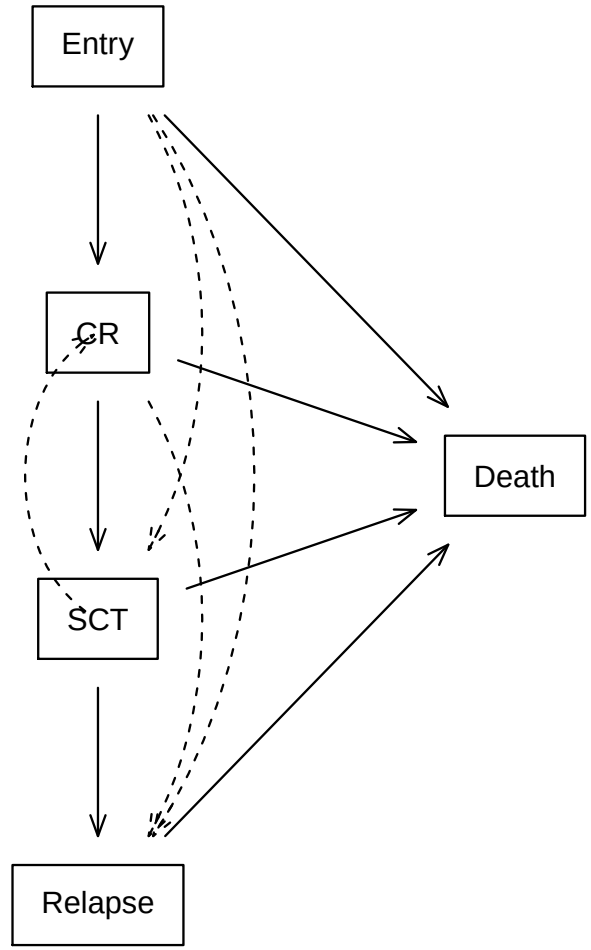
- Build the data set, and check it
- Start simple
  - total endpoints of each type
  - transtion rates = number/(person years at risk)
  - LOOK at the data
- Non-parametric
  - Aalen-Johansen estimate
- Multi-state models

Myeloid data

Ideal model

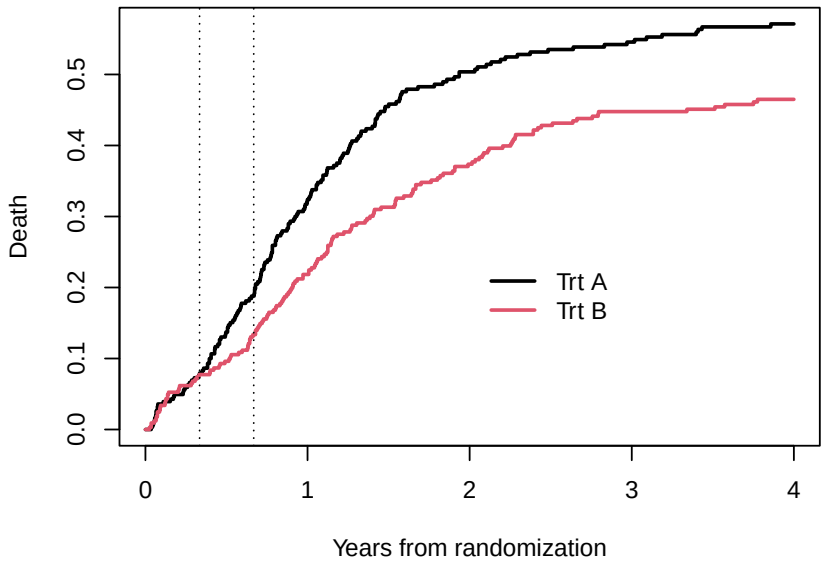


Reality

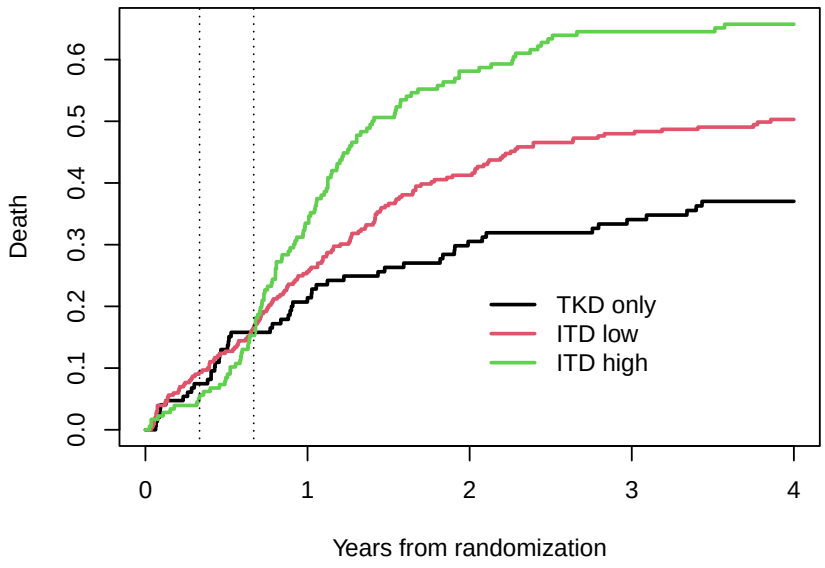


```
load('data/myeloid.rda')
myeloid[1:5,]
```

	id	trt	sex	flt3	futime	death	txtime	crtime	rltime
1	1	B	f	ITD >=.7	235	1	NA	44	113
2	2	A	m	ITD <.7	286	1	200	NA	NA
3	3	A	f	TKD	1983	0	NA	38	NA
4	4	B	f	TKD	2137	0	245	25	NA
5	5	B	f	ITD >=.7	326	1	112	56	200


Death
0.0
0.1
0.2
0.3
0.4
0.5
0
1
2
3
4
Years from randomization
Trt A
Trt B


Death
0.0
0.1
0.2
0.3
0.4
0.5
0
1
2
3
4
Years from randomization
Female
Male


Death
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
1
2
3
4
Years from randomization
TKD only
ITD low
ITD high

Multistate data

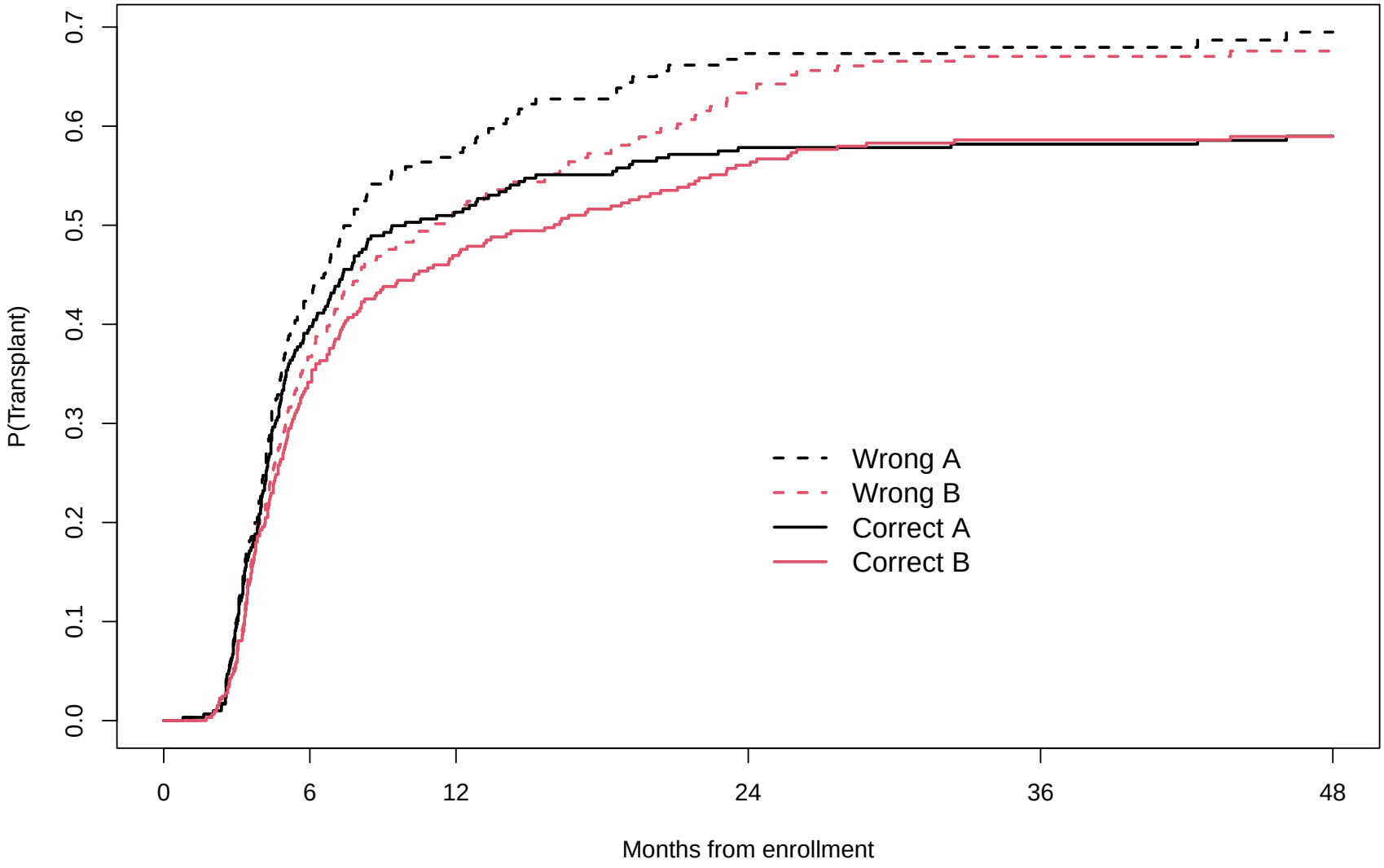
- Rows with id, time1, time2, state, covariates, strata, cstate
- Over the interval (time1, time2] these are the covariates, strata, current state
- at time2 there is a transition to a new state 'state'
  - a factor variable whose first level is 'no change occured' (censoring)
  - labels can be anything you wish
- Looks a lot like time-dependent covariate data
- The set of rows for a subject describes a feasable path
  - can't be two places at once (overlapping intervals)
  - have to be somewhere (disconnected intervals)
  - time in any state is > 0
  - no teleporting
- combat immortal time bias, easier code

```
mdata <- tmerge(myeloid[,1:4], myeloid,  id=id,  death= eve
                sct = event(txtime), cr = event(crtime),
                relapse = event(rltime))
temp <- with(mdata, cr + 2*sct  + 4*relapse + 8*death)
table(temp)

temp
  0   1   2   3   4   8
325 453 363   1 226 320
```

```
temp
  0   1   2   4   8
325 454 364 226 320
```

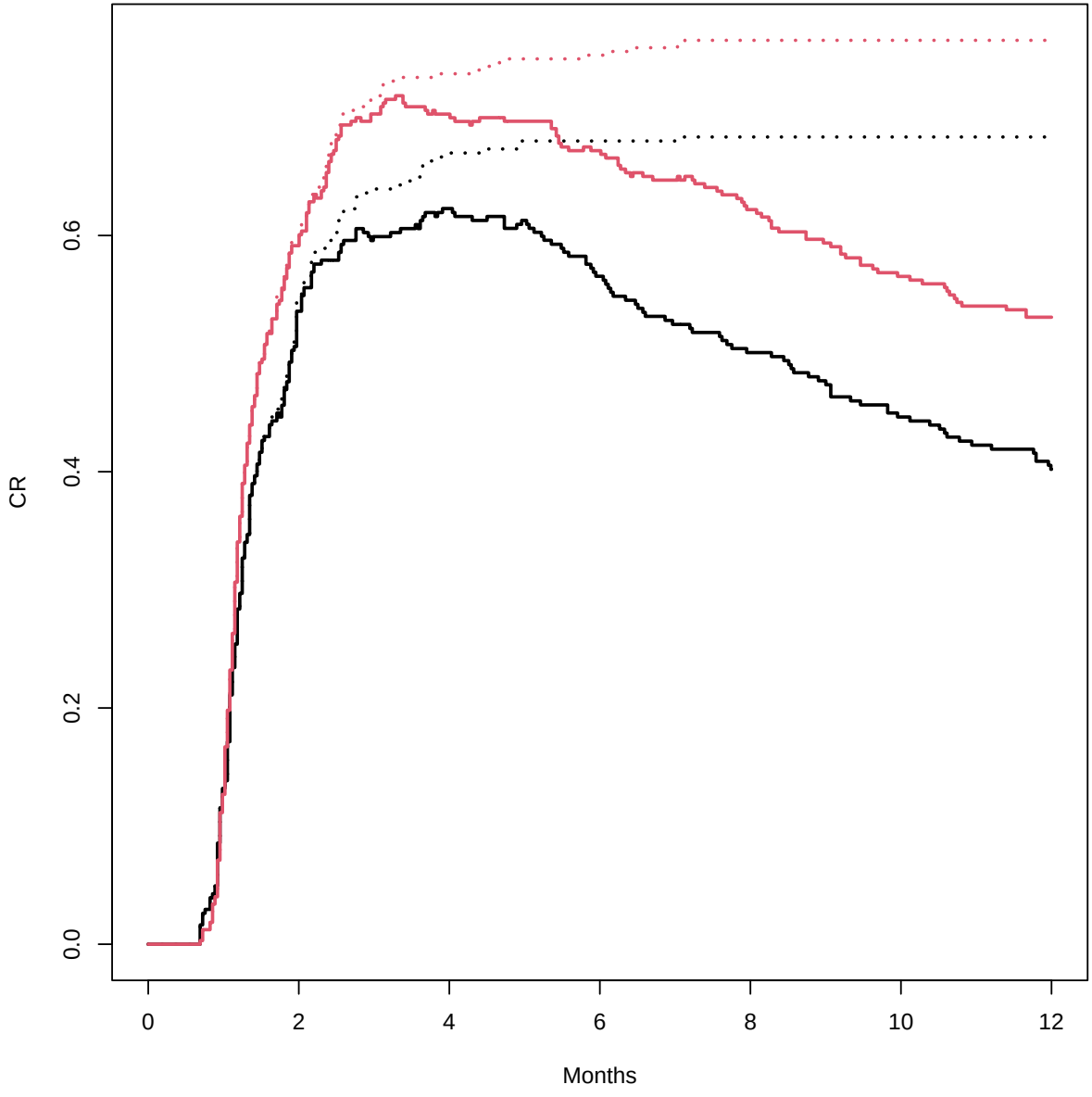
	id	trt	tstart	tstop	event	priorcr	priortx
1	1	B	0	44	CR	0	0
2	1	B	44	113	relapse	1	0
3	1	B	113	235	death	1	0
4	2	A	0	200	SCT	0	0
5	2	A	200	286	death	0	1
6	3	A	0	38	CR	0	0
7	3	A	38	1983	none	1	0
8	4	B	0	25	CR	0	0

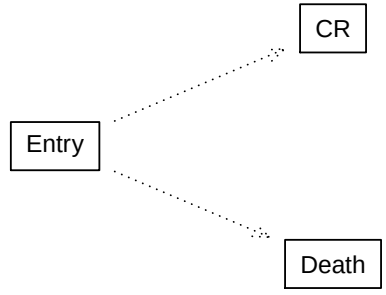
```
survcheck(Surv(tstart, tstop, event) ~1, mdata, id=id)

Call:
survcheck(formula = Surv(tstart, tstop, event) ~ 1, data =
    id = id)

Unique identifiers       Observations        Transitions
               646               1689               1364

Transitions table:
         to
from       CR SCT relapse death (censored)
  (s0)    443 106      13    55         29
  CR        0 159     168    17        110
  SCT      11   0      45   149        158
  relapse   0  99       0    99         28
  death     0   0       0     0          0

Number of subjects with 0, 1, ... transitions to each state
         count
```

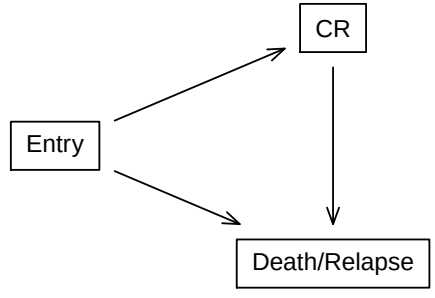

P(Transplant)
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
6
12
24
36
48
Months from enrollment
Wrong A
Wrong B
Correct A
Correct B

```
tfit <- coxph(Surv(tstart, tstop, txstat) ~ trt + flt3, mda
print(tfit, digits=2)

Call:
coxph(formula = Surv(tstart, tstop, txstat) ~ trt + flt3, c
    id = id)


1:2                coef exp(coef) se(coef) robust se    z
  trtB           -0.14      0.87     0.11       0.11 -1.4 0.17
  flt3ITD <.7     0.44      1.55     0.14       0.14  3.1 0.00
  flt3ITD >=.7    0.49      1.63     0.15       0.15  3.2 0.00


1:3                coef exp(coef) se(coef) robust se    z
  trtB           -0.39      0.68     0.17       0.17 -2.3  0.0
  flt3ITD <.7     0.41      1.51     0.23       0.24  1.7  0.0
  flt3ITD >=.7    0.85      2.35     0.24       0.24  3.5 4e-0

 States:  1= (s0), 2= SCT, 3= death
```

Duration of CR







```
print(crsurv, rmean=48, digits=2)

Call: survfit(formula = Surv(tstart, tstop, cr2) ~ trt, dat
    id = id, influence = TRUE)

                          n nevent rmean se(rmean)*
trt=A, (s0)             693      0   7.1        0.78
trt=B, (s0)             739      0   5.6        0.65
trt=A, CR               693    206  16.3        1.13
trt=B, CR               739    248  21.2        1.12
trt=A, Death/Relapse    693    194  24.6        1.13
trt=B, Death/Relapse    739    184  21.1        1.07
   *restricted mean time in state (max time = 48 )
```