Survival Analysis

Randy Johnson

3/9/2017

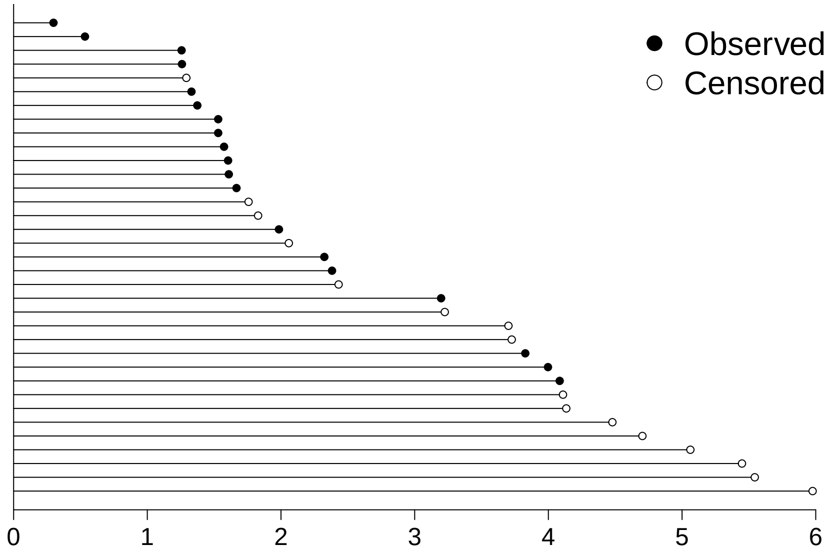
Setup

```r
library(car)
library(cowplot)
library(survival)
library(tidyverse)
theme_set(theme_classic() +
          theme(axis.line.x = element_line(color = 'black'),
                axis.line.y = element_line(color = 'black'),
                text = element_text(size = 20)))

# colorbline palette
cbbPalette <- c("#000000", "#E69F00", "#56B4E9", "#009E73",
                "#F0E442", "#0072B2", "#D55E00", "#CC79A7")
```

Survival: Definition

Survival time is the amount of time that elapses from a common starting point until an event occurs. Examples of events include: time to HIV acquisition, time to recurrence of cancer, time to successful pregnancy.

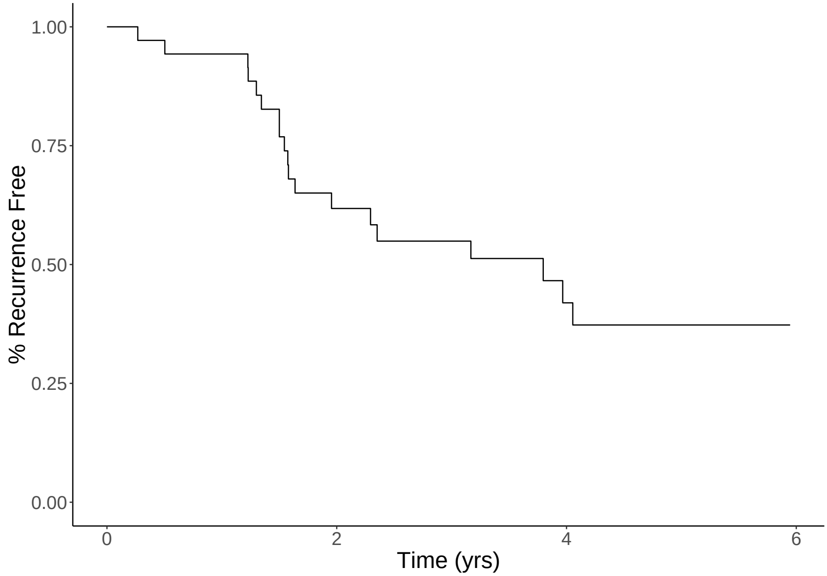
Survival: Graphical example of raw survival data (GBSG)



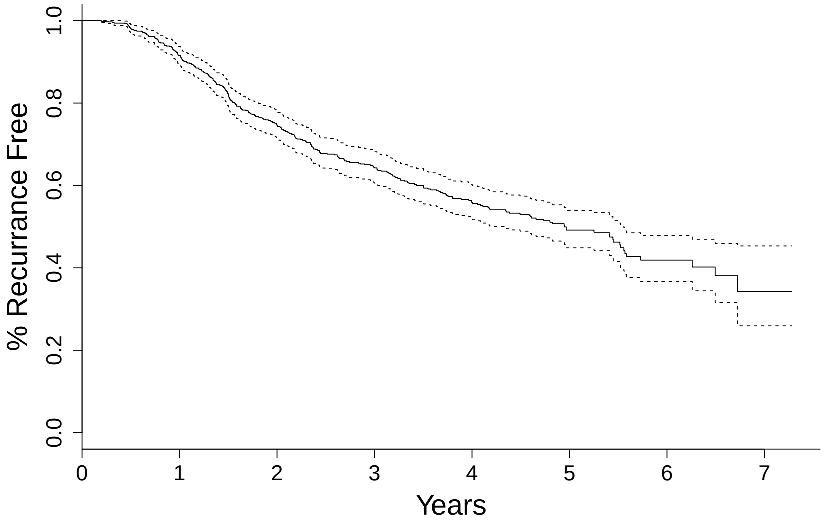
Kaplan-Meier Curves: Creation

```r
# derivation of a survival curve, the hard way
curve <- data.frame(t = 0, # time
                    s = 1) # survival proportion

# go through random sample of 65 individuals
for(i in 1:dim(gbsg.sub)[1])
{
  # proportion who survived until now
  curve <- bind_rows(curve,
          data_frame(t = gbsg.sub$t[i],
                     s = min(curve$s)))

  # if there is an event, decrease survival
  if(gbsg.sub$d[i] == 1)
  {
    t.curr <- gbsg.sub$t[i]
    prop_lost_to_event = 1 / dim(filter(gbsg.sub, t >= t.curr))[1]
    curve <- bind_rows(curve,
            data_frame(t = t.curr,
                       s = min(curve$s) * (1 - prop_lost_to_event)))
  }
}
```

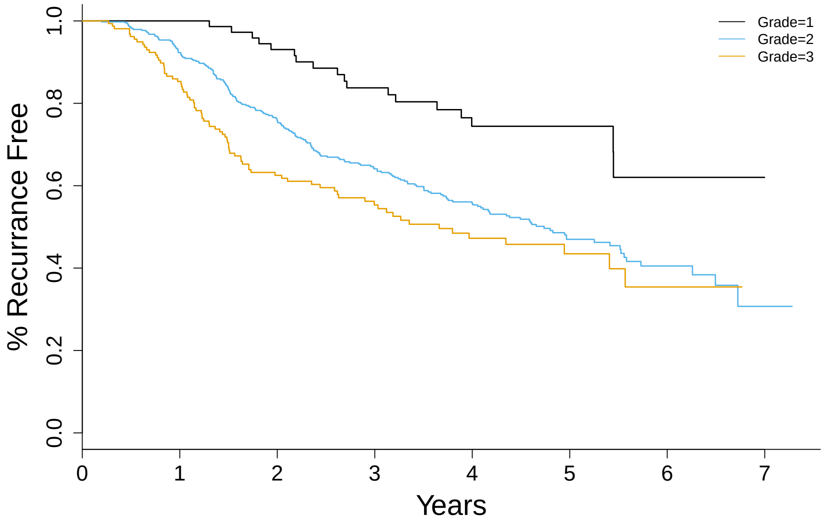
Kaplan-Meier Curve: Creation



Kaplan-Meier Curves: Full GBSG data set



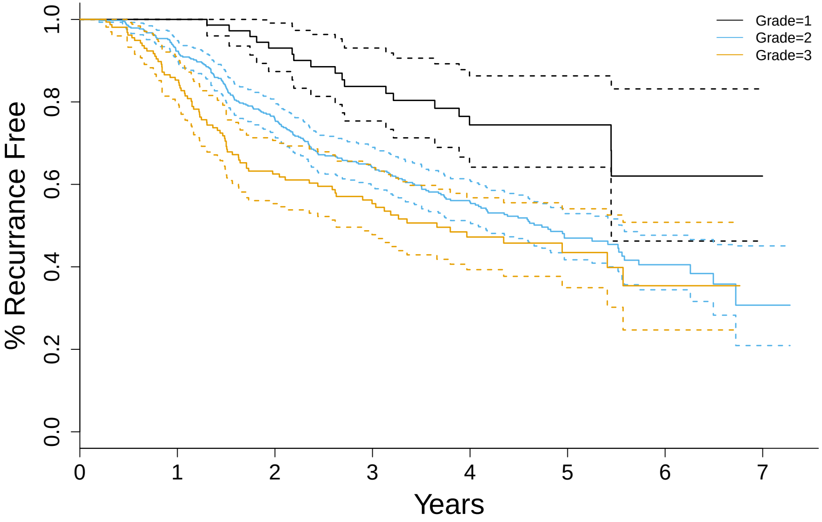
Kaplan-Meier Curves



Kaplan-Meier Curves: Code

```r
# create survival variable
gbsg$surv <- Surv(gbsg$t, gbsg$d)

# plot single curve
survfit(surv ~ 1, data = gbsg) %>%
  plot(bty = 'l', cex.axis = 1.5, ylab = '% Recurrance Free',
       cex.lab = 2, xlab = 'Years')

# plot curves by tumor grade
survfit(surv ~ grade, data = gbsg) %>%
  plot(bty='l', col=cbbPalette[c(1,3,2)], lwd=1.5, cex.lab=2,
       cex.axis=1.5, ylab='% Recurrance Free', xlab='Years')

legend('topright', c('Grade=1', 'Grade=2', 'Grade=3'),
       col = cbbPalette[c(1,3,2)], lty = 1, bty = 'n')

# noisy plot by tumor grade
survfit(surv ~ grade, data = gbsg) %>%
  plot(bty='l', col=cbbPalette[c(1,3,2)], conf.int=TRUE, lwd=1.5
       cex.axis=1.5, cex.lab=2, ylab='% Recurrance Free', xlab='
```

Kaplan-Meier Curves



Survival Analysis

The analysis of survival until a particular event of interest needs to take 4 key measures into account (with definitions for the GBSG data set):

- Target event (recurrence of breast cancer)
- Time origin (tumor resection)
- Time Metric (years - continuous)
- Censoring Causes (mostly at end of study? - would need to look more into this)

Survival Analysis: Time Metric

The time metric can fall two different categories, continuous or discrete. There are four main ways in which we will get discrete time values:

- Truly discrete: Events that may only occur at discrete time points (e.g. graduation time).
- Partially discrete: These events can occur continuously, but tend to clump into discrete groups (e.g. teachers leaving jobs at the end of the year).
- Discrete due to measurement & data collection constraints: These event times are continuous, but we only discrete times are available to us (e.g. CD4 measurements, or recall of event times on a questionnaire).
- Aggregation: Summary of incident events over short periods of time (e.g. clinical events since last visit).

Survival Analysis: Censoring Causes

We need to be aware of the cause of censoring whenever possible. One important thing to watch out for is emmigrative selection bias, when the risk among individuals being censored is somehow different than among those not censored (e.g. a side effect of the treatment is myocardial infarction, that causes patients to drop out of the study).

- Administrative censoring: Censoring at the end of the study (no worries about emmigrative selection bias).
- Drop-out or Loss to follow-up: Emmigrative bias must be addressed, but need not be a problem.
- Competing risks: This can cause emmigrative bias, and the assumption that un/censored groups are similar is indefensible.

Survival Analysis: Hazard Function

The discrete hazard function can be approximated by the following formula when no censoring or late entries exist.

$$\begin{aligned} h(t) &= \mathrm{P}(T = t | T \geq t) \\ &\approx \frac{\#\ \textit{events during time period } t}{\#\ \textit{individuals surviving up to time period } t}. \end{aligned}$$

Survival Analysis: Survival Function

The discrete survival function is the probability of surviving until time t without the event occurring.

$$S(t) = \prod_{i=0}^{t} 1 - h(i)$$

Cox Proportional Hazards Model

- Response: Continuous, time to event
- Interpretation:
  - $\beta_{1\ldots n}$ are the log hazard ratios comparing the likelihood of an event in those exposed to X to the reference group.
- Caveats: Hazards are assumed to be proportional. This can be checked using the `cox.zph()` function or by looking at the log cumulative hazard function. This can be done in R by using the fun = 'cloglog' option in `plot.survfit` (see ?plot.survfit for more details). The baseline hazard, though, is arbitrary and isn't even estimated. One important thing to keep in mind when analyzing survival data is that late entries should have their immortal person time censored from the analysis. For time varying markers, we can split an individual's time into pieces and include later time periods as late entries.

$$\log h(t) = \log h_0(t) + \beta_1 X_1 + \cdots + \beta_n X_n$$

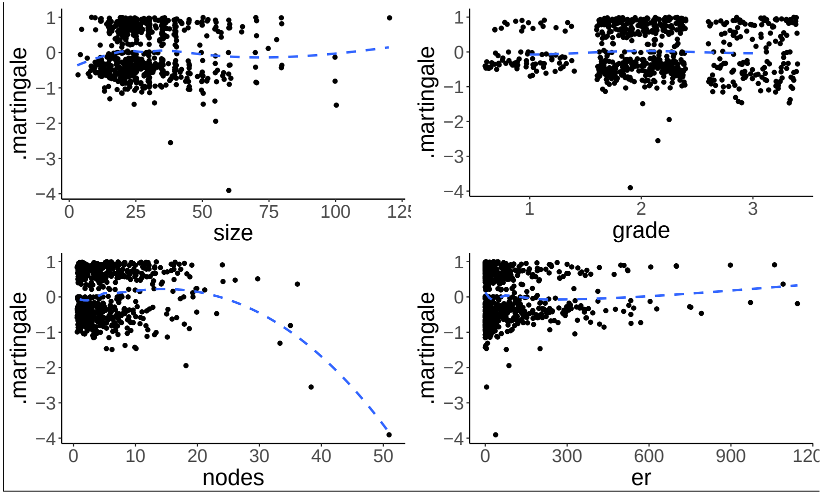
Cox Proportional Hazards Model: Example

```
(model <- coxph(surv ~ size + grade + nodes + er, data = gbsg)) %>%
  summary()
```

```
## Call:
## coxph(formula = surv ~ size + grade + nodes + er, data = gbsg)
##
##   n= 686, number of events= 299
##
##             coef  exp(coef)   se(coef)      z Pr(>|z|)
## size   0.0061306  1.0061495  0.0038223  1.604    0.109
## grade  0.4011126  1.4934854  0.1025339  3.912 9.15e-05 ***
## nodes  0.0525033  1.0539060  0.0074769  7.022 2.19e-12 ***
## er    -0.0006043  0.9993958  0.0004321 -1.399    0.162
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##       exp(coef) exp(-coef) lower .95 upper .95
## size     1.0061     0.9939    0.9986     1.014
## grade    1.4935     0.6696    1.2216     1.826
## nodes    1.0539     0.9489    1.0386     1.069
## er       0.9994     1.0006    0.9985     1.000
##
## Concordance= 0.66  (se = 0.018 )
## Rsquare= 0.099   (max possible= 0.995 )
## Likelihood ratio test= 71.61  on 4 df,   p=1.044e-14
```

Cox Proportional Hazards Model: Linearity Assumption

The Martingale plot should have a flat trend line. Most of these look OK, but the nodes should probably be log transformed.



Cox Porportional Hazards Model: Linearity Assumption

R code:

```
gbsg$.martingale <- residuals(model)

plot_grid(
  ggplot(gbsg, aes(size, .martingale)) +
    geom_jitter() +
    geom_smooth(linetype = 2, se = FALSE),

  ggplot(gbsg, aes(grade, .martingale)) + geom_jitter() + g

  ggplot(gbsg, aes(nodes, .martingale)) + geom_jitter() + g

  ggplot(gbsg, aes(er, .martingale)) + geom_jitter() + geon
```

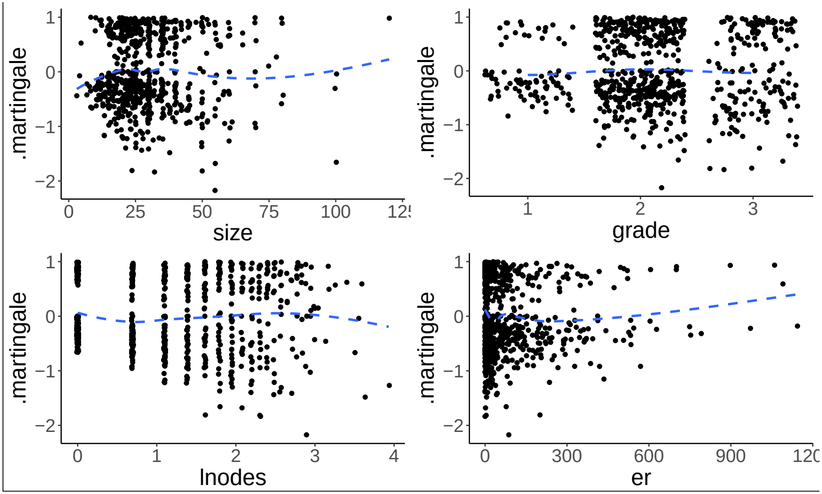
Model Update

```r
gbsg <- mutate(gbsg,
               lnodes = log(nodes),
               ler = log(er + 0.1))

(model1 <- coxph(surv ~ size + grade + lnodes + er, data = gbsg)) %>%
  summary()
```

```
## Call:
## coxph(formula = surv ~ size + grade + lnodes + er, data = gbsg)
##
##   n= 686, number of events= 299
##
##              coef  exp(coef)   se(coef)      z Pr(>|z|)
## size    0.0041428  1.0041514  0.0038139  1.086 0.277378
## grade   0.3810853  1.4638725  0.1027616  3.708 0.000209 ***
## lnodes  0.4977040  1.6449401  0.0670083  7.427 1.11e-13 ***
## er     -0.0005632  0.9994370  0.0004460 -1.263 0.206646
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##        exp(coef) exp(-coef) lower .95 upper .95
## size      1.0042     0.9959    0.9967     1.012
## grade     1.4639     0.6831    1.1968     1.790
## lnodes    1.6449     0.6079    1.4425     1.876
## er        0.9994     1.0006    0.9986     1.000
```

Cox Proportional Hazards Model: Linearity Re-check

The Martingale plot should have a flat trend line. Most of these look OK, but the nodes should probably be log transformed.



Cox Proportional Hazards Model: Collinearity

We are familiar with this function. There are some potential problems with it, but as long as the VIFs are greater than 1, we can interpret them as before.

```
vif(model1)
```

```
## Warning in vif.default(model1): No intercept: vifs may not be
```

```
##     size    grade   lnodes       er
## 1.128980 1.014686 1.127678 1.014429
```

Cox Proportional Hazards Model: Proportionality

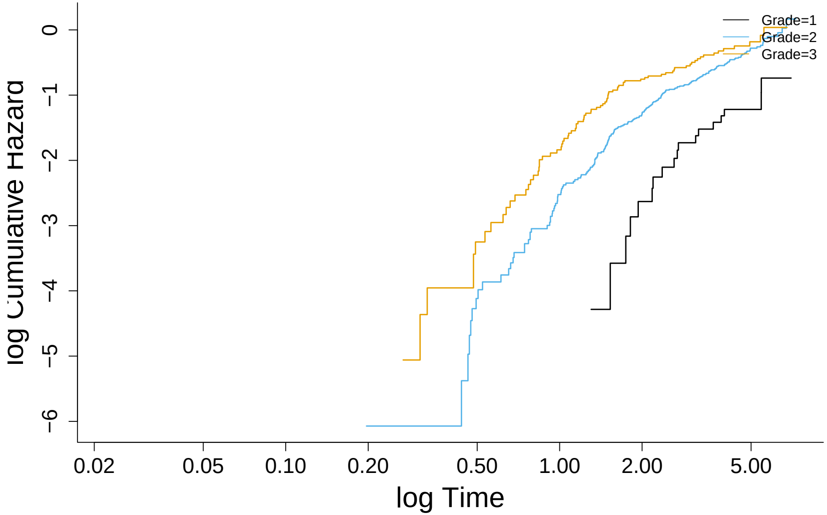
Proportionality assumption seems to be violated for grade and er measure.

```
cox.zph(model1)
```

```
##            rho   chisq       p
## size   -0.0136  0.0541 0.81611
## grade  -0.1657  6.8301 0.00896
## lnodes -0.0531  0.9218 0.33701
## er      0.0912  3.3243 0.06826
## GLOBAL      NA 13.1732 0.01046
```

Cox Proportional Hazards Model: Proportionality

This isn't really necessary, but is interesting to take a look at. We will use the same code as above, but use the `fun='cloglog'` argument. These lines should be roughly parallel.



Model Update 2

```
## Call:
## coxph(formula = surv ~ size + (grade == 1) + lnodes + ler, data = gbsg)
##
##   n= 686, number of events= 299
##
##                      coef exp(coef)  se(coef)      z Pr(>|z|)
## size             0.003993  1.004001  0.003781  1.056 0.290929
## grade == 1TRUE -0.710503  0.491397  0.246308 -2.885 0.003919 **
## lnodes           0.495588  1.641464  0.066913  7.406  1.3e-13 ***
## ler             -0.083003  0.920348  0.022808 -3.639 0.000274 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                exp(coef) exp(-coef) lower .95 upper .95
## size              1.0040     0.9960    0.9966    1.0115
## grade == 1TRUE    0.4914     2.0350    0.3032    0.7963
## lnodes            1.6415     0.6092    1.4397    1.8715
## ler               0.9203     1.0865    0.8801    0.9624
##
## Concordance= 0.675  (se = 0.018 )
## Rsquare= 0.136   (max possible= 0.995 )
## Likelihood ratio test= 100.5  on 4 df,   p=0
## Wald test            = 99  on 4 df,   p=0
## Score (logrank) test = 104.3  on 4 df,   p=0
```

Proportionality Re-check

Probably will stick with untransformed er… or perhaps just drop it altogether.

```
cox.zph(model2)
```

```
##                      rho   chisq      p
## size             -0.0134  0.0515 0.8205
## grade == 1TRUE    0.0888  2.3875 0.1223
## lnodes           -0.0594  1.1610 0.2812
## ler               0.1427  6.0439 0.0140
## GLOBAL                NA 11.6159 0.0204
```

Model Update 3

```
## Call:
## coxph(formula = surv ~ size + (grade == 1) + lnodes, data = gbsg)
##
##   n= 686, number of events= 299
##
##                       coef exp(coef)  se(coef)      z Pr(>|z|)
## size              0.004710  1.004721  0.003764  1.251 0.210866
## grade == 1TRUE -0.815722  0.442320  0.244101 -3.342 0.000833 ***
## lnodes            0.491468  1.634715  0.066971  7.339 2.16e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                exp(coef) exp(-coef) lower .95 upper .95
## size              1.0047     0.9953    0.9973    1.0122
## grade == 1TRUE    0.4423     2.2608    0.2741    0.7137
## lnodes            1.6347     0.6117    1.4336    1.8640
##
## Concordance= 0.666  (se = 0.018 )
## Rsquare= 0.12   (max possible= 0.995 )
## Likelihood ratio test= 87.95  on 3 df,   p=0
## Wald test            = 86.21  on 3 df,   p=0
## Score (logrank) test = 90.3  on 3 df,   p=0
```

Cox Regression: Time dependent variables

In some instances, we have variables that change over time. In the table below, individual 5 has several creatinine measurements over a 12 month period prior to death, as well as a disease related infection toward the end.

subject	time1	time2	death	infection	creatinine
5	0	60	0	0	0.9
5	60	90	0	0	1.0
5	90	120	0	0	1.5
5	120	160	0	0	1.2
5	160	180	0	0	0.9
5	180	201	0	0	1.0
5	201	210	0	1	1.0
5	210	213	1	1	1.3

Cox Regression: Knowing the future

When we set up our regression model, survival time goes on the left-hand side of the equation, and predictors go on the right-hand side. Infection in the previous slide needs to be a time dependent variable, because it doesn't occur until day 201. Including it as a time invariant variable at earlier time intervals would be like knowing the future. We can't know the future in real life, and models that suggest otherwise are not very useful.

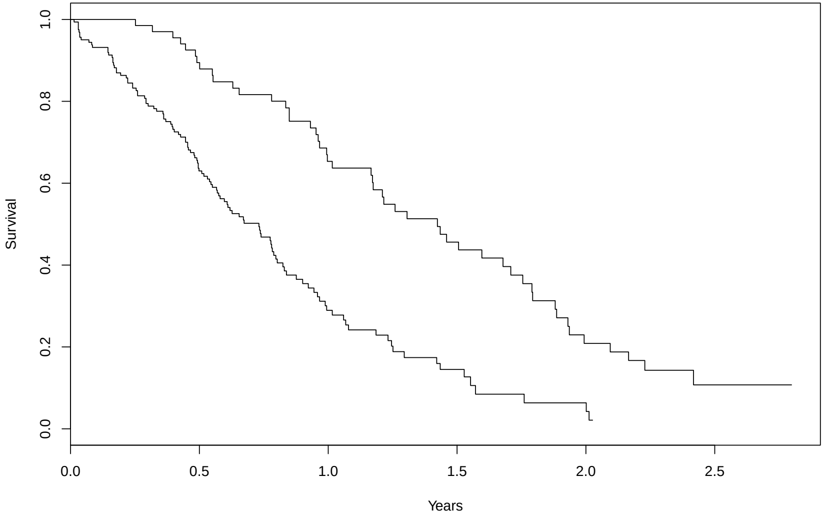
Cox Regression: Knowing the future (Example)

Using a future outcome to predict survival. For every month an individual remains in the study, they get a 10% chance of having a positive response to some treatment we give them.

```r
set.seed(238947)
lung$response <- rbinom(dim(lung)[1],
                        ceiling(lung$time / 30.5), 0.1) > 1

model <- survfit(Surv(time/365.25, status) ~ response,
                     data = lung)
```

Cox Regression: Knowing the future (Example)

Is this surprising?



Cox Regression: Knowing the future (Example)

A more appropriate way to code this is to make a time dependent variable that starts at 0 for everyone and changes to 1 when the treatment response is observed.

Another Example (from survival vignette)

Chronic Granulotomous Disease (CGD) data

"CGD is a heterogeneous group of uncommon inherited disorders characterized by recurrent pyogenic infections. . . "

"In 1986, Genentech, Inc. conducted a randomized, double-blind, placebo- controlled trial in 128 CGD patients."

Another Example (from survival vignette)

```
## 'data.frame':    128 obs. of  20 variables:
##  $ id      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ center  : int  204 204 204 204 238 245 245 245 238 238 ...
##  $ random  : int  82888 82888 82988 91388 92888 93088 93088 93088 100488 100
##  $ treat   : int  1 0 1 1 0 1 0 1 0 1 ...
##  $ sex     : int  2 1 1 1 1 2 1 1 1 1 ...
##  $ age     : int  12 15 19 12 17 44 22 7 27 5 ...
##  $ height  : num  147 159 171 142 162 ...
##  $ weight  : num  62 47.5 72.7 34 52.7 45 59.7 17.4 82.8 19.5 ...
##  $ inherit : int  2 2 1 1 1 2 1 1 2 1 ...
##  $ steroids: int  2 2 2 2 2 2 2 2 2 2 ...
##  $ propylac: int  2 1 1 1 1 2 1 1 1 1 ...
##  $ hos.cat : int  2 2 2 2 1 2 2 2 1 1 ...
##  $ futime  : int  414 439 382 388 383 364 364 363 349 371 ...
##  $ etime1  : int  219 8 NA NA 246 NA 292 NA 294 NA ...
##  $ etime2  : int  373 26 NA NA 253 NA NA NA NA NA ...
##  $ etime3  : int  NA 152 NA NA NA NA NA NA NA NA ...
##  $ etime4  : int  NA 241 NA NA NA NA NA NA NA NA ...
##  $ etime5  : int  NA 249 NA NA NA NA NA NA NA NA ...
##  $ etime6  : int  NA 322 NA NA NA NA NA NA NA NA ...
##  $ etime7  : int  NA 350 NA NA NA NA NA NA NA NA ...
```

Preparing the CGD data

```r
cgd0 <- tmerge(cgd0[,1:13], cgd0, id = id, tstop = futime,
               infect = event(etime1),
               infect = event(etime2),
               infect = event(etime3),
               infect = event(etime4),
               infect = event(etime5),
               infect = event(etime6),
               infect = event(etime7))
```

Preparing the CGD data

```
## 'data.frame':    203 obs. of  16 variables:
##  $ id      : int  1 1 1 2 2 2 2 2 2 2 ...
##  $ center  : int  204 204 204 204 204 204 204 204 204 20
##  $ random  : int  82888 82888 82888 82888 82888 82888 82
##  $ treat   : int  1 1 1 0 0 0 0 0 0 0 ...
##  $ sex     : int  2 2 2 1 1 1 1 1 1 1 ...
##  $ age     : int  12 12 12 15 15 15 15 15 15 15 ...
##  $ height  : num  147 147 147 159 159 159 159 159 159 15
##  $ weight  : num  62 62 62 47.5 47.5 47.5 47.5 47.5 47.5
##  $ inherit : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ steroids: int  2 2 2 2 2 2 2 2 2 2 ...
##  $ propylac: int  2 2 2 1 1 1 1 1 1 1 ...
##  $ hos.cat : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ futime  : int  414 414 414 439 439 439 439 439 439 43
##  $ tstart  : num  0 219 373 0 8 26 152 241 249 322 ...
##  $ tstop   : int  219 373 414 8 26 152 241 249 322 350
##  $ infect  : num  1 1 0 1 1 1 1 1 1 1 ...
##  - attr(*, "tname")=List of 3
```

Analyzing the CGD data

See 12_SurvivalAnalysis.R