Cox Model Building and Diagnostics

Model building

Load the data

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------
```

```
## v ggplot2 3.2.0     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## Warning: package 'dplyr' was built under R version 3.6.1
```

```
## -- Conflicts ------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(survival)
library(asaur)

dat <- pharmacoSmoking
```

The 4 candidate models

```
M0 <- coxph(Surv(ttr, relapse) ~ 1, data = dat)
MA <- coxph(Surv(ttr, relapse) ~ ageGroup4, data = dat)
MB <- coxph(Surv(ttr, relapse) ~ employment, data = dat)
MC <- coxph(Surv(ttr, relapse) ~ ageGroup4 + employment, data = dat)
```

```
table(dat$employment)
```

```
##
##    ft other    pt
##    72    39    14
```

```
table(dat$ageGroup4)
```

```
##
## 21-34 35-49 50-64   65+
##    16    50    48    11
```

Comparing nested models: LRT

```
anova(MA, MC)
```

```
## Analysis of Deviance Table
##  Cox model: response is  Surv(ttr, relapse)
##  Model 1: ~ ageGroup4
##  Model 2: ~ ageGroup4 + employment
##    loglik  Chisq Df P(>|Chi|)
## 1 -380.04
## 2 -377.76 4.5666  2    0.1019
```

$MA = B_1A_2 + B_2A_3 + B_3A_4$

$MC = MA + B_4E_o + B_5E_{pt}$

$H_0 : (B_4, B_5) = (0, 0)$

we do not reject H0 so that MA is accurate enoug. Note that here model selection in ANOVA we don't need all categorical data.

Comparing non-nested models: AIC

AIC smaller the better. $AIC = -2loglik(\hat{B}) + 2k$

```
fits <- list(MA = MA, MB = MB, MC = MC)
sapply(fits, AIC)
```

```
##       MA       MB       MC
## 766.0860 774.2464 765.5194
```

Automatic model selection based on AIC

```
Mfull <- coxph(Surv(ttr, relapse) ~ grp + gender + race +
                 employment + yearsSmoking + levelSmoking +
                 ageGroup4 + priorAttempts + longestNoSmoke,
               data = dat)
```

backward step, stop when AIC inrease

```
MAIC <- step(Mfull) #backward #return final model
```

```
## Start:  AIC=770.2
## Surv(ttr, relapse) ~ grp + gender + race + employment + yearsSmoking +
##     levelSmoking + ageGroup4 + priorAttempts + longestNoSmoke
##
##                   Df    AIC
## - race             3 766.98
## - yearsSmoking     1 768.20
```

```
## - gender          1 768.20
## - priorAttempts   1 768.24
## - levelSmoking    1 768.47
## - longestNoSmoke  1 769.04
## <none>              770.20
## - employment      2 772.45
## - ageGroup4       3 774.11
## - grp             1 776.80
##
## Step:  AIC=766.98
## Surv(ttr, relapse) ~ grp + gender + employment + yearsSmoking +
##     levelSmoking + ageGroup4 + priorAttempts + longestNoSmoke
##
##                  Df    AIC
## - levelSmoking    1 764.98
## - gender          1 765.00
## - priorAttempts   1 765.01
## - yearsSmoking    1 765.04
## - longestNoSmoke  1 766.29
## <none>              766.98
## - employment      2 768.37
## - ageGroup4       3 770.16
## - grp             1 773.88
##
## Step:  AIC=764.98
## Surv(ttr, relapse) ~ grp + gender + employment + yearsSmoking +
##     ageGroup4 + priorAttempts + longestNoSmoke
##
##                  Df    AIC
## - gender          1 763.00
## - priorAttempts   1 763.01
## - yearsSmoking    1 763.06
## - longestNoSmoke  1 764.29
## <none>              764.98
## - employment      2 766.37
## - ageGroup4       3 768.18
## - grp             1 771.88
##
## Step:  AIC=763
## Surv(ttr, relapse) ~ grp + employment + yearsSmoking + ageGroup4 +
##     priorAttempts + longestNoSmoke
##
##                  Df    AIC
## - priorAttempts   1 761.02
## - yearsSmoking    1 761.08
## - longestNoSmoke  1 762.31
## <none>              763.00
## - employment      2 764.42
## - ageGroup4       3 766.32
## - grp             1 769.91
##
## Step:  AIC=761.02
## Surv(ttr, relapse) ~ grp + employment + yearsSmoking + ageGroup4 +
##     longestNoSmoke
```

```
##
##                   Df    AIC
## - yearsSmoking     1 759.10
## - longestNoSmoke   1 760.34
## <none>               761.02
## - employment       2 762.42
## - ageGroup4        3 764.50
## - grp              1 767.93
##
## Step:  AIC=759.1
## Surv(ttr, relapse) ~ grp + employment + ageGroup4 + longestNoSmoke
##
##                   Df    AIC
## - longestNoSmoke   1 758.42
## <none>               759.10
## - employment       2 760.42
## - grp              1 765.94
## - ageGroup4        3 766.90
##
## Step:  AIC=758.42
## Surv(ttr, relapse) ~ grp + employment + ageGroup4
##
##              Df    AIC
## <none>          758.42
## - employment  2 760.31
## - grp         1 765.52
## - ageGroup4   3 767.24
```

Predictive power: concordance index

```
summary(MA)
```

```
## Call:
## coxph(formula = Surv(ttr, relapse) ~ ageGroup4, data = dat)
##
##   n= 125, number of events= 89
##
##                    coef exp(coef) se(coef)      z Pr(>|z|)
## ageGroup435-49  0.0293    1.0297   0.3093  0.095   0.9245
## ageGroup450-64 -0.7914    0.4532   0.3361 -2.355   0.0185 *
## ageGroup465+   -0.3173    0.7281   0.4435 -0.715   0.4744
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                exp(coef) exp(-coef) lower .95 upper .95
## ageGroup435-49    1.0297     0.9711    0.5616    1.8880
## ageGroup450-64    0.4532     2.2066    0.2345    0.8757
## ageGroup465+      0.7281     1.3734    0.3053    1.7367
##
## Concordance= 0.593  (se = 0.032 )
## Likelihood ratio test= 12.22  on 3 df,   p=0.007
## Wald test            = 11.36  on 3 df,   p=0.01
```

```
## Score (logrank) test = 11.93  on 3 df,   p=0.008
```

```
summary(MAIC)
```

```
## Call:
## coxph(formula = Surv(ttr, relapse) ~ grp + employment + ageGroup4,
##     data = dat)
##
##   n= 125, number of events= 89
##
##                     coef exp(coef) se(coef)      z Pr(>|z|)
## grppatchOnly      0.6564    1.9278   0.2198  2.986  0.00283 **
## employmentother   0.6231    1.8648   0.2764  2.254  0.02418 *
## employmentpt      0.5214    1.6844   0.3320  1.570  0.11631
## ageGroup435-49   -0.1119    0.8942   0.3216 -0.348  0.72792
## ageGroup450-64   -1.0233    0.3594   0.3597 -2.845  0.00444 **
## ageGroup465+     -0.7071    0.4931   0.5017 -1.410  0.15868
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                  exp(coef) exp(-coef) lower .95 upper .95
## grppatchOnly        1.9278     0.5187    1.2529    2.9661
## employmentother     1.8648     0.5363    1.0848    3.2057
## employmentpt        1.6844     0.5937    0.8787    3.2289
## ageGroup435-49      0.8942     1.1184    0.4761    1.6793
## ageGroup450-64      0.3594     2.7825    0.1776    0.7273
## ageGroup465+        0.4931     2.0281    0.1845    1.3180
##
## Concordance= 0.647  (se = 0.033 )
## Likelihood ratio test= 25.89  on 6 df,   p=2e-04
## Wald test            = 24.59  on 6 df,   p=4e-04
## Score (logrank) test = 25.54  on 6 df,   p=3e-04
```

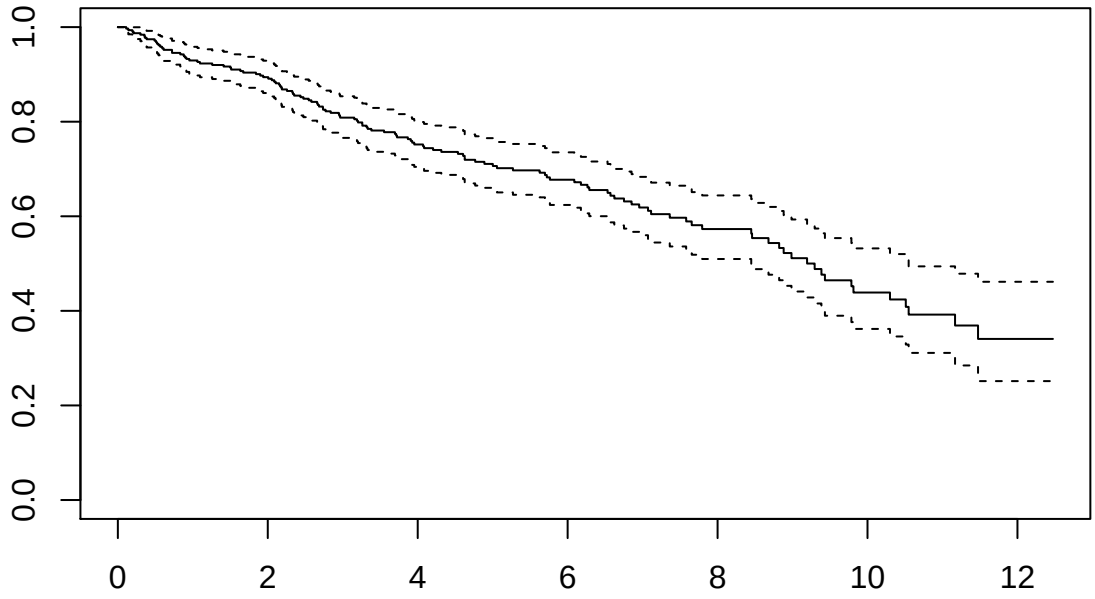
Predictive power: AUC

```
library(survivalROC)
data(mayo)
head(mayo)
```

```
##   time censor mayoscore5 mayoscore4
## 1   41      1  11.251850  10.629450
## 2  179      1  10.136070  10.185220
## 3  334      1  10.095740   9.422995
## 4  400      1  10.189150   9.567799
## 5  130      1   9.770148   9.039419
## 6  223      1   9.226429   9.033388
```

plot survival curve

```r
plot(survfit(Surv(time / 365.25, censor) ~ 1, data = mayo))
```




```r
survfit(Surv(time / 365.25, censor) ~ 1, data = mayo) #see medium
```

```
## Call: survfit(formula = Surv(time/365.25, censor) ~ 1, data = mayo)
##
##       n  events  median 0.95LCL 0.95UCL
##  312.00  125.00    9.30    8.45   10.55
```

```r
#5 years time point
#time and censor gave true value score, so score4 and score5 is the predicted score Y
ROC.4 <- survivalROC(Stime = mayo$time,
                     status = mayo$censor,
                     marker = mayo$mayoscore4,
                     predict.time = 365.25 * 5,
                     method="KM")
ROC.5 <- survivalROC(Stime = mayo$time,
                     status = mayo$censor,
                     marker = mayo$mayoscore5,
                     predict.time = 365.25*5,
                     method = "KM")
```

```
ROC <- list(mayo4 = ROC.4, mayo5 = ROC.5)
map_dbl(ROC, "AUC") #map the result and extract AUC
```

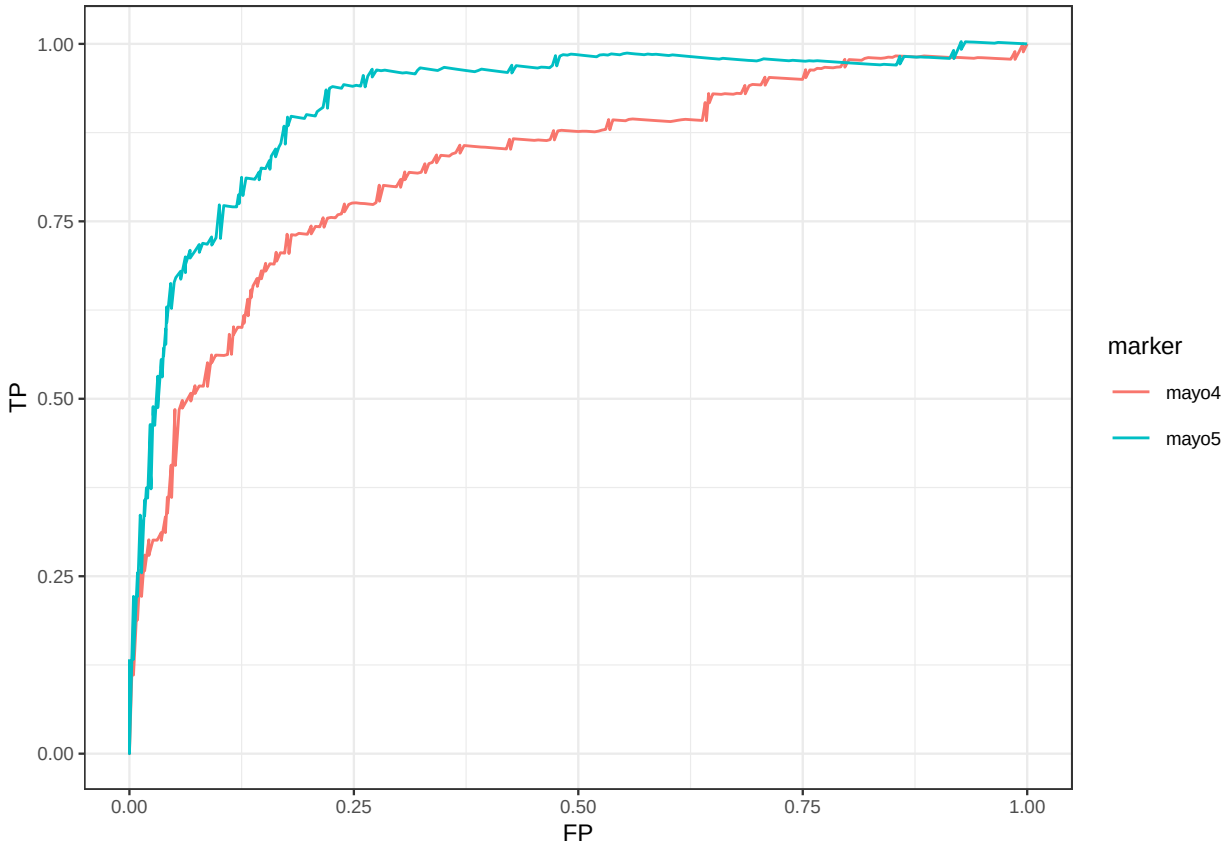
```
##     mayo4     mayo5
## 0.8257006 0.9182824
```

```
dfl <- map(ROC, ~ with(., tibble(cutoff = cut.values, FP, TP)))
for(nm in names(dfl)) {
  dfl[[ nm ]]$marker <- nm
}
dat <- do.call(rbind, dfl)
#list(T1,T2)
#make do.call to put them into one df
```

```
dat
```

```
## # A tibble: 626 x 4
##     cutoff    FP    TP marker
##  *   <dbl> <dbl> <dbl> <chr>
##  1 -Inf    1     1     mayo4
##  2    4.58 0.995 1.00  mayo4
##  3    4.90 0.996 0.989 mayo4
##  4    4.93 0.991 0.989 mayo4
##  5    4.93 0.986 0.989 mayo4
##  6    4.95 0.986 0.978 mayo4
##  7    4.97 0.982 0.978 mayo4
##  8    4.98 0.977 0.979 mayo4
##  9    5.06 0.972 0.979 mayo4
## 10    5.09 0.968 0.979 mayo4
## # ... with 616 more rows
```

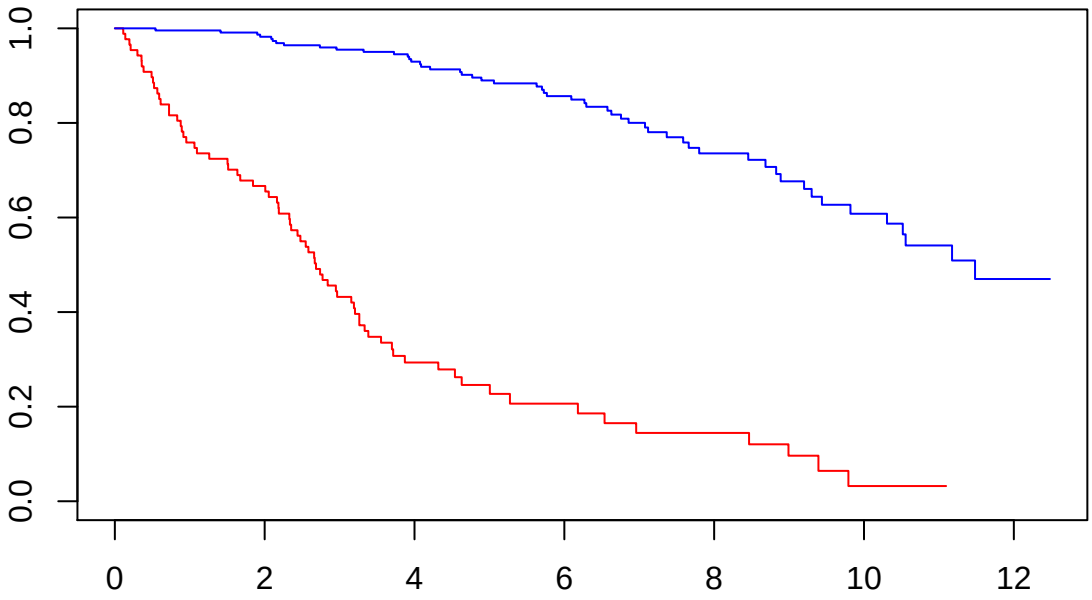
```
ggplot(dat, aes(FP, TP, color = marker)) +
  geom_line() +
  theme_bw(base_size = 9)
```




```
cutoff <- min(filter(dat, marker == "mayo5", FP <= 0.1)$cutoff)
```

```
mayo$prediction <-
  ifelse(mayo$mayoscore5 <= cutoff,
         "low_risk", "high_risk")
```

```
plot(survfit(Surv(time/365, censor) ~ prediction, data = mayo),
     col = c("red", "blue"))
```



Model diagnostics

Martingale residuals

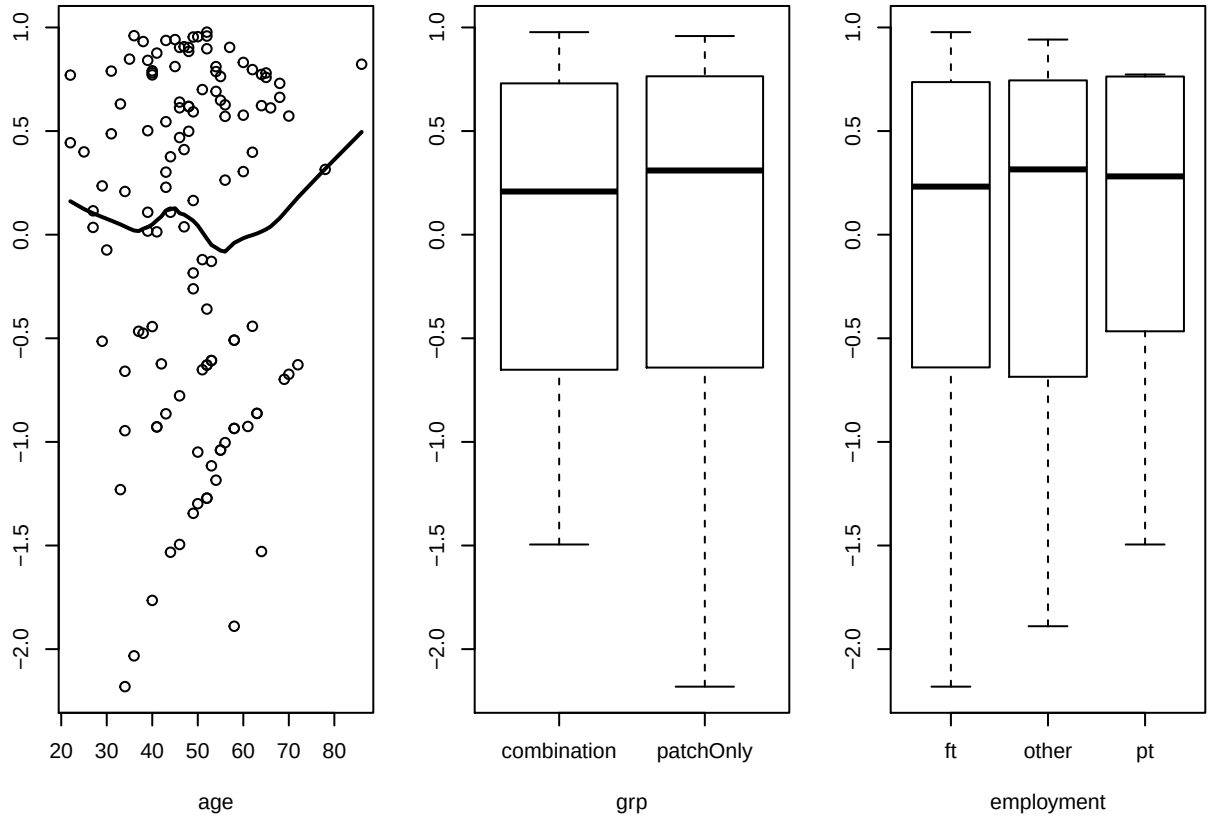
```
library(survival)
library(asaur) ## dataset

data(pharmacoSmoking)
dat <- pharmacoSmoking
```

```
fit <- coxph(Surv(ttr, relapse) ~ grp + age + employment, data = dat)
dat$residual <- residuals(fit, type = "martingale")
```

there are no strong patterns

```
par(mfrow = c(1, 3), mar = c(4.2, 2, 2, 2))
with(dat, {

  plot(age, residual)
  lines(lowess(age, residual), lwd = 2)

  plot(residual ~ grp)
```

```
  plot(residual ~ employment)

})
```



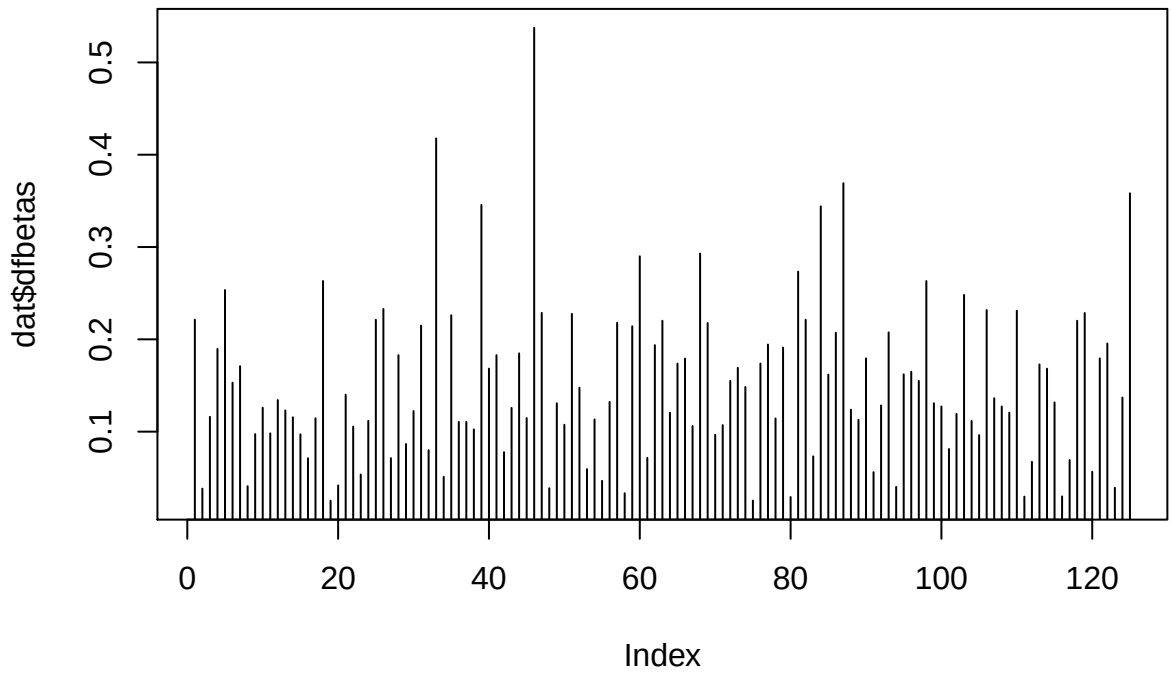

```
d1<-mutate(dat,agesq=age*age)
fit<-coxph(Surv(ttr,relapse)~grp+age+agesq+employment,data=d1)
summary(fit)
```

```
## Call:
## coxph(formula = Surv(ttr, relapse) ~ grp + age + agesq + employment,
##     data = d1)
##
##   n= 125, number of events= 89
##
##                       coef  exp(coef)   se(coef)      z Pr(>|z|)
## grppatchOnly     0.6206075  1.8600577  0.2188288  2.836  0.00457 **
## age             -0.1001902  0.9046654  0.0549849 -1.822  0.06843 .
## agesq            0.0006729  1.0006732  0.0005572  1.208  0.22713
## employmentother  0.6800741  1.9740240  0.2754600  2.469  0.01355 *
## employmentpt     0.6757762  1.9655581  0.3278821  2.061  0.03930 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                 exp(coef) exp(-coef) lower .95 upper .95
```

```
## grppatchOnly       1.8601     0.5376    1.2113     2.856
## age                0.9047     1.1054    0.8122     1.008
## agesq              1.0007     0.9993    0.9996     1.002
## employmentother    1.9740     0.5066    1.1505     3.387
## employmentpt       1.9656     0.5088    1.0337     3.737
##
## Concordance= 0.633  (se = 0.031 )
## Likelihood ratio test= 23.36  on 5 df,   p=3e-04
## Wald test            = 24.19  on 5 df,   p=2e-04
## Score (logrank) test = 24.68  on 5 df,   p=2e-04
```

```
dfbetas <- residuals(fit, type = 'dfbetas')
dat$dfbetas <- sqrt(rowSums(dfbetas^2))
```

```
plot(dat$dfbetas, type = 'h')
abline(h = 0)
```



Proportionality of hazards

Pancreatic cancer dataset

```
library(survival)
library(asaur) ## dataset
library(plyr)
```

```
## ------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)

## ------------------------------------------------------------------------

##
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following object is masked from 'package:purrr':
##
##     compact
```

```
library(ggplot2)

fmt <- "%m/%d/%Y"
dat <- as.tibble(pancreatic) %>%
  mutate(
  onstudy = as.Date(as.character(onstudy), format = fmt),
  progression = as.Date(as.character(progression), format = fmt),
  death = as.Date(as.character(death), format = fmt),
  OS = death - onstudy,
  PFS = ifelse(is.na(progression), OS, pmin(progression - onstudy, OS))) %>%
  mutate(
  PFS = Surv(as.numeric(PFS / 30.5)),
  OS = Surv(as.numeric(OS / 30.5))
  )
```

```
## Warning: `as.tibble()` is deprecated, use `as_tibble()` (but mind the new semantics).
## This warning is displayed once per session.
```

```
dat
```

```
## # A tibble: 41 x 6
##    stage onstudy    progression death      OS[,"time"] [,"status"]
##    <fct> <date>     <date>      <date>           <dbl>       <dbl>
##  1 M     2005-12-16 2006-02-02  2006-10-19       10.1            1
##  2 M     2006-01-06 2006-02-26  2006-04-19        3.38           1
##  3 LA    2006-02-03 2006-08-02  2007-01-19       11.5            1
##  4 M     2006-03-30 NA          2006-05-11        1.38           1
##  5 LA    2006-04-27 2007-03-11  2007-05-29       13.0            1
##  6 M     2006-05-07 2006-06-25  2006-10-11        5.15           1
##  7 LA    2006-08-20 NA          2007-01-24        5.15           1
##  8 M     2007-01-22 2007-03-20  2007-04-14        2.69           1
```

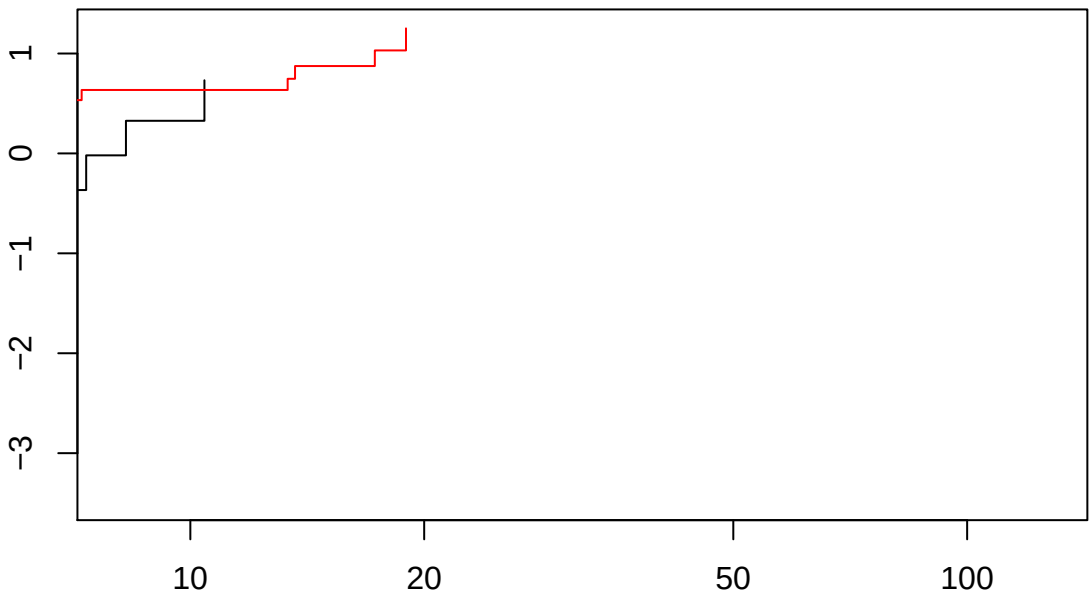
```
##  9 LA    2007-03-02 NA          2008-11-01       20              1
## 10 M     2007-03-27 NA          2007-05-15        1.61           1
## # ... with 31 more rows, and 2 more variables: PFS[,"time"] <dbl>,
## #   [,"status"] <dbl>
```

```
fit <- coxph(PFS ~ stage, data = dat)
summary(fit)
```

```
## Call:
## coxph(formula = PFS ~ stage, data = dat)
##
##   n= 41, number of events= 41
##
##          coef exp(coef) se(coef)    z Pr(>|z|)
## stageM 0.5931    1.8095   0.4007 1.48    0.139
##
##        exp(coef) exp(-coef) lower .95 upper .95
## stageM      1.81     0.5526    0.8251     3.969
##
## Concordance= 0.589  (se = 0.033 )
## Likelihood ratio test= 2.43  on 1 df,   p=0.1
## Wald test            = 2.19  on 1 df,   p=0.1
## Score (logrank) test = 2.25  on 1 df,   p=0.1
```

```
fit.KM <- survfit(PFS ~ stage, data = dat)
plot(fit.KM, fun= "cloglog", col = 1:2)
```

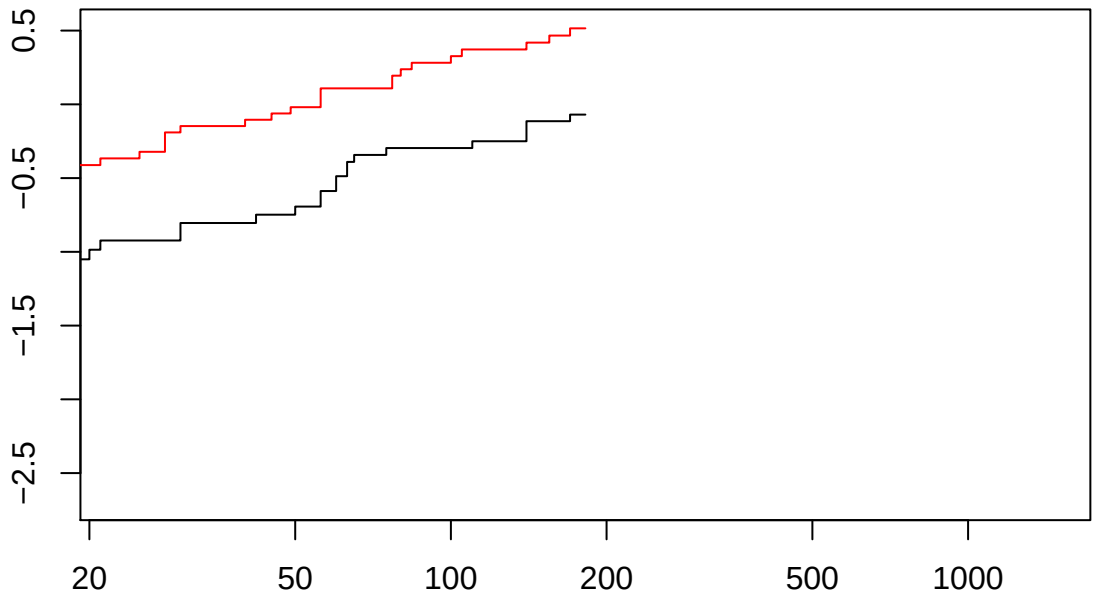
```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 1 x value <= 0 omitted
## from logarithmic plot
```



if we plot the survival function in c-log-log scale, we have to reject the H0 of the risk is proportional, the cox model would not be a good description. (but if we don't have enough sample size hard to tell) another limitation it's only for single categorical variable.

```
#head(pharmacoSmoking)
fit.KM <- survfit(Surv(ttr, relapse) ~ grp, data = pharmacoSmoking)

plot(fit.KM, fun = "cloglog", col = 1:2)
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 1 x value <= 0 omitted
## from logarithmic plot
```



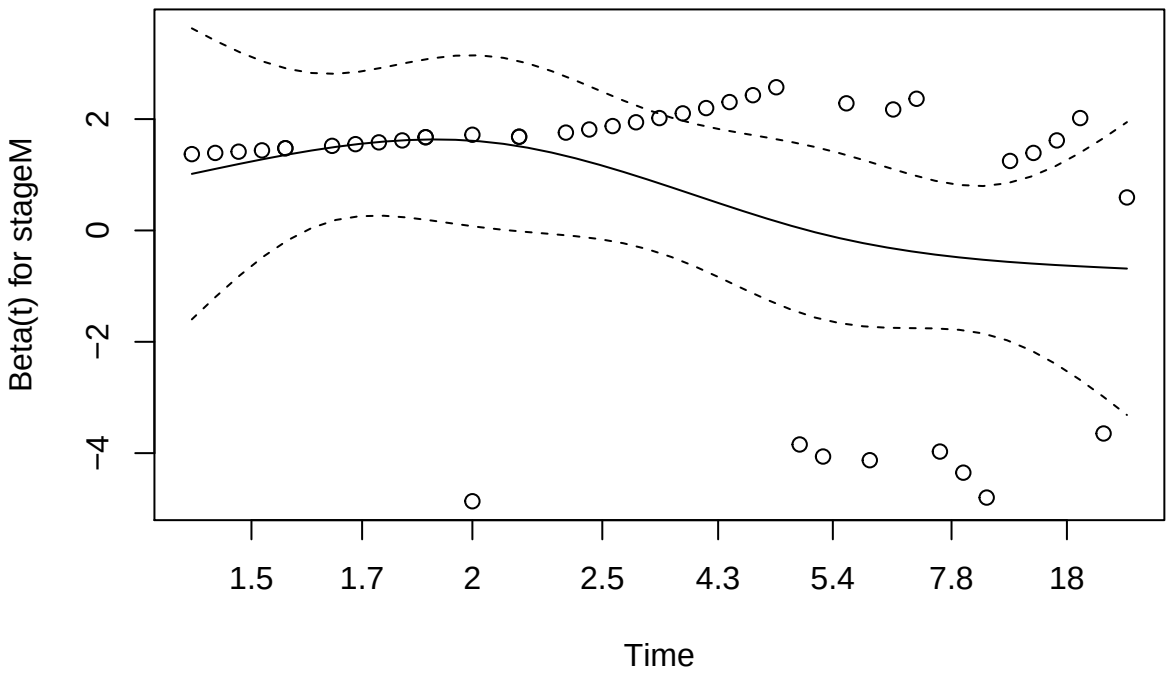
Schoenfeld residuals

```
fit <- coxph(PFS ~ stage, data = dat)
residual.sch <- cox.zph(fit)
residual.sch
```

```
##           rho chisq      p
## stageM -0.328  3.86 0.0496
```

- the hypothesis is that the slope=0, time independant. here is at the boarderline we would still reject it...

```
plot(residual.sch)
```



Dealing with assumptions violations

Stratification

```
library(asaur)
d <- pharmacoSmoking
d$employment <- ifelse(d$employment == "ft", "ft", "other")

table(d$employment)
```

```
##
##    ft other
##    72    53
```

Stratified Cox model:

all we know is we would take employment into account but we don't quantify its B since it's non-proportional

```
fit <- coxph(Surv(ttr, relapse) ~ grp + strata(employment), data = d)
summary(fit)
```

```
## Call:
```

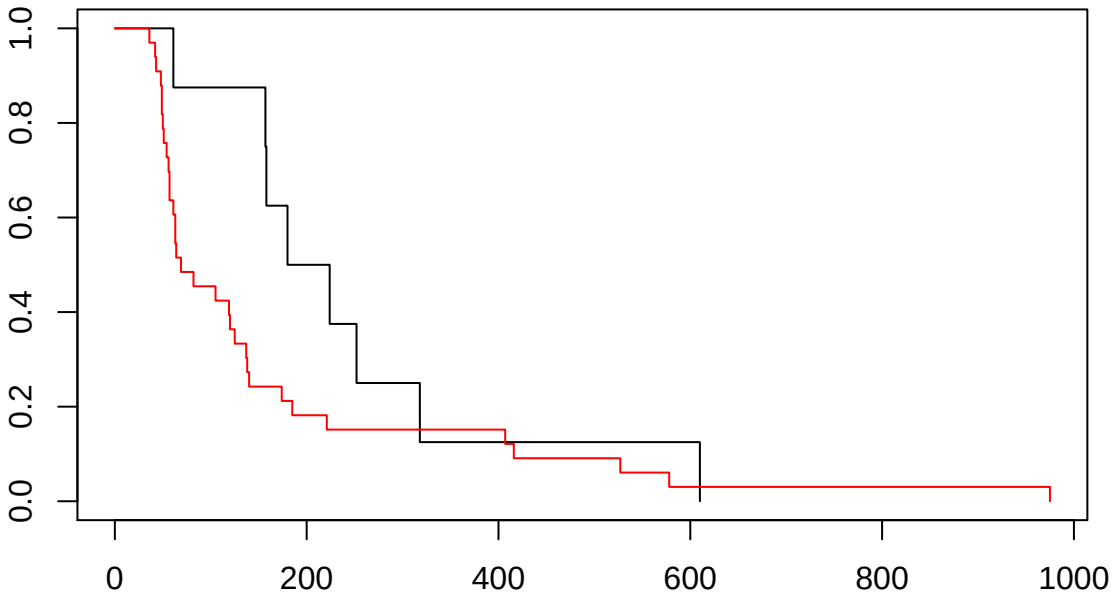
```
## coxph(formula = Surv(ttr, relapse) ~ grp + strata(employment),
##     data = d)
##
##   n= 125, number of events= 89
##
##                coef exp(coef) se(coef)     z Pr(>|z|)
## grppatchOnly 0.6391    1.8947   0.2187 2.922  0.00348 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##              exp(coef) exp(-coef) lower .95 upper .95
## grppatchOnly     1.895     0.5278     1.234     2.909
##
## Concordance= 0.577  (se = 0.029 )
## Likelihood ratio test= 8.71  on 1 df,   p=0.003
## Wald test            = 8.54  on 1 df,   p=0.003
## Score (logrank) test = 8.81  on 1 df,   p=0.003
```

```
#fit <- coxph(Surv(ttr, relapse) ~ grp + employment, data = d)
#summary(fit)
```

Note how there is no estimate associated with 'employment'.

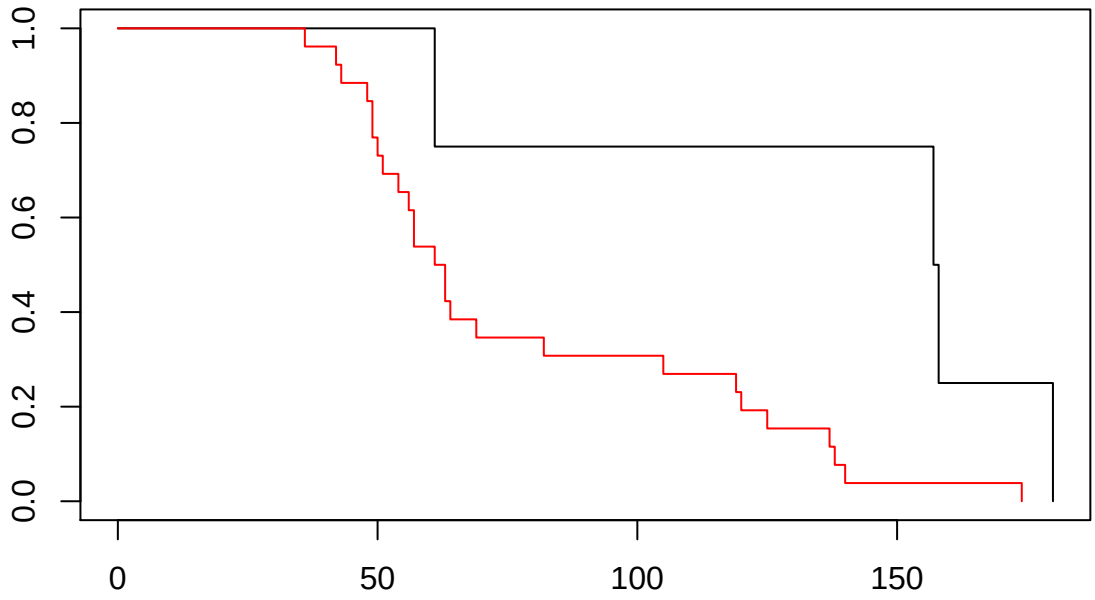
Truncation

```
library(asaur)
library(survival)
d <- pancreatic2

plot(survfit(Surv(pfs, status) ~ stage, data = d), col = 1:2)
```



THIS IS *NOT* HOW IT IS DONE:

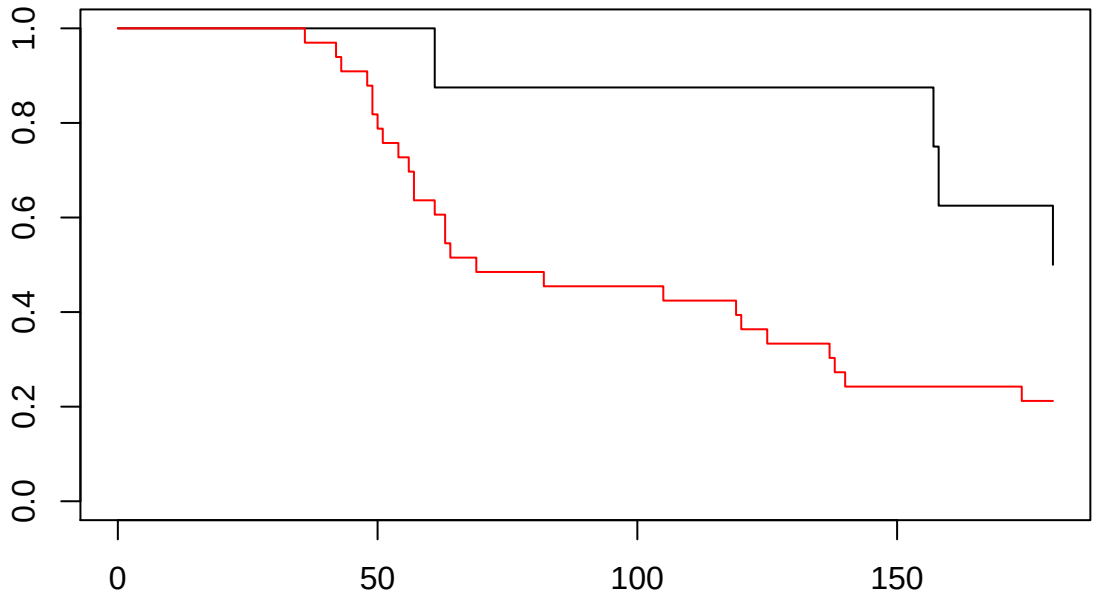
```
d_WRONG <- subset(d, pfs <= 180)

plot(survfit(Surv(pfs, status) ~ stage, data = d_WRONG), col = 1:2)
```



Here is how you do it:

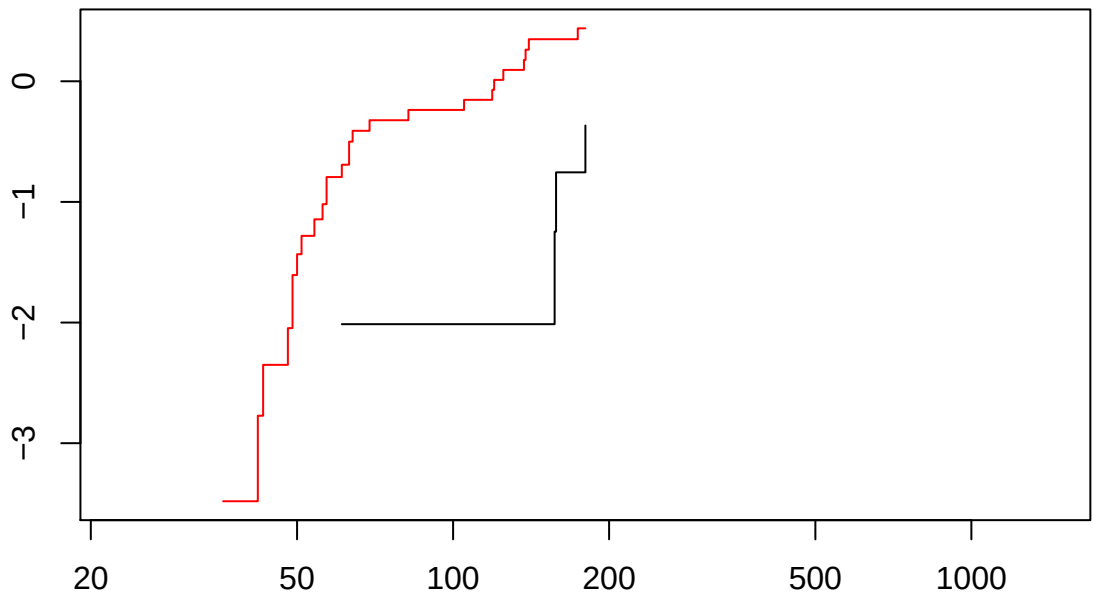
```
d_RIGHT <- within(d, {
  status_truncated <- ifelse(pfs > 180, 0, status)
  pfs_truncated <- ifelse(pfs > 180, 180, pfs)
})
```

```
plot(survfit(Surv(pfs_truncated, status_truncated) ~ stage, data = d_RIGHT),
     col = 1:2)
```




```
plot(survfit(Surv(pfs_truncated, status_truncated) ~ stage, data = d_RIGHT),
     fun = "cloglog",
     col = 1:2)
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 1 x value <= 0 omitted
## from logarithmic plot
```




```
summary(coxph(Surv(pfs_truncated, status_truncated) ~ stage, data = d_RIGHT))
```

```
## Call:
## coxph(formula = Surv(pfs_truncated, status_truncated) ~ stage,
##     data = d_RIGHT)
##
##   n= 41, number of events= 30
##
##          coef exp(coef) se(coef)     z Pr(>|z|)
## stageM 1.0466    2.8479   0.5418 1.932   0.0534 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##        exp(coef) exp(-coef) lower .95 upper .95
## stageM     2.848     0.3511    0.9848     8.236
##
## Concordance= 0.598  (se = 0.035 )
## Likelihood ratio test= 4.71  on 1 df,   p=0.03
## Wald test            = 3.73  on 1 df,   p=0.05
## Score (logrank) test = 4.07  on 1 df,   p=0.04
```