

# Simulation Study

```
library(survival)
library(glmnet)

## Loading required package: Matrix
## Loading required package: foreach
## Loaded glmnet 2.0-16

library(polspline)
library(knitr)
library(EnvStats)

##
## Attaching package: 'EnvStats'

## The following object is masked from 'package:Matrix':
##
##      print
##
## The following objects are masked from 'package:stats':
##
##      predict, predict.lm
##
## The following object is masked from 'package:base':
##
##      print.default
```

## Simulating Survival Time with a Weibull Distribution

This function is based on `simulate_data` in <https://cran.r-project.org/web/packages/rsimsum/vignettes/relhaz.html>

```
## Simulate survival times with censoring, based on a weibull baseline hazard
##
## This function simulates survival times with censoring, according to a weibull
## baseline hazard that the user parameterizes. The survival/censoring times are
## simulated for user-given covariates and coefficients.
## @param x model matrix (including intercept) of x-values for the Cox model of survival times
## @param fcts_select subset of fcts from a hare object containing the coefficients of interest.
## @param params parameters shape and scale for the baseline Weibull distribution, by default the expon
## @param FUN random generation function for the distribution of censoring times, expected to be unifor
## @param ... arguments for FUN, the random generation function
## @return dataframe appending survival time and censoring indicator to the model matrix x
## @export
simulate_weibull <- function(x, fcts_select, params = list(shape = 1, scale = 1), FUN, ...) {

  n <- nrow(x)

  ## extract unique list of covariates selected
  cov_nums <- sort(fcts_select[,1][fcts_select[,1] != 0])
  cov_names <- colnames(x)[cov_nums]
  x_select <- x[,cov_names]
```

```

# extract the coefficient values from fcts_select
betas <- fcts_select[,5][fcts_select[,1] != 0]

# simulate survival times according to Bender et al. (2005)
u <- runif(n)
time <- (-log(u) / (params$scale * exp(x_select %*% betas)))^(1 / params$shape)

# Censoring
# **
cid <- sample(c(0, 1), size = n, replace = TRUE, prob = c(.75, .25))

time <- abs(time)

# return a dataframe
data.frame(time, cid, x, junk1 = rnorm(n), junk2 = rnorm(n), junk3 = rnorm(n), junk4 = rnorm(n), junk5 = rnorm(n))
}

```

```

load("actg175.RData")

x <- model.matrix( ~ trt + age + wtkg + hemo + drugs +
                  karnof + oprior + preanti + race +
                  gender + symptom + offtrt + cd40 +
                  cd80, actg175)[,-1]

# x <- readRDS("actg175_mat.rds")

nphm_hare <- readRDS("nphm_hare.rds")

# extracting the coefficients for basis functions
# that do not correspond to knots and/or tensor products
fcts <- nphm_hare$fcts
fcts_select <- fcts[fcts[,2] == 0 & is.na(fcts[,3]),]

set.seed(513)

sim_mat <- simulate_weibull(x, fcts_select, params = list(shape = 500, scale = 1))

summary(sim_mat$time)

```

```

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.0000  0.7928   1.2550   1.1828  1.5158   3.7156

```

```
head(sim_mat)
```

```

##           time cid trtZDV.ddi trtZDV.ZAL trtddi age   wtkg hemo1 drugs1
## 1502 1.1593974   0         0         0      1  30 64.6380     0      0
## 1432 0.8909326   1         0         0      0  36 74.3904     0      0
## 1832 0.6438830   0         0         1      0  38 78.0192     0      0
## 1331 1.2065819   0         0         1      0  43 64.4112     0      0
##  886 0.4364583   0         0         0      0  46 86.1000     0      1
##  221 1.4061854   1         0         0      0  38 69.5000     0      0
##           karnof oprior1 preanti race1 gender1 symptom1 offtrt1 cd40 cd80
## 1502     90      0      0      0      1      1      0  131  227
## 1432     80      0    381      0      1      1      1  173  724
## 1832     90      0   1099      0      1      0      0  416 1663

```

```
## 1331      70      0      29      0      1      1      1 164 944
## 886      90      0     1423      0      1      0      1 311 1867
## 221     100      1      113      1      1      0      0 375 986
##          junk1      junk2      junk3      junk4      junk5      junk6
## 1502  0.9469981 -0.72027888  0.9011602 -0.1536779 -0.6121733  0.9671346
## 1432  0.9222080 -0.08939882 -0.5224114 -0.4843940 -0.1925645  0.2427216
## 1832  0.1919309 -0.07347145 -0.6279713 -1.9205628 -0.4626918 -0.8194799
## 1331 -1.3880487  1.72703969 -0.1320258  0.3296765 -0.7375773 -0.4792051
## 886  -1.0154564 -0.87474633  0.2437329  0.5040923 -0.1949745 -0.3471337
## 221   0.1208223 -0.33058755  1.1561764 -1.0981282  1.4132886 -0.5645395
##          junk7      junk8      junk9      junk10
## 1502 -0.5512485  0.03130879 -1.32809134 -0.29256896
## 1432 -0.3001599 -0.44732937  0.01787963 -1.72277089
## 1832 -1.8384854 -0.65893814  0.56367410 -1.21106116
## 1331  0.3946907 -1.32633071  0.08993031 -0.64060782
## 886   0.1721040  0.84334703  0.40605657  1.63752720
## 221  -1.4993685 -0.73646868 -1.59608991 -0.03726863
```

## Coxph Simulation

```
phm_sim_mat <- coxph(Surv(time, cid) ~ ., data = sim_mat)
```

```
phm_sim_mat
```

```
## Call:
## coxph(formula = Surv(time, cid) ~ ., data = sim_mat)
##
##              coef exp(coef) se(coef)      z      p
## trtZDV.ddi  0.0055907  1.0056064  0.1387106   0.040 0.967850
## trtZDV.ZAL -0.5312810  0.5878514  0.1347144  -3.944 8.02e-05
## trtddi      -0.0064619  0.9935590  0.1341914  -0.048 0.961593
## age         0.0027636  1.0027674  0.0058699   0.471 0.637776
## wtkg        -0.0035083  0.9964978  0.0038358  -0.915 0.360393
## hemo1        0.0435492  1.0445114  0.1790380   0.243 0.807819
## drugs1       0.0632071  1.0652475  0.1436084   0.440 0.659839
## karnof       -0.0029426  0.9970617  0.0084085  -0.350 0.726370
## oprrior1    -0.1694123  0.8441608  0.3457121  -0.490 0.624107
## preanti     0.2704868  1.3106023  0.0108955  24.826 < 2e-16
## race1       0.1487414  1.1603729  0.1110168   1.340 0.180307
## gender1     -0.0205015  0.9797072  0.1388708  -0.148 0.882635
## symptom1    -0.4512960  0.6368023  0.1301839  -3.467 0.000527
## offtrt1     -0.4876660  0.6140580  0.1051608  -4.637 3.53e-06
## cd40        -0.3378537  0.7132996  0.0135886 -24.863 < 2e-16
## cd80        -0.0113355  0.9887285  0.0004691 -24.162 < 2e-16
## junk1       0.0145746  1.0146813  0.0496146   0.294 0.768944
## junk2      -0.0511066  0.9501774  0.0476101  -1.073 0.283073
## junk3       0.0190166  1.0191986  0.0476278   0.399 0.689690
## junk4      -0.0052038  0.9948097  0.0487176  -0.107 0.914935
## junk5      -0.0626196  0.9393008  0.0472149  -1.326 0.184752
## junk6      -0.0126170  0.9874623  0.0462242  -0.273 0.784890
## junk7      -0.0386363  0.9621006  0.0491513  -0.786 0.431827
## junk8      -0.0521978  0.9491411  0.0482793  -1.081 0.279625
```

```
## junk9      0.0071612  1.0071869  0.0485450   0.148 0.882725
## junk10     0.0078081  1.0078386  0.0483059   0.162 0.871591
##
## Likelihood ratio test=4762 on 26 df, p=< 2.2e-16
## n= 2139, number of events= 535
```

## Repeat Simulation Many times

```
set.seed(407)

num1 <- 0
num2 <- 0
num3 <- 0
num4 <- 0
num5 <- 0
num6 <- 0
num7 <- 0
num8 <- 0
num9 <- 0

for (i in 1:100) {

  sim <- simulate_weibull(x, fcts_select, params = list(shape = 500, scale = 1))
  phm_sim <- coxph(Surv(time, cid) ~ ., data = sim_mat)

  phm_sum <- summary(phm_sim)

  min(phm_sum$coefficients[17:26,5])

  phm_sum
}
```