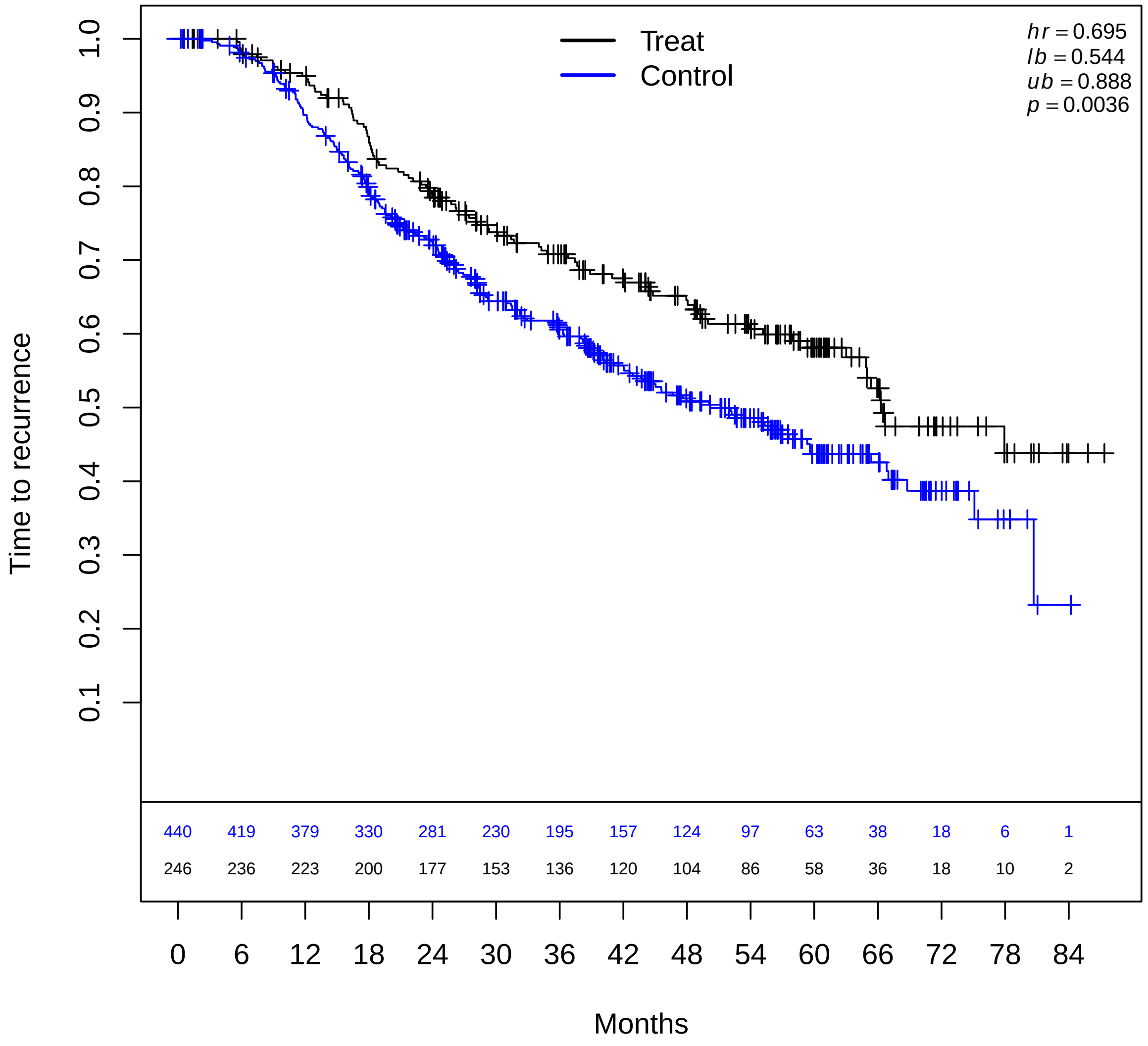
```r
rm(list = ls())

# Local github popOS codepath<-c('/media/larryleon/My
# Projects/GitHub/Forest-Search/R/')

# On MAC Mac Studio: M1 Ultra 2022, 64GB, 20 cores (16 performance, 4
# efficiency)
codepath <- c("/Users/larryleon/Documents/GitHub/Forest-Search/R/")

source(paste0(codepath, "source_forestsearch_v0.R"))
source_fs_functions(file_loc = codepath)

library(kableExtra)
library(knitr)
library(ggplot2)
library(gridExtra)
library(cubature)
# library(aVirtualTwins)
library(randomForest)
library(survival)
library(survminer)
library(grf)
library(policytree)
library(data.table)
library(plyr)
library(dplyr)
library(glmnet)

library(cli)
library(corrplot)
library(table1)
```

```r
maxFollow <- 84
cens.type <- "weibull"

# m1 -censoring adjustment
muC.adj <- log(1.5)
k.z3 <- 1
k.treat <- 0.9

z1_frac <- 0.25  # Default model index 'm1' (The 1st quartile of z1=er)
pH_super <- 0.125  # non-NULL re-defines z1_frac
```

```r
if (is.null(pH_super)) {
    # pH_check<-with(gbsg,mean(pgr<=quantile(pgr,c(z3_frac),1,0) &
    # er<=quantile(er,z1_frac)))
    pH_check <- with(gbsg, mean(meno == 0 & er <= quantile(er, z1_frac)))
    cat("Underlying pH_super", c(pH_check), "\n")
}
# pH_super specified If pH_super then override z1_frac and find z1_frac to
# yield pH_super

if (!is.null(pH_super)) {
    # Approximate Z1 quantile to yield pH proportion
    z1_q <- uniroot(propH.obj4, c(0, 1), tol = 1e-04, pH.target = pH_super)$root
    # pH_check<-with(gbsg,mean(pgr<=quantile(pgr,c(z3_frac),1,0) &
    # er<=quantile(er,z1_q)))
    pH_check <- with(gbsg, mean(meno == 0 & er <= quantile(er, z1_q)))
    cat("pH", c(pH_check), "\n")
    rel_error <- (pH_super - pH_check)/pH_super
    if (abs(rel_error) >= 0.1)
        stop("pH_super approximation relative error exceeds 10%")
    z1_frac <- z1_q
    cat("Underlying pH_super", c(pH_check), "\n")
}

## pH 0.122449
## Underlying pH_super 0.122449

# Bootstrap on log(hr) scale converted to HR (est.loghr=TRUE & est.scale='hr')
t.start.all <- proc.time()[3]
```

```r
######################## Forest search criteria

hr.threshold <- 1.25  # Initital candidates
hr.consistency <- 1  # Candidates for many splits
pconsistency.threshold <- 0.9
stop.threshold <- 0.95
maxk <- 2
nmin.fs <- 60
pstop_futile <- 0.7

# Limit timing for forestsearch
max.minutes <- 3
m1.threshold <- Inf  # Turning this off (Default)
# pconsistency.threshold<-0.70 # Minimum threshold (will choose max among
# subgroups satisfying)
fs.splits <- 400  # How many times to split for consistency
# vi is % factor is selected in cross-validation --> higher more important
# Null, turns off grf screening
d.min <- 10  # Min number of events for both arms (d0.min=d1.min=d.min)
# default=5 Virtual twins analysis Counter-factual difference (C-E) >=
# vt.threshold Large values in favor of C (control)
vt.threshold <- 0.225  # For VT delta
treat.threshold <- 0

maxdepth <- 2
n.min <- 60
ntree <- 1000
# GRF criteria
dmin.grf <- 12  # For GRF delta
# Note: For CRT this represents dmin.grf/2 RMS for control (-dmin.grf/2 for
# treatment)
frac.tau <- 0.6

outcome.name <- c("y.sim")
event.name <- c("event.sim")
id.name <- c("id")
treat.name <- c("treat")

cox.formula.sim <- as.formula(paste("Surv(y.sim,event.sim)~treat"))
cox.formula.adj.sim <- as.formula(paste("Surv(y.sim,event.sim)~treat+v1+v2+v3+v4+v5"))
```

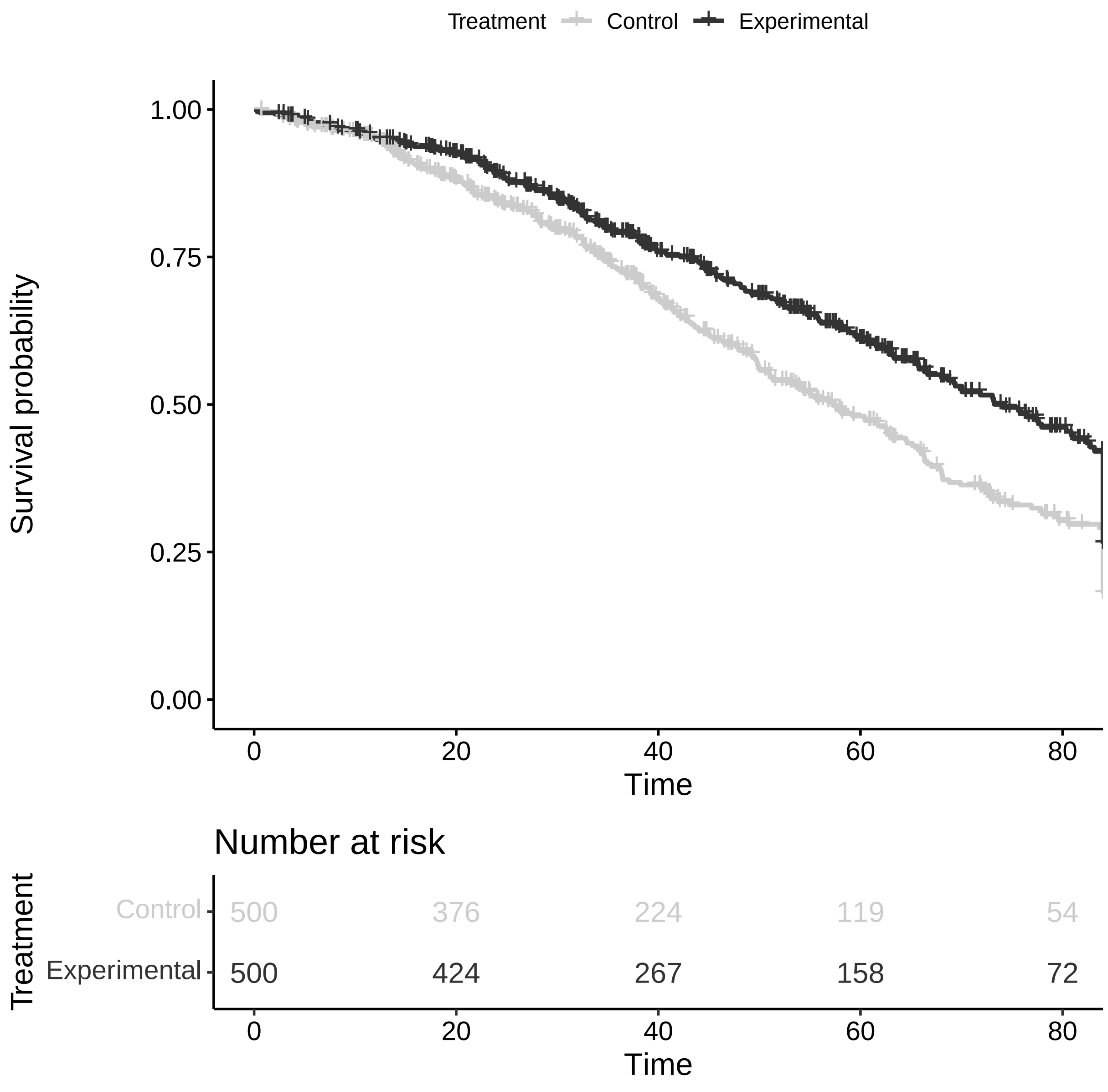
```
mod.harm <- "alt"
hrH.target <- 2
# out.loc = NULL turns off file creation
N <- 1000
this.dgm <- get.dgm4(mod.harm = mod.harm, N = N, k.treat = k.treat, hrH.target = hrH.ta
    cens.type = cens.type, out.loc = NULL, details = TRUE, parms_torand = FALSE)
```



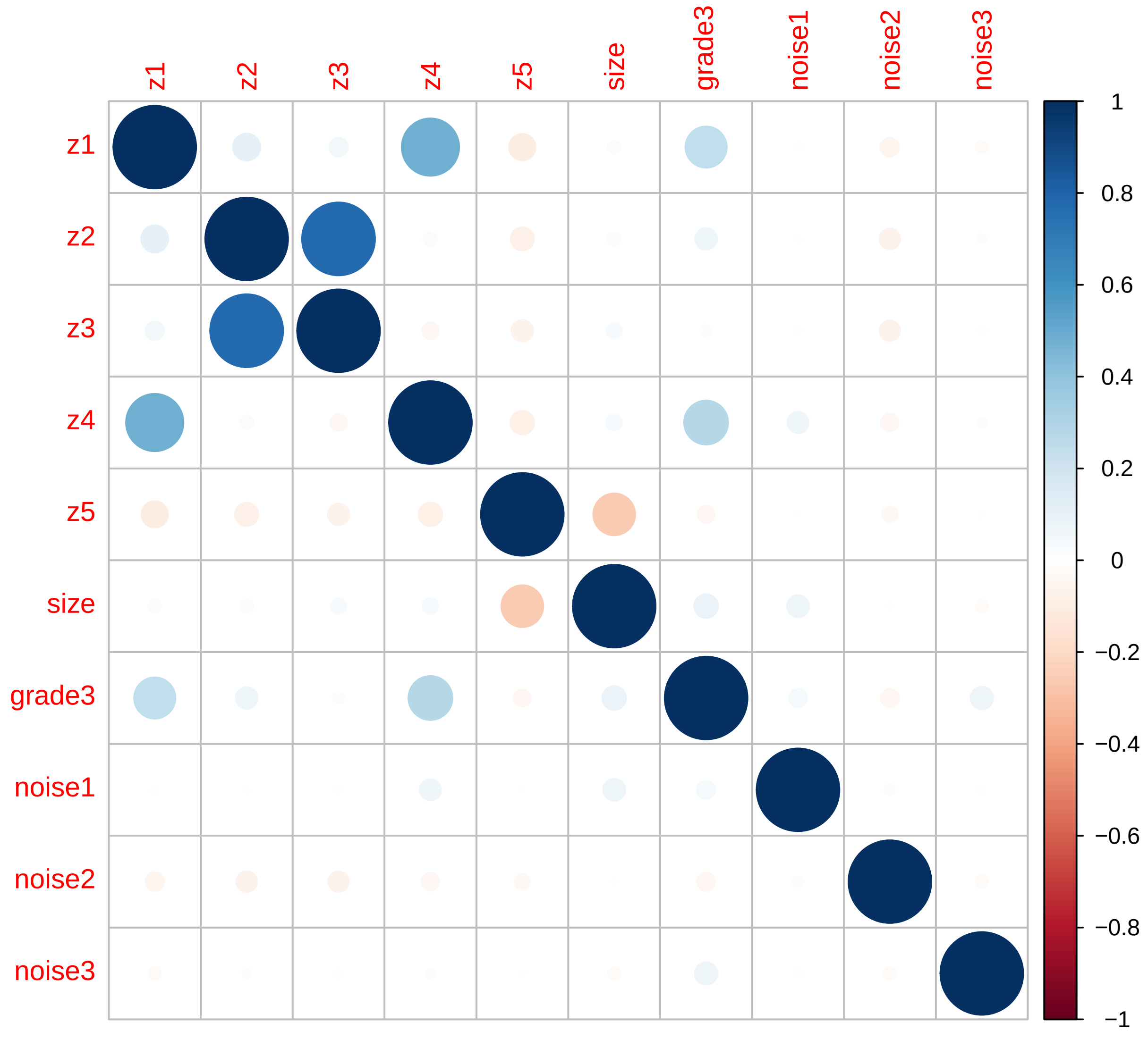
```
## Super-population empirical harm and non-harm hazard ratios= 2.000007 0.6466405
## Causal HR (empirical ITT)= 0.7057463

file_out <- NULL
# file_out <- c('output/sim-FS4_N1k_Noise=3_hrH=2_sim99.Rdata')
```

```
n_add_noise <- 3

sim <- 99
```

```r
dgm <- this.dgm$dgm

x <- sim_aftm4_gbsg(dgm = dgm, n = N, maxFollow = maxFollow, muC.adj = muC.adj, simid =

kmfit <- survfit(Surv(y.sim, event) ~ treat, data = x)

ggsurvplot(kmfit, data = x, main = "K-M curves for simulated data", legend = "top",
    legend.title = "Treatment", legend.labs = c("Control", "Experimental"), palette = "g
    risk.table = TRUE, risk.table.col = "strata")
```



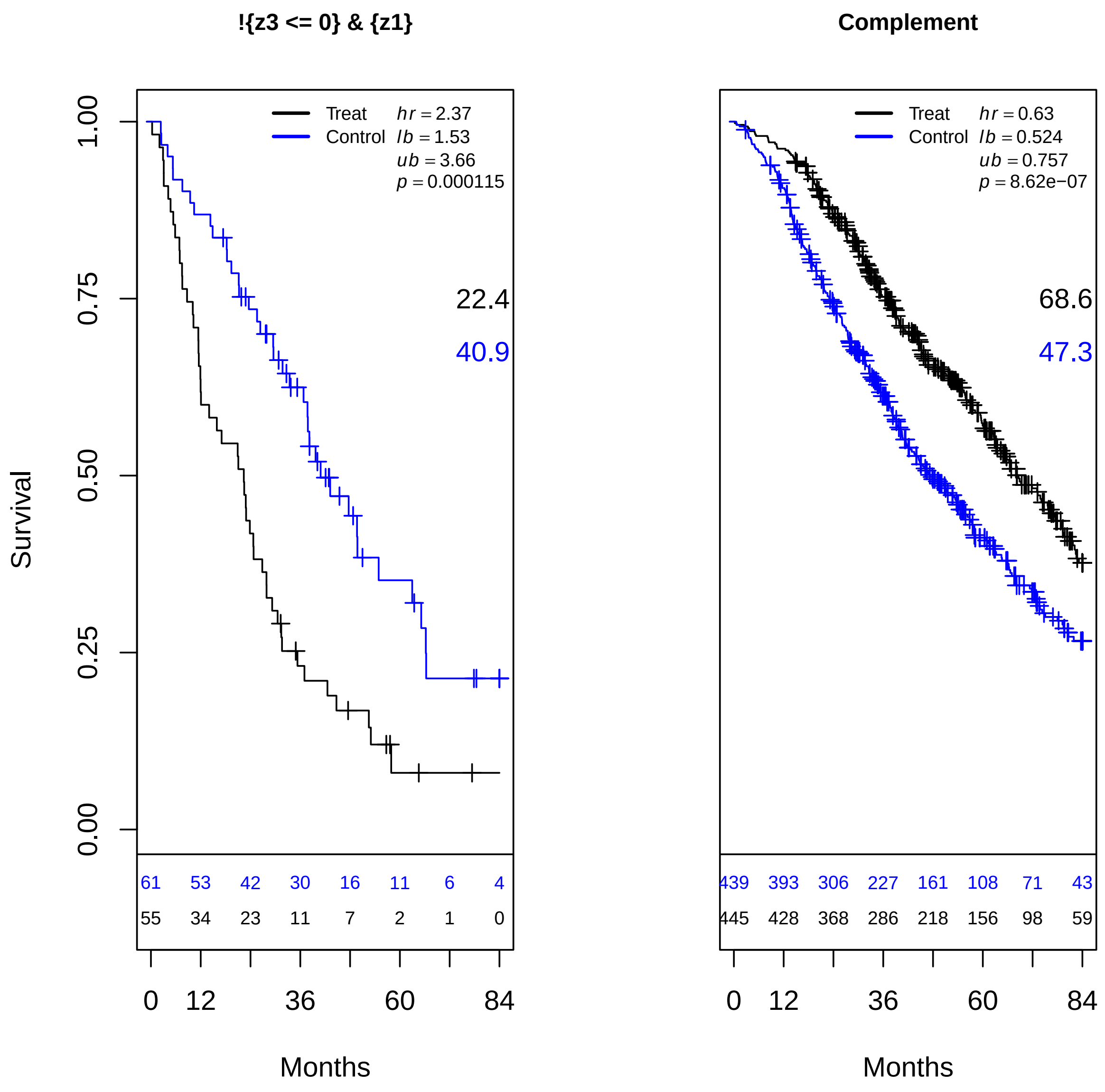
```
if (n_add_noise == 0) {
    confounders.name <- c("z1", "z2", "z3", "z4", "z5", "size", "grade3")
}

if (n_add_noise == 5) {
    set.seed(8316951 + 1000 * sim)
    # Add 5 noise
    x$noise1 <- rnorm(N)
    x$noise2 <- rnorm(N)
    x$noise3 <- rnorm(N)
    x$noise4 <- rnorm(N)
    x$noise5 <- rnorm(N)
    confounders.name <- c("z1", "z2", "z3", "z4", "z5", "size", "grade3", "noise1",
        "noise2", "noise3", "noise4", "noise5")
}

if (n_add_noise == 3) {
    set.seed(8316951 + 1000 * sim)
    # Add 3 noise
    x$noise1 <- rnorm(N, sd = 1)
    x$noise2 <- rnorm(N, sd = 1)
    x$noise3 <- rnorm(N, sd = 1)
    confounders.name <- c("z1", "z2", "z3", "z4", "z5", "size", "grade3", "noise1",
        "noise2", "noise3")
}


cox.formula.check <- as.formula(paste("Surv(y.sim,event.sim)~treat+z1+z2+z3+z4+z5+size+g
coxph(cox.formula.check, data = x)

## Call:
## coxph(formula = cox.formula.check, data = x)
##
##              coef  exp(coef)  se(coef)      z        p
## treat  -0.3407766  0.7112178  0.0882373 -3.862 0.000112
## z1      0.1454910  1.1566074  0.1074947  1.353 0.175905
## z2     -0.2671307  0.7655730  0.1447062 -1.846 0.064889
## z3      0.2355882  1.2656531  0.1481887  1.590 0.111883
## z4      0.5788410  1.7839696  0.1008861  5.738  9.6e-09
## z5     -0.9130836  0.4012849  0.0920592 -9.918  < 2e-16
## size   -0.0008376  0.9991627  0.0030755 -0.272 0.785347
## grade3 -0.1138499  0.8923919  0.1076257 -1.058 0.290132
## noise1  0.0565835  1.0582150  0.0447809  1.264 0.206386
## noise2 -0.0469701  0.9541160  0.0442398 -1.062 0.288365
## noise3  0.0063237  1.0063437  0.0402755  0.157 0.875237
##
## Likelihood ratio test=181.3  on 11 df, p=< 2.2e-16
## n= 1000, number of events= 546

Zm <- cor(as.matrix(x[, c(confounders.name)]))
corrplot(Zm)
```



```
suppressWarnings(table1(~z1 + z2 + z3 + z4 + z5 + size + grade3 + noise1 + noise2 +
    noise3 | treat, data = x))
```

	0	1	Overall
	(N=500)	(N=500)	(N=1000)
z1			
Mean (SD)	0.260 (0.439)	0.250 (0.433)	0.255 (0.436)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]
z2			
Mean (SD)	0.518 (0.500)	0.530 (0.500)	0.524 (0.500)
Median [Min, Max]	1.00 [0, 1.00]	1.00 [0, 1.00]	1.00 [0, 1.00]
z3			
Mean (SD)	0.422 (0.494)	0.402 (0.491)	0.412 (0.492)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]
z4			
Mean (SD)	0.512 (0.500)	0.478 (0.500)	0.495 (0.500)
Median [Min, Max]	1.00 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]
z5			
Mean (SD)	0.524 (0.500)	0.528 (0.500)	0.526 (0.500)
Median [Min, Max]	1.00 [0, 1.00]	1.00 [0, 1.00]	1.00 [0, 1.00]
size			
Mean (SD)	29.1 (14.7)	27.6 (12.7)	28.4 (13.7)
Median [Min, Max]	25.0 [3.00, 120]	25.0 [4.00, 100]	25.0 [3.00, 120]
grade3			
Mean (SD)	0.252 (0.435)	0.200 (0.400)	0.226 (0.418)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]
noise1			
Mean (SD)	0.0194 (0.993)	-0.0855 (0.983)	-0.0331 (0.989)
Median [Min, Max]	0.0198 [-3.14, 3.24]	-0.0674 [-3.65, 3.10]	-0.0172 [-3.65, 3.24]
noise2			
Mean (SD)	-0.00954 (0.953)	0.0251 (0.976)	0.00779 (0.964)
Median [Min, Max]	-0.0297 [-2.65, 2.95]	0.0771 [-3.77, 2.73]	0.0221 [-3.77, 2.95]
noise3			
Mean (SD)	0.0238 (1.05)	-0.0404 (0.998)	-0.00830 (1.02)
Median [Min, Max]	0.0381 [-3.07, 2.79]	-0.0336 [-3.00, 2.70]	0.000560 [-3.07, 2.79]

```
# Options Allconfounders.name is list of confounders within analysis dataset
# (1) use_lasso=TRUE & use_grf=FALSE Lasso used to possibly reduce dimension
# Any continuous factors are cut at medians (2) use_lasso=TRUE & use_grf=TRUE
# Lasso used to reduce dimension Continuous covariates are cut at medians
# However, if GRF selects a covariate cut then only that cut is used: For
# example if 'age <= median(ag)' is called for per Lasso but GRF includes 'age
# <= 54', then only the latter is used (3) use_grf_only = TRUE (overrides
# use_lasso and use_grf) Only factors selected via GRF are used (4) use_lasso =
# F & use_grf =T Median cuts (unless selected via GRF) as in (2) However no
# possible dimension reduction via lasso All categorical factors included (5)
# If all options set to false, then all factors are included with continuous
# factors cut at medians

use_lasso <- TRUE
use_grf <- TRUE
use_grf_only <- FALSE

fs.est <- forestsearch(df.analysis = x, Allconfounders.name = confounders.name, details
    use_lasso = use_lasso, use_grf = use_grf, use_grf_only = use_grf_only, dmin.grf = 12
    frac.tau = 0.6, conf_force = NULL, outcome.name = outcome.name, treat.name = treat.n
    event.name = event.name, id.name = id.name, n.min = nmin.fs, hr.threshold = hr.thres
    hr.consistency = hr.consistency, fs.splits = fs.splits, d0.min = d.min, d1.min = d.m
    pstop_futile = pstop_futile, pconsistency.threshold = pconsistency.threshold,
    stop.threshold = stop.threshold, max.minutes = max.minutes, maxk = maxk, by.risk = 1
    plot.sg = TRUE)

## ---------------------------------------------------------------------------------
## FS: GRF stage for cut selection with dmin,tau= 12 0.6
## ---------------------------------------------------------------------------------
## tau, maxdepth= 49.15213 2
##   leaf.node control.mean control.size control.se treated.mean treated.size
## 1         2     2.609069   189.000000   1.804834    -2.609069   189.000000
## 2         3    -4.271507   811.000000   1.011465     4.271507   811.000000
## 3         4    -7.455452   488.000000   1.215950     7.455452   488.000000
## 4         5     2.691612   100.000000   2.928641    -2.691612   100.000000
## 5         6    -3.447022   296.000000   1.554938     3.447022   296.000000
## 6         7    12.227063   116.000000   2.842148   -12.227063   116.000000
##   treated.se       diff Nsg depth
## 1   1.804834   5.218138 189     1
## 2   1.011465  -8.543015 811     1
## 3   1.215950 -14.910904 488     2
## 4   2.928641   5.383223 100     2
## 5   1.554938  -6.894044 296     2
## 6   2.842148  24.454126 116     2
##   leaf.node control.mean control.size control.se treated.mean treated.size
## 6         7    12.227063   116.000000   2.842148   -12.227063   116.000000
##   treated.se     diff Nsg depth
## 6   2.842148 24.45413 116     2
## ---------------------------------------------------------------------------------
## GRF subgroup found
## All splits
## [1] "z3 <= 0"        "noise3 <= 0.89" "z1 <= 0"
## Terminating node at max.diff (sg.harm.id)
## [1] "z1 <= 0"
## -----------------------------------------------------------------------------
## # of continuous/categorical characteristics 4 6
## Continuous characteristics: size noise1 noise2 noise3
## Categorical characteristics: z1 z2 z3 z4 z5 grade3
## CV lambda = 0.02782646
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                 s0
## z1      0.04420163
## z2      .
## z3      .
## z4      0.49151069
## z5     -0.77713914
## size    .
## grade3  .
## noise1  0.01624411
## noise2  .
## noise3  .
## Cox-LASSO selected: z1 z4 z5 noise1
## Cox-LASSO not selected: z2 z3 size grade3 noise2 noise3
## Median cuts after Lasso: noise1
## Categorical after Lasso: z1 z4 z5
## Factors per GRF: z3 <= 0 noise3 <= 0.89 z1 <= 0
## Medians prior to removing if also in GRF: noise1
## Factors after removing any duplicates also in GRF: noise1
## ***Factors per lasso after omitting GRF dups***=z1
## ***Factors per lasso after omitting GRF dups***=z4
## ***Factors per lasso after omitting GRF dups***=z5
## ***Factors per lasso after omitting GRF dups***=noise1 <= median(noise1)
## Initial GRF cuts included z3 <= 0 noise3 <= 0.89 z1 <= 0
## Factors included per GRF (not in lasso) z3 <= 0 noise3 <= 0.89
## -----------------------------------------------------------------------------
## # of candidate subgroup factors= 6
## [1] "noise1 <= median(noise1)" "noise3 <= 0.89"
## [3] "z1"                       "z4"
## [5] "z5"                       "z3 <= 0"
## -----------------------------------------------------------------------------
## ***FSdata completed***=
## LMAX= 6
## Confounders per grf screening q6 q1 q5 q3 q2 q4
##   FSconfounders.name      vi.cs
## 6                 q6 0.39160768
## 1                 q1 0.17613854
## 5                 q5 0.13321382
## 3                 q3 0.12723306
## 2                 q2 0.08827609
## 4                 q4 0.08353080
## Number of unique levels (L) and possible subgroups= 12 4095
## # of subgroups based on # variables > k.max and excluded (per million) 0.004017
## k.max= 2
## Events criteria for control,exp= 10 10
## # of subgroups with events less than criteria: control, experimental 7 7
## # of subgroups with sample size less than criteria 8
## # of subgroups meeting all criteria = 70
## # of subgroups fitted (Cox model estimable) = 70
## *Subgroup Searching Minutes=* 0.002
## Number of subgroups meeting HR threshold 2
## -------------------------------------------------------------------------------
## Subgroup candidate(s) found (FS)
## -------------------------------------------------------------------------------
## # of candidate subgroups (meeting HR criteria) =  2
## Subgroups (1st 10) meeting overall screening thresholds (HR, m1) sorted by HRs= Inf
##      K   n   E d1    m1    m0   HR L(HR) U(HR) q6.0 q6.1 q1.0 q1.1 q5.0 q5.1
##  1:  2 116  85 48 22.39 40.90 2.36  1.53  3.66    1    0    0    0    0    0
##  2:  2 194 135 65 29.65 40.52 1.30  0.93  1.83    1    0    0    0    0    0
##  3: NA  NA  NA NA    NA    NA   NA    NA    NA   NA   NA   NA   NA   NA   NA
##  4: NA  NA  NA NA    NA    NA   NA    NA    NA   NA   NA   NA   NA   NA   NA
##  5: NA  NA  NA NA    NA    NA   NA    NA    NA   NA   NA   NA   NA   NA   NA
##  6: NA  NA  NA NA    NA    NA   NA    NA    NA   NA   NA   NA   NA   NA   NA
##  7: NA  NA  NA NA    NA    NA   NA    NA    NA   NA   NA   NA   NA   NA   NA
##  8: NA  NA  NA NA    NA    NA   NA    NA    NA   NA   NA   NA   NA   NA   NA
##  9: NA  NA  NA NA    NA    NA   NA    NA    NA   NA   NA   NA   NA   NA   NA
## 10: NA  NA  NA NA    NA    NA   NA    NA    NA   NA   NA   NA   NA   NA   NA
##     q3.0 q3.1 q2.0 q2.1 q4.0 q4.1
##  1:    0    1    0    0    0    0
##  2:    0    0    0    0    0    1
##  3:   NA   NA   NA   NA   NA   NA
##  4:   NA   NA   NA   NA   NA   NA
##  5:   NA   NA   NA   NA   NA   NA
##  6:   NA   NA   NA   NA   NA   NA
##  7:   NA   NA   NA   NA   NA   NA
##  8:   NA   NA   NA   NA   NA   NA
##  9:   NA   NA   NA   NA   NA   NA
## 10:   NA   NA   NA   NA   NA   NA
## Consistency 1
## # of splits= 400
## Model, % Consistency Met= !{z3 <= 0} {z1} 1
## Number of subgroups meeting consistency criteria=
##    Pcons   N g m K        M.1  M.2
## 1:      1 116 1 1 2 !{z3 <= 0} {z1}
```



```
## [1] "!{z3 <= 0}" "{z1}"
## % consistency criteria met= 1
## SG focus= hr
## Subgroup Consistency Minutes= 0.012
## ------------------------------------------------------------------------------
## Subgroup found (FS)
## ------------------------------------------------------------------------------
## Minutes overall= 0.03741667

grf.est <- grf.subg.harm.survival(data = x, confounders.name = confounders.name,
    outcome.name = outcome.name, event.name = event.name, id.name = id.name, treat.name
    n.min = n.min, dmin.grf = dmin.grf, frac.tau = 0.6, details = TRUE)

## tau, maxdepth= 49.15213 2
##   leaf.node control.mean control.size control.se treated.mean treated.size
## 1         2     2.609069   189.000000   1.804834    -2.609069   189.000000
## 2         3    -4.271507   811.000000   1.011465     4.271507   811.000000
## 3         4    -7.455452   488.000000   1.215950     7.455452   488.000000
## 4         5     2.691612   100.000000   2.928641    -2.691612   100.000000
## 5         6    -3.447022   296.000000   1.554938     3.447022   296.000000
## 6         7    12.227063   116.000000   2.842148   -12.227063   116.000000
##   treated.se       diff Nsg depth
## 1   1.804834   5.218138 189     1
## 2   1.011465  -8.543015 811     1
## 3   1.215950 -14.910904 488     2
## 4   2.928641   5.383223 100     2
## 5   1.554938  -6.894044 296     2
## 6   2.842148  24.454126 116     2
##   leaf.node control.mean control.size control.se treated.mean treated.size
## 6         7    12.227063   116.000000   2.842148   -12.227063   116.000000
##   treated.se     diff Nsg depth
## 6   2.842148 24.45413 116     2
## ------------------------------------------------------------------------------
## GRF subgroup found
## All splits
## [1] "z3 <= 0"        "noise3 <= 0.89" "z1 <= 0"
## Terminating node at max.diff (sg.harm.id)
## [1] "z1 <= 0"
## ------------------------------------------------------------------------------

# plot(grf.est£tree)
```

```
library(doParallel)
registerDoParallel(parallel::detectCores(logical = FALSE))

cox.formula.boot <- cox.formula.sim
max.minutes <- 7
```

```
# Suggest running 50, first ... to get timing estimate

NB <- 1000

df_boot_analysis <- fs.est$df.est

fitH <- get_Cox_sg(df_sg = subset(df_boot_analysis, treat.recommend == 0), cox.formula =
H_obs <- fitH$est_obs  # log(hr) scale
seH_obs <- fitH$se_obs
# Hc observed estimates
fitHc <- get_Cox_sg(df_sg = subset(df_boot_analysis, treat.recommend == 1), cox.formula
Hc_obs <- fitHc$est_obs
seHc_obs <- fitHc$se_obs
rm("fitH", "fitHc")

Ystar_mat <- bootYstar({
    ystar <- get_Ystar(boot)
}, boots = NB, seed = 8316951, counter = "boot", export = fun_arg_list_boot)
# Check dimension
if (dim(Ystar_mat)[1] != NB | dim(Ystar_mat)[2] != nrow(df_boot_analysis)) stop("Dimensi

# Check 1st 5 bootstraps
ansB <- NULL
for (bb in 1:5) {
    boot <- bb
    ans <- fsboot_forparallel(boot)
    cat_line("***Bootstrap done, B***=", c(boot), col = "blue")
    print(ans)
    ansB <- rbind(ansB, c(bb, ans))
}

## ***Bootstrap done, B***=1
##    H_biasadj_1 H_biasadj_2 Hc_biasadj_1 Hc_biasadj_2 tmins_search max_sg_est
## 1:   0.7924622    0.500695   -0.4091576   -0.3382356   0.01913333   3.166213
## ***Bootstrap done, B***=2
##    H_biasadj_1 H_biasadj_2 Hc_biasadj_1 Hc_biasadj_2 tmins_search max_sg_est
## 1:    1.077061    1.067168   -0.6064418   -0.7229886   0.02041667   2.388492
## ***Bootstrap done, B***=3
##    H_biasadj_1 H_biasadj_2 Hc_biasadj_1 Hc_biasadj_2 tmins_search max_sg_est
## 1:   0.9415545   0.8986191   -0.5447398   -0.6132861   0.06728333   2.468732
## ***Bootstrap done, B***=4
##    H_biasadj_1 H_biasadj_2 Hc_biasadj_1 Hc_biasadj_2 tmins_search max_sg_est
## 1:   0.8727046   0.4818302    -0.514505   -0.6023089       0.0195   3.496084
## ***Bootstrap done, B***=5
##    H_biasadj_1 H_biasadj_2 Hc_biasadj_1 Hc_biasadj_2 tmins_search max_sg_est
## 1:    1.045569    1.266187   -0.4058867   -0.4289281       0.0017   1.896767

print(ansB)

##        H_biasadj_1 H_biasadj_2 Hc_biasadj_1 Hc_biasadj_2 tmins_search
## [1,] 1 0.7924622   0.500695    -0.4091576   -0.3382356   0.01913333
## [2,] 2 1.077061    1.067168    -0.6064418   -0.7229886   0.02041667
## [3,] 3 0.9415545   0.8986191   -0.5447398   -0.6132861   0.06728333
## [4,] 4 0.8727046   0.4818302   -0.514505    -0.6023089   0.0195
## [5,] 5 1.045569    1.266187    -0.4058867   -0.4289281   0.0017
##      max_sg_est
## [1,] 3.166213
## [2,] 2.388492
## [3,] 2.468732
## [4,] 3.496084
## [5,] 1.896767

tB.start <- proc.time()[3]
# Bootstraps
resB <- bootPar({
    ans <- fsboot_forparallel(boot)
}, boots = NB, seed = 8316951, counter = "boot", export = fun_arg_list_boot)
tB.now <- proc.time()[3]
tB.min <- (tB.now - tB.start)/60

doParallel::stopImplicitCluster()

cat("Minutes for Boots", c(NB, tB.min), "\n")

## Minutes for Boots 1000 7.9841

cat("Projection per 1000", c(tB.min * (1000/NB)), "\n")

## Projection per 1000 7.9841

cat("Propn bootstrap subgroups found =", c(sum(!is.na(resB$H_biasadj_1))/NB), "\n")

## Propn bootstrap subgroups found = 0.966

# How many timmed out
cat("Number timmed out=", c(sum(is.na(resB$H_biasadj_1) & resB$tmins_search > max.minute
    "\n")

## Number timmed out= 0
```

```
H_estimates <- get_dfRes(Hobs = H_obs, seHobs = seH_obs, H1_adj = resB$H_biasadj_1,
    H2_adj = resB$H_biasadj_2, ystar = Ystar_mat, cov_method = "standard", cov_trim = 0)
Hc_estimates <- get_dfRes(Hobs = Hc_obs, seHobs = seHc_obs, H1_adj = resB$Hc_biasadj_1,
    H2_adj = resB$Hc_biasadj_2, ystar = Ystar_mat, cov_method = "standard", cov_trim = (

print(H_estimates)

##          H0      sdH0 H0_lower H0_upper       H1       sdH1 H1_lower H1_upper
## 1: 2.364979 0.5277364 1.527161 3.662434 2.100723 0.09880031 1.915735 2.303573
##          H2      sdH2 H2_lower H2_upper
## 1: 2.032208 0.5519679 1.193365 3.460692

print(Hc_estimates)

##           H0       sdH0  H0_lower  H0_upper        H1       sdH1  H1_lower
## 1: 0.6298563 0.05917019 0.5239352 0.7571909 0.6357446 0.02794041 0.5832746
##     H1_upper        H2       sdH2  H2_lower  H2_upper
## 1: 0.6929345 0.6341271 0.08657696 0.4852464 0.8286866

bootit <- list(H_estimates = H_estimates, Hc_estimates = Hc_estimates)
ans <- fsBoot.parallel.out(df = df_boot_analysis, FSboots = bootit, cox.formula.boot = c
    sim_number = sim)
tall.min <- (tB.now - t.start.all)/60

cat("Overall minutes for analysis", c(tall.min), "\n")

## Overall minutes for analysis 8.332567

# save(dgm, x, fs.est, grf.est, bootit, ans, tall.min, resB, cox.formula.boot,
# file=file_out)
```

```
## Confounders evaluated in Forestsearch:
## [1] "noise1 <= median(noise1)" "noise3 <= 0.89"
## [3] "z1"                       "z4"
## [5] "z5"                       "z3 <= 0"
```

Table 1: Simulated data example: Cox hazard ratio (hr) estimates for the ITT population and subgroups H and H^c. Cox model estimates are based on subgroups: H true (knowing the actual subgroup, a-priori); the estimated subgroup $\hat{H}$; and the boostrap $(1,000$ resamples) bias-correction to $\hat{H}$ estimates, denoted $\hat{H}_{bc}$. Estimates for the complement H^c are defined analogously. The number of subjects in each population (# Subjects) and the number of subjects correctly classified (# Correct) in the true subgroups H and H^c, respectively, are listed.

	Causal(adj)	Estimate	Lower	Upper	# Subjects	# Correct
ITT	0.795	0.733	0.619	0.867	1000	.
H subgroup estimates						
H_{oracle}	2.249	2.365	1.527	3.663	116	.
$\hat{H}$	2.249	2.365	1.527	3.663	116	116
$\hat{H}_{bc}$	2.249	2.032	1.193	3.461	116	116
H-complement subgroup estimates						
H^c_{oracle}	0.604	0.63	0.524	0.757	884	.
$\hat{H}^c$	0.604	0.63	0.524	0.757	884	884
$\hat{H}^c_{bc}$	0.604	0.634	0.484	0.83	884	884