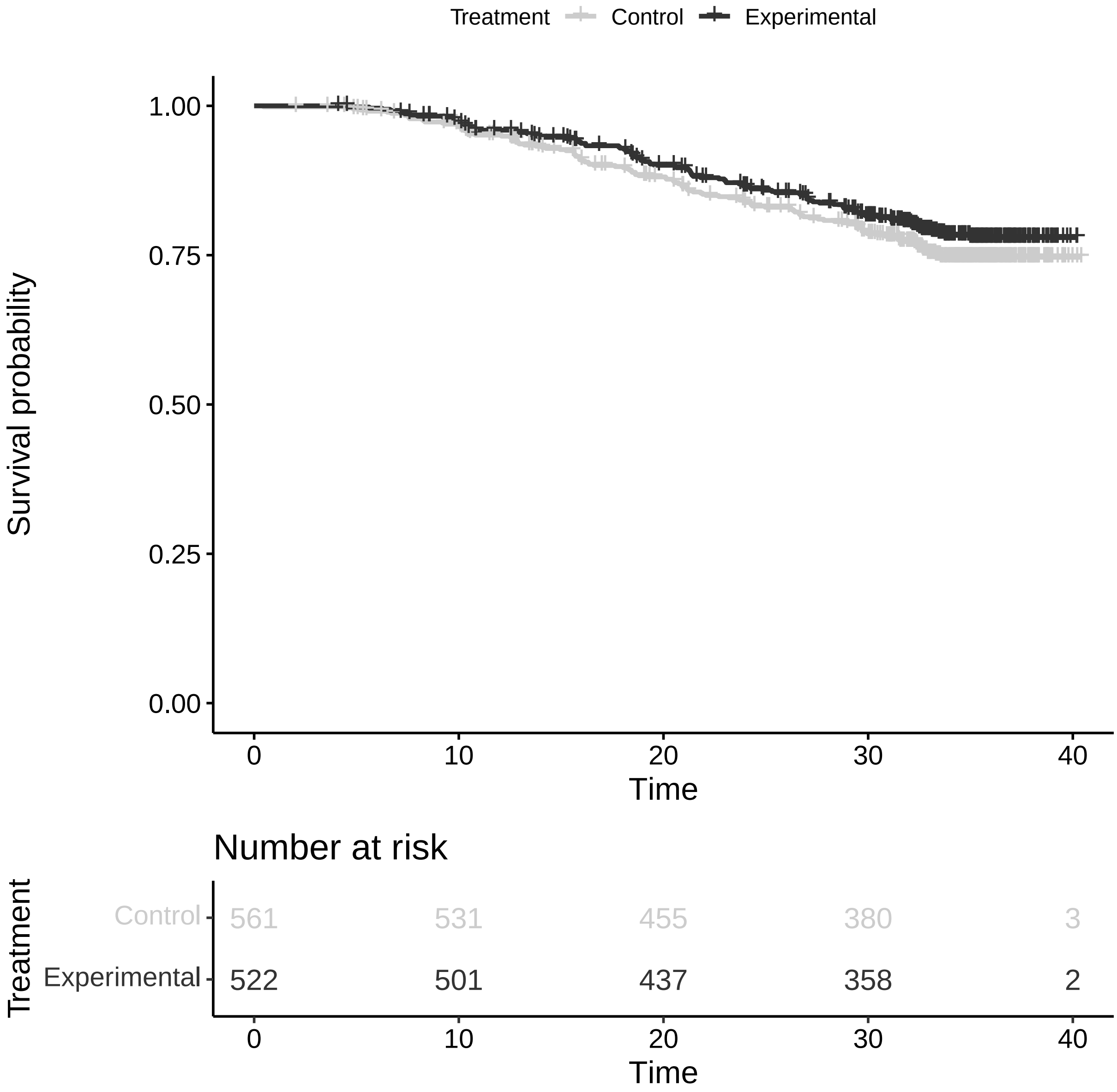
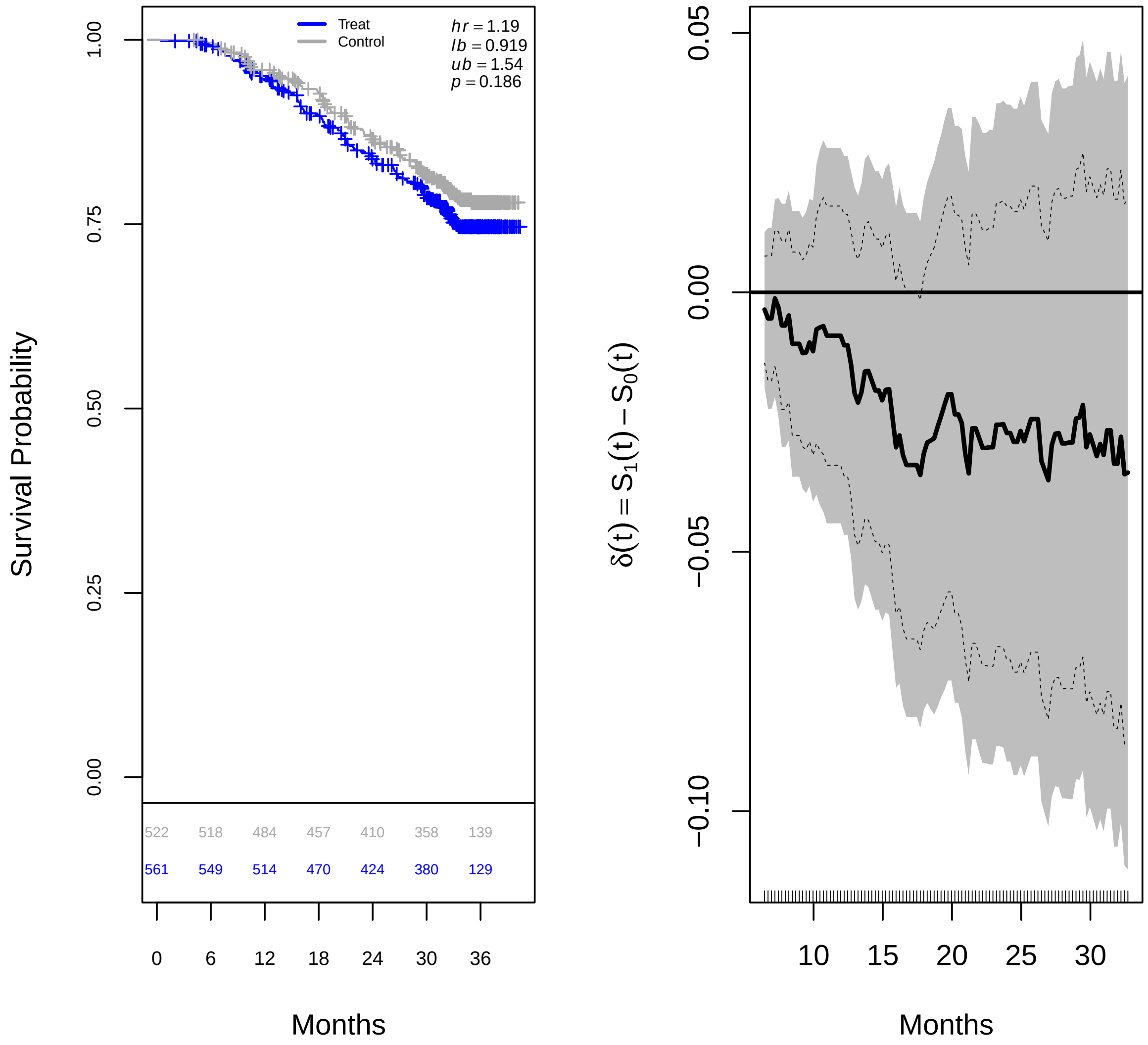
```r
rm(list = ls())

# Local github (on Linux) codepath<-c('/media/larryleon/My
# Projects/GitHub/Forest-Search/R/') On MAC Mac Studio: M1 Ultra 2022, 64GB, 20
# cores (16 performance, 4 efficiency)

# Revise paths as needed (Originally run in private directory) This file will
# be placed in public github forestSearch

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(corrplot)

library(table1)
library(cli)  # for colors in cat

library(speff2trial)
```

```r
t.start.all <- proc.time()[3]
df.analysis <- subset(ACTG175, arms %in% c(1, 3))

# Here we are 'reversing the roles of treatment' looking for large benefit by
# looking for harm with treatment reversed

df.analysis <- within(df.analysis, {
    id <- as.numeric(c(1:nrow(df.analysis)))
    time_months <- days/30.4375
    treat <- ifelse(arms == 3, 1, 0)
    # arm1 for 1 vs 3
    arm1 <- ifelse(arms == 1, 1, 0)
})

# plot(survfit(Surv(time_months,cens)~treat,data=df.analysis))
# coxph(Surv(time_months,cens)~ arm1, data=df.analysis)

survfit(Surv(time_months, cens) ~ treat, data = df.analysis)

## Call: survfit(formula = Surv(time_months, cens) ~ treat, data = df.analysis)
##
##           n events median 0.95LCL 0.95UCL
## treat=0 522    103     NA      NA      NA
## treat=1 561    128     NA      NA      NA

coxph(Surv(time_months, cens) ~ treat, data = df.analysis)

## Call:
## coxph(formula = Surv(time_months, cens) ~ treat, data = df.analysis)
##
##         coef exp(coef) se(coef)     z     p
## treat 0.1751    1.1913   0.1324 1.323 0.186
##
## Likelihood ratio test=1.76  on 1 df, p=0.1849
## n= 1083, number of events= 231

confounders.name <- c("age", "wtkg", "karnof", "cd40", "cd80", "hemo", "homo", "drugs",
    "race", "gender", "oprior", "zprior", "symptom", "preanti", "str2", "z30")

outcome.name <- c("time_months")
event.name <- c("cens")
id.name <- c("id")
treat.name <- c("treat")

kmfit <- survfit(Surv(time_months, cens) ~ arm1, data = df.analysis)

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



```
# Calculate RMST
itt_tab <- SGtab(df = df.analysis, SG_flag = "ITT", outcome.name = outcome.name,
    event.name = event.name, treat.name = treat.name, draws = 1000)

par(mfrow = c(1, 2))
plot_twosample(df = df.analysis, tte.name = outcome.name, treat.name = treat.name,
    event.name = event.name, col.treat = "blue", col.control = "darkgrey", ylab = "Survi
    xlab = "Months", show.Y.axis = TRUE, byrisk = 6, show.med = FALSE, legend.cex = 0.5
    risk.cex = 0.5, censor.cex = 0.7, cox.cex = 0.55, cex_Yaxis = 0.65)

plotband_survdiff(res = itt_tab$rmst_fit)
```

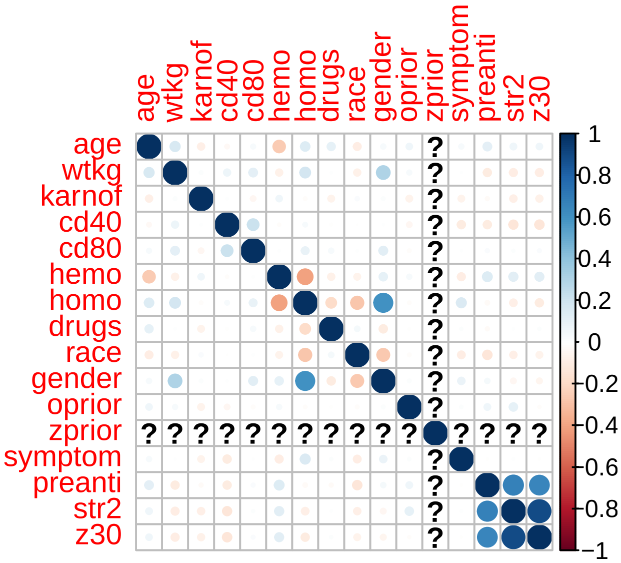


```
print(itt_tab$res_out)

##      Subgroup n      n1    n0    m1 m0 HR (95% CI)        RMST (95% CI)
## [1,] "ITT"    "1083" "561" "522" NA NA "1.19 (0.92,1.54)" "-0.63 (-1.45,0.19)"

cat("RMSTs for individual arms (m1,m0)", c(itt_tab$rmst_fit$m1L, itt_tab$rmst_fit$m0L),
    "\n")

## RMSTs for individual arms (m1,m0) 30.17336 30.8021

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

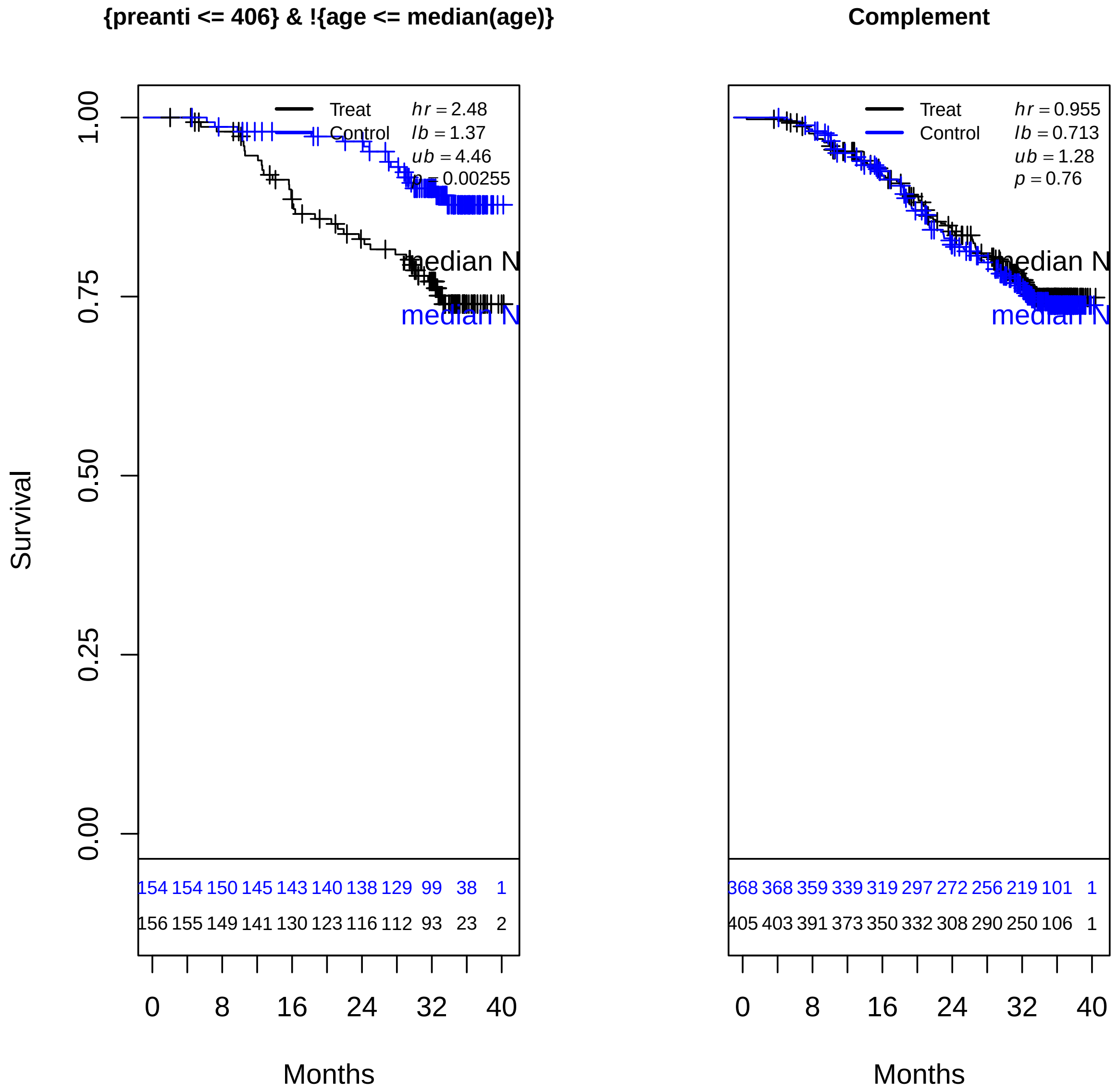
```
suppressWarnings(table1(~age + wtkg + karnof + cd40 + cd80 + hemo + homo + drugs +
    race + gender + oprior + symptom | treat, data = df.analysis))
```

	0	1	Overall
	(N=522)	(N=561)	(N=1083)
age			
Mean (SD)	35.2 (8.70)	35.1 (8.51)	35.2 (8.60)
Median [Min, Max]	34.0 [12.0, 69.0]	34.0 [12.0, 70.0]	34.0 [12.0, 70.0]
wtkg			
Mean (SD)	74.9 (13.6)	74.9 (13.0)	74.9 (13.3)
Median [Min, Max]	74.4 [31.0, 160]	74.4 [41.1, 129]	74.4 [31.0, 160]
karnof			
Mean (SD)	95.5 (5.76)	95.1 (5.95)	95.3 (5.86)
Median [Min, Max]	100 [80.0, 100]	100 [70.0, 100]	100 [70.0, 100]
cd40			
Mean (SD)	349 (130)	347 (114)	348 (122)
Median [Min, Max]	336 [0, 1200]	340 [110, 911]	338 [0, 1200]
cd80			
Mean (SD)	1000 (488)	972 (503)	988 (496)
Median [Min, Max]	917 [40.0, 3390]	890 [177, 5010]	895 [40.0, 5010]
hemo			
Mean (SD)	0.0824 (0.275)	0.0873 (0.283)	0.0849 (0.279)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]
homo			
Mean (SD)	0.663 (0.473)	0.676 (0.469)	0.669 (0.471)
Median [Min, Max]	1.00 [0, 1.00]	1.00 [0, 1.00]	1.00 [0, 1.00]
drugs			
Mean (SD)	0.140 (0.347)	0.123 (0.329)	0.131 (0.338)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]
race			
Mean (SD)	0.264 (0.441)	0.308 (0.462)	0.287 (0.453)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]
gender			
Mean (SD)	0.831 (0.375)	0.838 (0.369)	0.835 (0.372)
Median [Min, Max]	1.00 [0, 1.00]	1.00 [0, 1.00]	1.00 [0, 1.00]
oprior			
Mean (SD)	0.0172 (0.130)	0.0160 (0.126)	0.0166 (0.128)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]
symptom			
Mean (SD)	0.184 (0.388)	0.171 (0.377)	0.177 (0.382)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]

```
# Searching for large positive effects Treatment roles reversed

hr.threshold <- 1/0.6  # Initital candidates
hr.consistency <- 1.25  # 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
# vi.grf.min <- (-1)*0.2 Use default which is -0.2 so that factors are NOT
# excluded via VI Null, turns off grf screening

d.min <- 10  # Min number of events for both arms (d0.min=d1.min=d.min)
# default=5

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

# Now run with stop.threshold
fs.est <- forestsearch(df.analysis = df.analysis, Allconfounders.name = confounders.name
    details = TRUE, use_lasso = use_lasso, use_grf = use_grf, use_grf_only = use_grf_onl
    conf_force = c("karnof <= mean(karnof)", "age <= median(age)"), dmin.grf = 4,
    frac.tau = 0.8, maxk = maxk, max_n_confounders = 11, sg_focus = "Nsg_only", stop.thr
    grf_depth = 2, outcome.name = outcome.name, treat.name = treat.name, event.name = ev
    id.name = id.name, n.min = nmin.fs, hr.threshold = hr.threshold, hr.consistency = hr
    fs.splits = fs.splits, d0.min = d.min, d1.min = d.min, pstop_futile = pstop_futile,
    pconsistency.threshold = pconsistency.threshold, max.minutes = max.minutes, by.risk
    plot.sg = TRUE)
```



```
## --------------------------------------------------------------------------------
## FS: GRF stage for cut selection with dmin,tau= 4 0.8
## --------------------------------------------------------------------------------
## tau, maxdepth= 26.83532 2
##   leaf.node control.mean control.size  control.se treated.mean treated.size
## 1         2    0.9025897  814.0000000   0.3067431   -0.9025897  814.0000000
## 2         3   -0.6852263  269.0000000   0.6135849    0.6852263  269.0000000
## 3         4   -1.1872087  121.0000000   0.8531539    1.1872087  121.0000000
## 4         5    2.3750908  206.0000000   0.6767198   -2.3750908  206.0000000
## 5         6    1.0449298  490.0000000   0.3679299   -1.0449298  490.0000000
## 6         7   -1.1550748  266.0000000   0.6030144    1.1550748  266.0000000
##    treated.se      diff Nsg depth
## 1   0.3067431  1.805179 814     1
## 2   0.6135849 -1.370453 269     1
## 3   0.8531539 -2.374417 121     2
## 4   0.6767198  4.750182 206     2
## 5   0.3679299  2.089860 490     2
## 6   0.6030144 -2.310150 266     2
##   leaf.node control.mean control.size  control.se treated.mean treated.size
## 4         5    2.3750908  206.0000000   0.6767198   -2.3750908  206.0000000
##    treated.se     diff Nsg depth
## 4   0.6767198 4.750182 206     2
## ---------------------------------------------------------------------------
## GRF subgroup found
## All splits
## [1] "wtkg <= 68.04"  "age <= 29"      "preanti <= 406"
## Terminating node at max.diff (sg.harm.id)
## [1] "age <= 29"
## ---------------------------------------------------------------------------
## # of continuous/categorical characteristics 6 10
## Continuous characteristics: age wtkg karnof cd40 cd80 preanti
## Categorical characteristics: hemo homo drugs race gender oprior zprior symptom str2 z
## CV lambda = 0.015266
## 16 x 1 sparse Matrix of class "dgCMatrix"
##                    s0
## age      .
## wtkg     .
## karnof  -0.0199521891
## cd40    -0.0028808609
## cd80     0.0001699201
## hemo     .
## homo     0.0234755013
## drugs    .
## race     .
## gender   .
## oprior   .
## zprior   .
## symptom  0.4945436841
## preanti  0.0001747329
## str2     .
## z30      .
## Cox-LASSO selected: karnof cd40 cd80 homo symptom preanti
## Cox-LASSO not selected: age wtkg hemo drugs race gender oprior zprior str2 z30
## Median cuts after Lasso: karnof cd40 cd80 preanti
## Categorical after Lasso: homo symptom
## Factors per GRF: wtkg <= 68.04 age <= 29 preanti <= 406
## Medians prior to removing if also in GRF: karnof cd40 cd80 preanti
## ***cMed_flag***=preanti
## ***to_exclude***=FALSE
## ***to_exclude***=FALSE
## ***to_exclude***=FALSE
## ***to_exclude***=TRUE
## ***conf.cont_medians***=karnof
## ***conf.cont_medians***=cd40
## ***conf.cont_medians***=cd80
## Factors after removing any duplicates also in GRF: karnof cd40 cd80
## ***Factors per lasso after omitting GRF dups***=homo
## ***Factors per lasso after omitting GRF dups***=symptom
## ***Factors per lasso after omitting GRF dups***=karnof <= median(karnof)
## ***Factors per lasso after omitting GRF dups***=cd40 <= median(cd40)
## ***Factors per lasso after omitting GRF dups***=cd80 <= median(cd80)
## Initial GRF cuts included wtkg <= 68.04 age <= 29 preanti <= 406
## Factors included per GRF (not in lasso) wtkg <= 68.04 age <= 29 preanti <= 406
## Dropping variables (flag_drop2) karnof <= median(karnof)
## --------------------------------------------------------------------------------
## # of candidate subgroup factors= 9
## [1] "cd40 <= median(cd40)"   "cd80 <= median(cd80)"   "wtkg <= 68.04"
## [4] "age <= 29"              "preanti <= 406"         "karnof <= mean(karnof)"
## [7] "age <= median(age)"     "homo"                   "symptom"
## --------------------------------------------------------------------------------
## At least 1 variable dropped
## ***A variable was dropped (check)***=
## ***FSdata completed***=
## LMAX= 9
## Confounders per grf screening q5 q3 q9 q7 q4 q8 q1 q2 q6
##   FSconfounders.name      vi.cs
## 5                 q5 0.20493841
## 3                 q3 0.13507666
## 9                 q9 0.11182226
## 7                 q7 0.11177660
## 4                 q4 0.11030479
## 8                 q8 0.09668145
## 1                 q1 0.08591628
## 2                 q2 0.07430446
## 6                 q6 0.06917910
## Number of unique levels (L) and possible subgroups= 18 262143
## # of subgroups based on # variables > k.max and excluded (per million) 0.261972
## k.max= 2
## Events criteria for control,exp= 10 10
## # of subgroups with events less than criteria: control, experimental 17 13
## # of subgroups with sample size less than criteria 13
## # of subgroups meeting all criteria = 151
## # of subgroups fitted (Cox model estimable) = 151
## *Subgroup Searching Minutes=* 0.02208333
## Number of subgroups meeting HR threshold 15
## --------------------------------------------------------------------------------
## Subgroup candidate(s) found (FS)
## --------------------------------------------------------------------------------
## # of candidate subgroups (meeting HR criteria) =  15
## Subgroups (1st 10) meeting overall screening thresholds (HR, m1) sorted by HRs= Inf
##     K   n  E d1 m1 m0   HR L(HR) U(HR) q5.0 q5.1 q3.0 q3.1 q9.0 q9.1 q7.0 q7.1
##  1: 2 488 88 55 NA NA 1.72  1.12  2.65    0    1    0    0    0    0    0    0
##  2: 2 310 52 36 NA NA 2.48  1.37  4.46    0    1    0    0    0    0    1    0
##  3: 2 263 41 26 NA NA 1.67  0.88  3.15    0    0    0    0    0    0    1    0
##  4: 2 246 53 36 NA NA 1.83  1.03  3.26    0    0    0    0    0    0    1    0
##  5: 2 240 41 27 NA NA 2.17  1.14  4.14    0    0    0    0    0    0    0    0
##  6: 2 233 67 43 NA NA 1.79  1.08  2.94    0    0    0    0    0    0    1    0
##  7: 2 206 52 34 NA NA 2.08  1.17  3.68    0    0    0    1    0    0    0    0
##  8: 2 206 39 24 NA NA 1.74  0.91  3.32    0    0    0    0    0    0    0    0
##  9: 2 192 42 29 NA NA 1.84  0.96  3.54    0    0    0    1    0    0    0    0
## 10: 2 154 33 21 NA NA 1.82  0.89  3.70    0    0    0    1    0    0    0    0
##     q4.0 q4.1 q8.0 q8.1 q1.0 q1.1 q2.0 q2.1 q6.0 q6.1
##  1:    1    0    0    0    0    0    0    0    0    0
##  2:    0    0    0    0    0    0    0    0    0    0
##  3:    0    0    0    0    1    0    0    0    0    0
##  4:    0    0    0    0    0    0    0    1    0    0
##  5:    1    0    1    0    0    0    0    0    0    0
##  6:    0    0    0    0    0    0    0    0    0    1
##  7:    1    0    0    0    0    0    0    0    0    0
##  8:    0    0    1    0    0    0    0    1    0    0
##  9:    0    0    0    0    0    0    0    1    0    0
## 10:    0    0    1    0    0    0    0    0    0    0
## Consistency 0.855
## Consistency 0.9875
## # of splits= 400
## Model, % Consistency Met= {preanti <= 406} !{age <= median(age)} 0.9875
## Number of subgroups meeting consistency criteria=
##     Pcons   N g m K              M.1                   M.2
## 1: 0.9875 310 4 2 2 {preanti <= 406} !{age <= median(age)}
```



```
## [1] "{preanti <= 406}"      "!{age <= median(age)}"
## % consistency criteria met= 0.9875
## SG focus= Nsg_only
## Subgroup Consistency Minutes= 0.02483333
## --------------------------------------------------------------------------------
## Subgroup found (FS)
## --------------------------------------------------------------------------------
## Minutes overall= 0.06913333
```

```
file_out <- NULL
# file_out <- c('output/actg_1-vs-3_results_Boots=2k_v0B.Rdata')

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

cox.formula.boot <- as.formula(paste("Surv(time_months,cens)~treat"))
max.minutes <- 6

# Suggest running 20, first ... to get timing estimate

NB <- 2000

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("Dimens

# Check 1st bootstrap
ansB <- NULL
for (bb in 1:1) {
    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:   0.9133076    0.671855  -0.02870733     -0.17853       0.2503   2.912498
```

```
print(ansB)

##        H_biasadj_1 H_biasadj_2 Hc_biasadj_1 Hc_biasadj_2 tmins_search
## [1,] 1 0.9133076   0.671855    -0.02870733  -0.17853     0.2503
##      max_sg_est
## [1,] 2.912498

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 2000 43.18335

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

## Projection per 1000 21.59167

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

## Propn bootstrap subgroups found = 0.886

# 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
    est.scale = "1/hr")
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 =
    est.scale = "1/hr")

print(H_estimates)
```

```
##           H0      sdH0  H0_lower  H0_upper        H1       sdH1  H1_lower
## 1: 0.4037082 0.1213374 0.2239921 0.7276163 0.4640309 0.06288109 0.3557952
##     H1_upper        H2      sdH2  H2_lower  H2_upper
## 1: 0.6051927 0.4907615 0.1426285 0.2776441 0.8674662

print(Hc_estimates)

##         H0      sdH0  H0_lower H0_upper        H1       sdH1  H1_lower H1_upper
## 1: 1.04673 0.1565406 0.7807929 1.403245 0.9602521 0.03714221 0.8901457  1.03588
##           H2      sdH2  H2_lower H2_upper
## 1: 0.9668173 0.1657666 0.6908772 1.352969

bootit <- list(H_estimates = H_estimates, Hc_estimates = Hc_estimates)

tall.min <- (tB.now - t.start.all)/60

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

## Overall minutes for analysis 43.72657

if (!is.null(file_out)) save(df.analysis, fs.est, bootit, tall.min, resB, cox.formula.bo
    file = file_out)
```

```
## H un-adjusted estimates-----:   0.4 (95% CI=0.22,0.73)
## H bias-corrected estimates--:   0.49 (95% CI=0.28,0.87)
## H^c un-adjusted estimates---:   1.05 (95% CI=0.78,1.4)
## H^c bias-corrected estimates:   0.97 (95% CI=0.69,1.35)
```