BIOST544_HW2_AWolf

Aaron Wolf

January 18, 2017

```
library(data.table)
library(ggplot2)
library(dplyr)
set.seed(1)

adaptive.trial = fread('~/Documents/Dropbox/2016:2017/BIOST544/HW2/HW2-adaptive-trial.txt')
```

1. Write a function that takes in clinical data, permutes the data based on tx/control status, and calculates difference in response proportion. Is there a *REAL* difference in the proportion responders for the tx group v. the ctl group, or is this just noise.

```
prop.tx.resp.empir = adaptive.trial %>% filter(tx==1) %>% summarise(resp.prop = mean(outcome)) %
>% unlist()
prop.ctl.resp.empir = adaptive.trial %>% filter(tx==0) %>% summarise(resp.prop = mean(outcome)) %
>% unlist()

prop.diff.resp.empir = prop.tx.resp.empir - prop.ctl.resp.empir
```

```
permute.resp.prop.fn=function(data){
  data.mut = data %>% mutate(tx.permute = sample(x = c(0,1), size = nrow(data), replace = TRUE))
  prop.tx.resp = data.mut %>% filter(tx.permute==1) %>% summarise(resp.prop = mean(outcome)) %>%
unlist()
  prop.ctl.resp = data.mut %>% filter(tx.permute==0) %>% summarise(resp.prop = mean(outcome)) %>%
 unlist()
  prop.diff.resp = prop.tx.resp - prop.ctl.resp
  return(as.data.table(prop.diff.resp))
}

simulation.resp.fn = function(data, nsimulations){
  sim = replicate(n = nsimulations, expr = permute.resp.prop.fn(data))
  sim = as.data.table(t(as.data.table(sim)))

  prop.tx.resp.empir = data %>% filter(tx==1) %>% summarise(resp.prop = mean(outcome)) %>%
unlist()
  prop.ctl.resp.empir = data %>% filter(tx==0) %>% summarise(resp.prop = mean(outcome)) %>% unlis
t()

  prop.diff.resp.empir = prop.tx.resp.empir - prop.ctl.resp.empir

  pval = mean(sim$V1 >= prop.diff.resp.empir)
  return(pval)
}
```

2.

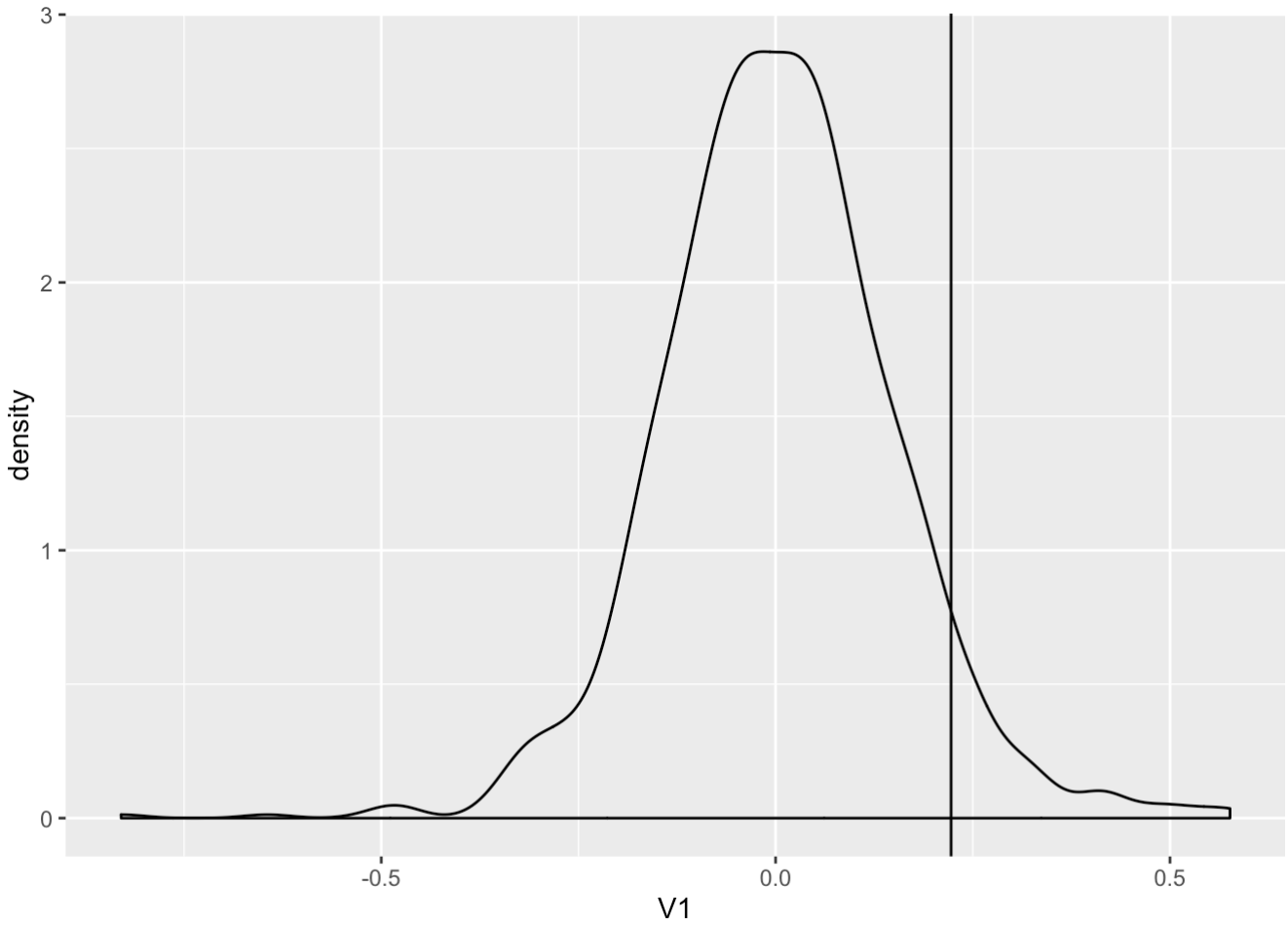
a. Write a function that takes in clinical data with information on tx/ctl, outcome, and order of enrollment. The function should rerandomize tx/ctl assignment based on the randomization probability function and then assess the difference in response proportion for *randomized.tx* v. *randomized.ctl*. The probability of assignment to the *new_tx* arm increases with the number of successfully treated tx individuals, and increases with the number of unsuccessfully treated ctl

individuals. The randomization probability is dependent on the order of enrollment, so randomizing the order of enrollment will then randomize the assignment of tx/ctl for individuals, while keeping thier outcome constant.

```
randomize.resp.prop.fn = function(data){
  PNEW = numeric()
  PNEW = c(PNEW, 0.5)
  data.mut = data %>% mutate(order.rand = sample(x = 1:nrow(data), size = nrow(data), replace=FAL
SE)) %>% arrange(order.rand) # randomize the enrollment order
  data.mut.new = data.mut[1, tx.rand:= sample(x = c(0,1), size = nrow(data.mut), replace = TRUE,
prob =c((1-PNEW),PNEW))] # randomly assigne a new treatment category to the first individual

  nsuccess = 0 # set counts
  nfail = 0
  npatients = 0

  for(i in 2:nrow(data.mut.new)){ # starting now on the second row
    line = data.mut.new[i-1] # look at the outcome and treatment of the previous individual
    if( (line$tx.rand==1) & (line$outcome==1) ){ # success_new = number of success on the newly a
ssigned tx arm
      nsuccess = nsuccess + 1
      } else if ((line$tx.rand==0) & (line$outcome==0) ) { # fail_old = number of failures on the
 newly assigned ctl arm
      nfail = nfail + 1
      }
    npatients = npatients + 1 # number of patients enrolled so far (not including the current on
e)
    pnew = (1 + 3*(nsuccess + nfail)) / (2 + 3*npatients)
    PNEW = c(PNEW, pnew)
    data.mut.new[i, tx.rand:=sample(x = c(0,1), size = nrow(data.mut), replace = TRUE, prob
=c((1-pnew),pnew))]
  }

  prop.tx.resp = data.mut.new %>% filter(tx.rand==1) %>% summarise(resp.prop = mean(outcome)) %>%
 unlist()
  prop.ctl.resp = data.mut.new %>% filter(tx.rand==0) %>% summarise(resp.prop = mean(outcome)) %
>% unlist()
  prop.diff.resp = prop.tx.resp - prop.ctl.resp
  return(as.data.table(prop.diff.resp))

}
```

b. using adaptive.trial data, evaluate if the data are consistent with the hypothesis that standard of care is at least as effective as the new treatment. I perform randomization of the enrollment order, and reassign tx/ctl status, then calculate difference in response proportion for the tx.rand and ctl.rand groups and compare with the empirical difference in response proportions.

```
s = replicate(n = 1000, expr = randomize.resp.prop.fn(data = adaptive.trial))

s = as.data.table(t(as.data.table(s)))

ggplot() + geom_density(data=s, aes(x=V1)) + geom_vline(xintercept = prop.diff.resp.empir)
```

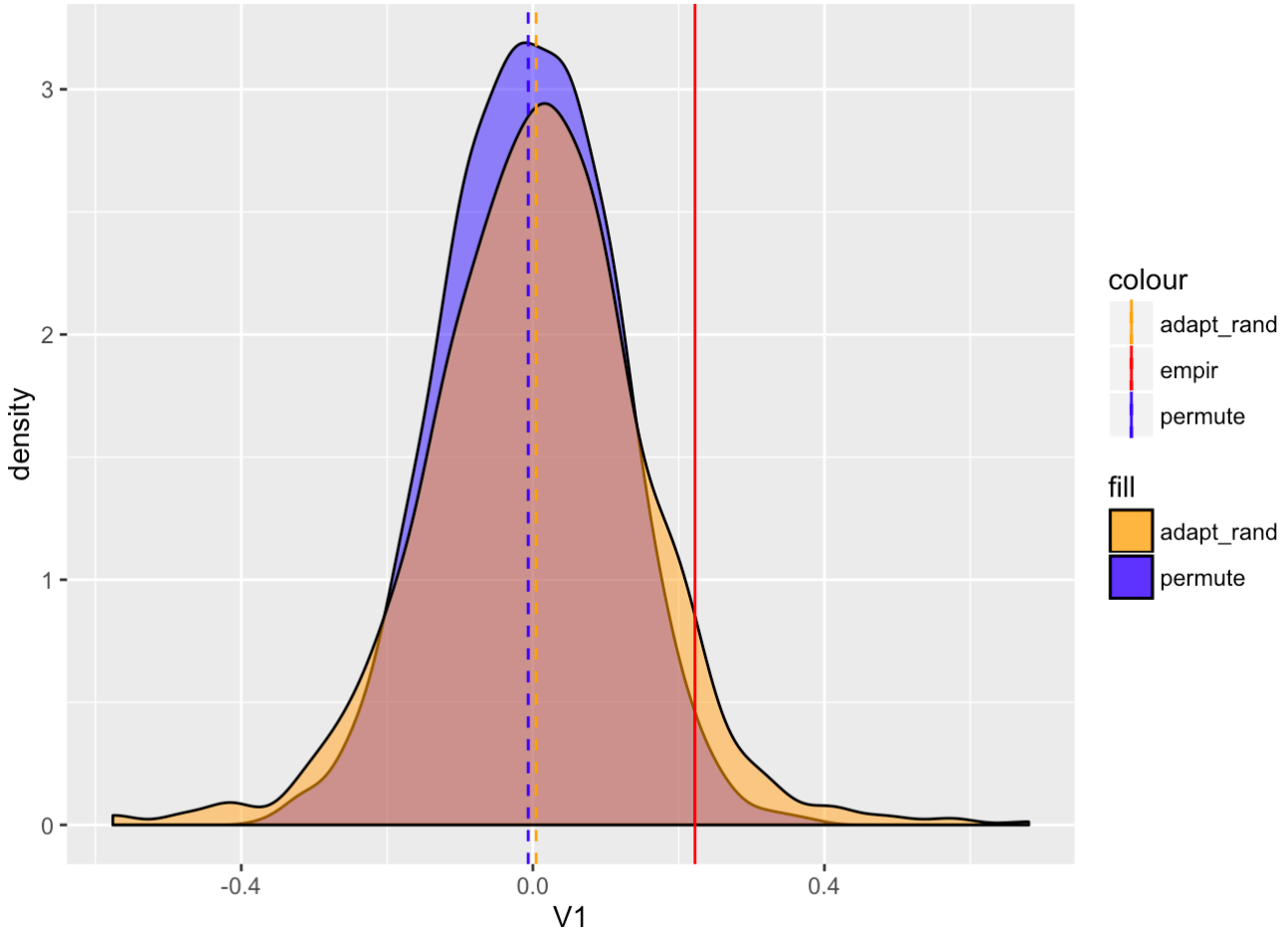



```
mean(s$V1 <= prop.diff.resp.empir)
```

```
## [1] 0.938
```

3. How do the sample distributions of resp.tx - resp.ctl differe between the permutation method and the adaptive_randomization method when applied to the trial data? This distributions appear to overlap almost identically, with the medians being nearly idistinguishable. The empirical response difference is greater than ~95% of the simulated data using both methods.

```
permuted.diff.resp.dt = as.data.table(t(as.data.table(replicate(n = 1000, expr = permute.resp.pro
p.fn(data = adaptive.trial)))))

randomized.diff.resp.dt = as.data.table(t(as.data.table(replicate(n = 1000, expr = randomize.res
p.prop.fn(data = adaptive.trial)))))

ggplot() +
  geom_density(data=permuted.diff.resp.dt, aes(x=V1, fill='permute'), alpha=0.5) +
  geom_density(data=randomized.diff.resp.dt, aes(x=V1, fill='adapt_rand'), alpha=0.5) +
  geom_vline(aes(xintercept = prop.diff.resp.empir, color='empir')) +
  geom_vline(aes(xintercept = mean(permuted.diff.resp.dt$V1), color='permute'),
linetype='dashed') +
  geom_vline(aes(xintercept = mean(randomized.diff.resp.dt$V1), color='adapt_rand'), linetype='da
shed') +
  scale_color_manual(values = c('permute' = 'blue', 'adapt_rand' = 'orange', 'empir'='red')) +
  scale_fill_manual(values = c('permute' = 'blue', 'adapt_rand' = 'orange', 'empir'='red'))
```




```
mean(permuted.diff.resp.dt$V1 <= prop.diff.resp.empir)
```

```
## [1] 0.981
```

```
mean(randomized.diff.resp.dt$V1 <= prop.diff.resp.empir)
```

```
## [1] 0.945
```