

Homework 3 on Bootstrap and resampling methods

Leave your name and uni here

Due: 04/15/2019, Wednesday, by 1pm

Please implement parallel computing into all your bootstrap algorithm and R codes.

Problem 1

Example 1: a randomized trial on eye treatment (See Lecture 7.pdf, page 2)

For the two sample trial, the null hypothesis is:

$$H_0 : \mu_{red} = \mu_{blue}$$

Corresponding t statistics:

$$t(X_1, X_2) = \frac{\bar{X}_2 - \bar{X}_1}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}}$$

```
# data
blue <- c(4,69,87,35,39,79,31,79,65,95,68,62,70,80,84,79,66,75,59,77,36,86,39,85,74,72,69,85,85,72)
red <-c(62,80,82,83,0,81,28,69,48,90,63,77,0,55,83,85,54,72,58,68,88,83,78,30,58,45,78,64,87,65)
acui = data.frame(str=c(rep(0,20),rep(1,10)),red,blue)
```

```
teststat1 <- function(x, y) {
  numerator = mean(y) - mean(x)
  denominator = sqrt(var(x)/length(x) + var(y)/length(y))
  return(numerator/denominator)
}

boottest <- function(x, y, nboot=200) {
  combmean <- mean(c(x,y))
  # The mean of the combined sample
  teststatvec <- rep(NA, nboot)
  adjx <- x - mean(x) + combmean
  # The adjusted X's will have mean=combmean
  adjy <- y - mean(y) + combmean
  # The adjusted X's will have mean=combmean
  for(b in 1:nboot)
    teststatvec[b] <- teststat1(sample(adjx, replace=T),
                               sample(adjy, replace=T))
  return(list(bootpval = sum(teststat1(x,y) < teststatvec)/
            nboot, teststatvec = teststatvec))
}
```

```
two_sample = acui[which(acui$str==1),]
```

Not using parallel computing

```
set.seed(123)
system.time({
  res1 = boottest(two_sample$blue, two_sample$red)
})
```

```
##      user  system elapsed
##    0.120   0.013   0.164
```

```
mean(res1$teststatvec)
```

```
## [1] 0.03825936
```

The mean of t is 0.0022 which is smaller than 0.05, so we fail to reject null hypothesis and conclude that two treatment do not have different effect.

Using parallel computing

```
set.seed(123)
library(parallel)
nCores<-detectCores() # detect numbers of available cores
cl = makeCluster(nCores)
system.time({
  res2 = boottest(two_sample$blue, two_sample$red)
})
```

```
##      user  system elapsed
##    0.029   0.000   0.029
```

```
stopCluster(cl)
mean(res2$teststatvec)
```

```
## [1] 0.03825936
```

We can find that when using parallel computing, we need less time. The mean of t is 0.0038 which is smaller than 0.05, so we fail to reject null hypothesis and conclude that two treatment do not have different effect.

For paired comparison trial, let d_i equals to the difference of visual acuity between two eyes. Then the null hypothesis:

$$H_0 : \bar{d} = 0$$

Corresponding t statistics:

$$t = \frac{\bar{d} - 0}{\frac{s}{\sqrt{n}}}$$

```
teststat2 <- function(d) {
  numerator = mean(d)
  denominator = sqrt(var(d)/length(d))
  return(numerator/denominator)
}
boottest2 <- function(d, nboot=200) {
  teststatvec <- rep(NA, nboot)
  adjd <- d - mean(d)
  for(b in 1:nboot)
    teststatvec[b] <- teststat2(sample(adjd, replace=T))
  return(list(bootpval = sum(teststat2(b) < teststatvec)/
    nboot, teststatvec = teststatvec))
}
```

```
paired_sample = acui[which(acui$str==0),]
d = paired_sample$red-paired_sample$blue
```

Not using parallel computing

```
set.seed(123)
system.time({
```

```
res1 = boottest2(d)
})
```

```
##      user  system elapsed
##    0.025   0.001   0.026
```

```
mean(res1$teststatvec)
```

```
## [1] 0.01563833
```

The mean of t is 0.016 which is larger than 0.005, so we reject null hypothesis and conclude that two treatment have different effect in according to paired comparison.

Using parallel computing

```
set.seed(123)
library(parallel)
nCores<-detectCores() # detect numbers of available cores
cl = makeCluster(nCores)
system.time({
  res2 = boottest2(d)
})
```

```
##      user  system elapsed
##    0.018   0.000   0.018
```

```
stopCluster(cl)
mean(res2$teststatvec)
```

```
## [1] 0.01563833
```

We can find that when using parallel computing, we need less time. The mean of t is 0.016 which is larger than 0.05, so we reject null hypothesis and conclude that two treatment have different effect in according to paired comparison.

Problem 2

Example 2 Number of modes of a density (See Lecture 7.pdf, page 4)

```
#R codes:
```

Problem 3

Recall in the lecture of EM algorithm, we studied an ABO blood type data, where we have $N_{obs} = (N_A, N_B, N_{AB}, N_O) = (26, 27, 42, 7)$, and designed EM algorithm to estimate the allele frequencies, P_A, P_B and P_O .

Please design a bootstrap algorithm to estimate the variances of the estimated \hat{P}_A, \hat{P}_B and \hat{P}_O . Implement your algorithms in R, and present your results..

Answer: your answer starts here...

```
#R codes:
```