HW 9 Bootstrap and Jackknife

STAT 5400

Due: Nov 1, 2024 9:30 AM

Problems

To help you prepare the midterm, the solution of this homework will be posted right after the deadline. Late homework will not be accepted without exceptions.

Submit your solutions as an .Rmd file and accompanying .pdf file.

1. Use `echo=TRUE, include=FALSE` to ensure that all the code are provided but only the important output is included. Try to write your homework in the form of a neat report and don't pile up any redundant and irrelevant output.
2. *Always interpret your result whenever it is necessary.* Try to make sure the interpretation can be understood by people with a moderate level of statistics knowledge.

Reading assignments.

Here is an undergraduate-level introduction to the bootstrap. https://statweb.stanford.edu/~tibs/stat315a/Supplements/bootstrap.pdf

Problems

1. Bootstrap and jackknife Consider the airconditioning data listed below:

`3, 5, 7, 18, 43, 85, 91, 98, 100, 130, 230, 487.`

Suppose the mean of the underlying distribution is μ and our interest is to estimate $\log(\mu)$. To estimate it, we use the log of the sample mean, i.e., $\log(\bar{X})$, as an estimator.

(a) Carry out a nonparametric bootstrap analysis to estimate the bias of $\log(\bar{X})$.

```r
set.seed(5400)

aircon <- c(3, 5, 7, 18, 43, 85, 91, 98, 100, 130, 230, 487)

# sample estimate of log X bar
rhat <- log(mean(aircon))

n <- length(aircon); B <- 200
boot.bias <- rep(NA, B)
```

```r
for (b in seq(B)) {
  id <- sample(n,n, replace = TRUE)
  boot.bias[b] <- log(mean(aircon[id]))
  boot.bias[b] <- boot.bias[b]- rhat
}

(bias <- mean(boot.bias))
```

```
## [1] -0.07729246
```

(b) Based on the bootstrap analysis, is the bias of $\log(\bar{X})$ positive or negative? (In other word, does $\log(\bar{X})$ overestimates or underestimates $\log(\mu)$) Can you explain the observation? (Hint: Jensen's inequality)

It is negative, i.e. it systematically underestimates log of mu which follows from Jensen's inequality for concave functions: $\log(E[X]) \geq E[\log(X)]$

(c) Also run a nonparametric bootstrap to estimate the standard error of the log of the sample mean. In terms of the mean square error of the estimator, do you think the bias is large given the standard error?

```r
set.seed(5400)
B <- 200; n <- length(aircon)
boot.rep <- replicate(B, log(mean(aircon[sample(n, replace=TRUE) ])))
(boot.se <- sd(boot.rep))
```

```
## [1] 0.373746
```

Compared to the standard error the Bias is quite small indicating that the standard deviation is a more important problem and bootstrap minimizes the bias well

(d) Carry out a parametric bootstrap analysis to estimate the bias of the log of sample mean. Assume that the population distribution of failure times of airconditioning equipment is exponential.

```r
set.seed(5400)

library(boot)
```

```
## Warning: package 'boot' was built under R version 4.3.3
```

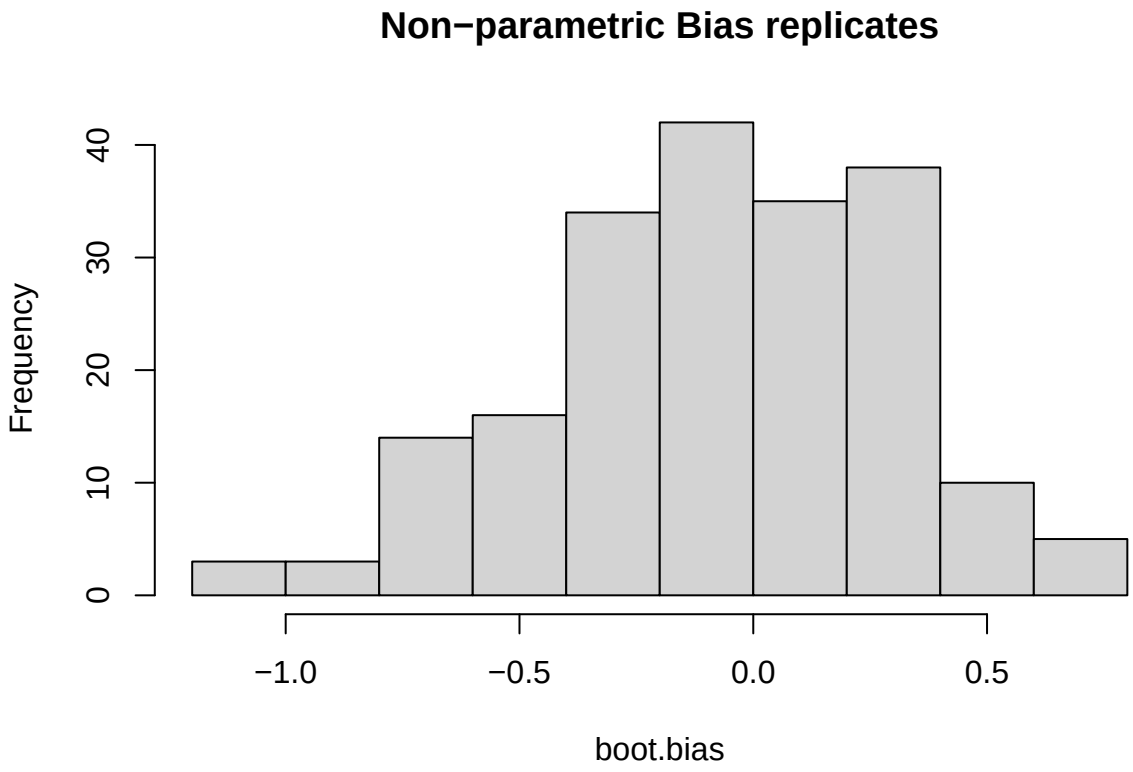
```r
B <- 200

n <- length(aircon)

# sample mean
xbar = mean(aircon)

# sample log mean
rhat <- log(mean(aircon))

boot.par.bias <- rep(NA, 200)
```

```r
for (b in seq(B)) {
   boot.par.bias[b] <- log(mean(rexp(n=n,rate = 1/xbar)))
   boot.par.bias[b] <- boot.par.bias[b] - rhat

}

(bias_par <- mean(boot.par.bias))
```

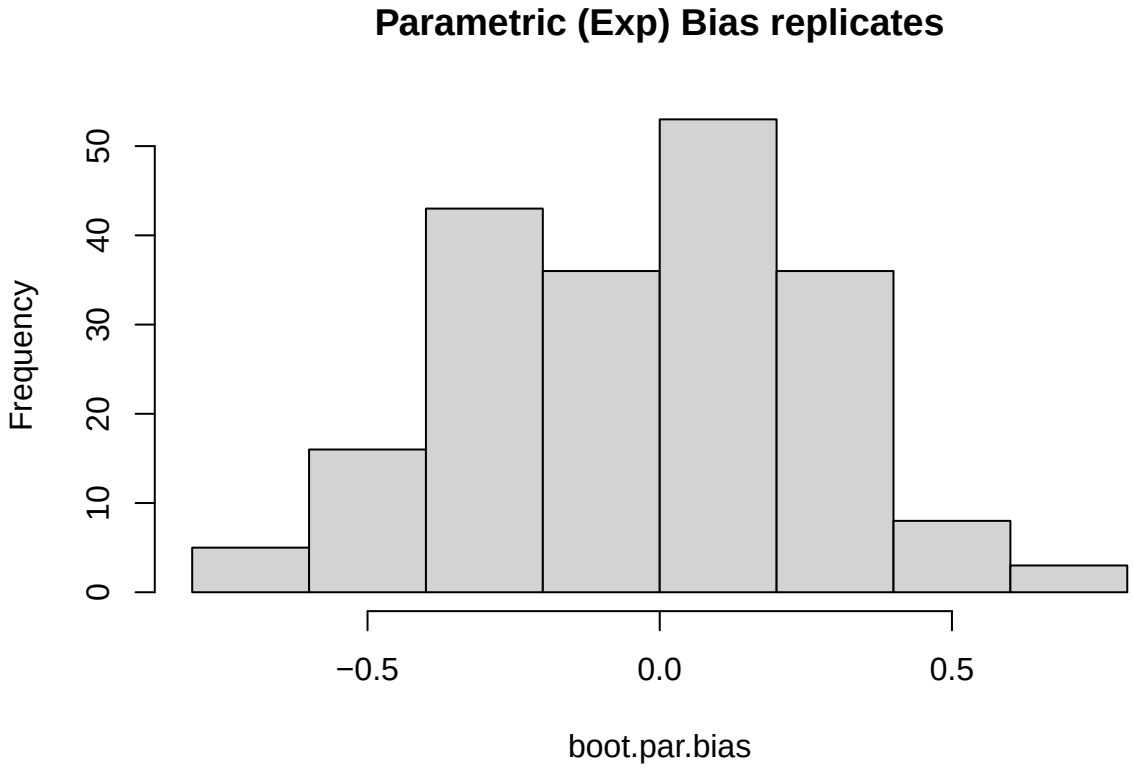
```
## [1] -0.02538925
```

(e) Plot both the histograms of the bootstrap replications from nonparametric and parametric bootstrap.

```r
hist(boot.bias,main="Non-parametric Bias replicates")
```




```r
hist(boot.par.bias,main="Parametric (Exp) Bias replicates")
```



As shown, the parametric methods reduce bias

(f) Produce 95% confidence intervals by the standard normal, basic, percentile, and Bca methods.

```
# Standard Normal Conf Int
set.seed(5400)

# sample estimate of log X bar
rhat <- log(mean(aircon))

boot.obj <- boot(data=aircon,
                 statistic = function(x,i) log(mean(x[i])),
                 R=B)

(boot.se <- sd(boot.obj$t))
```

```
## [1] 0.3778917
```

```
conf_int <- rhat + c(-1, 1) * qnorm(0.975) * boot.se

cat('Normal Conf Int',conf_int)
```

```
## Normal Conf Int 3.942248 5.423557
```

```
# Basic Conf Int
boot.quan <- quantile(boot.obj$t, c(0.975, 0.025), type=6)
conf_int <- 2 * rhat - boot.quan

cat('Basic Conf Int',conf_int)
```

```
## Basic Conf Int 4.124402 5.55086
```

```
# Percentile Conf Int
cat('Percentile Conf Int',boot.quan)
```

```
## Percentile Conf Int 5.241403 3.814945
```

```
# BCA Conf Int
bca <- boot.ci(boot.obj, type="bca")
```

```
## Warning in norm.inter(t, adj.alpha): extreme order statistics used as endpoints
```

```
cat('BCA Conf Int',bca$bca[4],bca$bca[5])
```

```
## BCA Conf Int 4.026941 5.339139
```

(g) Use jackknife to estimate the standard error and bias of the log of the sample mean.

```
set.seed(5400)

# sample estimate of correlation
rhat <- log(mean(aircon))

n <- length(aircon)

# jackknife estimate of bias
jack.r <- rep(NA, n)
for (i in seq(n)) {
 jack.r[i] <- log(mean(aircon[-i]))
 }
(jack.bias <- (n-1) * (mean(jack.r) - rhat))
```

```
## [1] -0.07889564
```

```
 # jackknife estimation of se
(jack.se <- sqrt((n - 1) / n * sum((jack.r - mean(jack.r))^2)))
```

```
## [1] 0.4154832
```

2. Failure of bootstrap

The bootstrap is not foolproof. To see this, consider analysis of a binomial model with n trials. You observe 0 successes. Discuss what would happen if you were to use the standard, non-parametric bootstrap in constructing a 95% C.I. for the binomial parameter p.

```
set.seed(5400)

(bin_obs <- rbinom(10,1, 0.01))
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0
```

Imagining a binomial model with two outcomes (0=failure and 1=success) with a very low probability of success (1%). This seed generates 0 successes for 10 observations. Any non-parametric bootstrap would fail due to no variance in the observed responses as follows:

```
# Bootstrap estimate of the mean
muhat <- (mean(bin_obs))
n <- length(bin_obs); B <- 200

boot.bias <- rep(NA, B)

for (b in seq(B)) {
  id <- sample(n,n, replace = TRUE)
  boot.rep[b] <- (mean(bin_obs[id]))
}

(bias <- mean(boot.rep))
```

```
## [1] 0
```

3. Bootstrap estimate of the standard error of trimmed mean.

Consider an artificial data set consisting of eight observations:

1, 3, 4.5, 6, 6, 6.9, 13, 19.2.

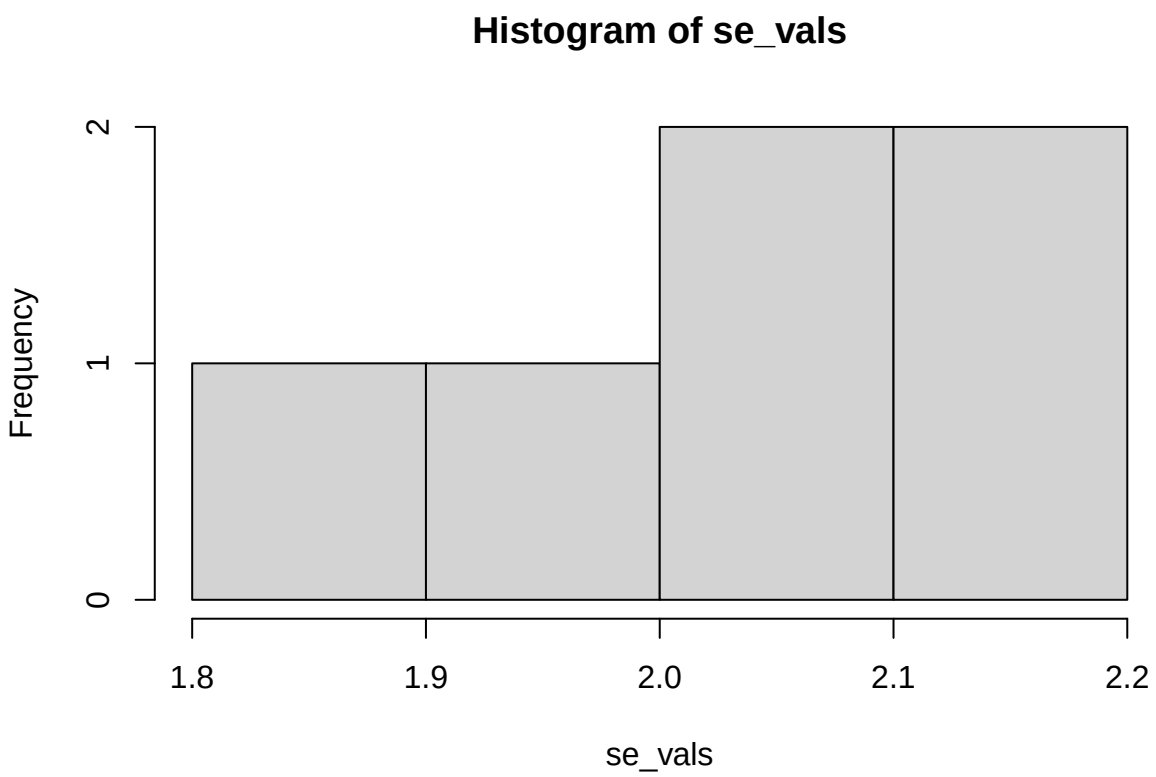
Let $\hat{\theta}$ be the 25% trimmed mean, which is computed by deleting two smallest numbers and two largest numbers, and then taking the average of the remaining four numbers.

(a) Calculate $\widehat{\text{se}}_B$ for $B = 25, 100, 200, 500, 1000, 2000$. From these results estimate the ideal bootstrap estimate $\widehat{\text{se}}_\infty$.

```
set.seed(5400)

obs <- c(1, 3, 4.5, 6, 6, 6.9, 13, 19.2)

thetahat <- mean(obs, trim = .25)

B <- c(25,100,200,500,1000,2000); n <- length(obs); se_vals <- vector()

for(b in B) {
  boot.rep <- replicate(b, (mean(obs[sample(n, replace=TRUE)], trim = 0.25)))
  boot.se <- sd(boot.rep)
  cat("St. Error Estimate for ", b, " replicates is: ", boot.se,"\n")
  se_vals <- c(se_vals, boot.se)
}
```

```
## St. Error Estimate for  25  replicates is:  1.852573
## St. Error Estimate for  100  replicates is:  2.01992
## St. Error Estimate for  200  replicates is:  2.188328
## St. Error Estimate for  500  replicates is:  2.132988
## St. Error Estimate for  1000  replicates is:  1.998223
## St. Error Estimate for  2000  replicates is:  2.001772
```

```r
hist(se_vals)
```



A standard error value of 2 seems appropriate

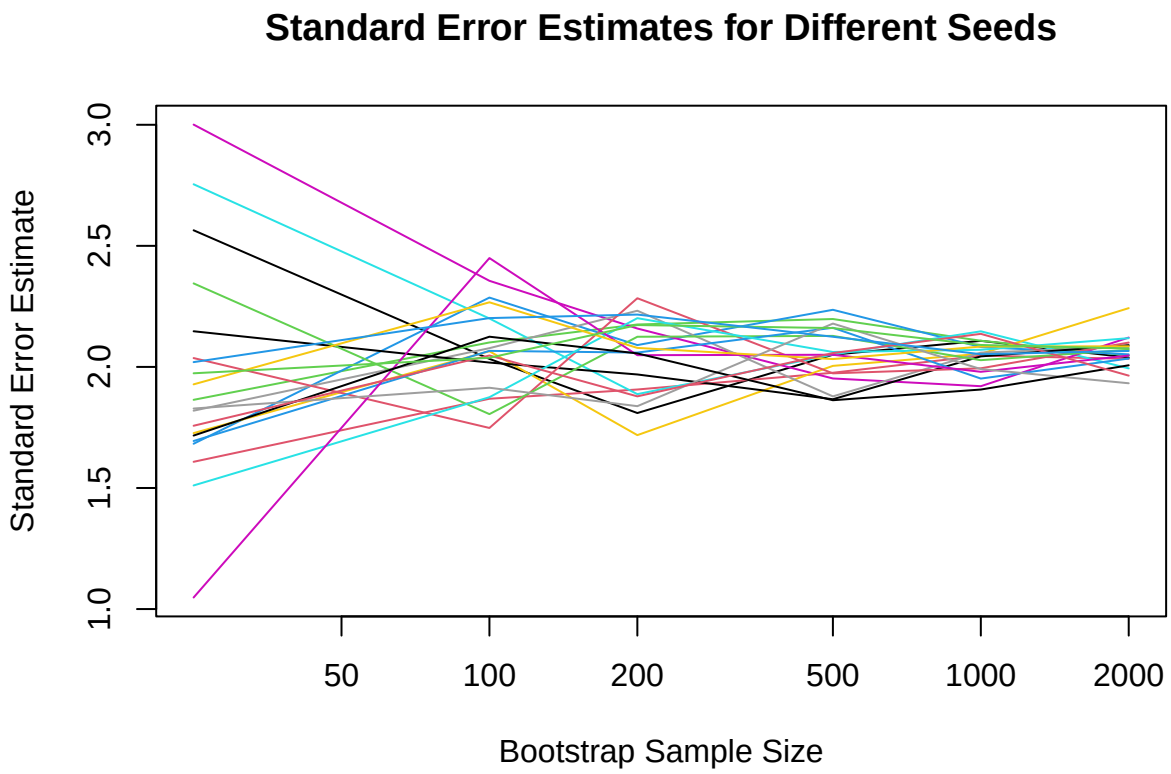
(a) Repeat part (a) using twenty different random number seeds. Comment on the trend of the variability of each $\widehat{\text{se}}_B$.

```r
set.seed(5400)
# Get 20 random seeds
seeds <- round(runif(20, 1000, 9999))

B <- c(25, 100, 200, 500, 1000, 2000)


cat('Set of seeds: ', seeds)
```

```
## Set of seeds:  9703 8659 9157 4682 8603 3265 3222 1330 9476 5901 5933 1810 5237 9674 4038 6536 2
```

```r
se_values <- matrix(nrow = length(B), ncol = length(seeds))

# plot(1:10, xlim=c(0, 10), ylim=c(0,10))

for (i in seq_along(seeds)) {
  set.seed(seeds[i])
  obs <- c(1, 3, 4.5, 6, 6, 6.9, 13, 19.2)
  thetahat <- mean(obs, trim = 0.25)
```

```r
  for (j in seq_along(B)) {
boot.rep <- replicate(B[j], mean(obs[sample(n, replace = TRUE)], trim = 0.25))
boot.se <- sd(boot.rep)
se_values[j, i] <- boot.se
  }

}
   plot(B, se_values[,1], type = "l", ylim = range(se_values), log = "x",
 xlab = "Bootstrap Sample Size", ylab = "Standard Error Estimate",
 main = "Standard Error Estimates for Different Seeds")
for (i in 2:length(seeds)) {
  lines(B, se_values[,i], col = i)
}
```



The seed does seem to vary the variance quite a bit for some variance values, sometimes even doubling the variance between replications, however larger sizes reduces the variations

4. Hypothesis testing using bootstrap Consider two independent random samples X and Y drawn from possibly different probability distributions:

$$X_1, \ldots, X_n \overset{iid}{\sim} F,$$
$$Y_1, \ldots, Y_m \overset{iid}{\sim} G,$$
$$X_i \text{ is indepdent of } Y_j, \ \forall i, j.$$

The goal is to perform a hypothesis test for

$$H_0 : F = G \text{ vs } H_1 : F \neq G$$

If H_0 is true, then there is no significant difference between random vectors $\mathbf{X}$ and $\mathbf{Y}$.

The bootstrap algorithm for performing such test is given as below:

- Compute a statistic on the original sample:

$$t(\mathbf{Z}) = |\bar{\mathbf{X}} - \bar{\mathbf{Y}}|.$$

- For each $b = 1, \dots B$:
  - Generate bootstrap samples, $\mathbf{Z}^{\star b}$, by drawing $(n+m)$ observations from **Z with replacement**.
  - Put $\mathbf{X}^{\star b} = (Z_1^{\star b}, \dots, Z_n^{\star b})$ and $\mathbf{Y}^{\star b} = (Z_{n+1}^{\star b}, \dots, Z_{n+m}^{\star b})$.
  - Compute bootstrap replications:
$$t(\mathbf{Z}^{\star b}) = |\bar{\mathbf{X}}^{\star b} - \bar{\mathbf{Y}}^{\star b}|.$$

- Estimate the achieved significance level (ASL):

$$\widehat{\text{ASL}}_B = \#\{t(\mathbf{Z}^{\star b}) \geq t(\mathbf{Z})\}/B.$$

- Reject H_0 if ASL $< \alpha$.

Below is an example to test $F = G$, where $F \sim \exp(\mu = 2)$ and $G \sim \exp(\mu = 1/2)$.

```
set.seed(5400)
x <- rexp(20, rate=1/2)
y <- rexp(10, rate=2)
B <- 2000
z <- c(x, y)
tstat <- abs(mean(x) - mean(y))
boot.r <- rep(NA, B)
for (i in seq(B)) {
  boot.samp <- z[sample(length(z), replace=TRUE)]
  m.boot.x <- mean(boot.samp[seq_along(x)])
  m.boot.y <- mean(boot.samp[-seq_along(x)])
  boot.r[i] <- abs(m.boot.x - m.boot.y)
}
mean(boot.r > tstat)
```

```
## [1] 0.013
```

Instead of using $|\bar{\mathbf{X}} - \bar{\mathbf{Y}}|$ as the test statistic, we may also use the t-statistic, if we assume equal variance:

$$t(\mathbf{Z}) = \frac{|\bar{\mathbf{X}} - \bar{\mathbf{Y}}|}{\hat{\sigma}_{\text{pool}}\sqrt{1/n + 1/m}},$$

where

$$\hat{\sigma}_{\text{pool}} = \sqrt{\frac{1}{n+m-2}\left[\sum_{i=1}^{n}(X_i - \bar{\mathbf{X}})^2 + \sum_{j=1}^{m}(Y_j - \bar{\mathbf{Y}})^2\right]}.$$

Please use bootstrap to test $F = G$ with the new statistics.

```r
set.seed(5400)
x <- rexp(20, rate=1/2)
y <- rexp(10, rate=2)
B <- 2000
z <- c(x, y)
n <- length(x)
m <- length(y)


sum_of_squared_deviations <- function(x) {
  # Calculate the mean of the vector
  mean_x <- mean(x)
  # Calculate the sum of squared deviations from the mean
  sum((x - mean_x)^2)
}

tstat <- abs(mean(x) - mean(y))/
  (sqrt((1/(n+m-2))*(sum_of_squared_deviations(x)+ sum_of_squared_deviations(y)))*(sqrt(1/n + 1/m)))

boot.r <- rep(NA, B)
for (i in seq(B)) {
boot.samp <- z[sample(length(z), replace=TRUE)]

m.boot.rep.x <- boot.samp[seq_along(x)]
m.boot.rep.y <- boot.samp[-seq_along(x)]

m.boot.x <- mean(boot.samp[seq_along(x)])
m.boot.y <- mean(boot.samp[-seq_along(x)])
boot.r[i] <- abs(m.boot.x - m.boot.y)

n <- length(boot.samp[seq_along(x)])
m <- length(boot.samp[-seq_along(x)])

boot.r[i] <- abs(m.boot.x - m.boot.y)/
  (sqrt((1/(n+m-2))*(sum_of_squared_deviations(m.boot.rep.x)+ sum_of_squared_deviations(m.boot.rep.y)))
}
mean(boot.r > tstat)
```

```
## [1] 0.0115
```

Since p = 0.0115, which is less than 0.05, we would reject the null hypothesis at the 5% significance level, suggesting a significant difference between the distributions F and G.