Homework 9 Elias Washor

Elias Washor

2024-10-31

Q1)

```
data <- c(3, 5, 7, 18, 43, 85, 91, 98, 100, 130, 230, 487)
sample_mean <- mean(data)
log_sample_mean <- log(sample_mean)

## (A)
set.seed(5400)
B <- 200
boot.rep <- rep(NA,B)

for (i in 1:B) {
  resample <- sample(data, replace = TRUE)
  boot.rep[i] <- log(mean(resample))
}

bootstrap_mean_log <- mean(boot.rep)
bias <- bootstrap_mean_log - log_sample_mean

## part b, c
std_ER <- (sd(boot.rep)) ## SE
bias_se_ratio <- abs(bias / std_ER) ## bias se ratio
root_MSE <- (std_ER * (1 + (bias_se_ratio^2)/2))
MSE_se_ratio <- (root_MSE / std_ER)
```

```
## Original log(sample mean): 4.682903
```

```
## Bootstrap mean of log(sample means): 4.60561
```

```
## Estimated bias of log(sample mean): -0.07729246
```

```
## Std Error Estimate:  0.373746
```

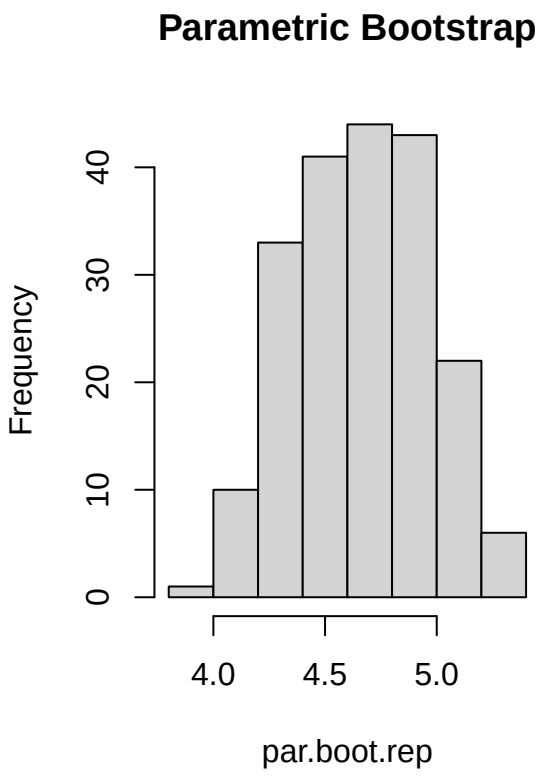
```
## Bias_SE_ratio       rootMSE  MSE_SE_Ratio
##     0.2068048     0.3817382     1.0213841
```

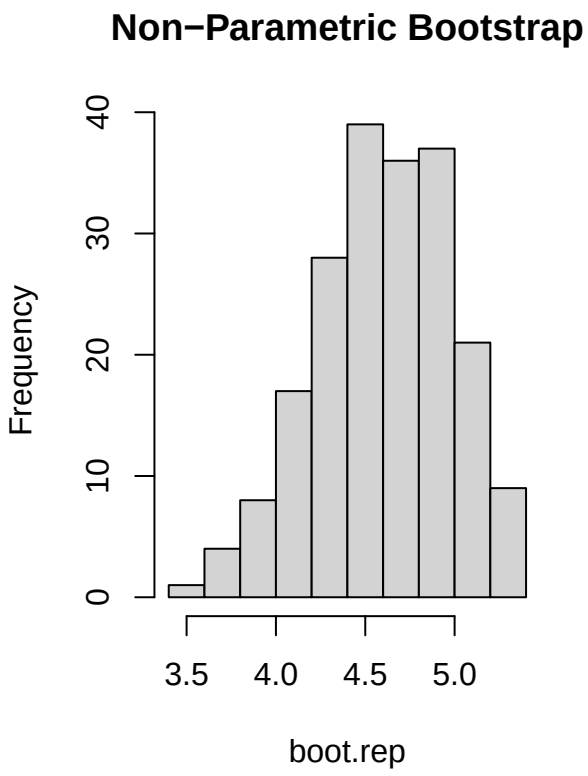
(b) The estimated bias for log(X Bar) is -0.077 (negative), therefore log(X Bar) overestimates log(Mu). The explanation lies in Jensen's inequality: $E[g(X)] \leq g(E[X])$ for a concave function.

$E[log(\bar{X})] \leq log(E[\bar{X}])$. The left side of the inequality is the overall mean of the bootstrap samples and the right side is the log of the sample mean, X bar. If we move the right side over to the left side then we are left with $E[log(\bar{X})] - log(E[\bar{X}]) \leq 0$. This makes intuitive sense since the bias was -0.077.

(c) The SE estimate is 0.374, and the ratio of Bias to SE is 0.207. The Bias is not very large given the Standard Error. The Bias/SE Ratio (0.207) is smaller than 0.25, the general rule of thumb for when Bias can be ignored. Also, the root MSE is only 2.14 % greater than the SE (MSE_SE ratio).

```
library(boot)
library(MASS)
set.seed(5400)

### (d)
B <- 200
par.boot.rep <- rep(NA,B)

### exp dist.
for (b in seq(B)) {
  par.boot.rep[b] <- log(mean(rexp(12, rate=1/sample_mean)))
  }

par.log.samp.mean <- mean(par.boot.rep)
par.bias <- par.log.samp.mean - log(sample_mean)
par.SE <- sd(par.boot.rep)

## (e)
par(mfrow=c(1,2))
hist(par.boot.rep, main= "Parametric Bootstrap")
hist(boot.rep, main = "Non-Parametric Bootstrap")
```






```r
## (f)
norm_CI <- round(par.log.samp.mean + c(-1,1) * qnorm(1 - .05/2) * par.SE, 3)
boot.quan <- quantile(par.boot.rep, c(0.975, 0.025), type = 6)
basic_CI <- 2 * par.log.samp.mean - boot.quan
pct_CI <- quantile(par.boot.rep, c(.025, 0.975), type = 6)

#BCA_CI
m <- function(x, i) log(mean(x[i]))
boot.obj <- boot(data = data, statistic = m, R = 200)
bca_CI <- boot.ci(boot.obj, type = "bca")$bca[4:5]

### (g) Jackknife
n <- length(data)
jack.r <- rep(NA, n)

for (i in 1:n) {
  jack.r[i] <- log(mean(data[-i]))
}

jackknife_mean <- mean(jack.r)
jackknife_bias <- (n - 1) * (jackknife_mean - log(sample_mean))
jackknife_se <- sqrt((n - 1) * mean((jack.r - jackknife_mean)^2))
```

The Non-Parametric bootstrap looks slightly more skewed left than the Parametric bootstrap.

```
##      Par.mean      Par.bias Parametric.SE
##    4.65751328   -0.02538925    0.29293363

##
## Normal CI:  4.083 5.232

## Basic CI:  4.062376 5.231279

## Percentile CI:  4.083748 5.252651

## BCa CI:  4.143 5.623

## Jackknife Bias:  -0.07889564

## Jackknife SE Estimate:  0.4154832
```

(f) The confidence intervals were close between the different methods. However, the BCA CI was the widest and had the highest interval for log(X bar).

(g) Jackknife Bias estimate was -0.079, while the Jackknife SE estimate was 0.415, the highest of the three SE estimates. Then follows the non-parametric bootstrap with a SE of 0.374, and finally the lowest was the parametric bootstrap SE estimate of 0.293. The Parametric Bootstrap Bias estimate was the smallest at -0.025.

2)

A binomial model with n trials and 0 successes observed would be problematic for the standard, non-parametric bootstrap to produce a 95% CI for the binom parameter, p. This is because since 0 successes were observed, if we were to continuously resample, we would have sample mean of 0 for all the bootstrap samples. With 0 as every sample mean, the CI would suggest that $p = 0$. We can't infer that $p = 0$ based on these specific bootstrap samples.

3)

```r
## (a)
set.seed(5400)
data2 <- c(1, 3, 4.5, 6, 6, 6.9, 13, 19.2)

trim.25.mean <- function(V) {
  n <- length(V)
  V <- sort(V)
  W <- V[3:n-2]
  return(mean(W))
}

samp_trim_mean <- trim.25.mean(data2)
B <- 200

get_se_boot <- function(dat, B) {
  boot.rep.2 <- rep(NA,B)
```

```
  for (i in 1:B) {
    boot.sample <- sample(dat, replace = TRUE)
    boot.rep.2[i] <- trim.25.mean(boot.sample)
  }
  trim.25.mean(boot.rep.2)
  sd(boot.rep.2)
}

S_ERRs <- rep(NA, 6)
B_vals <- c(25,100,200,500,1000,2000)

### (B)
for (b in seq_along(B_vals)) {
  S_ERRs[b]<- get_se_boot(data2, B_vals[b])
}

S_ERRs <- cbind(B_vals, S_ERRs)

SE_df <- as.data.frame(replicate(20,
          expr = {
          set.seed(sample.int(10000, size = 1))
          S_ERRs <- rep(NA, 6)
          B_vals <- c(25,100,200,500,1000,2000)

          for (b in seq_along(B_vals)) {
            S_ERRs[b]<- get_se_boot(data2, B_vals[b])
          }
          S_ERRs}))

### (PART A) S_ERRs of first bootstrap ###
print(S_ERRs)
```

```
##      B_vals   S_ERRs
## [1,]     25 1.508471
## [2,]    100 1.624551
## [3,]    200 1.732679
## [4,]    500 1.681025
## [5,]   1000 1.576696
## [6,]   2000 1.572421
```

```
### show first five cols
print(SE_df[,1:5])
```

```
##         V1       V2       V3       V4       V5
## 1 1.317914 1.432103 1.556275 1.576813 1.540123
## 2 1.356614 1.440661 1.752186 1.573646 1.626004
## 3 1.631749 1.814975 1.667518 1.672972 1.690284
## 4 1.569863 1.603738 1.474309 1.644389 1.651897
## 5 1.593191 1.686266 1.521152 1.618732 1.544242
## 6 1.640025 1.585415 1.556408 1.628793 1.612055
```

```
### average for each B value
SE_MEANS <- rowMeans(SE_df)
(rbind(B=(B_vals), SE_MEANS))
```

```
##               [,1]       [,2]     [,3]      [,4]        [,5]        [,6]
## B        25.000000 100.000000 200.0000 500.00000 1000.000000 2000.000000
## SE_MEANS  1.545849   1.602733   1.6625   1.58284    1.583338    1.614562
```

(a) Interestingly at B = 200, the SE estimate got as high as 1.73. The ideal SE estimate is roughly 1.60 based on the SE estimates at large values of B. Reference S_ERRs above.

(b) The variability of SE_B fluctuates a lot until a value of B = 500 or even 1000. At that number of bootstrap samples, the SE converges towards 1.60.

Q4)

```
set.seed(5400)

x <- rexp(20, rate = 1/2)
y <- rexp(10, rate = 2)

n <- length(x)
m <- length(y)
x_bar <- mean(x)
y_bar <- mean(y)
pooled_sd <- sqrt(((sum((x - x_bar)^2) + sum((y - y_bar)^2)) / (n + m - 2)))
tstat <- abs(x_bar - y_bar) / (pooled_sd * sqrt(1/n + 1/m))

B <- 2000
z <- c(x, y)
boot.r <- rep(NA, B)

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

  boot_x <- boot.samp[seq_along(x)]
  boot_y <- boot.samp[-seq_along(x)]

  m.boot.x <- mean(boot_x)
  m.boot.y <- mean(boot_y)

  pooled_sd_boot <- sqrt(((sum((boot_x - m.boot.x)^2) +
                             sum((boot_y - m.boot.y)^2)) / (n + m - 2)))

  boot.r[i] <- abs(m.boot.x - m.boot.y) / (pooled_sd_boot * sqrt(1/n + 1/m))
}

# Achieved Significance Level
ASL <- mean(boot.r >= tstat)
cat("(ASL):", ASL, "\n")
```

```
## (ASL): 0.0115
```

H0: $F = G$ vs. H1: $F \neq G$

Since, ASL is 0.0115 $<$ alpha (0.05) we reject H0 and have significant evidence that $F \neq G$.