Homework 05 - STAT440

Joseph Sepich (jps6444)

10/04/2020

```r
set.seed(42)
```

Problem 1

```r
# load dataset
scores <- data.matrix(read.csv('./data/score.csv'))
scores[0:10,]
```

```
##       HW1 HW2 HW3 HW4 HW5
##  [1,]  93  99  81  81  98
##  [2,]  76  94  97  85  98
##  [3,]  91  88  86  80  98
##  [4,]  66  87  76  85  82
##  [5,]  76  76  78  85  76
##  [6,]  64  74  87  77  88
##  [7,]  62  81  78  78  82
##  [8,]  71  85  82  75  68
##  [9,]  75  72  70  75  85
## [10,]  77  87  72  75  54
```

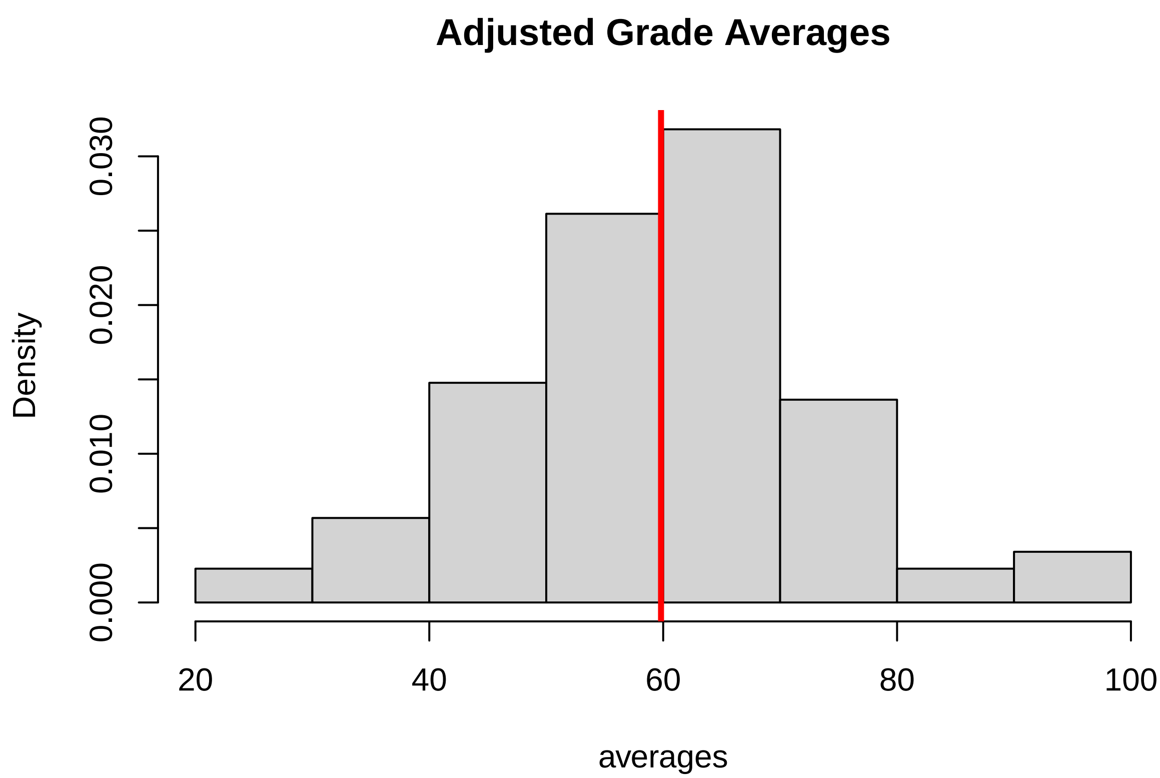
Adjust the scores by dropping the lowest.

```r
adjusted_scores <- t(as.matrix(apply(scores, 1, function(x) x[-(match(min(x), x))])))
adjusted_scores[0:10,]
```

```
##       [,1] [,2] [,3] [,4]
##  [1,]   93   99   81   98
##  [2,]   94   97   85   98
##  [3,]   91   88   86   98
##  [4,]   87   76   85   82
##  [5,]   76   78   85   76
##  [6,]   74   87   77   88
##  [7,]   81   78   78   82
##  [8,]   71   85   82   75
##  [9,]   75   72   75   85
## [10,]   77   87   72   75
```

Part a

```
averages <- rowMeans(adjusted_scores)
sample_avg <- mean(averages)
hist(averages, freq=FALSE, main = "Adjusted Grade Averages")
abline(v=sample_avg, col="red", lw=3)
```



Part b

You could express θ as an expectation of an indicator function: $E[1_{X>C}(x)] = P(X > C) = \theta$. This value can be approximated by sampling from X and you obtain a 0 or 1 depending on whether it is greater than C or not. You would divide the sum of samples (0 or 1) by N to get the expected value of the indicator, which is also the approximation of the probability $P(X > C)$. In this specific problem we are sampling from the student's average adjusted scores. If the average adjusted score we randomly select is greater than C the indicator is a 1, otherwise it is a zero.

Part c

```
C <- 70
n <- 10000
# sample from data (X)
samples <- sample(averages, n, replace=TRUE)
```

```r
# apply indicator function
samples[samples <= C] <- 0
samples[samples > C] <- 1
# expectation of indicator
theta_hat <- mean(samples)
print(theta_hat)
```

```
## [1] 0.1961
```

The Monte Carlo estimate states that $P(X > 70) \approx 0.1961$, so about 19.61% of students have an average adjust test score about 70%.

Part d

The histogram looks similar to a normal distribution. Often grades of students fall upon a normal distribution and instructors usually use a normal distribution when grading on a curve. For this reason we will use a normal distribution for our parametric bootstrap.

Part e

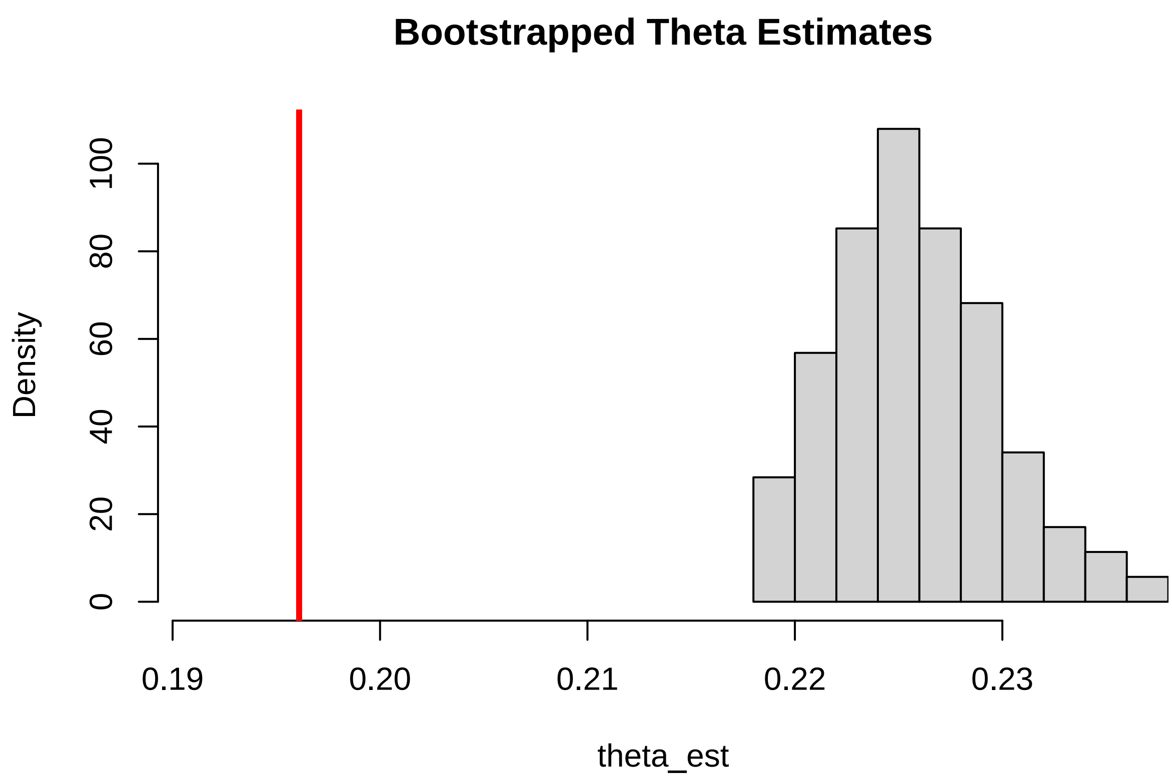
```r
k <- 10000
n <- length(averages)

# parameter estimates
mu_hat <- mean(averages)
sigma_hat <- sd(averages)

# parametric bootstrap from estimated distribution
boot_samples <- matrix(rnorm(k*n, mean=mu_hat, sd=sigma_hat), nrow=k)

# apply indicator function
boot_samples <- t(as.matrix(apply(boot_samples, c(1,2), function(x) if (x <= C) {0} else {1})))
theta_est <- rowMeans(boot_samples)

hist(theta_est, freq = FALSE, main="Bootstrapped Theta Estimates", xlim = c(theta_hat - 0.005, max(theta
abline(v=theta_hat, col="red", lw=3)
```



This results makes sense due to our choice of a normal distribution. The histogram of the actual test scores shows fewer students past 70 then you would expect if you traced a normal distribution over it.

Part f

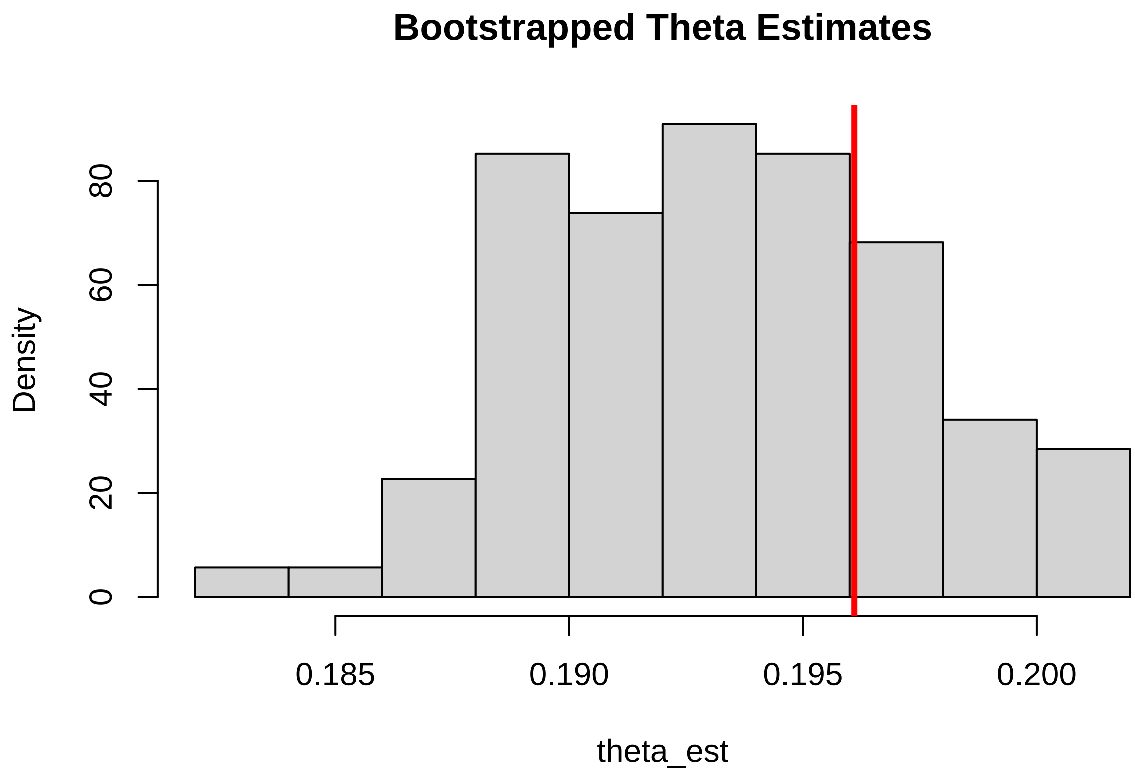
```
k <- 10000
n <- length(averages)

# parameter estimates
mu_hat <- mean(averages)
sigma_hat <- sd(averages)

# parametric bootstrap from estimated distribution
boot_samples <- matrix(sample(averages, k*n, replace=TRUE), nrow=k)

# apply indicator function
boot_samples <- t(as.matrix(apply(boot_samples, c(1,2), function(x) if (x <= C) {0} else {1})))
theta_est <- rowMeans(boot_samples)

hist(theta_est, freq = FALSE, main="Bootstrapped Theta Estimates")
abline(v=theta_hat, col="red", lw=3)
```


Bootstrapped Theta Estimates
Density
0
20
40
60
80
0.185
0.190
0.195
0.200
theta_est