Homework 3

Akhil Havaldar

10/11/2022

Question 1

```r
library(boot)
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.1.3
```

```r
library(ipred)
```

```
## Warning: package 'ipred' was built under R version 4.1.3
```

```r
x <- c(70, 75, 80, 80, 85, 90)    # creating data vectors
y <- c(75, 82, 80, 86, 90, 91)
mse <- vector("numeric", 6)       # initializing empty MSE vector
for (i in 1:6){         # loop through all values in vector
  xval <- x[i]          # storing the validation x and y
  yval <- y[i]
  xtrain <- x[-match(xval, x)]  # dropping the validation x and y from training data
  ytrain <- y[! y %in% yval]
  lm <- lm(ytrain~xtrain)       # fitting model
  pred <- lm$coefficients[1] + lm$coefficients[2]*xval   # predictions
  error <- (yval - pred)^2        # calculating MSE for each prediction
  mse[i] <- error                 # storing values
}

mse_test <- mean(mse)     # taking mean of MSE's to find test MSE
print(mse_test)           # 9.054487
```

```
## [1] 9.054487
```

```r
# df <- data.frame(x =c(70, 75, 80, 80, 85, 90),
# y = c(75, 82, 80, 86, 90, 91))
# result.lin<-glm(y ~ x, data = df, family="gaussian")
# loocv.lin<-boot::cv.glm(df,result.lin)
# loocv.lin$delta
```

Question 2)

```r
gpa <- c(3.68, 3.42, 3.23, 3.87, 3.91, 3.67, 2.89, 3.14, 3.70, 3.56, 3.49, 3.02, 3.16, 3.76, 3.41)
load("hw3_bootstrap_samples.RData")
boot.samples
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
##  [1,] 3.41 3.68 3.49 2.89 3.14 3.41 3.49 3.70 3.14  3.41  3.70  3.49  3.14
##  [2,] 3.67 3.70 3.56 3.23 3.42 3.87 3.87 2.89 3.70  3.70  3.67  3.68  3.16
##  [3,] 3.87 3.02 3.41 3.67 3.14 3.41 3.91 3.70 3.76  3.76  3.14  3.91  3.76
##  [4,] 3.42 3.56 3.49 3.49 3.67 3.41 3.87 3.02 3.91  3.02  3.56  3.02  3.68
##  [5,] 3.49 3.23 3.41 3.67 2.89 3.42 3.42 3.23 3.23  3.87  3.68  3.41  3.23
##  [6,] 2.89 3.16 3.76 3.76 3.42 3.67 3.56 3.68 3.16  3.68  3.87  3.76  3.16
##  [7,] 3.87 3.67 3.68 3.67 3.41 3.76 3.68 3.14 3.23  3.87  2.89  3.42  3.23
##  [8,] 3.41 3.02 2.89 3.23 3.87 3.23 3.67 3.56 3.14  3.76  3.68  3.42  3.56
##  [9,] 3.49 3.68 2.89 3.76 3.14 3.70 3.87 3.87 3.87  3.02  3.02  3.14  3.14
## [10,] 3.16 3.67 3.16 3.41 3.67 3.67 3.41 3.42 3.76  3.87  3.68  3.91  3.14
## [11,] 3.14 3.23 3.14 3.87 3.68 2.89 3.16 3.42 3.76  3.14  3.70  3.23  3.56
## [12,] 3.23 3.41 3.49 3.42 3.87 3.14 3.91 3.41 3.41  3.76  3.14  3.87  3.49
## [13,] 3.41 3.49 3.67 3.56 3.91 3.70 3.91 3.87 3.16  3.67  3.49  3.56  3.16
## [14,] 3.49 3.67 3.14 3.91 3.14 3.76 2.89 3.70 3.68  3.41  3.49  3.42  3.41
## [15,] 3.49 3.02 3.67 3.14 3.42 3.42 3.02 3.56 3.14  2.89  3.70  3.87  3.41
##       [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
##  [1,]  3.76  3.68  3.16  3.76  3.67  3.76  3.02  3.02  2.89  3.87  3.14  3.02
##  [2,]  3.42  3.68  3.91  3.14  3.87  3.42  3.14  3.02  3.42  3.68  3.42  3.16
##  [3,]  3.56  3.23  3.91  3.16  3.67  3.68  3.23  3.02  3.49  3.23  3.91  3.23
##  [4,]  3.41  3.67  3.49  3.42  3.91  3.68  3.16  3.67  3.02  3.87  3.87  3.87
##  [5,]  3.56  3.76  3.87  3.41  2.89  2.89  3.56  3.76  3.02  2.89  3.02  3.67
##  [6,]  3.56  3.87  3.67  3.68  3.68  3.41  3.14  3.76  3.49  3.67  3.49  3.14
##  [7,]  3.87  3.41  3.49  3.49  3.49  3.49  3.41  3.14  3.70  3.49  3.42  3.91
##  [8,]  3.14  2.89  3.42  2.89  3.56  3.91  3.87  3.16  3.76  3.41  3.70  3.16
##  [9,]  3.41  3.76  3.42  2.89  3.02  3.67  3.49  3.56  2.89  3.87  3.87  3.70
## [10,]  3.42  3.91  3.02  3.41  3.87  3.02  3.41  3.16  3.16  3.23  3.76  2.89
## [11,]  3.67  3.41  3.91  3.42  3.23  3.49  3.41  3.02  3.49  3.56  2.89  3.67
## [12,]  3.67  3.41  3.49  3.02  3.67  3.42  3.02  3.76  3.70  3.76  3.02  3.02
## [13,]  2.89  3.41  3.70  3.23  3.56  3.76  3.14  3.67  3.41  3.16  3.68  3.56
## [14,]  3.91  3.76  3.02  3.49  3.49  3.87  3.91  3.87  3.70  3.87  3.91  3.23
## [15,]  3.70  3.67  3.16  3.41  3.76  3.14  3.02  3.56  3.14  2.89  3.56  3.16
##       [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37]
##  [1,]  3.91  3.70  3.16  3.91  3.56  3.87  3.16  3.41  3.02  3.49  3.16  3.56
##  [2,]  3.56  3.49  3.42  3.91  3.56  3.16  3.23  3.41  3.23  3.14  2.89  3.56
##  [3,]  3.49  3.56  2.89  3.67  3.14  3.16  3.02  3.87  3.67  3.49  3.42  3.56
##  [4,]  3.14  2.89  3.14  3.56  3.70  3.91  3.49  3.16  3.14  3.87  3.14  3.87
##  [5,]  3.49  3.23  3.91  3.23  3.76  3.87  3.68  3.14  3.68  3.76  3.16  2.89
##  [6,]  3.41  3.87  3.67  3.76  3.41  3.91  3.56  3.70  3.16  3.87  3.67  3.68
##  [7,]  3.16  3.16  2.89  3.70  3.02  3.76  3.02  3.02  3.68  3.23  3.67  3.76
##  [8,]  3.49  3.91  3.91  3.42  3.76  3.16  3.68  3.23  3.91  3.02  3.87  3.56
##  [9,]  3.16  3.56  3.23  3.68  3.91  3.91  3.41  3.91  2.89  3.02  3.87  3.16
## [10,]  3.49  3.87  3.68  2.89  3.70  3.67  3.70  3.23  3.67  3.87  2.89  3.23
## [11,]  3.23  3.87  3.68  3.41  3.14  3.67  3.68  3.87  3.14  3.76  3.91  3.16
## [12,]  3.70  3.41  3.42  3.87  3.91  3.42  3.14  3.02  3.02  3.14  3.16  3.41
## [13,]  3.14  3.56  3.56  3.02  3.23  3.70  3.41  3.23  3.16  3.42  2.89  3.41
## [14,]  3.42  3.02  3.70  3.91  3.02  3.14  2.89  3.56  3.02  3.56  3.70  3.42
```

```
## [15,]  3.42  3.87  3.70  3.87  3.56  3.16  3.76  3.42  3.76  3.76  2.89  3.23
##       [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48] [,49]
##  [1,]  3.14  3.76  3.02  3.67  3.14  3.67  3.23  3.56  3.91  2.89  3.76  3.91
##  [2,]  3.02  3.56  3.68  3.91  3.67  3.76  3.41  3.91  3.68  3.23  3.41  2.89
##  [3,]  3.23  3.49  2.89  3.87  3.76  3.42  3.49  3.68  3.16  3.56  3.02  3.42
##  [4,]  3.42  3.87  2.89  3.14  3.67  3.56  3.67  3.14  2.89  3.23  3.42  3.49
##  [5,]  3.02  2.89  3.41  3.91  3.16  3.67  2.89  3.42  2.89  3.76  3.56  3.42
##  [6,]  2.89  3.68  3.70  3.70  3.41  3.76  3.14  3.23  2.89  3.23  3.02  3.87
##  [7,]  3.68  3.56  3.76  3.91  3.02  3.14  3.49  3.23  3.91  3.91  3.67  3.16
##  [8,]  3.76  3.49  3.49  3.23  3.67  3.91  3.56  3.23  3.16  3.14  3.41  3.41
##  [9,]  3.41  3.87  3.42  3.02  3.67  3.68  3.49  3.02  3.70  3.49  3.14  3.14
## [10,]  2.89  3.14  3.16  3.56  3.68  3.49  3.23  3.87  3.02  3.23  3.91  3.76
## [11,]  2.89  3.91  3.42  3.49  3.56  3.41  3.87  3.02  3.02  3.23  3.23  3.41
## [12,]  3.67  3.14  2.89  3.70  3.14  3.56  3.49  3.41  2.89  3.70  3.87  3.76
## [13,]  3.70  3.02  3.67  3.70  3.23  3.56  3.49  3.67  3.70  3.49  3.49  3.14
## [14,]  3.68  3.42  3.16  3.56  3.91  3.16  3.49  3.76  3.67  3.41  3.76  3.02
## [15,]  3.87  3.56  3.16  3.49  3.23  3.56  3.56  3.91  3.76  3.56  3.02  3.70
##       [,50]
##  [1,]  3.16
##  [2,]  3.14
##  [3,]  3.68
##  [4,]  3.14
##  [5,]  3.14
##  [6,]  3.56
##  [7,]  3.70
##  [8,]  3.23
##  [9,]  3.14
## [10,]  3.49
## [11,]  3.16
## [12,]  3.23
## [13,]  3.70
## [14,]  3.49
## [15,]  3.49
```

```
# a)
quant <- quantile(gpa)
iqr.gpa <- quant[4] - quant[2]    # calculating IQR by q3-q1
#0.495

#b)
iqr.boot <- array(0,50)    # initializing array
for (i in 1:50){
  q <- quantile(boot.samples[,i])      # finding the quantiles for each boot sample
  iq <- q[4] - q[2]                     # calculating iqr
  iqr.boot[i] <- iq
}

print(sd(iqr.boot)) #0.1367592
```

```
## [1] 0.1367592
```

```
#c)
delta <- iqr.boot - iqr.gpa      # calculating delta
```

```r
conf <- quantile(delta, c(0.04,0.96))       # calculating 92% CI
ci <- c(iqr.gpa-conf[2], iqr.gpa-conf[1])
print(ci)         # (0.3184, 0.7706)
```

```
##    75%    75%
## 0.3184 0.7706
```

d)

- To improve estimates, we would have to increase sample size, and increase the number of bootstrap samples.

Question 6

```r
library(ISLR2)
```

```
## Warning: package 'ISLR2' was built under R version 4.1.3

##
## Attaching package: 'ISLR2'

## The following object is masked from 'package:MASS':
##
##     Boston
```

```r
dat <- Wage

#a)
samp.mean <- mean(dat$wage)
samp.sd <- sd(dat$wage)
print(samp.mean)            # 111.7036
```

```
## [1] 111.7036
```

```r
print(samp.sd)              # 41.7286
```

```
## [1] 41.7286
```

```r
#b)
samp.se <- samp.sd/(sqrt(nrow(dat)))
print(samp.se)               # 0.7618564
```

```
## [1] 0.7618564
```

```r
#c)
t.mult <- qt(0.975, df=2999)
lower <- samp.mean - (t.mult * samp.se)
upper <- samp.mean + (t.mult * samp.se)
ci <- c(lower, upper)
print(ci)                 # (110.2098 113.1974)
```

```
## [1] 110.2098 113.1974
```

```r
#d.i)
set.seed(4630)
boot.10 <- replicate(10, sample(dat$wage, replace=TRUE))

#d.ii)
mean.boot.10 <- array(0,10)    # initializing array
for (i in 1:10){
  m <- mean(boot.10[,i])      # finding the mean for each boot
  mean.boot.10[i] <- m
}

#d.iii)
se.boot.10 <- sd(mean.boot.10)
print(se.boot.10)           # 0.89869
```

```
## [1] 0.7720473
```

d.iv)

- The two values are relatively similar. In this instance, I would increase the number of bootstrap samples to improve the SE.

```r
#e.100)
set.seed(4630)
boot.100 <- replicate(100, sample(dat$wage, replace=TRUE))
mean.boot.100 <- array(0,100)    # initializing array
for (i in 1:100){
  m <- mean(boot.100[,i])      # finding the mean for each boot
  mean.boot.100[i] <- m
}

se.boot.100 <- sd(mean.boot.100)
print(se.boot.100)           # 0.8897307
```

```
## [1] 0.8897307
```

```r
#e.1000)
set.seed(4630)
boot.1000 <- replicate(1000, sample(dat$wage, replace=TRUE))

mean.boot.1000 <- array(0,1000)    # initializing array
for (i in 1:1000){
  m <- mean(boot.1000[,i])      # finding the mean for each boot
  mean.boot.1000[i] <- m
}

se.boot.1000 <- sd(mean.boot.1000)
print(se.boot.1000)           # 0.8021534
```

```
## [1] 0.8059771
```

```r
#e.10000)
set.seed(4630)
boot.10000 <- replicate(10000, sample(dat$wage, replace=TRUE))

mean.boot.10000 <- array(0,10000)    # initializing array
for (i in 1:10000){
  m <- mean(boot.10000[,i])      # finding the mean for each boot
  mean.boot.10000[i] <- m
}

se.boot.10000 <- sd(mean.boot.10000)
print(se.boot.10000)          # 0.7700513
```

```
## [1] 0.7650972
```

- When looking at the different bootstrap SE in relation to the sample SE, we can see that the bootstrap estimate gets closer and closer to the true value in (6b). With B=10000, the SE is only 0.01 off the actual SE.

f)

- Since the values of SE, as the bootstrap sizes get larger, converge to the estimation calculated in (6b) we know that we calculated the SE in (6b) correctly.

```r
#g)
set.seed(4630)
delta <- mean.boot.10000 - samp.mean      # calculating delta
conf <- quantile(delta, c(0.025,0.975))       # calculating 95% CI
ci <- c(samp.mean-conf[2], samp.mean-conf[1])
print(ci)         # (110.1883,113.1947)
```

```
##    97.5%     2.5%
## 110.1761 113.2084
```

- We can see that the bootstrap CI is very close to the CI we found from the sample.