Midterm II Solution

Yidong Zhou

5/21/2021

Q1

```
plastic <- read.table("http://users.stat.ufl.edu/~rrandles/sta4210/Rclassnotes/data/textdatasets/KutnerI
colnames(plastic) <- c('Y', 'X')
str(plastic)
```

```
## 'data.frame':    16 obs. of  2 variables:
##  $ Y: num  199 205 196 200 218 220 215 223 237 234 ...
##  $ X: num  16 16 16 16 24 24 24 24 32 32 ...
```

```
n <- nrow(plastic)
p <- 2
```

(a) 15 points

- $(\mathbf{X}'\mathbf{X})^{-1}$

Here we have only one predictor. The design matrix $\mathbf{X}$ should be n by 2, where n is the number of observations ($n = 16$ in this case).

```
X <- cbind(rep(1, n), plastic$X)# n by 2
Y <- plastic$Y# n by 1
solve(t(X)%*%X)
```

```
##             [,1]        [,2]
## [1,]  0.675000 -0.02187500
## [2,] -0.021875  0.00078125
```

- **b**

To make sure that `b` is a vector, you can either use `as.vector()` or `as.numeric()`. See `class(solve(t(X)%*%X)%*%t(X)%*%Y)`.

```
b <- as.vector(solve(t(X)%*%X)%*%t(X)%*%Y)
b
```

```
## [1] 168.600000   2.034375
```

- SSE

The same for SSE, `as.vector()` or `as.numeric()` can be used.

```
H <- X%*%solve(t(X)%*%X)%*%t(X) # n by n
I <- diag(n)
SSE <- as.vector(t(Y)%*%(I-H)%*%Y)
MSE <- SSE/(n-p)
SSE
```

```
## [1] 146.425
```

(b) 10 points

$s\{b_i\}$ is the square root of $s^2\{b_i\}$, i.e., the square root of the (i, i) entry of the variance covariance matrix of $\mathbf{b}$. $s\{b_i, b_j\}$ is the (i, j) entry of the variance covariance matrix of $\mathbf{b}$.

```
s2b <- MSE*solve(t(X)%*%X)
sqrt(diag(s2b)[1])
```

```
## [1] 2.657024
```

```
sqrt(diag(s2b)[2])
```

```
## [1] 0.09039379
```

```
s2b[1, 2]# or s2b[2, 1]
```

```
## [1] -0.2287891
```

(c) 5 points

Note that $\frac{1}{n}\mathbf{J}$ is equal to $\mathbf{1}(\mathbf{1}'\mathbf{1})^{-1}\mathbf{1}'$, where $\mathbf{1}$ is the n-vector of ones.

```
J <- rep(1, n)%*%t(rep(1, n))
H-J/n
```

```
##          [,1]    [,2]    [,3]    [,4]    [,5]    [,6]    [,7]    [,8]    [,9]
##  [1,]  0.1125  0.1125  0.1125  0.1125  0.0375  0.0375  0.0375  0.0375 -0.0375
##  [2,]  0.1125  0.1125  0.1125  0.1125  0.0375  0.0375  0.0375  0.0375 -0.0375
##  [3,]  0.1125  0.1125  0.1125  0.1125  0.0375  0.0375  0.0375  0.0375 -0.0375
##  [4,]  0.1125  0.1125  0.1125  0.1125  0.0375  0.0375  0.0375  0.0375 -0.0375
##  [5,]  0.0375  0.0375  0.0375  0.0375  0.0125  0.0125  0.0125  0.0125 -0.0125
##  [6,]  0.0375  0.0375  0.0375  0.0375  0.0125  0.0125  0.0125  0.0125 -0.0125
##  [7,]  0.0375  0.0375  0.0375  0.0375  0.0125  0.0125  0.0125  0.0125 -0.0125
##  [8,]  0.0375  0.0375  0.0375  0.0375  0.0125  0.0125  0.0125  0.0125 -0.0125
##  [9,] -0.0375 -0.0375 -0.0375 -0.0375 -0.0125 -0.0125 -0.0125 -0.0125  0.0125
## [10,] -0.0375 -0.0375 -0.0375 -0.0375 -0.0125 -0.0125 -0.0125 -0.0125  0.0125
## [11,] -0.0375 -0.0375 -0.0375 -0.0375 -0.0125 -0.0125 -0.0125 -0.0125  0.0125
## [12,] -0.0375 -0.0375 -0.0375 -0.0375 -0.0125 -0.0125 -0.0125 -0.0125  0.0125
## [13,] -0.1125 -0.1125 -0.1125 -0.1125 -0.0375 -0.0375 -0.0375 -0.0375  0.0375
## [14,] -0.1125 -0.1125 -0.1125 -0.1125 -0.0375 -0.0375 -0.0375 -0.0375  0.0375
## [15,] -0.1125 -0.1125 -0.1125 -0.1125 -0.0375 -0.0375 -0.0375 -0.0375  0.0375
## [16,] -0.1125 -0.1125 -0.1125 -0.1125 -0.0375 -0.0375 -0.0375 -0.0375  0.0375
##         [,10]   [,11]   [,12]   [,13]   [,14]   [,15]   [,16]
##  [1,] -0.0375 -0.0375 -0.0375 -0.1125 -0.1125 -0.1125 -0.1125
##  [2,] -0.0375 -0.0375 -0.0375 -0.1125 -0.1125 -0.1125 -0.1125
##  [3,] -0.0375 -0.0375 -0.0375 -0.1125 -0.1125 -0.1125 -0.1125
##  [4,] -0.0375 -0.0375 -0.0375 -0.1125 -0.1125 -0.1125 -0.1125
##  [5,] -0.0125 -0.0125 -0.0125 -0.0375 -0.0375 -0.0375 -0.0375
##  [6,] -0.0125 -0.0125 -0.0125 -0.0375 -0.0375 -0.0375 -0.0375
##  [7,] -0.0125 -0.0125 -0.0125 -0.0375 -0.0375 -0.0375 -0.0375
##  [8,] -0.0125 -0.0125 -0.0125 -0.0375 -0.0375 -0.0375 -0.0375
##  [9,]  0.0125  0.0125  0.0125  0.0375  0.0375  0.0375  0.0375
## [10,]  0.0125  0.0125  0.0125  0.0375  0.0375  0.0375  0.0375
## [11,]  0.0125  0.0125  0.0125  0.0375  0.0375  0.0375  0.0375
## [12,]  0.0125  0.0125  0.0125  0.0375  0.0375  0.0375  0.0375
## [13,]  0.0375  0.0375  0.0375  0.1125  0.1125  0.1125  0.1125
## [14,]  0.0375  0.0375  0.0375  0.1125  0.1125  0.1125  0.1125
## [15,]  0.0375  0.0375  0.0375  0.1125  0.1125  0.1125  0.1125
```

```
## [16,]  0.0375  0.0375  0.0375  0.1125  0.1125  0.1125  0.1125
```

(d) 10 points

See (6.50) in the textbook.

```
alpha <- 1-0.95
c(L = b[2] - qt(1-alpha/2, n-p)*sqrt(diag(s2b)[2]),
  U = b[2] + qt(1-alpha/2, n-p)*sqrt(diag(s2b)[2]))
```

```
##       L       U
## 1.84050 2.22825
```

Q2

```
df <- read.table('/Users/easton/Google Drive/Teaching/TA/STA-108B-SQ-2021/Midterm II/Demographic.txt')
df[, 5] <- df[, 5]/df[, 4]
df <- df[, c(10, 5, 15, 11, 16, 14, 17)]
colnames(df) <- c('Y', paste0('X', 1:5), 'Region')
dfRegion <- list()
for(i in 1:4) dfRegion[[i]] <- df[df$Region==i, -7]
n <- sapply(dfRegion, nrow)
p <- 6
```

(a)

```
fit <- list()
for(i in 1:4) fit[[i]] <- lm(Y~., data = dfRegion[[i]])
beta <- matrix(nrow = 4, ncol = p)
colnames(beta) <- names(fit[[1]]$coefficients)
rownames(beta) <- paste0('Region ', 1:4)
for(i in 1:4) beta[i, ] <- fit[[i]]$coefficients
beta
```

```
##          (Intercept)         X1        X2         X3       X4         X5
## Region 1   -50120.11 16.0232957 -2.821920 1307.25893 1.230042 -511.26983
## Region 2    11090.44  6.0907523 -3.631120  526.20046 3.759611  834.65976
## Region 3    44323.32  2.0087409 -2.242039 -165.57798 5.174021  166.48199
## Region 4    34971.14  0.6153439 -1.971027  -23.02375 3.707403  -24.64323
```

(b)

The answer should elaborate the difference between the four estimated regression functions, especially the sign of the coefficients.

(c)

Remember to state the decision rule and conclusion

```
MSE <- rep(1, 4)
MSR <- rep(1, 4)
for(i in 1:4){
  MSE[i] <- anova(fit[[i]])['Residuals', 'Mean Sq']
  MSR[i] <- sum(anova(fit[[i]])[paste0('X', 1:5), 'Sum Sq'])/(p-1)
}
```

```r
MSE
```

```
## [1] 752502442  96845117 164024379 170629475
```

```r
MSR
```

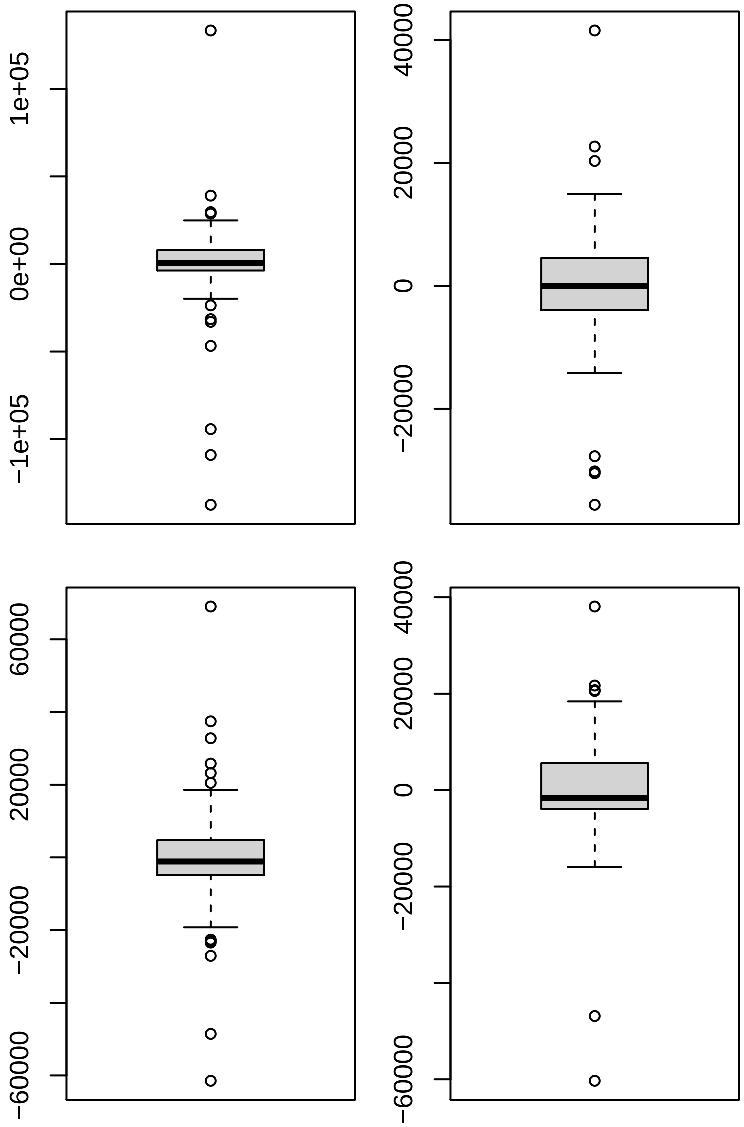
```
## [1]  80006644853  46005642480  39801976726 104591260179
```

```r
1-pf(MSR/MSE, df1 = p-1, df2 = n-p)
```

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

(d)

```r
res <- list()
for(i in 1:4) res[[i]] <- fit[[i]]$residuals
opar <- par(mar=c(1, 2, 1, 1))
par(mfrow = c(2, 2))
for(i in 1:4) boxplot(res[[i]])
```



```
par(mfrow = c(1, 1))
par(opar)
```

(e)

This question is much more open. Possible directions include the coefficients, F-test, or residual plots.

Code Appendix

```
plastic <- read.table("http://users.stat.ufl.edu/~rrandles/sta4210/Rclassnotes/data/textdatasets/KutnerI
colnames(plastic) <- c('Y', 'X')
str(plastic)
n <- nrow(plastic)
p <- 2
X <- cbind(rep(1, n), plastic$X)# n by 2
Y <- plastic$Y# n by 1
solve(t(X)%*%X)
b <- as.vector(solve(t(X)%*%X)%*%t(X)%*%Y)
b
H <- X%*%solve(t(X)%*%X)%*%t(X) # n by n
I <- diag(n)
SSE <- as.vector(t(Y)%*%(I-H)%*%Y)
MSE <- SSE/(n-p)
SSE
s2b <- MSE*solve(t(X)%*%X)
sqrt(diag(s2b)[1])
sqrt(diag(s2b)[2])
s2b[1, 2]# or s2b[2, 1]
J <- rep(1, n)%*%t(rep(1, n))
H-J/n
alpha <- 1-0.95
c(L = b[2] - qt(1-alpha/2, n-p)*sqrt(diag(s2b)[2]),
  U = b[2] + qt(1-alpha/2, n-p)*sqrt(diag(s2b)[2]))
df <- read.table('/Users/easton/Google Drive/Teaching/TA/STA-108B-SQ-2021/Midterm II/Demographic.txt')
df[, 5] <- df[, 5]/df[, 4]
df <- df[, c(10, 5, 15, 11, 16, 14, 17)]
colnames(df) <- c('Y', paste0('X', 1:5), 'Region')
dfRegion <- list()
for(i in 1:4) dfRegion[[i]] <- df[df$Region==i, -7]
n <- sapply(dfRegion, nrow)
p <- 6
fit <- list()
for(i in 1:4) fit[[i]] <- lm(Y~., data = dfRegion[[i]])
beta <- matrix(nrow = 4, ncol = p)
colnames(beta) <- names(fit[[1]]$coefficients)
rownames(beta) <- paste0('Region ', 1:4)
for(i in 1:4) beta[i, ] <- fit[[i]]$coefficients
beta
MSE <- rep(1, 4)
MSR <- rep(1, 4)
for(i in 1:4){
  MSE[i] <- anova(fit[[i]])['Residuals', 'Mean Sq']
  MSR[i] <- sum(anova(fit[[i]])[paste0('X', 1:5), 'Sum Sq'])/(p-1)
```

```r
}
MSE
MSR
1-pf(MSR/MSE, df1 = p-1, df2 = n-p)
res <- list()
for(i in 1:4) res[[i]] <- fit[[i]]$residuals
opar <- par(mar=c(1, 2, 1, 1))
par(mfrow = c(2, 2))
for(i in 1:4) boxplot(res[[i]])
par(mfrow = c(1, 1))
par(opar)
```