HW6

Zeqiu.Yu

2022-11-13

HW 6

9.11

```r
input1 <- read.table("./BookDataSets/Chapter  9 Data Sets/CH09PR10.txt")
names(input1) <- c("Y", "X1", "X2", "X3", "X4")
```

(a.)

```r
library(leaps)
Result.adjr2 <- leaps(x = input1[,2:5], y = input1[,1], names = names(input1)[2:5], method = "adjr2")
Result.adjr2 <- data.frame(Result.adjr2$which, Result.adjr2$adjr2)
colnames(Result.adjr2)[5] <- "adjr2"
Result.adjr2[order(Result.adjr2$adjr2, decreasing = TRUE), ]
```

```
##         X1    X2    X3    X4     adjr2
## X3    TRUE FALSE  TRUE  TRUE 0.9560482
## X4    TRUE  TRUE  TRUE  TRUE 0.9554702
## X2    TRUE FALSE  TRUE FALSE 0.9269043
## X3.1  TRUE  TRUE  TRUE FALSE 0.9246779
## X2.1 FALSE FALSE  TRUE  TRUE 0.8660988
## X3.2 FALSE  TRUE  TRUE  TRUE 0.8616797
## X3.3  TRUE  TRUE FALSE  TRUE 0.8232664
## X2.2  TRUE FALSE FALSE  TRUE 0.7984716
## X1   FALSE FALSE  TRUE FALSE 0.7962344
## X2.3 FALSE  TRUE  TRUE FALSE 0.7884436
## X2.4 FALSE  TRUE FALSE  TRUE 0.7635916
## X1.1 FALSE FALSE FALSE  TRUE 0.7452170
## X2.5  TRUE  TRUE FALSE FALSE 0.4154853
## X1.2  TRUE FALSE FALSE FALSE 0.2326452
## X1.3 FALSE  TRUE FALSE FALSE 0.2142762
```

Hence, according to the $R^2_{a,p}$ criterion, the four best subset regression models are:

Subset	$R^2_{a,p}$
X1, X3, X4	0.9560482
X1, X2, X3, X4	0.9554702
X1, X3	0.9269043
X1, X2, X3	0.9246779

(b.)
I'd like to use Mallow's C_p Criterion.

```r
Result.Cp <- leaps(x = input1[,2:5], y = input1[,1], names = names(input1)[2:5], method = c("Cp"))
Result.Cp <- data.frame(Result.Cp$which, Result.Cp$Cp)
colnames(Result.Cp)[5] <- "Cp"
Result.Cp[order(Result.Cp$Cp, decreasing = FALSE), ]
```

```
##         X1    X2    X3    X4         Cp
## X3    TRUE FALSE  TRUE  TRUE   3.727399
## X4    TRUE  TRUE  TRUE  TRUE   5.000000
## X2    TRUE FALSE  TRUE FALSE  17.112978
## X3.1  TRUE  TRUE  TRUE FALSE  18.521465
## X2.1 FALSE FALSE  TRUE  TRUE  47.153985
## X3.2 FALSE  TRUE  TRUE  TRUE  48.231020
## X3.3  TRUE  TRUE FALSE  TRUE  66.346500
## X2.2  TRUE FALSE FALSE  TRUE  80.565307
## X1   FALSE FALSE  TRUE FALSE  84.246496
## X2.3 FALSE  TRUE  TRUE FALSE  85.519650
## X2.4 FALSE  TRUE FALSE  TRUE  97.797790
## X1.1 FALSE FALSE FALSE  TRUE 110.597414
## X2.5  TRUE  TRUE FALSE FALSE 269.780029
## X1.2  TRUE FALSE FALSE FALSE 375.344689
## X1.3 FALSE  TRUE FALSE FALSE 384.832454
```

p = 5, the models are considered to be "good" if $C_p \approx p$ or $C_p \leq p$.
Hence, according to the C_p criterion, the best subset regression models are:

Subset	C_p
X1, X3, X4	3.727399
X1, X2, X3, X4	5.000000

(c.)

```r
fit.Full <- lm(Y~., data = input1)
selected.model <- step(fit.Full, scope = list(upper = fit.Full, lower = ~1), direction = "both", trace =
selected.model
```

```
##
## Call:
## lm(formula = Y ~ X1 + X3 + X4, data = input1)
##
## Coefficients:
## (Intercept)           X1           X3           X4
##   -124.2000       0.2963       1.3570       0.5174
```

```r
Y.predict <- predict(selected.model, input1)
output <- data.frame("Y"=input1$Y, "Y_hat"=Y.predict)
output
```

```
##      Y     Y_hat
## 1   88  81.99641
## 2   80  80.25410
## 3   96 101.45788
## 4   76  81.08190
## 5   80  79.87756
## 6   73  71.28870
## 7   58  58.20567
## 8  116 117.09940
```

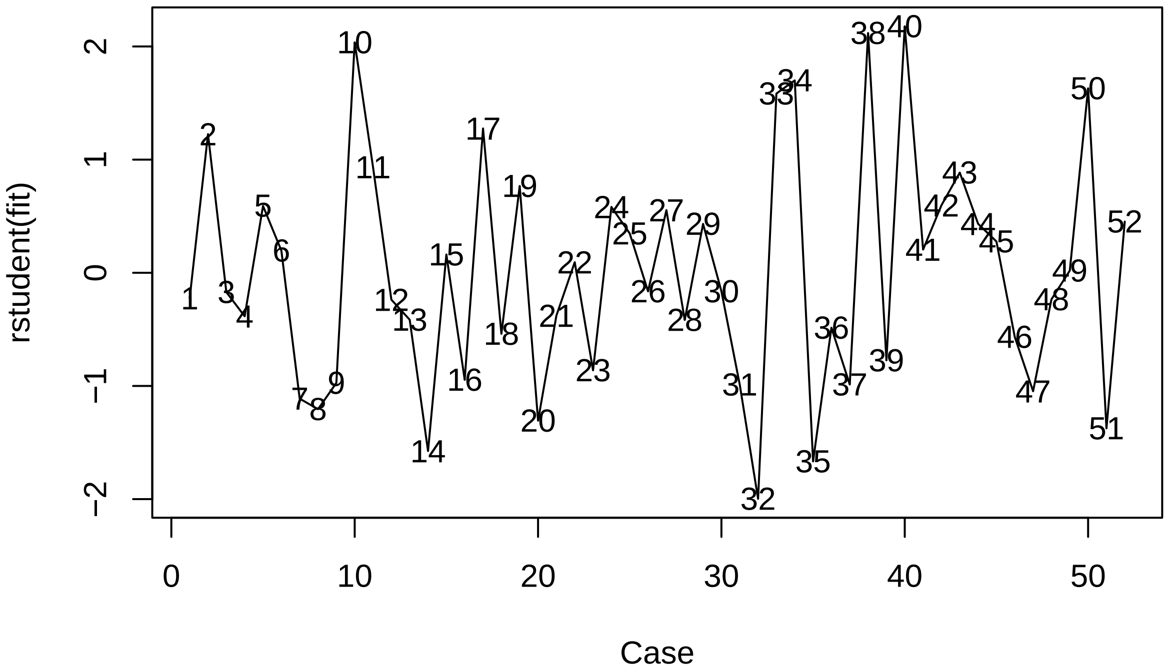
```
## 9  104 107.15627
## 10  99 104.03055
## 11  64  69.24048
## 12 126 124.43016
## 13  94  93.60499
## 14  71  74.03304
## 15 111 111.51891
## 16 109 102.39169
## 17 100  95.09160
## 18 127 122.36073
## 19  99 100.54743
## 20  82  86.83892
## 21  67  65.19295
## 22 109 112.23596
## 23  78  74.20956
## 24 115 113.51769
## 25  83  77.33746
```

10.10

```r
input2 <- read.table("./BookDataSets/Chapter  6 Data Sets/CH06PR09.txt", header = FALSE)
colnames(input2) <- c("Y", "X1", "X2","X3")
n <- dim(input2)[1]
p <- 4
fit <- lm(Y~X1+X2+X3, data = input2)
```

(a.)

```r
Case <- c(1:n)
plot(Case, rstudent(fit), type = "l")
text(Case, rstudent(fit), Case)
```



```r
rstudent(fit)
```

```
##           1           2           3           4           5           6
## -0.22408724  1.22549009 -0.17058921 -0.38465346  0.59079243  0.19612403
##           7           8           9          10          11          12
## -1.11220903 -1.20529304 -0.97317140  2.03651764  0.93459516 -0.23775605
##          13          14          15          16          17          18
## -0.41516269 -1.57563574  0.16177701 -0.94585538  1.27571169 -0.53946536
##          19          20          21          22          23          24
##  0.76695374 -1.30688333 -0.37866873  0.09348669 -0.86199416  0.58204380
##          25          26          27          28          29          30
##  0.34737819 -0.16427705  0.55395260 -0.41694583  0.43222641 -0.16381817
##          31          32          33          34          35          36
## -0.98793522 -1.99766654  1.58403006  1.70041654 -1.66686277 -0.48548061
##          37          38          39          40          41          42
## -0.98726030  2.11878596 -0.77401415  2.17827186  0.20535372  0.59660183
##          43          44          45          46          47          48
##  0.88556630  0.43246959  0.27680521 -0.56690329 -1.04682238 -0.23443689
##          49          50          51          52
##  0.01909697  1.63020460 -1.37470514  0.45278959
```

```r
alpha <- 0.05
crit <- qt(1-alpha/2/n, n-p-1)
paste("The Boferroni critical value is: ", crit)
```

```
## [1] "The Boferroni critical value is:  3.52308019248865"
```

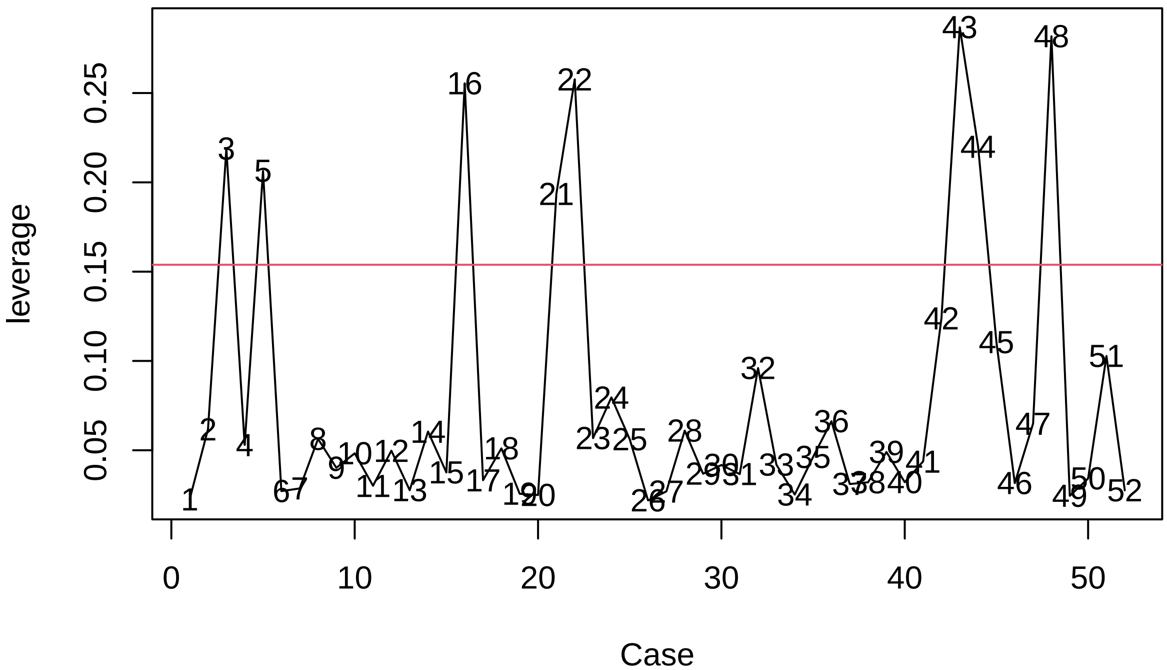
Decision Rule: for the test statistics t^*, if $|t_i|<t^*$, conclude H_0 : no outliers. Otherwise, conclude H_a.
Conclusion: There are no outliers with $\alpha = 0.05$

(b.)

```r
leverage <- hatvalues(fit)
leverage[which(leverage>2*p/n)]
```

```
##         3         5        16        21        22        43        44        48
## 0.2188773 0.2063282 0.2554249 0.1936047 0.2577199 0.2868586 0.2200236 0.2817766
```

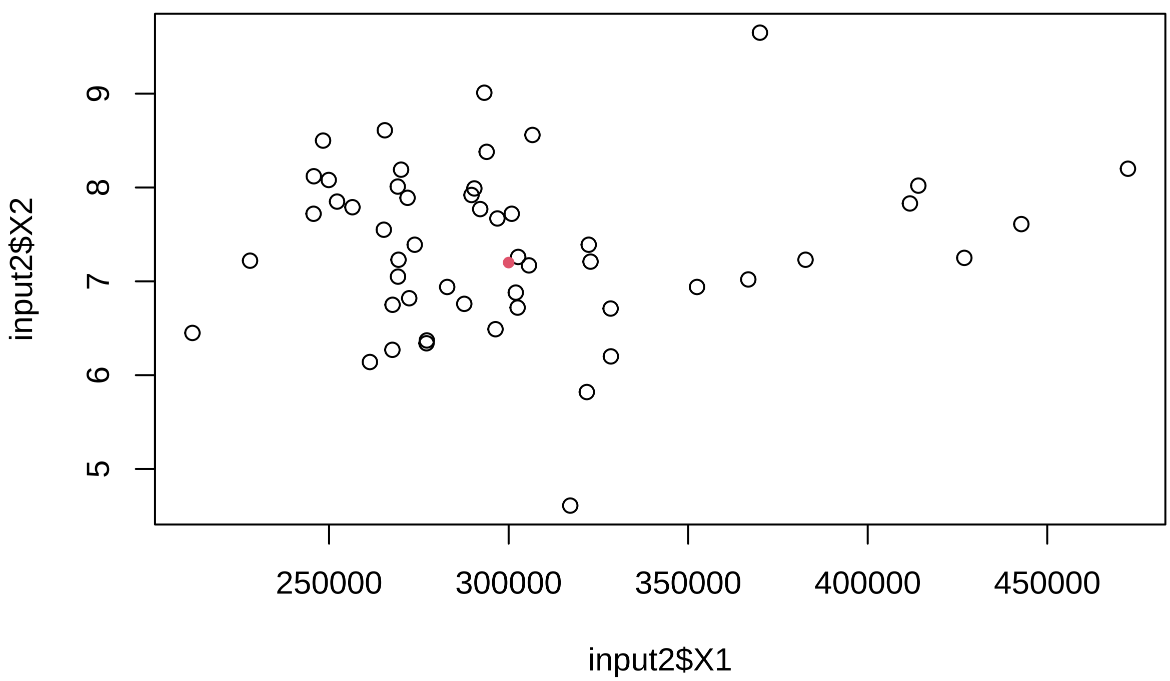
```r
plot(Case, leverage, type = "l")
text(Case, leverage, Case)
abline(h=2*p/n, col =2)
```



Rule of thumb determining if h_{ii} is "larger":
according to the red line in the figure above, $h_{ii} = 2p/n$, Case 3, 5, 16, 21, 22, 43, 44, 48 are outling X observations.

(c.)

```
plot(input2$X1, input2$X2)
points(300000, 7.2, col = 2, pch = 20)
```




```
X <- cbind(rep(1,n), input2$X1, input2$X2, input2$X3)
head(X)
```

```
##       [,1]    [,2] [,3] [,4]
```

```
## [1,]    1 305657 7.17    0
## [2,]    1 328476 6.20    0
## [3,]    1 317164 4.61    0
## [4,]    1 366745 7.02    0
## [5,]    1 265518 8.61    1
## [6,]    1 301995 6.88    0
```

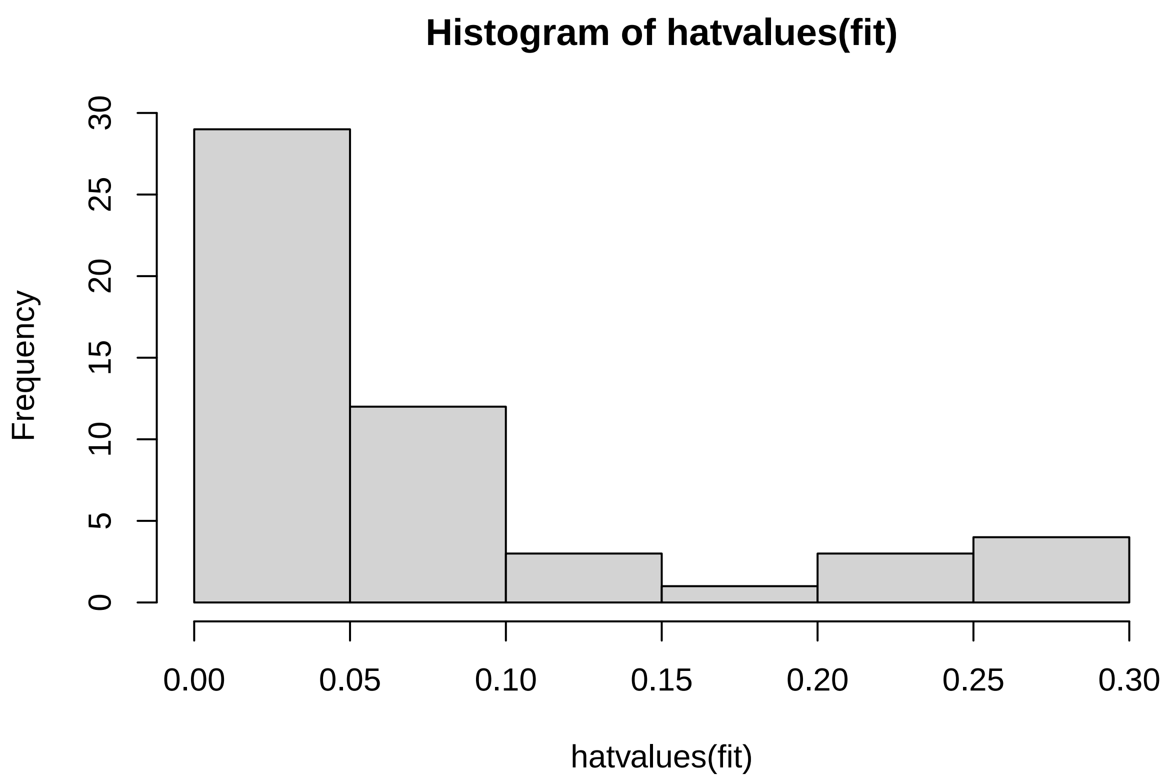
```r
X.new <- c(1,300000, 7.2, 0)
h.new.new <- t(X.new)%*%solve(t(X)%*%X)%*%X.new
h.new.new
```

```
##            [,1]
## [1,] 0.02221728
```

```r
summary(hatvalues(fit))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.02189 0.03147 0.04724 0.07692 0.06956 0.28686
```

```r
hist(hatvalues(fit))
```



$h_{new,new}$ is within [0.02189, 0.06956], then no extrapolaion is indicated. Visually, according to the plot of X2 against X1, the new observation shows no extreme. Hence, the conclusions agree from these two methods.

(d.)

```r
# Cook's distance values
CD <- cooks.distance(fit)[c(16, 22, 43, 48, 10, 32, 38, 40)]
CD
```

```
##           16           22           43           48           10           32
## 0.0768950835 0.0007746088 0.0792193145 0.0054988670 0.0493501187 0.0997597379
##           38           40
```

```
## 0.0346380312 0.0364991539
```

```r
pf(CD, p, n-p)
```

```
##           16           22           43           48           10           32
## 1.103491e-02 1.248642e-06 1.167356e-02 6.249683e-05 4.726750e-03 1.798234e-02
##           38           40
## 2.378087e-03 2.633476e-03
```

```r
# DFFITS
DFFITS <- dffits(fit)[c(16, 22, 43, 48, 10, 32, 38, 40)]
DFFITS
```

```
##          16          22          43          48          10          32
## -0.55399026  0.05508583  0.56165186 -0.14684146  0.45863297 -0.65107706
##          38          40
##  0.38551766  0.39672030
```

```r
DFFITS > 1 # small data set
```

```
##    16    22    43    48    10    32    38    40
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```r
# DFBETAS
DFBETAS <- dfbetas(fit)[c(16, 22, 43, 48, 10, 32, 38, 40)]
DFBETAS
```

```
## [1] -0.24768867  0.03042319 -0.35779734  0.04498580  0.36407495  0.40954150
## [7] -0.09961479  0.07379876
```

```r
CDDvalues <- cbind("X" = c(16, 22, 43, 48, 10, 32, 38, 40), CD, DFFITS, DFBETAS)
CDDvalues
```

```
##     X           CD      DFFITS     DFBETAS
## 16 16 0.0768950835 -0.55399026 -0.24768867
## 22 22 0.0007746088  0.05508583  0.03042319
## 43 43 0.0792193145  0.56165186 -0.35779734
## 48 48 0.0054988670 -0.14684146  0.04498580
## 10 10 0.0493501187  0.45863297  0.36407495
## 32 32 0.0997597379 -0.65107706  0.40954150
## 38 38 0.0346380312  0.38551766 -0.09961479
## 40 40 0.0364991539  0.39672030  0.07379876
```

Considering all CDs are below 20%, it indicates little influence on the fitted values. All the DFFITS and absolute values of DFBETAS are smaller than 1, which indicates no influential observations.

(e.)

```r
# Cook's distance values
pred1 <- fitted.values(fit)
AAPD <- numeric(8)
a <- 1
for(i in c(16, 22, 43, 48, 10, 32, 38, 40)){
  fit2 <- lm(Y~X1+X2+X3, data = input2[-i,])
  pred2 <- predict(fit2, input2)
  AAPD[a] <- mean(abs((pred2-pred1)/pred1*100))
  a <- a+1
}
AAPD_df <- data.frame(AAPD)
rownames(AAPD_df) <- c(16, 22, 43, 48, 10, 32, 38, 40)
```

```
AAPD_df
```

```
##          AAPD
## 16 0.16059358
## 22 0.01498540
## 43 0.16364834
## 48 0.04207119
## 10 0.16671403
## 32 0.22748377
## 38 0.15202218
## 40 0.15653220
```

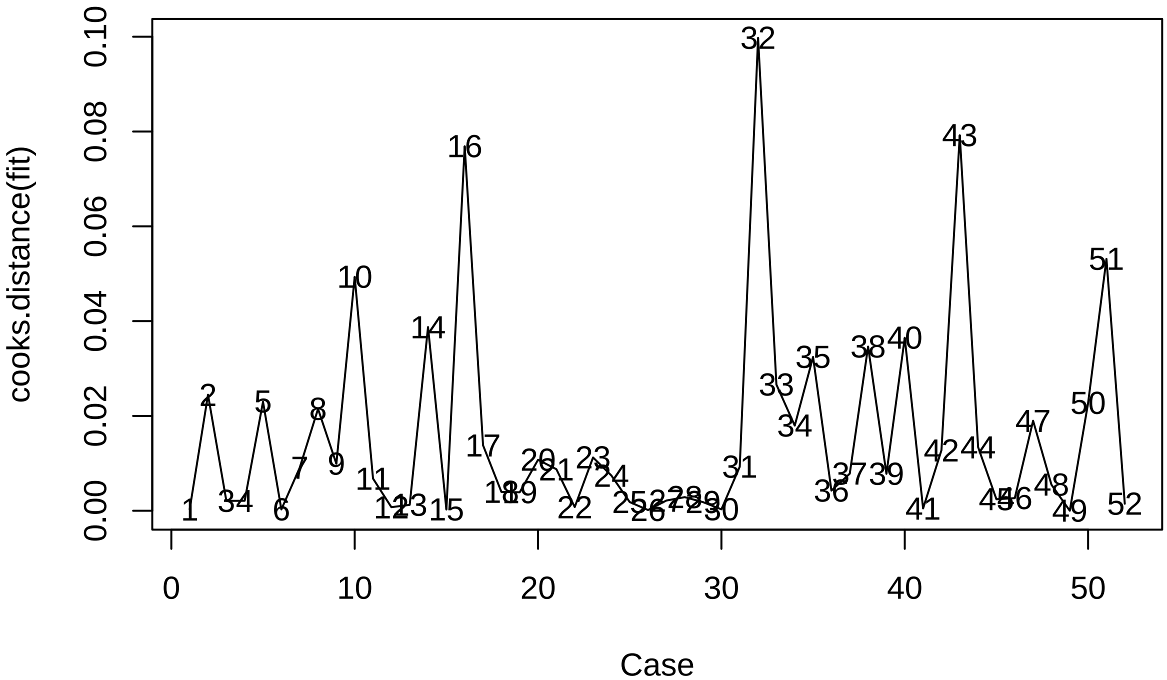
All the mean absolute difference percents are small, which means little influence and no remedical is needed.

(f.)

```
cooks.distance(fit)
```

```
##            1            2            3            4            5            6 
## 2.959337e-04 2.447558e-02 2.080650e-03 2.106453e-03 2.299628e-02 2.735616e-04 
##            7            8            9           10           11           12 
## 9.066686e-03 2.148586e-02 9.920293e-03 4.935012e-02 6.798544e-03 7.550338e-04 
##           13           14           15           16           17           18 
## 1.245024e-03 3.875147e-02 2.606639e-04 7.689508e-02 1.381267e-02 3.972613e-03 
##           19           20           21           22           23           24 
## 3.899679e-03 1.075330e-02 8.762889e-03 7.746088e-04 1.124087e-02 7.426012e-03 
##           25           26           27           28           29           30 
## 1.827605e-03 1.541471e-04 2.157761e-03 2.871488e-03 1.817555e-03 2.983322e-04 
##           31           32           33           34           35           36 
## 9.283401e-03 9.975974e-02 2.661839e-02 1.796257e-02 3.245103e-02 4.246344e-03 
##           37           38           39           40           41           42 
## 7.821682e-03 3.463803e-02 7.787509e-03 3.649915e-02 4.919467e-04 1.276193e-02 
##           43           44           45           46           47           48 
## 7.921931e-02 1.341707e-02 2.426422e-03 2.658843e-03 1.898976e-02 5.498867e-03 
##           49           50           51           52 
## 2.335319e-06 2.274285e-02 5.313716e-02 1.476023e-03
```

```
plot(Case,cooks.distance(fit),type="l")
text(Case,cooks.distance(fit))
```

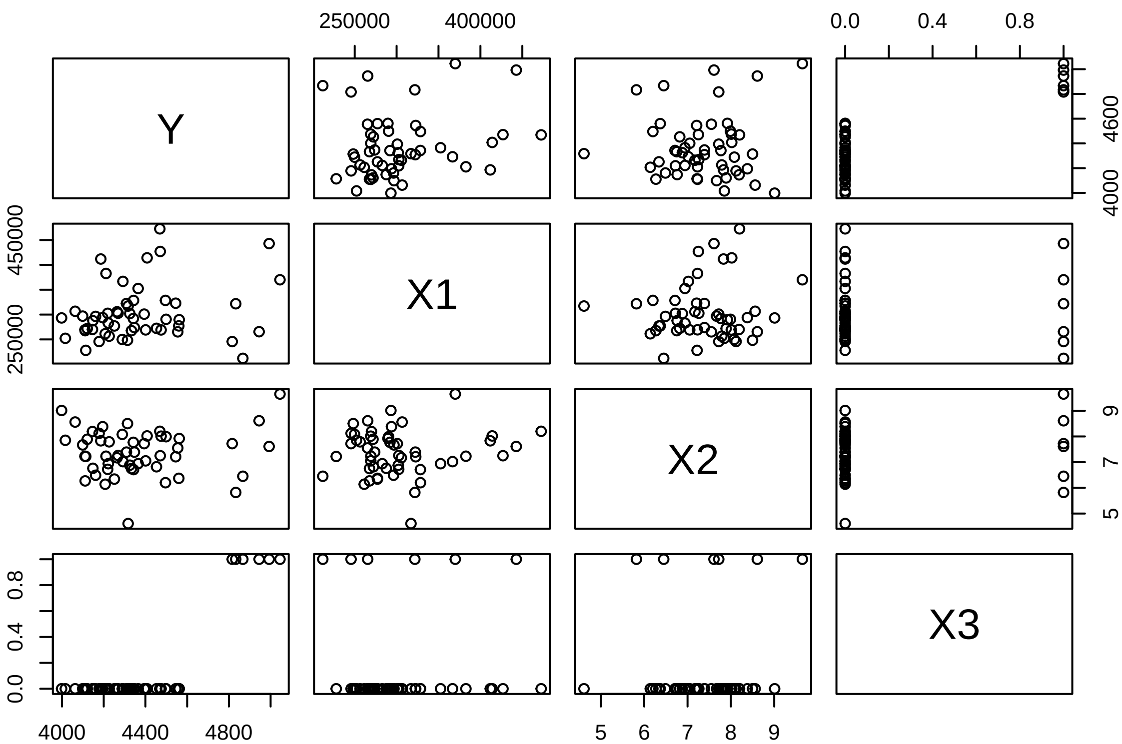


The Cook's distances indicates there are no influential cases.

10.16

(a.)

```
pairs(input2)
```




```
cor(input2)
```

```
##            Y         X1         X2         X3
## Y  1.0000000 0.20766494 0.06002960 0.81057940
## X1 0.2076649 1.00000000 0.08489639 0.04565698
## X2 0.0600296 0.08489639 1.00000000 0.11337076
## X3 0.8105794 0.04565698 0.11337076 1.00000000
```

The scatter plot matrix and the correlation matrix shows no obvious correlations between Y and X1, X2. However, there is a correlation between Y and X3 because the correlation is close to 1.X1, X2 and X3 have low correlation.

(b.)

```
library(faraway)
vif(fit)
```

```
##       X1       X2       X3
## 1.008596 1.019598 1.014364
```

Hence, there is no multiplicity problem (All VIF values are close to 1).

Extra credit problem

```
input3 <- read.csv("./BookDataSets/DataSet.csv", header = TRUE)
Data1_index <- sample(1:168, 130, replace = TRUE)
Data1 <- input3[Data1_index, ]
Data2 <- input3[-Data1_index, ]
names(input3)
```

```
## [1] "Price"   "Food"    "Decor"   "Service" "East"
```

(1.)
Use Data1 to establish the model, I choose Mallow's C_p criterion.

```
library(leaps)
Data1.Cp <- leaps(x=Data1[,2:5], y=Data1[,1], names=names(Data1)[2:5], method="Cp")

Data1.Cp <- data.frame(Data1.Cp$which, Data1.Cp$Cp)
colnames(Data1.Cp)[5] <- "Cp"
Data1.Cp[order(Data1.Cp$Cp, decreasing = FALSE), ]
```

```
##       Food Decor Service  East         Cp
## X4    TRUE  TRUE    TRUE  TRUE   5.000000
## X3    TRUE  TRUE    TRUE FALSE   5.217269
## X3.1 FALSE  TRUE    TRUE  TRUE   6.060896
## X3.2  TRUE  TRUE   FALSE  TRUE   6.460999
## X2    TRUE  TRUE   FALSE FALSE   6.578906
## X2.1 FALSE  TRUE    TRUE FALSE   7.764373
## X2.2 FALSE  TRUE   FALSE  TRUE  27.159074
## X1   FALSE  TRUE   FALSE FALSE  34.848856
## X2.3  TRUE FALSE    TRUE FALSE  50.630348
## X3.3  TRUE FALSE    TRUE  TRUE  52.252077
## X1.1 FALSE FALSE    TRUE FALSE  55.232648
## X2.4 FALSE FALSE    TRUE  TRUE  57.222214
## X1.2  TRUE FALSE   FALSE FALSE  71.110206
## X2.5  TRUE FALSE   FALSE  TRUE  71.533126
## X1.3 FALSE FALSE   FALSE  TRUE 230.416156
```

$C_p \leq p$ indicates a good model. Hence, one good model is needed, I choose Food(X1), Decor(X2) and East(X4) as my model subset.

```r
Data1.fit <- lm(formula = Price ~ Food + Decor + East, data = Data1)
summary(Data1.fit)
```

```
##
## Call:
## lm(formula = Price ~ Food + Decor + East, data = Data1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.1148  -3.7037   0.4867   3.5053  16.5671
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -23.3713     5.2746  -4.431 2.02e-05 ***
## Food          1.5209     0.3223   4.718 6.21e-06 ***
## Decor         1.9193     0.2366   8.111 3.83e-13 ***
## East          1.4794     1.0264   1.441    0.152
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.363 on 126 degrees of freedom
## Multiple R-squared:  0.6401, Adjusted R-squared:  0.6315
## F-statistic: 74.68 on 3 and 126 DF,  p-value: < 2.2e-16
```

(2.)

```r
model1 <- lm(Price~Food+Decor+East, data=Data1)
lm(model1, Data2)
```

```
##
## Call:
## lm(formula = model1, data = Data2)
##
## Coefficients:
## (Intercept)         Food        Decor         East
##     -22.838        1.523        1.810        2.089
```

```r
pred1 <- predict(model1, Data2)
sqrt(mean((Data2$Price-pred1)^2))
```

```
## [1] 5.813263
```

```r
model2 <- lm(Price~., data=Data1)
lm(model2, Data2)
```

```
##
## Call:
## lm(formula = model2, data = Data2)
##
## Coefficients:
## (Intercept)         Food        Decor      Service         East
##    -22.4982       1.7602       1.9332      -0.3892       2.3118
```

```r
pred2 <- predict(model2, Data2)
sqrt(mean((Data2$Price-pred2)^2))
```

```
## [1] 6.03829
```

Using the square root of MSPE, the model that I choose in step 1 behaves much better with smaller value in compare with uing all predictors.

(3.)
Define the full model Price ~ Food + Decor + East + Food*Decor + Food*East. In case of multicolinearity, we do transformation first.

```r
Data1.trans <- data.frame("Price" = Data1$Price-mean(Data1$Price),
                                  "Food" = Data1$Food-mean(Data1$Food),
                                  "Decor" = Data1$Decor-mean(Data1$Decor),
                                  "East" = Data1$East-mean(Data1$East))
Full <- lm(Price~Food+Decor+East + Food*East, data = Data1.trans)
model <- lm(Price~Food+Decor+East, data = Data1.trans)
anova(model,Full)
```

```
## Analysis of Variance Table
##
## Model 1: Price ~ Food + Decor + East
## Model 2: Price ~ Food + Decor + East + Food * East
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    126 3623.8
## 2    125 3579.1  1    44.714 1.5616 0.2138
```

Choose $\alpha = 0.05$.
Let $H_0: $ the coefficients of interaction terms are zero and $H_a: $ at least one of them is not zero.
Decision rule: we conclude H_0 if p-value is larger than $\alpha = 0.05$. Otherwise, conclude H_a.
The p-value > 0.05, we conculde H_0.