STT481 HW1 solution

Q3

(3a) The Euclidean distances are computed as follows.

$$\sqrt{(1-0)^2+(0-3)^2+(1-0)^2} = \sqrt{11},$$
$$\sqrt{(1-2)^2+(0-0)^2+(1-0)^2} = \sqrt{2},$$
$$\sqrt{(1-0)^2+(0-1)^2+(1-3)^2} = \sqrt{6},$$
$$\sqrt{(1-0)^2+(0-1)^2+(1-2)^2} = \sqrt{3},$$
$$\sqrt{(1+1)^2+(0-0)^2+(1-1)^2} = \sqrt{4},$$
$$\sqrt{(1-1)^2+(0-1)^2+(1-1)^2} = \sqrt{1},$$

(3b) The closest neighbor is the observation 6 so the prediction is Red.

(3c) The closest three neighbors are the observations 2, 4, 6 so the prediction is Red (majority is Red).

(3d) The best value of K should be smaller to be able to capture more of the non-linear decision boundary.

(3e)

```
library(class)
X.train <- rbind(c(0,3,0),c(2,0,0),c(0,1,3),c(0,1,2),c(-1,0,1),c(1,1,1))
y.train <- c("Red", "Red", "Red", "Green", "Green", "Red")
X.test <- matrix(c(1,0,1),nrow=1)
pred.out <- knn(train=X.train, test=X.test, cl=y.train, k=1)
pred.out # prediction of (b)
```

```
## [1] Red
## Levels: Green Red
```

```
pred.out <- knn(train=X.train, test=X.test, cl=y.train, k=3)
pred.out # prediction of (c)
```

```
## [1] Red
## Levels: Green Red
```

Q4

(4a) load data.

```
college <- read.csv("College.csv")
```

(4b) View data.

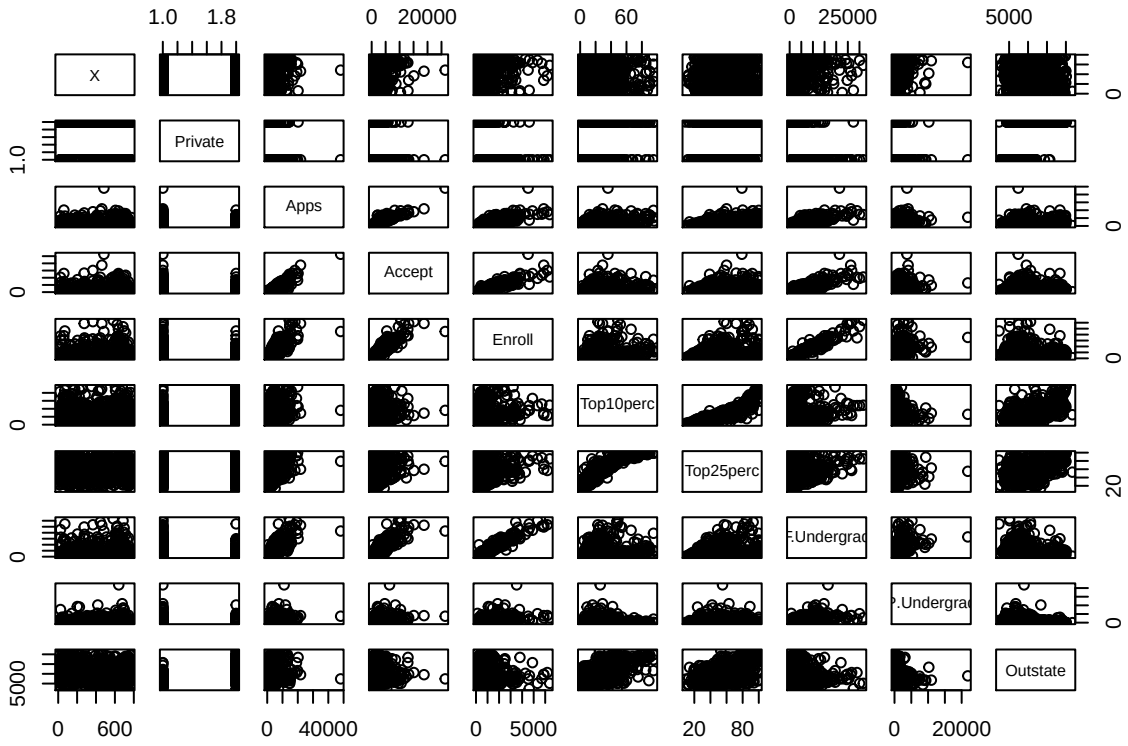
```
rownames(college) <- college[,1]
View(college)
college <- college[,-1]
View(college)
```

(4c)

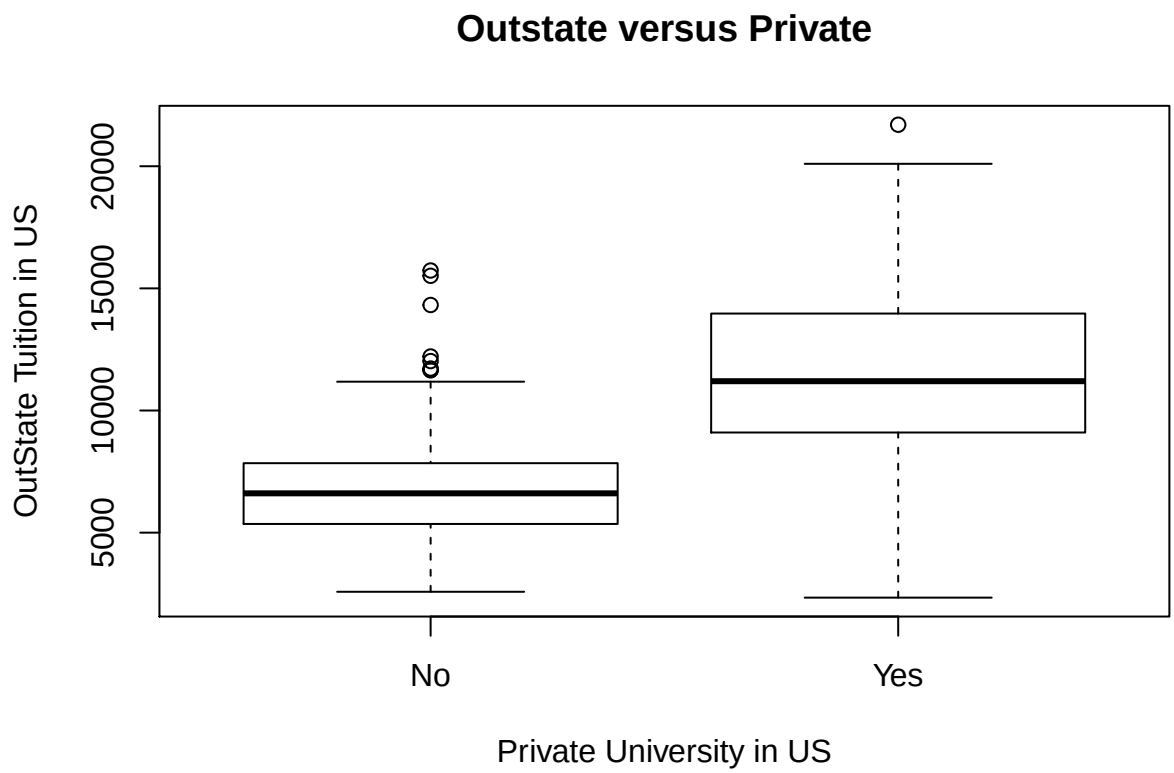
```
#(i)
summary(college)
```

```
##  Private        Apps           Accept          Enroll       Top10perc
##  No :212   Min.   :   81   Min.   :   72   Min.   :  35   Min.   : 1.00
##  Yes:565   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242   1st Qu.:15.00
##            Median : 1558   Median : 1110   Median : 434   Median :23.00
##            Mean   : 3002   Mean   : 2019   Mean   : 780   Mean   :27.56
##            3rd Qu.: 3624   3rd Qu.: 2424   3rd Qu.: 902   3rd Qu.:35.00
##            Max.   :48094   Max.   :26330   Max.   :6392   Max.   :96.00
##    Top25perc      F.Undergrad     P.Undergrad         Outstate
##  Min.   :  9.0   Min.   :  139   Min.   :    1.0   Min.   : 2340
##  1st Qu.: 41.0   1st Qu.:  992   1st Qu.:   95.0   1st Qu.: 7320
##  Median : 54.0   Median : 1707   Median :  353.0   Median : 9990
##  Mean   : 55.8   Mean   : 3700   Mean   :  855.3   Mean   :10441
##  3rd Qu.: 69.0   3rd Qu.: 4005   3rd Qu.:  967.0   3rd Qu.:12925
##  Max.   :100.0   Max.   :31643   Max.   :21836.0   Max.   :21700
##    Room.Board       Books           Personal         PhD
##  Min.   :1780   Min.   :  96.0   Min.   : 250   Min.   :  8.00
##  1st Qu.:3597   1st Qu.: 470.0   1st Qu.: 850   1st Qu.: 62.00
##  Median :4200   Median : 500.0   Median :1200   Median : 75.00
##  Mean   :4358   Mean   : 549.4   Mean   :1341   Mean   : 72.66
##  3rd Qu.:5050   3rd Qu.: 600.0   3rd Qu.:1700   3rd Qu.: 85.00
##  Max.   :8124   Max.   :2340.0   Max.   :6800   Max.   :103.00
##     Terminal       S.F.Ratio      perc.alumni        Expend
##  Min.   : 24.0   Min.   : 2.50   Min.   : 0.00   Min.   : 3186
##  1st Qu.: 71.0   1st Qu.:11.50   1st Qu.:13.00   1st Qu.: 6751
##  Median : 82.0   Median :13.60   Median :21.00   Median : 8377
##  Mean   : 79.7   Mean   :14.09   Mean   :22.74   Mean   : 9660
##  3rd Qu.: 92.0   3rd Qu.:16.50   3rd Qu.:31.00   3rd Qu.:10830
##  Max.   :100.0   Max.   :39.80   Max.   :64.00   Max.   :56233
##    Grad.Rate
##  Min.   : 10.00
##  1st Qu.: 53.00
##  Median : 65.00
##  Mean   : 65.46
##  3rd Qu.: 78.00
##  Max.   :118.00
```

```
#(ii)
A <- read.csv("College.csv")
pairs(A[,1:10])
```




```
#(iii)
plot(A$Private, A$Outstate, main = "Outstate versus Private", xlab = "Private University in US", ylab =
```

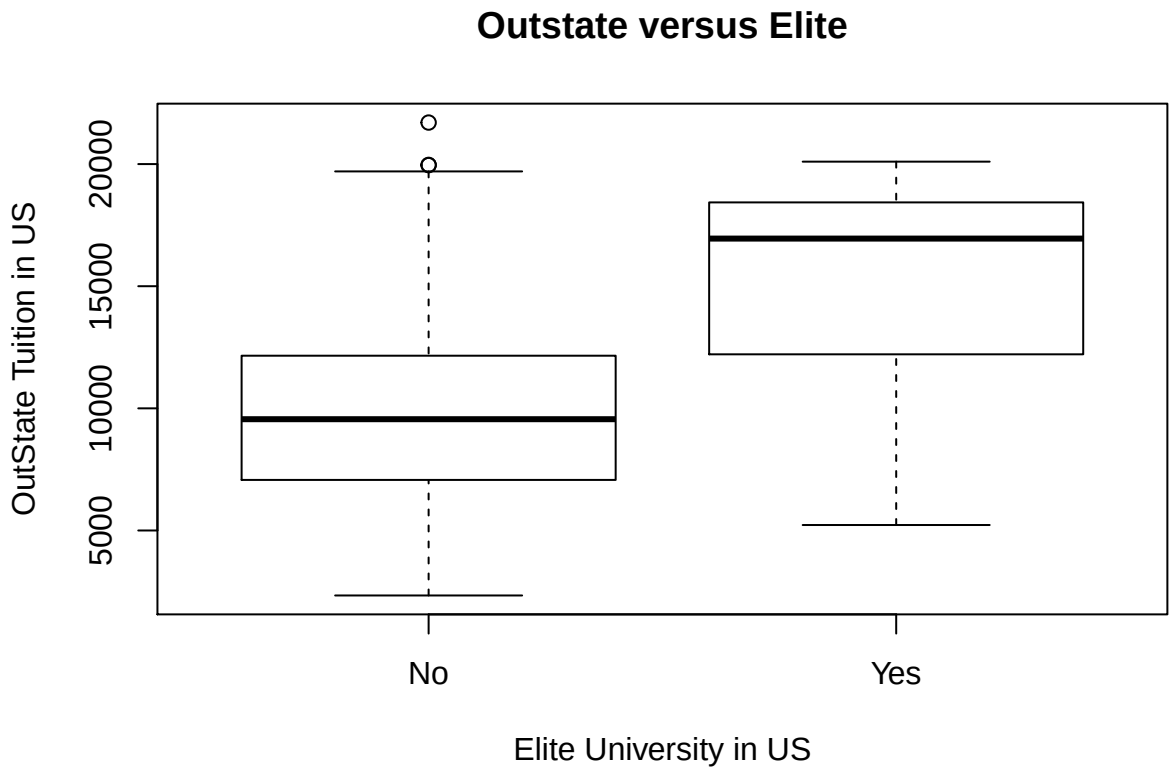



```
#(iv)
Elite <- rep("No", nrow(college))
Elite[college$Top10perc>50] <- "Yes"
```

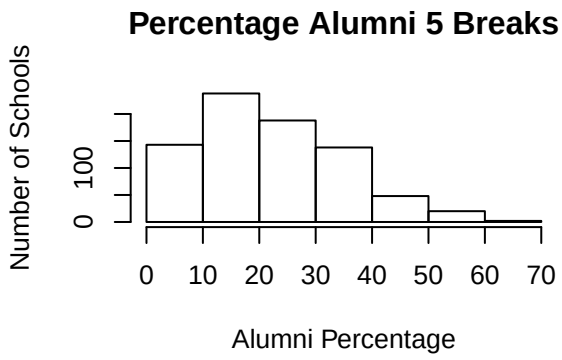
```
Elite = as.factor(Elite)
college = data.frame(college, Elite)
summary(college$Elite)
```

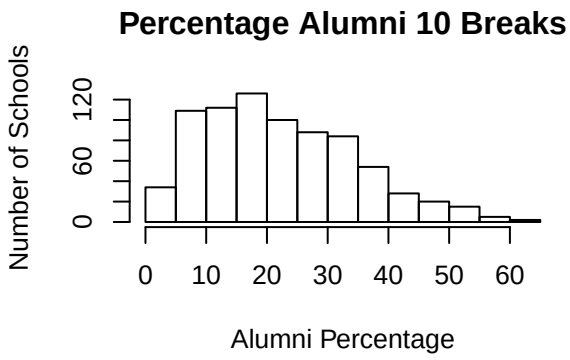
```
##  No Yes
## 699  78
```

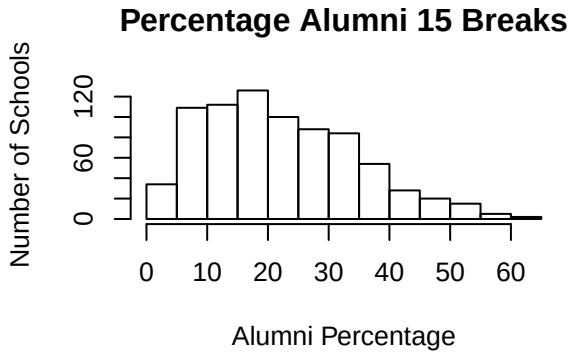
```
plot(college$Elite, college$Outstate, main = "Outstate versus Elite", xlab = "Elite University in US", y
```

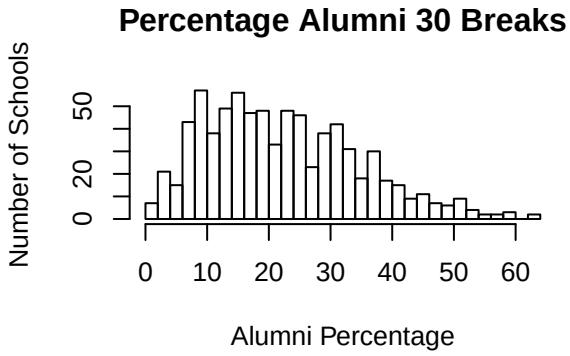



```
# (v) Percentage of Alumni who donate histograms
par(mfrow = c(2,2))
hist(college$perc.alumni, breaks = 5, main = "Percentage Alumni 5 Breaks",
     ylab = "Number of Schools", xlab= "Alumni Percentage")
hist(college$perc.alumni, breaks = 10, main = "Percentage Alumni 10 Breaks",
     ylab = "Number of Schools", xlab= "Alumni Percentage")
hist(college$perc.alumni, breaks = 15, main = "Percentage Alumni 15 Breaks",
     ylab = "Number of Schools", xlab= "Alumni Percentage")
hist(college$perc.alumni, breaks = 30, main = "Percentage Alumni 30 Breaks",
     ylab = "Number of Schools", xlab= "Alumni Percentage")
```

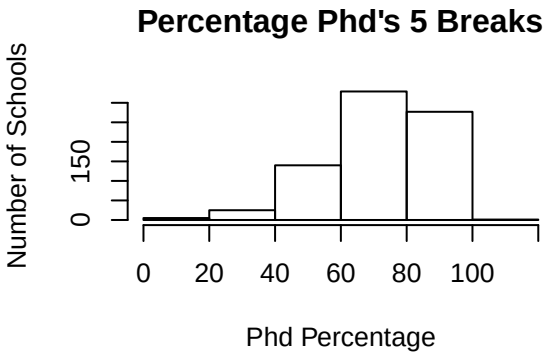


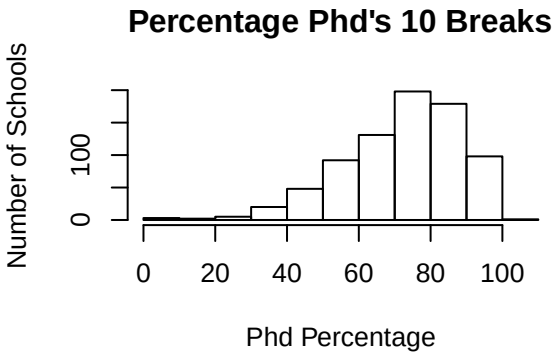


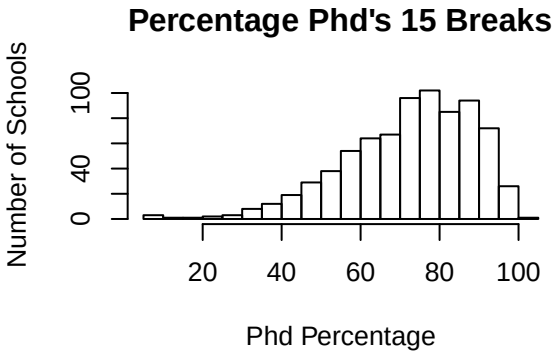


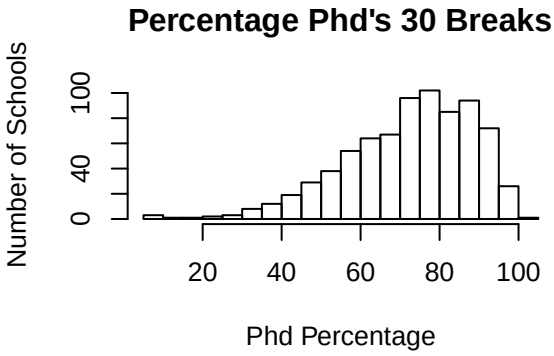



```r
# (v) Percent of faculty with Ph.D's histograms
par(mfrow = c(2,2))
hist(college$PhD, breaks = 5, main = "Percentage Phd's 5 Breaks",
     ylab = "Number of Schools", xlab= "Phd Percentage")
hist(college$PhD, breaks = 10, main = "Percentage Phd's 10 Breaks",
     ylab = "Number of Schools", xlab= "Phd Percentage")
hist(college$PhD, breaks = 15, main = "Percentage Phd's 15 Breaks",
     ylab = "Number of Schools", xlab= "Phd Percentage")
hist(college$PhD, breaks = 30, main = "Percentage Phd's 30 Breaks",
     ylab = "Number of Schools", xlab= "Phd Percentage")
```

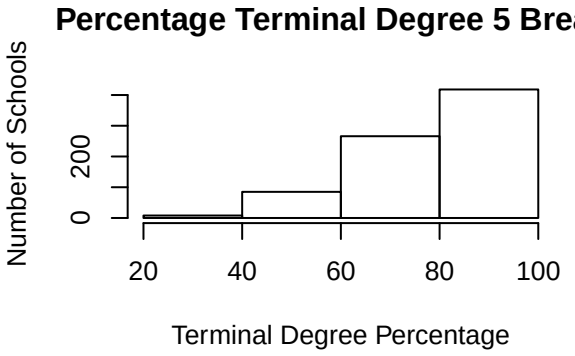


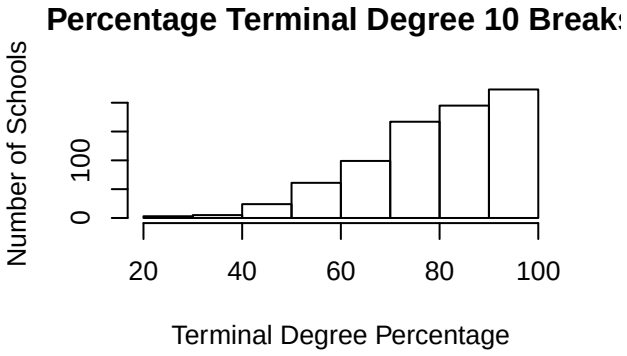


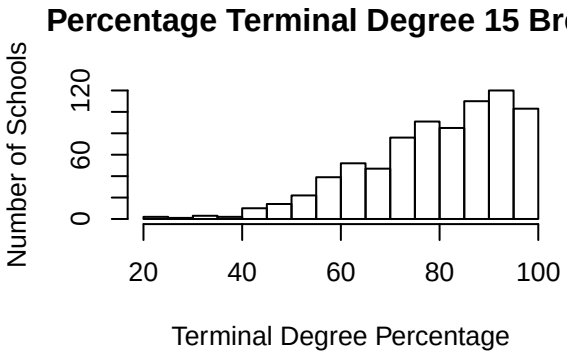


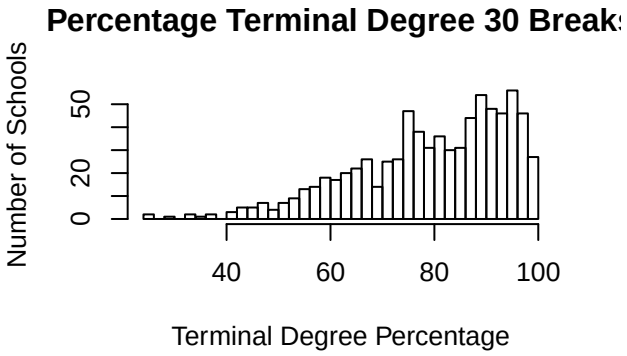



```r
# (v) Percent of faculty with terminal degree (highest degree in a given field)
par(mfrow = c(2,2))
hist(college$Terminal, breaks = 5, main = "Percentage Terminal Degree 5 Breaks",
     ylab = "Number of Schools", xlab= "Terminal Degree Percentage")
hist(college$Terminal, breaks = 10, main = "Percentage Terminal Degree 10 Breaks",
     ylab = "Number of Schools", xlab= "Terminal Degree Percentage")
hist(college$Terminal, breaks = 15, main = "Percentage Terminal Degree 15 Breaks",
     ylab = "Number of Schools", xlab= "Terminal Degree Percentage")
hist(college$Terminal, breaks = 30, main = "Percentage Terminal Degree 30 Breaks",
     ylab = "Number of Schools", xlab= "Terminal Degree Percentage")
```










```
# (vi)
summary(college$Top25perc)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     9.0    41.0    54.0    55.8    69.0   100.0
```

```
summary(college$PhD)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    8.00   62.00   75.00   72.66   85.00  103.00
```

```
summary(college$Expend)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    3186    6751    8377    9660   10830   56233
```

```
summary(college$Grad.Rate)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   10.00   53.00   65.00   65.46   78.00  118.00
```

For (vi), it can be seen that the college PhD, and the Graduation Rate has exceeded hundred percent, and the colleges that have stranges values are:

```
weird.phd = college[college$PhD == 103,]
print(rownames(weird.phd))
```

```
## [1] "Texas A&M University at Galveston"
```

```
weird.grad.rate = college[college$Grad.Rate == 118,]
print(rownames(weird.grad.rate))
```

```
## [1] "Cazenovia College"
```

Q5

(5a) quantitative: mpg, cylinders, displacement, horsepower, weight, acceleration, year; qualitative: name, origin.

(5b)

```
library(ISLR)
Auto <- na.omit(Auto)
head(Auto)
```

```
##   mpg cylinders displacement horsepower weight acceleration year origin
## 1  18         8          307        130   3504         12.0   70      1
## 2  15         8          350        165   3693         11.5   70      1
## 3  18         8          318        150   3436         11.0   70      1
## 4  16         8          304        150   3433         12.0   70      1
## 5  17         8          302        140   3449         10.5   70      1
## 6  15         8          429        198   4341         10.0   70      1
##                       name
## 1 chevrolet chevelle malibu
## 2         buick skylark 320
## 3        plymouth satellite
## 4             amc rebel sst
## 5               ford torino
## 6          ford galaxie 500
```

```
summary(Auto)
```

```
##       mpg          cylinders      displacement     horsepower        weight
##  Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0   Min.   :1613
##  1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0   1st Qu.:2225
##  Median :22.75   Median :4.000   Median :151.0   Median : 93.5   Median :2804
##  Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5   Mean   :2978
##  3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0   3rd Qu.:3615
##  Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0   Max.   :5140
##
##   acceleration        year           origin                      name
##  Min.   : 8.00   Min.   :70.00   Min.   :1.000   amc matador       :  5
##  1st Qu.:13.78   1st Qu.:73.00   1st Qu.:1.000   ford pinto        :  5
##  Median :15.50   Median :76.00   Median :1.000   toyota corolla    :  5
##  Mean   :15.54   Mean   :75.98   Mean   :1.577   amc gremlin       :  4
##  3rd Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.000   amc hornet        :  4
##  Max.   :24.80   Max.   :82.00   Max.   :3.000   chevrolet chevette:  4
##                                                  (Other)           :365
```

```
sapply(Auto[, 1:7], range)
```

```
##       mpg cylinders displacement horsepower weight acceleration year
## [1,]  9.0         3           68         46   1613          8.0   70
## [2,] 46.6         8          455        230   5140         24.8   82
```

(5c)

```
sapply(Auto[, 1:7], mean)
```

```
##          mpg    cylinders displacement   horsepower       weight acceleration
##    23.445918     5.471939   194.411990   104.469388  2977.584184    15.541327
##         year
```

```
##    75.979592
```

```
sapply(Auto[, 1:7], sd)
```

```
##          mpg    cylinders displacement   horsepower       weight acceleration
##     7.805007     1.705783   104.644004    38.491160   849.402560     2.758864
##         year
##     3.683737
```

(5d)

```
newAuto <- Auto[-(10:85),]
dim(newAuto) == dim(Auto) - c(76,0)
```

```
## [1] TRUE TRUE
```

```
newAuto[9,] <- Auto[9,]
newAuto[10,] <- Auto[86,]

sapply(newAuto[, 1:7], range)
```

```
##       mpg cylinders displacement horsepower weight acceleration year
## [1,] 11.0         3           68         46   1649          8.5   70
## [2,] 46.6         8          455        230   4997         24.8   82
```

```
sapply(newAuto[, 1:7], mean)
```

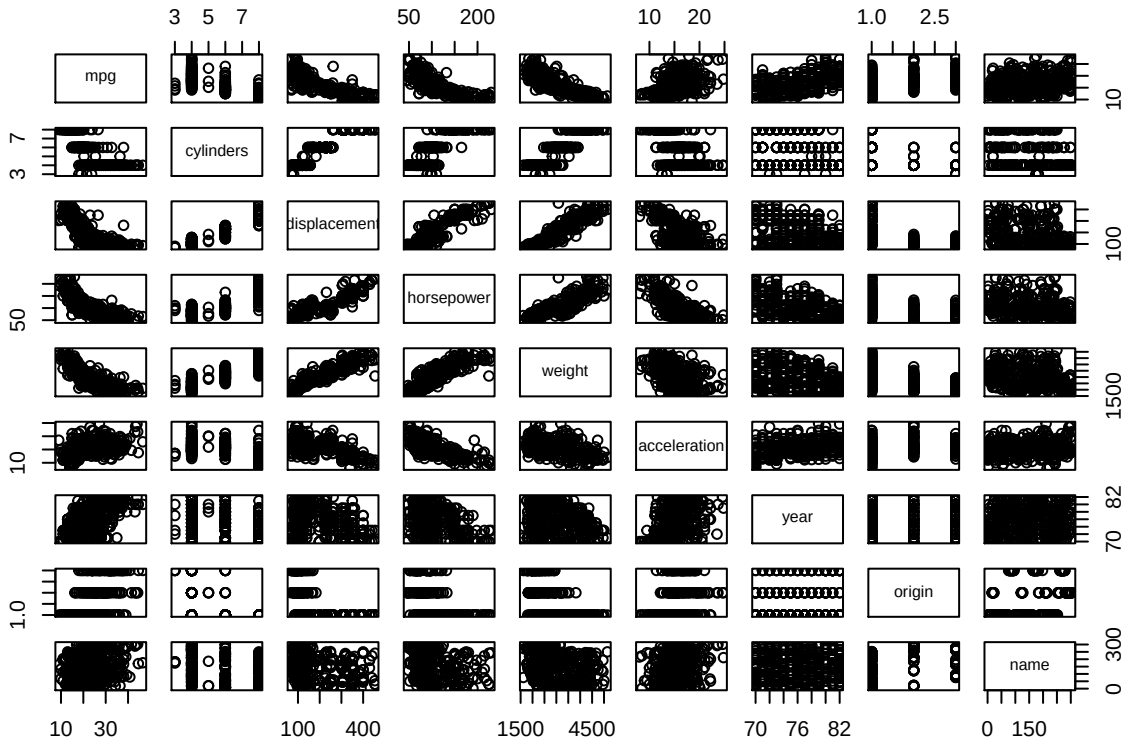
```
##          mpg    cylinders displacement   horsepower       weight acceleration
##    24.404430     5.373418   187.240506   100.721519  2935.971519    15.726899
##         year
##    77.145570
```

```
sapply(newAuto[, 1:7], sd)
```

```
##          mpg    cylinders displacement   horsepower       weight acceleration
##     7.867283     1.654179    99.678367    35.708853   811.300208     2.693721
##         year
##     3.106217
```

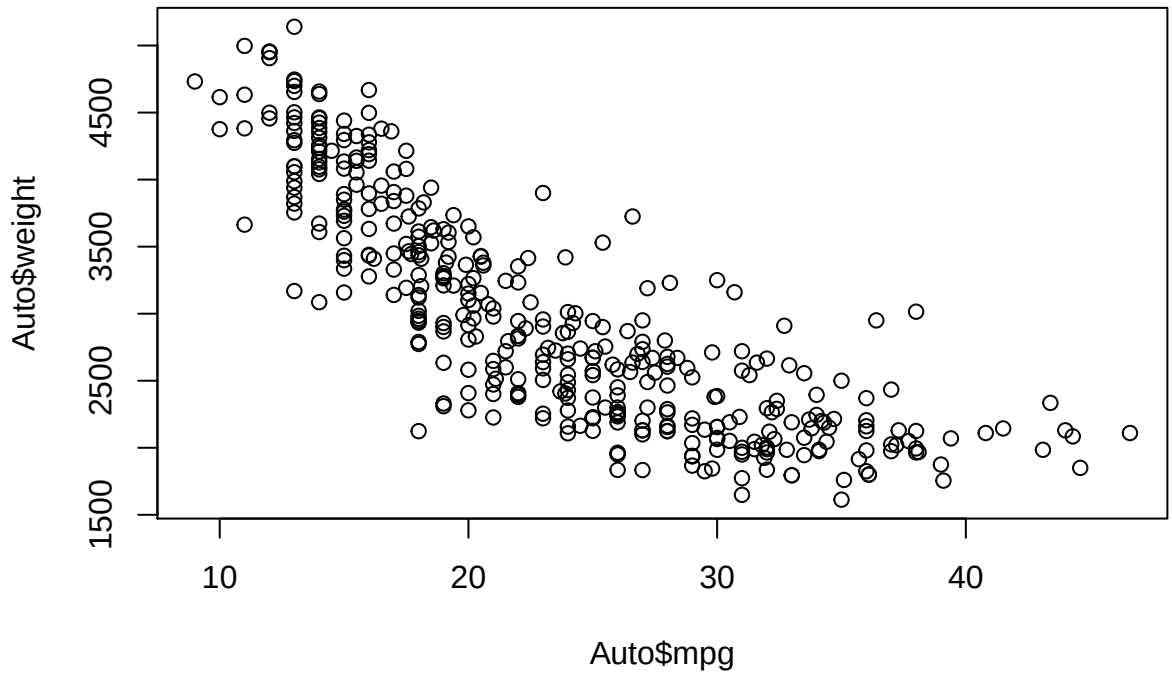
(5e)

```
pairs(Auto)
```



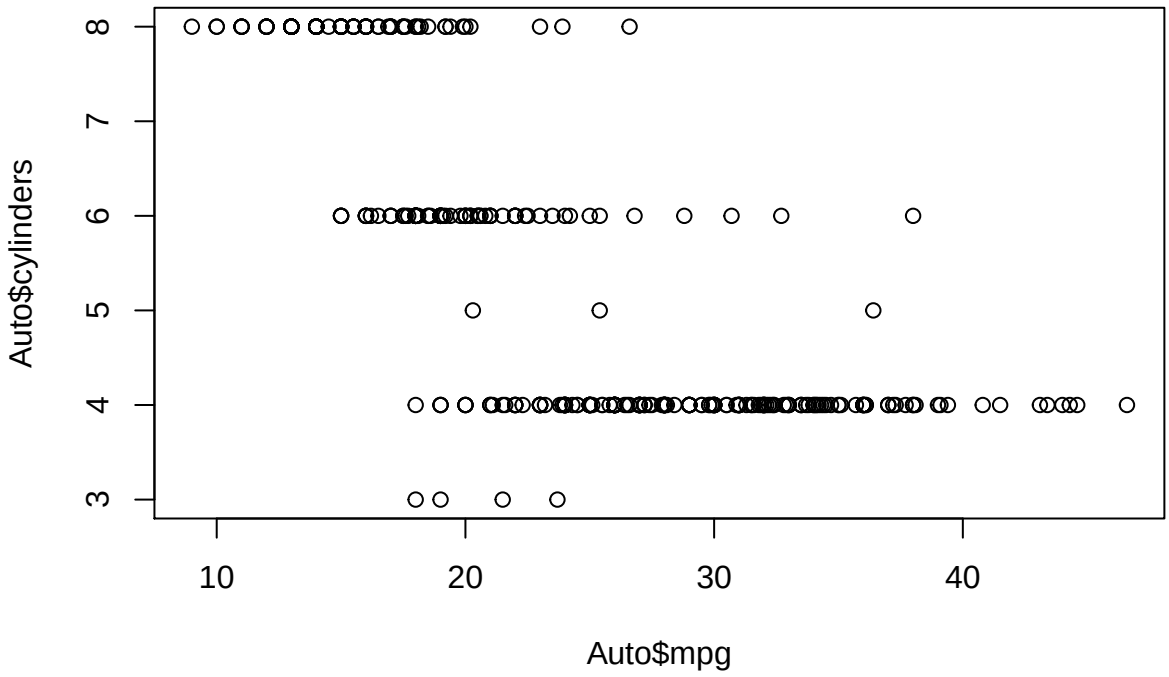
From the plot, it is observed that mpg under study has an inverse relation between weight, horsepower and displacement, while a direct relation with that of years. There is more mileage on four-cylinder engines, and a better mpg in Japanses made than US or European made.

```
plot(Auto$mpg, Auto$weight)
```



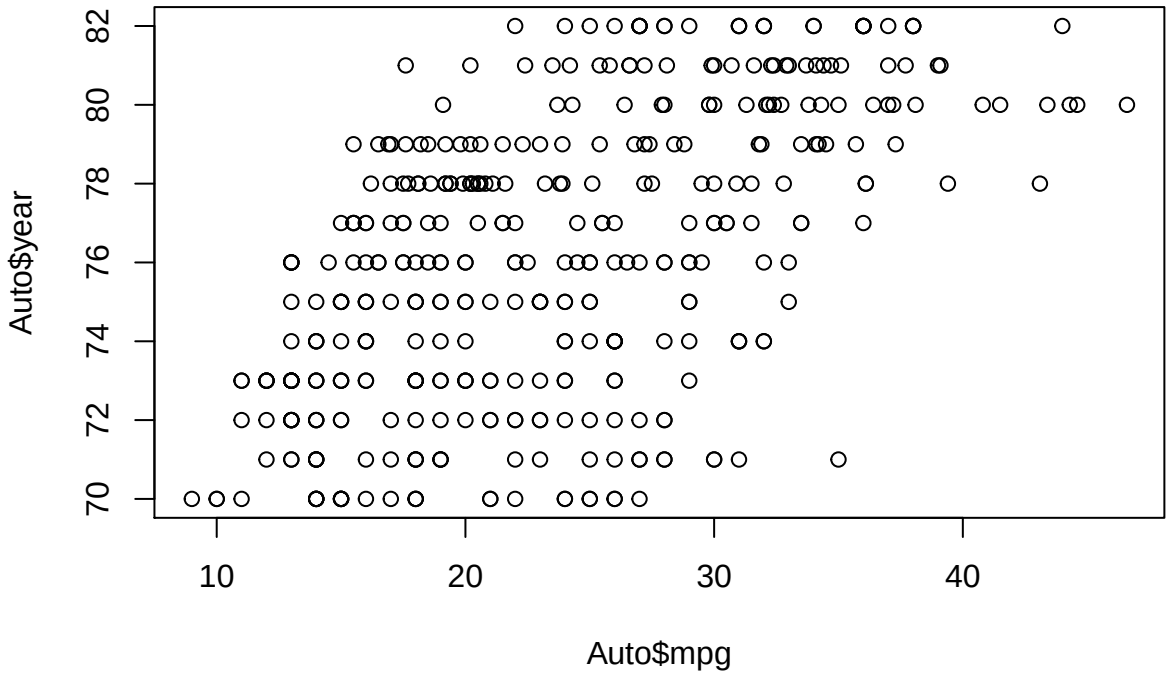
Heavier weight correlates with lower mpg.

```
plot(Auto$mpg, Auto$cylinders)
```



More cylinders, less mpg.

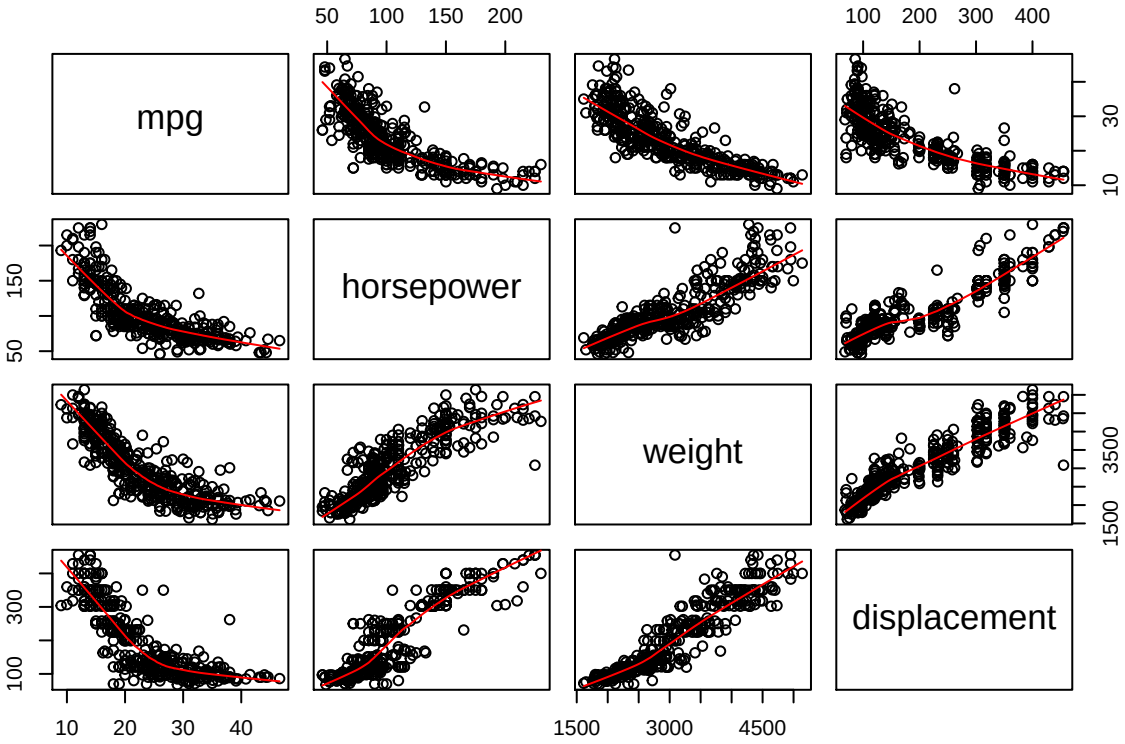
```
plot(Auto$mpg, Auto$year)
```



Cars become more efficient over time.

(5f)

```
pairs(~mpg+horsepower+weight+displacement, data=Auto, panel=panel.smooth)
```



From the plot of mpg vs horsepower, all data is in decreasing trend.

From the plot of mpg vs weight, all data is in decreasing trend.

From the plot of mpg vs displacement, all data is in decreasing trend.

From all the plots, it is observed that the mpg with any variable is in decreasing trend.

Q6

(6a) The least square equation is:

$$\hat{Y} = 50 + 20(gpa) + 0.07(iq) + 35(level) + 0.01(gpa * iq) - 10(gpa * level)$$

high school: (level = 0) 50+20(gpa) +0.07(iq) +0.01(gpa*iq)

college: (level = 1) 50+20(gpa) +0.07(iq) +35+0.01(gpa*iq)-10(gpa)

Once the GPA is high enough (GPA>3.5), high school earn more on average than college. Therefore, answer iii is correct.

(6b)

$$50 + 20 \times 4.0 + 0.07 \times 110 + 35 \times 1 + 0.01 \times (110 \times 4.0) - 10 \times (4.0 \times 1) = 137.1$$

Thus the predicted salary is $137,100.

(6c) The given statement is a false one. We must examine the p-value of the regression coefficient to determine if the interaction term is statistically significant or not.

Q7

(7a) Having more predictors generally means better (lower) RSS on training data, because it could make a tighter fit against data that matched with a wider irreducible error (Var(epsilon)).

(7b) If the additional predictors lead to overfitting, the testing RSS could be worse (higher) for the cubic regression fit.

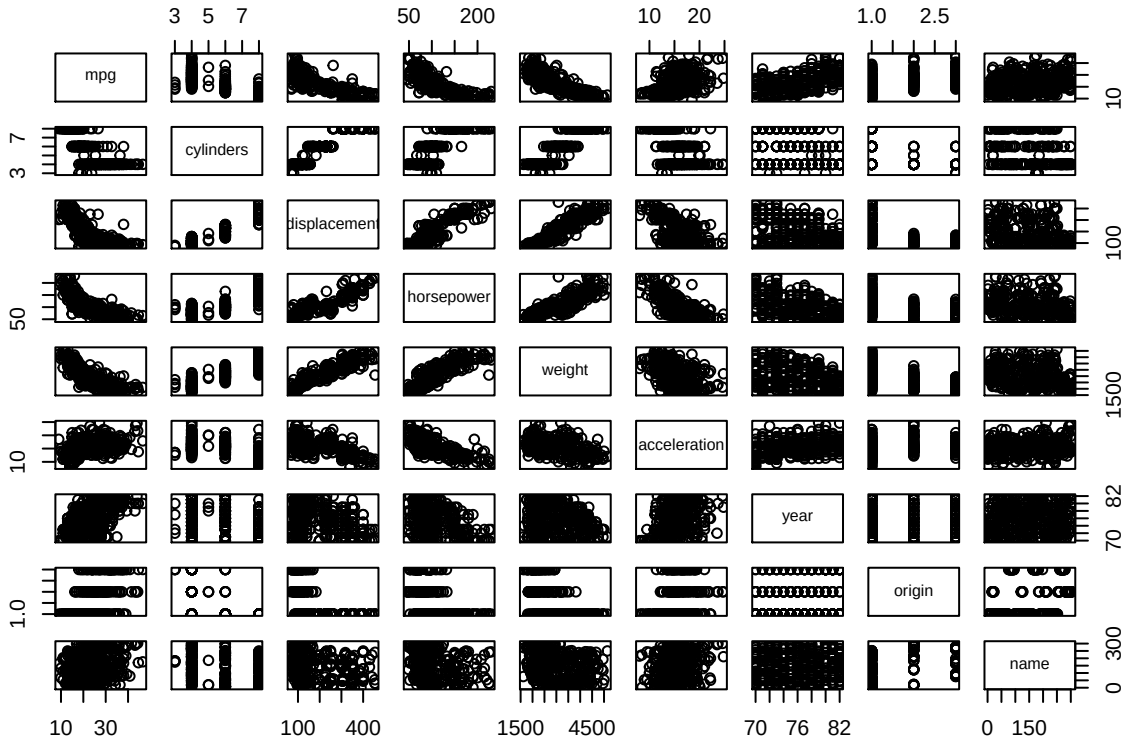
(7c) The cubic regression fit should produce a better RSS on the training set because it can adjust for the non-linearity.

(7d) There is not enough information to tell which test RSS would be lower for either regression given the problem statement is defined as not knowing "how far it is from linear". If it is closer to linear than cubic, the linear regression test RSS could be lower than the cubic regression test RSS. Or, if it is closer to cubic than linear, the cubic regression test RSS could be lower than the linear regression test RSS. It is due to bias-variance tradeoff: it is not clear what level of flexibility will fit data better.

Q8

(8a)

```
pairs(Auto)
```



(8b)

```
cor(Auto[1:8])
```

```
##                     mpg  cylinders displacement horsepower     weight
## mpg           1.0000000 -0.7776175   -0.8051269 -0.7784268 -0.8322442
## cylinders    -0.7776175  1.0000000    0.9508233  0.8429834  0.8975273
## displacement -0.8051269  0.9508233    1.0000000  0.8972570  0.9329944
## horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377
## weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000
## acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392
## year          0.5805410 -0.3456474   -0.3698552 -0.4163615 -0.3091199
## origin        0.5652088 -0.5689316   -0.6145351 -0.4551715 -0.5850054
##              acceleration       year     origin
## mpg             0.4233285  0.5805410  0.5652088
## cylinders      -0.5046834 -0.3456474 -0.5689316
## displacement   -0.5438005 -0.3698552 -0.6145351
## horsepower     -0.6891955 -0.4163615 -0.4551715
## weight         -0.4168392 -0.3091199 -0.5850054
## acceleration    1.0000000  0.2903161  0.2127458
## year            0.2903161  1.0000000  0.1815277
## origin          0.2127458  0.1815277  1.0000000
```

(8c)

```
lm.fit1 = lm(mpg~.-name, data=Auto)
summary(lm.fit1)
```

```
##
## Call:
## lm(formula = mpg ~ . - name, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5903 -2.1565 -0.1169  1.8690 13.0604
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -17.218435   4.644294  -3.707  0.00024 ***
## cylinders     -0.493376   0.323282  -1.526  0.12780
## displacement   0.019896   0.007515   2.647  0.00844 **
## horsepower    -0.016951   0.013787  -1.230  0.21963
## weight        -0.006474   0.000652  -9.929  < 2e-16 ***
## acceleration   0.080576   0.098845   0.815  0.41548
## year           0.750773   0.050973  14.729  < 2e-16 ***
## origin         1.426141   0.278136   5.127 4.67e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.328 on 384 degrees of freedom
## Multiple R-squared:  0.8215, Adjusted R-squared:  0.8182
## F-statistic: 252.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

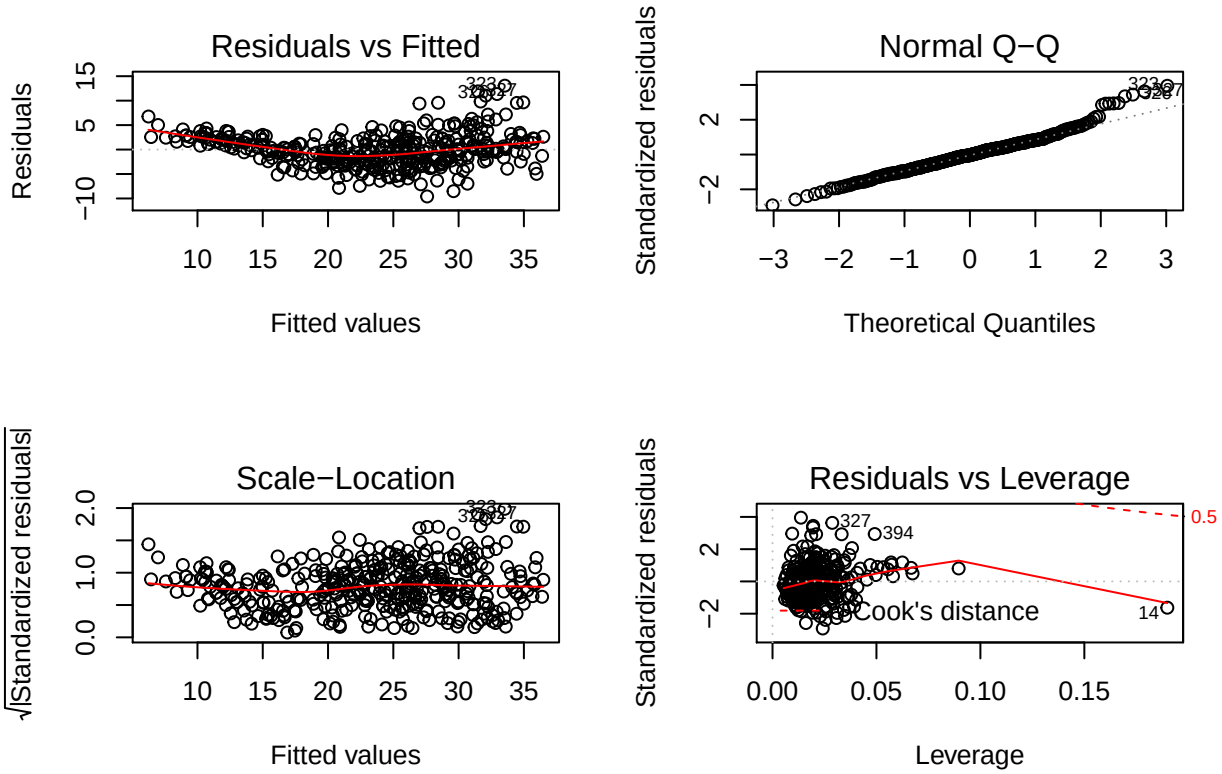
(8c-i) Yes, there is a relatioship between the predictors and the response by testing the null hypothesis of whether all the regression coefficients are zero. The F -statistic is far from 1 (with a small p-value), indicating evidence against the null hypothesis.

(8c-ii) Looking at the p-values associated with each predictor's t-statistic, we see that displacement, weight, year, and origin have a statistically significant relationship, while cylinders, horsepower, and acceleration do not.

(8c-iii) The regression coefficient for year is 0.750773, suggests that for every one year, mpg increases by the coefficient. In other words, cars become more fuel efficient every year by almost 1 mpg per year.

(8d)

```
par(mfrow=c(2,2))
plot(lm.fit1)
```



From the bottom right panel, it appears there are some outliers, but it doesn't appear that there are any high-leverage points, because the all the Cook's distances are smaller than 1.

(option) We can use `rstudent` to see if there are any outliers, and from the result below, there are 4 outliers if we use 3 as the cutoff.

```
any(abs(rstudent(lm.fit1)) > 3)
```

```
## [1] TRUE
```

```
sum(abs(rstudent(lm.fit1)) > 3)
```

```
## [1] 4
```

(8e) From the p-values, we can see that the interaction between displacement and weight is statistically signifcant, while the interaction between cylinders and displacement, and the interaction between cylinders and weight, are both not statistically signifcant.

```
lm.fit2 = lm(mpg~cylinders*displacement+cylinders*weight+
               displacement*weight,data = Auto[,1:8])
summary(lm.fit2)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders * displacement + cylinders * weight +
##     displacement * weight, data = Auto[, 1:8])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.1599  -2.5204  -0.3546   1.7851  17.8829
##
## Coefficients:
```

```
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              4.903e+01  6.743e+00   7.271 2.01e-12 ***
## cylinders                1.851e+00  2.075e+00   0.892  0.37289
## displacement            -9.357e-02  3.919e-02  -2.387  0.01746 *
## weight                  -8.351e-03  3.026e-03  -2.759  0.00607 **
## cylinders:displacement -2.026e-03  3.826e-03  -0.529  0.59682
## cylinders:weight        -3.801e-04  6.720e-04  -0.566  0.57197
## displacement:weight      2.499e-05  8.250e-06   3.029  0.00262 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.106 on 385 degrees of freedom
## Multiple R-squared:  0.7275, Adjusted R-squared:  0.7232
## F-statistic: 171.3 on 6 and 385 DF,  p-value: < 2.2e-16
```

(8f)

```
lm.fit_21 = lm(mpg~horsepower+log(horsepower), data = Auto)
summary(lm.fit_21)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower + log(horsepower), data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.5118  -2.5018  -0.2533   2.4446  15.3102
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     156.04057   12.08267  12.914  < 2e-16 ***
## horsepower        0.11846    0.02929   4.044 6.34e-05 ***
## log(horsepower) -31.59815    3.28363  -9.623  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.415 on 389 degrees of freedom
## Multiple R-squared:  0.6817, Adjusted R-squared:  0.6801
## F-statistic: 416.6 on 2 and 389 DF,  p-value: < 2.2e-16
```

```
lm.fit_22 = lm(mpg~horsepower+ sqrt(horsepower), data = Auto)
summary(lm.fit_22)
```

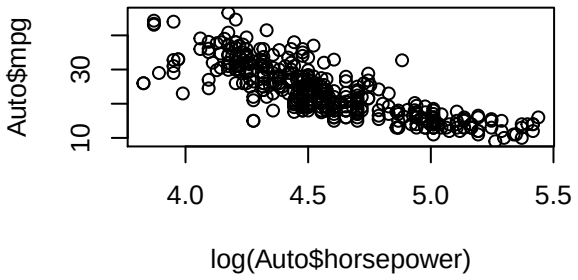
```
##
## Call:
## lm(formula = mpg ~ horsepower + sqrt(horsepower), data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.5479  -2.5677  -0.2663   2.2998  15.5098
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      105.31581    6.64657  15.845  < 2e-16 ***
## horsepower         0.41913    0.05867   7.144 4.49e-12 ***
## sqrt(horsepower) -12.48574    1.26337  -9.883  < 2e-16 ***
```

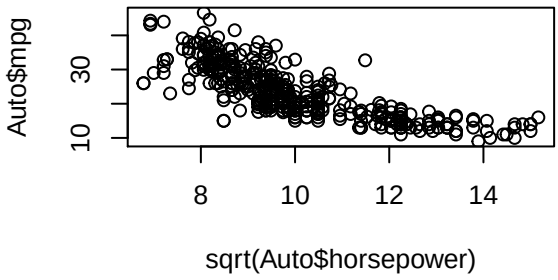
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.392 on 389 degrees of freedom
## Multiple R-squared:  0.685,  Adjusted R-squared:  0.6834
## F-statistic:   423 on 2 and 389 DF,  p-value: < 2.2e-16
```

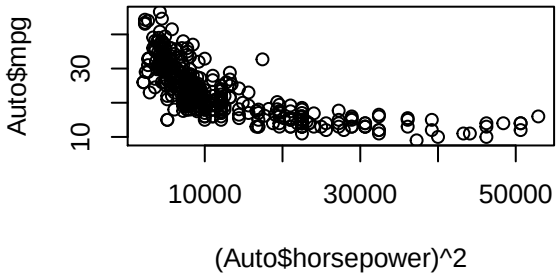
```
lm.fit_23 = lm(mpg~horsepower+I(horsepower^2), data = Auto)
summary(lm.fit_23)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower + I(horsepower^2), data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.7135  -2.5943  -0.0859   2.2868  15.8961
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     56.9000997  1.8004268   31.60   <2e-16 ***
## horsepower      -0.4661896  0.0311246  -14.98   <2e-16 ***
## I(horsepower^2)  0.0012305  0.0001221   10.08   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.374 on 389 degrees of freedom
## Multiple R-squared:  0.6876, Adjusted R-squared:  0.686
## F-statistic:   428 on 2 and 389 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(log(Auto$horsepower), Auto$mpg)
plot(sqrt(Auto$horsepower), Auto$mpg)
plot((Auto$horsepower)^2, Auto$mpg)
```







Apparently from the p-values, the log, sqrt, and square all have statistical significance of some sort. The log transformation shows the maximum linear characteristic.

Q9

(9a)

```
data(Carseats)
lm.fit3 <- lm(Sales ~ Price + Urban + US, data=Carseats)
summary(lm.fit3)
```

```
##
## Call:
## lm(formula = Sales ~ Price + Urban + US, data = Carseats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9206 -1.6220 -0.0564  1.5786  7.0581
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.043469   0.651012  20.036  < 2e-16 ***
## Price       -0.054459   0.005242 -10.389  < 2e-16 ***
## UrbanYes    -0.021916   0.271650  -0.081    0.936
## USYes        1.200573   0.259042   4.635 4.86e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.472 on 396 degrees of freedom
## Multiple R-squared:  0.2393, Adjusted R-squared:  0.2335
## F-statistic: 41.52 on 3 and 396 DF,  p-value: < 2.2e-16
```

(9b) Sales: sales in thousands at each location;

Price: price charged for car seats at each location;

Urban: No/Yes by location;

US: No/Yes by location.

Coefficients: Price (-0.054459): Sales drop by 54 for each dollar increase in Price - statistically significant;

UrbanYes (-0.021916): Sales are 22 lower for urban locations than rural locations - not statistically significant;

USYes (1.200573): Sales are 1,201 higher in US locations than non-US locations - statistically significant.

(9c)

$$Sales = 13.043469 - 0.054459X_{Price} - 0.021916X_{Urban} + 1.200573X_{US} + \epsilon,$$

where X_{Urban} and X_{US} are dummy variables. $X_{Urban} = 1$ if it's Yes, otherwise $X_{Urban} = 0$. Similarly, $X_{US} = 1$ if it's Yes, otherwise $X_{US} = 0$.

(9d) We can reject null hypothesis for "Price" and "US", since the p-value is very small.

(9e)

```
lm.fit4 = lm(Sales ~ Price + US, data=Carseats)
summary(lm.fit4)
```

```
##
## Call:
## lm(formula = Sales ~ Price + US, data = Carseats)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -6.9269 -1.6286 -0.0574  1.5766  7.0515
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.03079    0.63098  20.652  < 2e-16 ***
## Price       -0.05448    0.00523 -10.416  < 2e-16 ***
## USYes        1.19964    0.25846   4.641 4.71e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.469 on 397 degrees of freedom
## Multiple R-squared:  0.2393, Adjusted R-squared:  0.2354
## F-statistic: 62.43 on 2 and 397 DF,  p-value: < 2.2e-16
```

(9f) Based on the RSE and R^2 of the linear regressions, they both fit the data similarly. If we look at the R^2 more closely (see below), then we can see that the R^2 of (a) is slightly lower, which indicates that, in terms of training MSE, the linear regression of (a) fits the data slightly better.

```
summary(lm.fit3)$r.squared
```

```
## [1] 0.2392754
```

```
summary(lm.fit4)$r.squared
```

```
## [1] 0.2392629
```

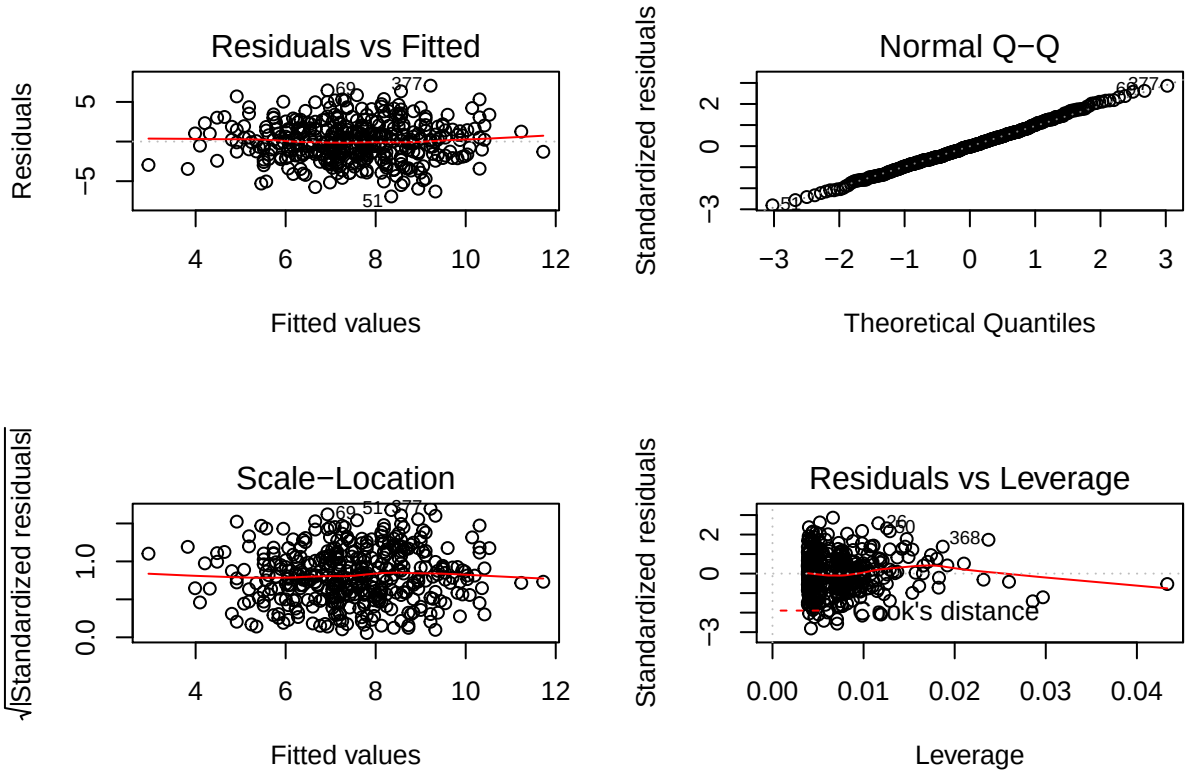
(9g)

```
confint(lm.fit4)
```

```
##                   2.5 %      97.5 %
## (Intercept) 11.79032020 14.27126531
## Price       -0.06475984 -0.04419543
## USYes        0.69151957  1.70776632
```

(9h)

```
par(mfrow=c(2,2))
plot(lm.fit4)
```



All studentized residuals appear to be bounded by -3 to 3 and their cook's distances are smaller than 1, so no potential outliers or high-leverage points are suggested from the linear regression. Again, we can use `rstudent` to see if there are any outliers, and from the result below, there are no outliers.

```
any(abs(rstudent(lm.fit4)) > 3)
```

```
## [1] FALSE
```

```
sum(abs(rstudent(lm.fit4)) > 3)
```

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

Q10

(10a)

```
set.seed(1)
x1 = runif(100)
x2 = 0.5 * x1 + rnorm(100)/10
y1 = 2 + 2*x1 + 0.3*x2 + rnorm(100)
```

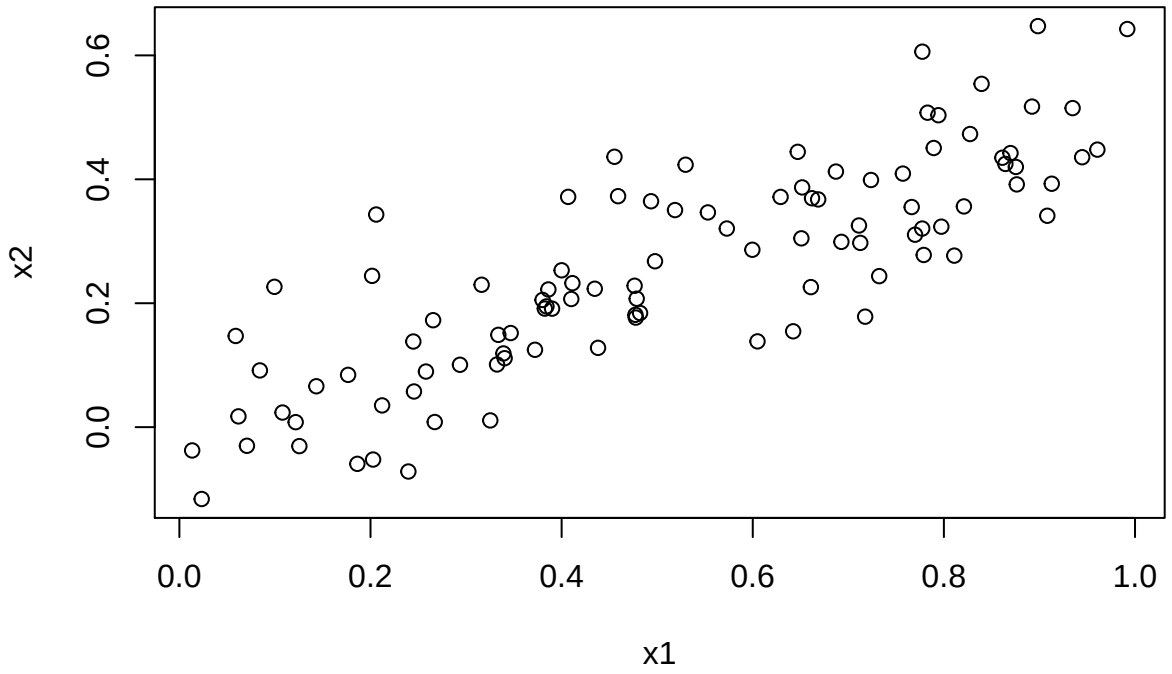
The data generated according to the model which is: $Y = 2 + 2x_1 + 0.3x_2 + \epsilon$

(10b)

```
cor(x1, x2)
```

```
## [1] 0.8351212
```

```
plot(x1, x2)
```



(10c)

```
lm.fit9 = lm(y1~x1+x2)
summary(lm.fit9)
```

```
##
## Call:
## lm(formula = y1 ~ x1 + x2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8311 -0.7273 -0.0537  0.6338  2.3359
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1305     0.2319   9.188 7.61e-15 ***
## x1            1.4396     0.7212   1.996   0.0487 *
## x2            1.0097     1.1337   0.891   0.3754
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.056 on 97 degrees of freedom
## Multiple R-squared:  0.2088, Adjusted R-squared:  0.1925
## F-statistic:  12.8 on 2 and 97 DF,  p-value: 1.164e-05
```

$\beta_0 = 2.1305, \beta_1 = 1.4396, \beta_3 = 1.0097$. The p-value for beta1 suggests that the null hypothesis for this coefficient should be rejected, while that of beta2 suggests that null is accepted.

(10d)

```
lm.fit10 = lm(y1~x1)
summary(lm.fit10)
```

```
##
## Call:
## lm(formula = y1 ~ x1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.89495 -0.66874 -0.07785  0.59221  2.45560
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1124     0.2307   9.155 8.27e-15 ***
## x1            1.9759     0.3963   4.986 2.66e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.055 on 98 degrees of freedom
## Multiple R-squared:  0.2024, Adjusted R-squared:  0.1942
## F-statistic: 24.86 on 1 and 98 DF,  p-value: 2.661e-06
```

Yes, we can reject the null hypothesis for the regression coefficient given the p-value for its t-statistic is too small.

(10e)

```
lm.fit11 = lm(y1~x2)
summary(lm.fit11)
```

```
##
## Call:
## lm(formula = y1 ~ x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.62687 -0.75156 -0.03598  0.72383  2.44890
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.3899     0.1949   12.26  < 2e-16 ***
## x2            2.8996     0.6330    4.58 1.37e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 1.072 on 98 degrees of freedom
## Multiple R-squared:  0.1763, Adjusted R-squared:  0.1679
## F-statistic: 20.98 on 1 and 98 DF,  p-value: 1.366e-05
```

Yes, we can reject the null hypothesis for the regression coefficient given the p-value for its t-statistic is too small.

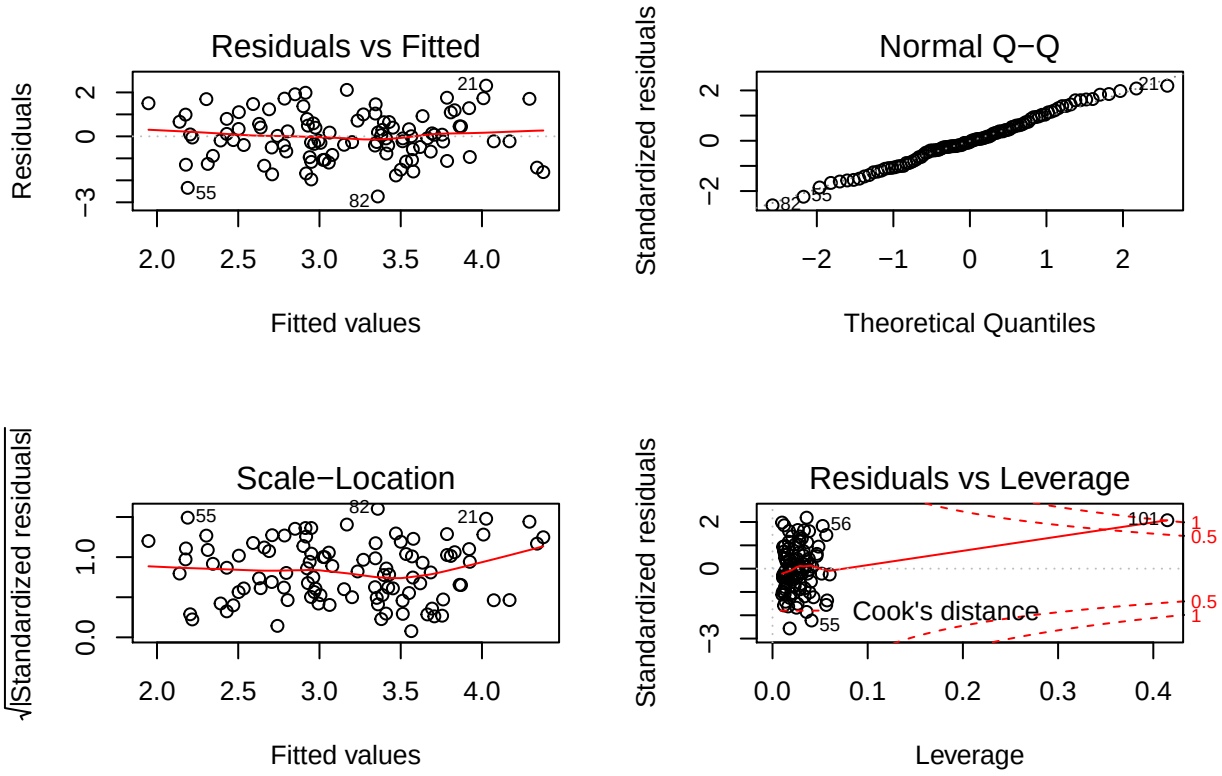
(10f) No, because x1 and x2 have collinearity, it is hard to distinguish their effects when regressed upon together.

(10g)

```
x1 <- c(x1, 0.1)
x2 <- c(x2, 0.8)
y2 <- c(y1, 6)
par(mfrow=c(2,2))
# regression with both x1 and x2
lm.fit12 <- lm(y2~x1+x2)
summary(lm.fit12)
```

```
##
## Call:
## lm(formula = y2 ~ x1 + x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.73348 -0.69318 -0.05263  0.66385  2.30619
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.2267     0.2314   9.624 7.91e-16 ***
## x1            0.5394     0.5922   0.911  0.36458
## x2            2.5146     0.8977   2.801  0.00614 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.075 on 98 degrees of freedom
## Multiple R-squared:  0.2188, Adjusted R-squared:  0.2029
## F-statistic: 13.72 on 2 and 98 DF,  p-value: 5.564e-06
```

```
plot(lm.fit12)
```

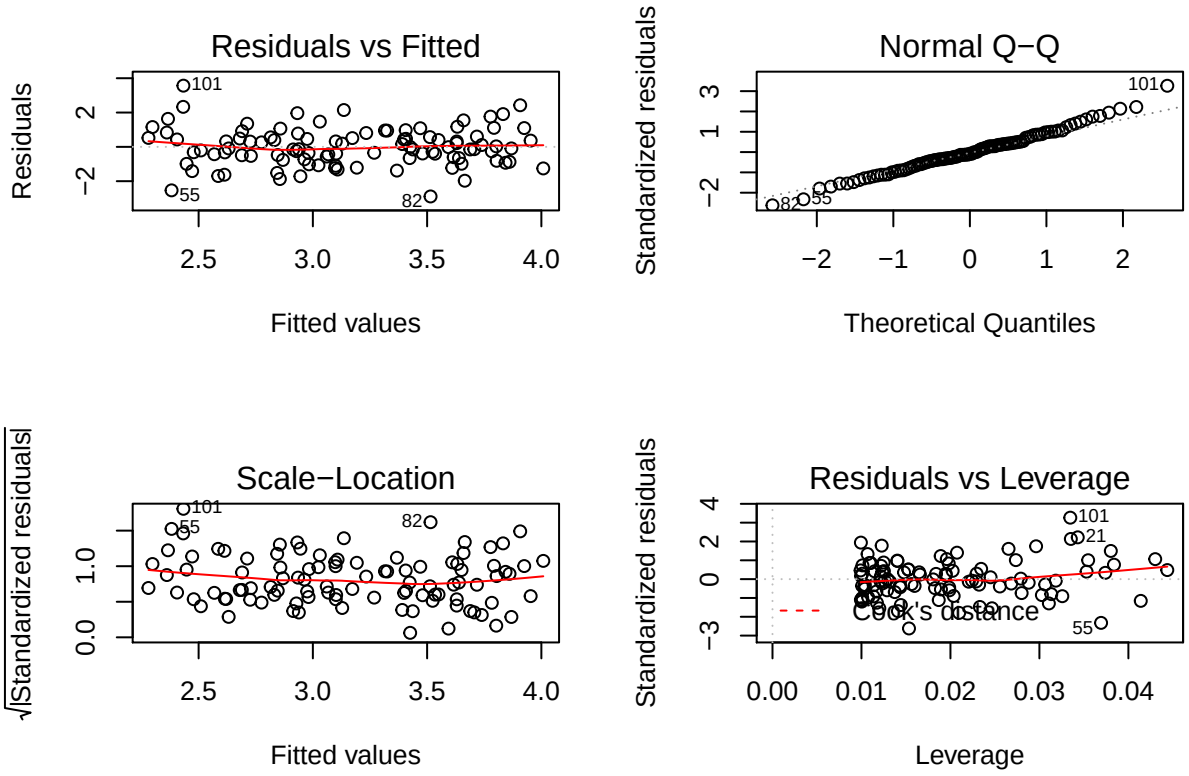


In the first model, it shifts x1 to statistically insignificance and shifts x2 to statistiscal significance from the change in p-values between the two linear regressions. Moreover, we can see the new data becomes a high-leverage point because its Cook's distance is great than 1.

```
# regression with x1 only
lm.fit13 <- lm(y2~x1)
summary(lm.fit13)
```

```
##
## Call:
## lm(formula = y2 ~ x1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8897 -0.6556 -0.0909  0.5682  3.5665
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.2569     0.2390   9.445 1.78e-15 ***
## x1            1.7657     0.4124   4.282 4.29e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.111 on 99 degrees of freedom
## Multiple R-squared:  0.1562, Adjusted R-squared:  0.1477
## F-statistic: 18.33 on 1 and 99 DF,  p-value: 4.295e-05
```

```
par(mfrow=c(2,2))
plot(lm.fit13)
```

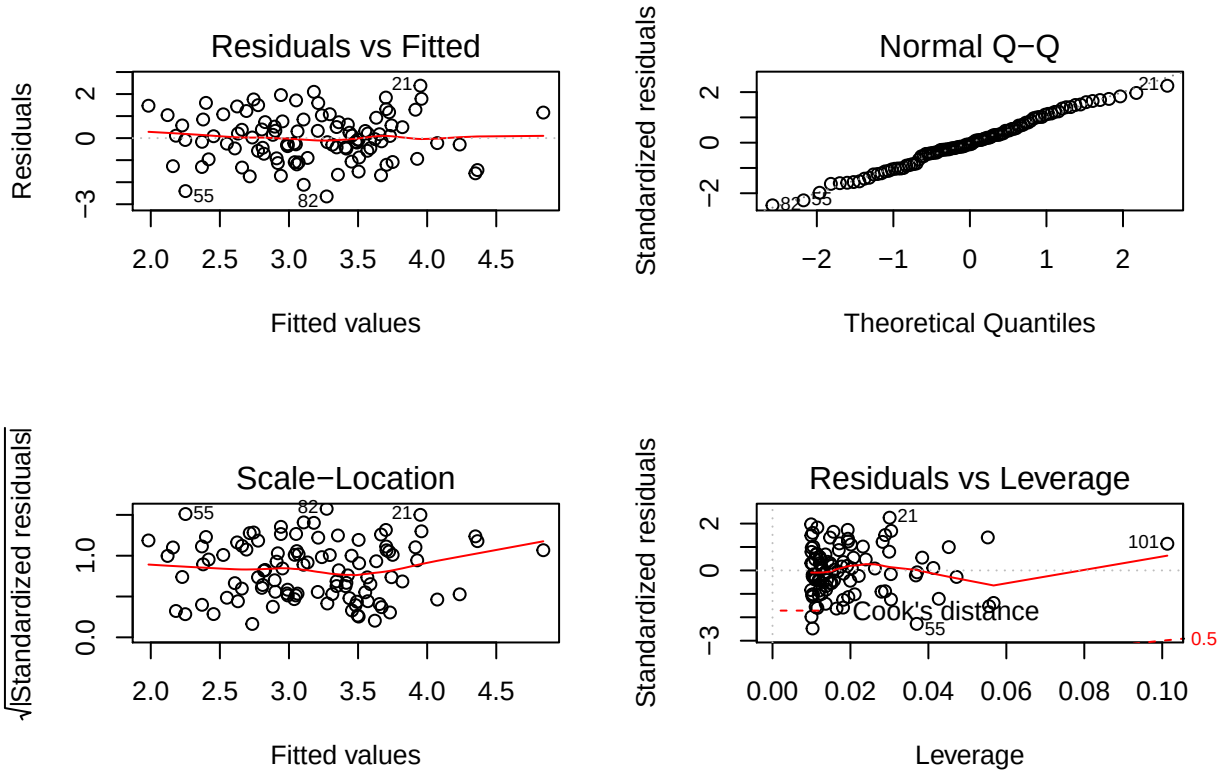


From the bottom-right residual plot, we see that the new point becomes an outlier point.

```
# regression with x2 only
lm.fit14 <- lm(y2~x2)
summary(lm.fit14)
```

```
##
## Call:
## lm(formula = y2 ~ x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.64729 -0.71021 -0.06899  0.72699  2.38074
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.3451     0.1912  12.264  < 2e-16 ***
## x2            3.1190     0.6040   5.164 1.25e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.074 on 99 degrees of freedom
## Multiple R-squared:  0.2122, Adjusted R-squared:  0.2042
## F-statistic: 26.66 on 1 and 99 DF,  p-value: 1.253e-06
```

```
par(mfrow=c(2,2))
plot(lm.fit14)
```



In this new model, we don't observe any outliers and high-leverage points, but we see the new point has a large value of leverage (although the Cook's distance is smaller than 1). The rationale is that most of original `x2` are in the region [0,0.5] (you can see from `hist(x2[1:100])`) but we add the new data 0.8, which is far from this region.

Q11

(11a) The predicted prices are 20893.9 and 26952.1.

```
library(FNN)
```

```
##
## Attaching package: 'FNN'
```

```
## The following objects are masked from 'package:class':
##
##     knn, knn.cv
```

```
train.df <- read.csv("toyota.csv")
test.df <- data.frame("model"=c("RAV4", "Camry"), "year"=c(2019, 2015),
                      "transmission"=c("Automatic","Automatic"), "mileage"=c(12345,50000),
                      "fuelType"=c("Diesel", "Hybrid"), "tax"=c(150,130),
                      "mpg"=c(25,50), "engineSize"=c(2,3))

# combine and then transform together. Will separate them later
toyata.all <- rbind(train.df[,-3], test.df) # -3 removing the output (Sales) column
X.all <- model.matrix( ~ ., data=toyata.all)[,-1]
X.all <- scale(X.all)
# Separate them to train and test (original size)
X.train <- X.all[1:nrow(train.df),]
X.test <- X.all[-(1:nrow(train.df)),]
pred.out <- knn.reg(train = X.train, test = X.test, y = train.df$price, k = 10)
print(pred.out$pred)
```

```
## [1] 20893.9 26952.1
```

(11b) The predicted prices are 25197.33 and 15175.07.

```
pred.out <- knn.reg(train = X.train, test = X.test, y = train.df$price, k = 100)
print(pred.out$pred)
```

```
## [1] 25197.33 15175.07
```

(11c) $K = 100$ because $K = 100$ results in a less flexible model which gives higher bias but lower variance.

(11d) The predicted prices are 19964.64 and 22504.62.

```
lm.fit <- lm(price ~ ., data=train.df)
predict(lm.fit, test.df)
```

```
##        1        2
## 19964.64 22504.62
```

(11e) For example, increasing one-unit of tax will decrease the price in 3.78 dollars. `Model`, `year`, `transmission`, `mileage`, `fuelType`, `tax`, `mpg`, `engineSize` are significant.

```
summary(lm.fit)
```

```
##
## Call:
## lm(formula = price ~ ., data = train.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15669.1   -826.3   -169.0    602.7  17990.7
##
```

```
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -1.566e+06  2.933e+04 -53.411  < 2e-16 ***
## modelAvensis           1.324e+03  1.960e+02   6.755 1.55e-11 ***
## modelAygo             -2.694e+03  1.228e+02 -21.942  < 2e-16 ***
## modelC-HR              5.559e+03  1.052e+02  52.850  < 2e-16 ***
## modelCamry             6.847e+03  5.349e+02  12.800  < 2e-16 ***
## modelCorolla           5.001e+03  1.288e+02  38.831  < 2e-16 ***
## modelGT86              6.185e+03  2.342e+02  26.411  < 2e-16 ***
## modelHilux             8.352e+03  2.934e+02  28.465  < 2e-16 ***
## modelIQ               -3.093e+02  6.144e+02  -0.503    0.615
## modelLandCruiser       2.275e+04  3.780e+02  60.201  < 2e-16 ***
## modelPrius             4.987e+03  1.387e+02  35.950  < 2e-16 ***
## modelPROACEVERSO       1.358e+04  4.768e+02  28.486  < 2e-16 ***
## modelRAV4              4.908e+03  1.577e+02  31.117  < 2e-16 ***
## modelSupra             3.037e+04  5.663e+02  53.620  < 2e-16 ***
## modelUrbanCruiser     -6.856e+00  8.585e+02  -0.008    0.994
## modelVerso             1.204e+03  1.862e+02   6.466 1.08e-10 ***
## modelVerso-S           2.536e+02  9.892e+02   0.256    0.798
## modelYaris            -1.557e+03  8.358e+01 -18.631  < 2e-16 ***
## year                   7.811e+02  1.455e+01  53.696  < 2e-16 ***
## transmissionManual    -1.221e+03  8.463e+01 -14.428  < 2e-16 ***
## transmissionOther      8.927e+02  1.707e+03   0.523    0.601
## transmissionSemi-Auto  7.322e+01  1.382e+02   0.530    0.596
## mileage               -6.233e-02  1.691e-03 -36.851  < 2e-16 ***
## fuelTypeHybrid         3.240e+03  1.590e+02  20.378  < 2e-16 ***
## fuelTypeOther          2.941e+03  2.253e+02  13.052  < 2e-16 ***
## fuelTypePetrol         1.587e+03  1.295e+02  12.261  < 2e-16 ***
## tax                   -3.780e+00  3.637e-01 -10.391  < 2e-16 ***
## mpg                   -9.591e+00  2.081e+00  -4.609 4.12e-06 ***
## engineSize             2.997e+03  1.698e+02  17.648  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1706 on 6709 degrees of freedom
## Multiple R-squared:  0.928,  Adjusted R-squared:  0.9277
## F-statistic:  3089 on 28 and 6709 DF,  p-value: < 2.2e-16
```

(11f) No, because we are not able to compute the test MSE error (we don't know the true prices of the these two cars).