Problem1

a) Prove $\frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$

$$= 2 \sum_{i \in C_k} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{kj})^2$$

given $\bar{x}_{kj} = \frac{1}{|C_k|} \sum_{i \in C_k} x_{ij}$

Expanding Squares on the Left hand Side

$$\frac{1}{C_k} \sum_{i \in C_k} \sum_{i' \in C_k} \sum_{j=1}^{p} \left(x_{ij}^2 - 2x_{ij} x_{i'j} + x_{i'j}^2 \right)$$

$$\sum_{i \in C_k} \sum_{j=1}^{p} \left(\frac{1}{|C_k|} \sum_{i' \in C_k} \left(x_{ij}^2 - 2x_{ij} x_{i'j} + x_{i'j}^2 \right) \right)$$

$$\sum_{i \in C_k} \sum_{j=1}^{p} \left[2 \times x_{ij}^2 - 4 x_{ij} \bar{x}_{kj} + 2 \bar{x}_{kj}^2 \right]$$

$\because$ as the Summation is over orderd pairs.

$$= 2 \sum_{i \in C_k} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{kj})^2$$

hence proved

Problem (2)

Given dissimilarity Matrix

$$\begin{array}{c|c} \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} - & 0.3 & 0.4 & 0.7 \\ 0.3 & - & 0.5 & 0.8 \\ 0.4 & 0.5 & - & 0.45 \\ 0.7 & 0.8 & 0.45 & - \end{bmatrix} \\ & \quad 1 \qquad\; 2 \qquad\; 3 \qquad\; 4 \end{array}$$

a) Hierarchial Clustering (Using Complete Linkage)

Step (1): minimum dissimilarity is 0.3 between (1) & (2).

⇒ so combine them to form a cluster at hieght 0.3.

new dissimilarity matrix

$$\Rightarrow \begin{array}{c|c} \begin{array}{c} (1,2) \\ 3 \\ 4 \end{array} & \begin{bmatrix} - & 0.5 & 0.8 \\ 0.5 & - & 0.45 \\ 0.8 & 0.45 & - \end{bmatrix} \\ & (1,2) \quad 3 \quad 4 \end{array}$$

Step (2): minimum dissimilarity in 0.45 between (3) & (4)

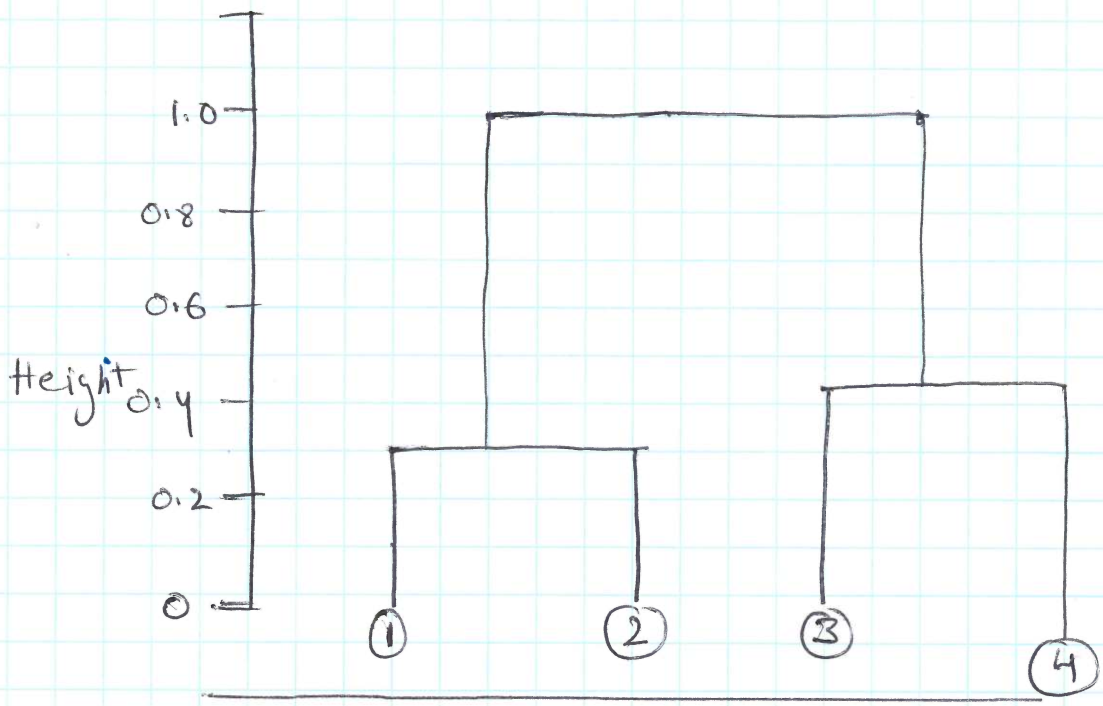
combine them to form a cluster at height = 0.45.

Reduced dissimilarity matrix

$$\Rightarrow \begin{array}{c|c} \begin{array}{c} (1,2) \\ (3,4) \end{array} & \begin{bmatrix} - & 0.8 \\ 0.8 & - \end{bmatrix} \\ & (1,2) \quad (3,4) \end{array}$$

final step: Combine (1,2) & (3,4) at hieght at 0.8.

Cluster dendrogram [Complete Linkage]



b) Hierarchial Clustering [Single Linkage]

Step 1: minum dissimilarity is 0.3 between 1 & 2 combine them to form a Cluster at hight = 0.3

$\Rightarrow$ New dissimilarity matrix

$$
\begin{array}{c}
(1,2) \\ 3 \\ 4
\end{array}
\begin{bmatrix}
- & 0.4 & 0.7 \\
0.4 & - & 0.45 \\
0.7 & 0.45 & -
\end{bmatrix}
$$

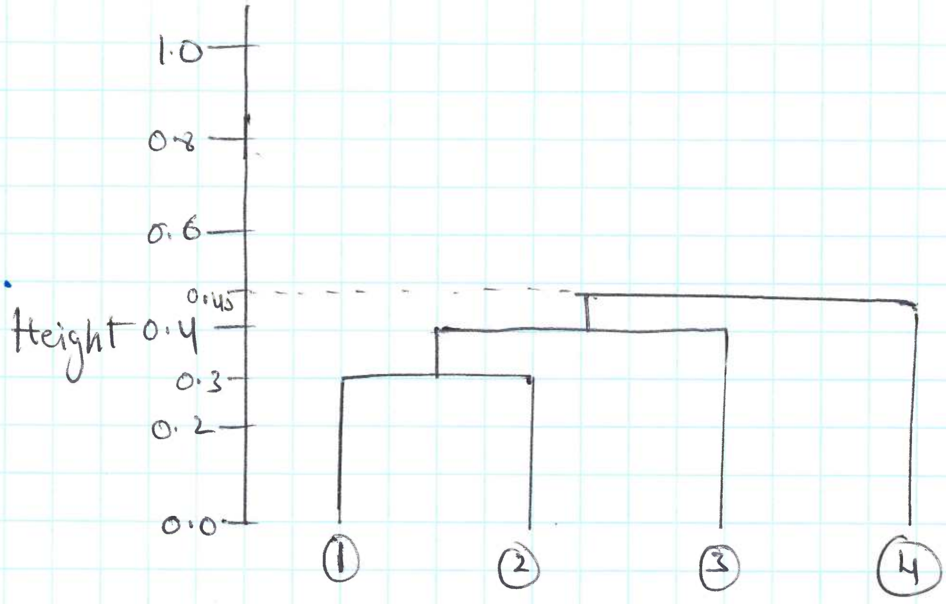
$\quad\quad (1,2) \quad 3, \quad 4$

$\Rightarrow$ Similarly clusters will created with

[(1, 2), 3] at height = 0.4 and

[{(1, 2), 3}, 4] at height = 0.45 .

Cluster dendrogram [Single Linkage]



c) Cut Dendogram from (a) to form two Clusters.

⇒ we have clusters two Clusters with

(1, 2) & (3, 4)

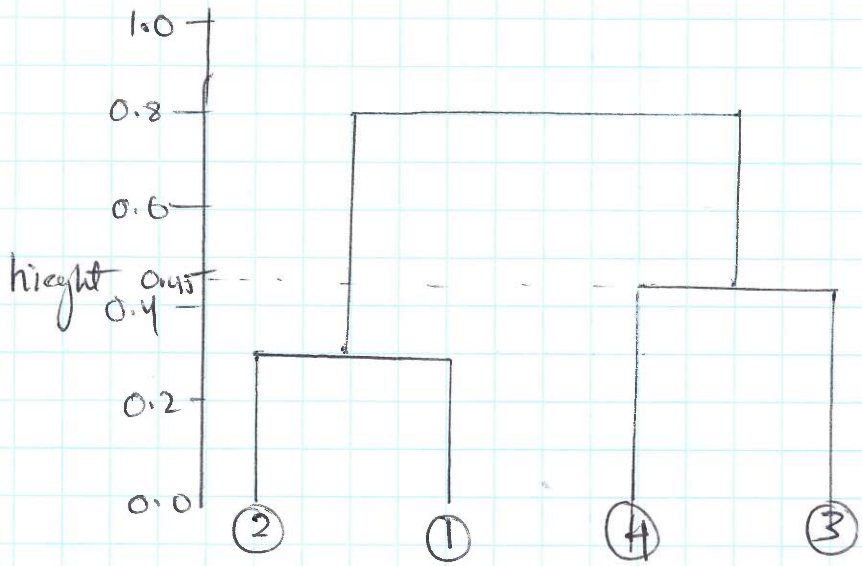
d) Cut Dendogram from (b) to form two Clusters.

⇒ we have two Clusters with

[(1, 2), 3] and 4

e) Swap observation such that no meaning is changed in dendogram with Complete linkage.

e) Cluster diagram with Swapped Observations [in Complete Linkage Case]

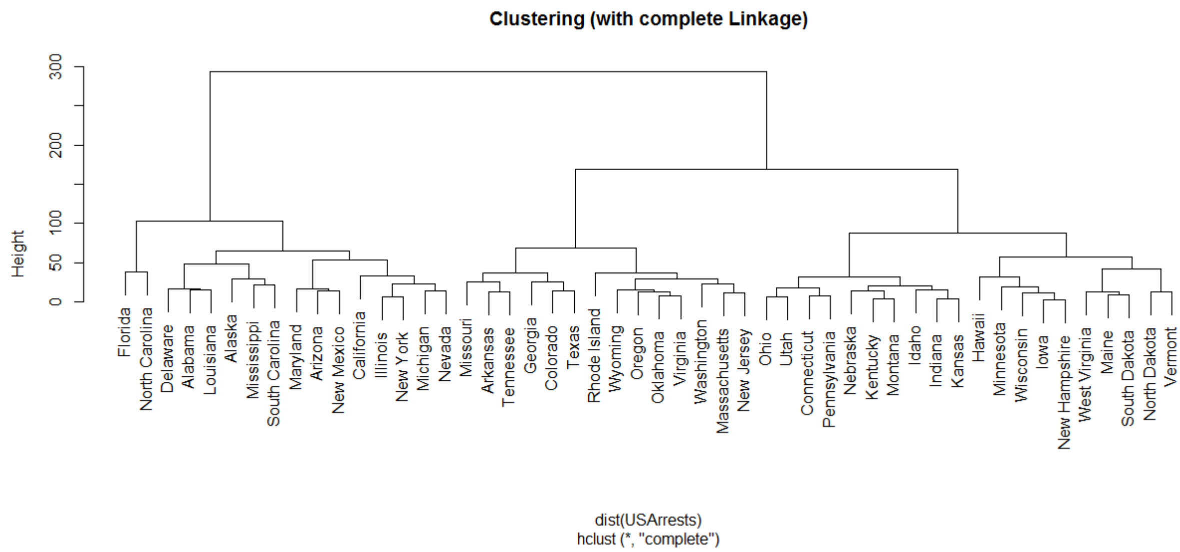


STATS 202 | HW:2 | Sagar Ganapaneni | SUID# 06167633

Problem: 3

a) Using hierarchical clustering with complete linkage and Euclidean distance, cluster the states.

```
set.seed(12)
arrests_complete <- hclust(dist(USArrests), method = "complete")
plot(arrests_complete,main='Clustering (with complete Linkage)')
```



b) Cut the dendrogram at a height that results in three distinct clusters. Which states belong to which clusters?

```
cluster_mapping<- cutree(arrests_complete, 3)
cluster1 <- USArrests[cluster_mapping == 1,]
cluster2 <- USArrests[cluster_mapping == 2,]
cluster3 <- USArrests[cluster_mapping == 3,]
print(cluster1)
```

```
##                Murder Assault UrbanPop Rape
## Alabama          13.2     236       58 21.2
## Alaska           10.0     263       48 44.5
## Arizona           8.1     294       80 31.0
## California        9.0     276       91 40.6
## Delaware          5.9     238       72 15.8
## Florida          15.4     335       80 31.9
## Illinois         10.4     249       83 24.0
```

```
## Louisiana        15.4     249       66 22.2
## Maryland         11.3     300       67 27.8
## Michigan         12.1     255       74 35.1
## Mississippi      16.1     259       44 17.1
## Nevada           12.2     252       81 46.0
## New Mexico       11.4     285       70 32.1
## New York         11.1     254       86 26.1
## North Carolina   13.0     337       45 16.1
## South Carolina   14.4     279       48 22.5
```

```
print(cluster2)
```

```
##                Murder Assault UrbanPop Rape
## Arkansas          8.8     190       50 19.5
## Colorado          7.9     204       78 38.7
## Georgia          17.4     211       60 25.8
## Massachusetts     4.4     149       85 16.3
## Missouri          9.0     178       70 28.2
## New Jersey        7.4     159       89 18.8
## Oklahoma          6.6     151       68 20.0
## Oregon            4.9     159       67 29.3
## Rhode Island      3.4     174       87  8.3
## Tennessee        13.2     188       59 26.9
## Texas            12.7     201       80 25.5
## Virginia          8.5     156       63 20.7
## Washington        4.0     145       73 26.2
## Wyoming           6.8     161       60 15.6
```

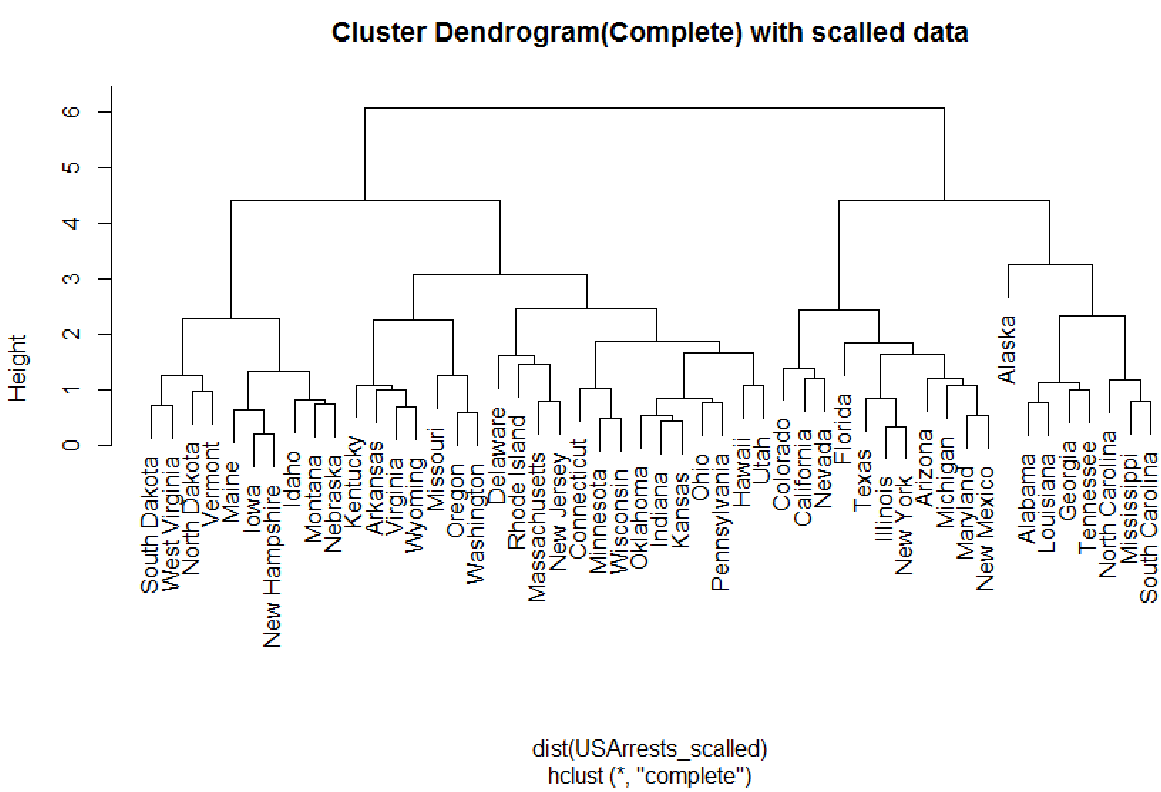
```
print(cluster3)
```

```
##                Murder Assault UrbanPop Rape
## Connecticut       3.3     110       77 11.1
## Hawaii            5.3      46       83 20.2
## Idaho             2.6     120       54 14.2
## Indiana           7.2     113       65 21.0
## Iowa              2.2      56       57 11.3
## Kansas            6.0     115       66 18.0
## Kentucky          9.7     109       52 16.3
## Maine             2.1      83       51  7.8
## Minnesota         2.7      72       66 14.9
## Montana           6.0     109       53 16.4
```

```
## Nebraska          4.3     102        62 16.5
## New Hampshire     2.1      57        56  9.5
## North Dakota      0.8      45        44  7.3
## Ohio              7.3     120        75 21.4
## Pennsylvania      6.3     106        72 14.9
## South Dakota      3.8      86        45 12.8
## Utah              3.2     120        80 22.9
## Vermont           2.2      48        32 11.2
## West Virginia     5.7      81        39  9.3
## Wisconsin         2.6      53        66 10.8
```

c) Hierarchically cluster the states using complete linkage and Euclidean distance, after scaling the variables to have standard deviation one

```r
USArrests_scalled <- scale(USArrests)
arrests_scalled_complete <- hclust(dist(USArrests_scalled), method = "complete")
plot(arrests_scalled_complete,main='Cluster Dendrogram(Complete) with scalled data')
```



```r
cluster_mapping<- cutree(arrests_scalled_complete, 3)
cluster1 <- USArrests[cluster_mapping == 1,]
cluster2 <- USArrests[cluster_mapping == 2,]
cluster3 <- USArrests[cluster_mapping == 3,]
print(cluster1)
```

```
##                  Murder Assault UrbanPop Rape
## Alabama            13.2     236       58 21.2
## Alaska             10.0     263       48 44.5
## Georgia            17.4     211       60 25.8
## Louisiana          15.4     249       66 22.2
## Mississippi        16.1     259       44 17.1
## North Carolina     13.0     337       45 16.1
## South Carolina     14.4     279       48 22.5
## Tennessee          13.2     188       59 26.9
```

```r
print(cluster2)
```

```
##              Murder Assault UrbanPop Rape
## Arizona         8.1     294       80 31.0
## California      9.0     276       91 40.6
## Colorado        7.9     204       78 38.7
## Florida        15.4     335       80 31.9
## Illinois       10.4     249       83 24.0
## Maryland       11.3     300       67 27.8
## Michigan       12.1     255       74 35.1
## Nevada         12.2     252       81 46.0
## New Mexico     11.4     285       70 32.1
## New York       11.1     254       86 26.1
## Texas          12.7     201       80 25.5
```

```r
print(cluster3)
```

```
##                 Murder Assault UrbanPop Rape
## Arkansas           8.8     190       50 19.5
## Connecticut        3.3     110       77 11.1
## Delaware           5.9     238       72 15.8
## Hawaii             5.3      46       83 20.2
## Idaho              2.6     120       54 14.2
## Indiana            7.2     113       65 21.0
## Iowa               2.2      56       57 11.3
```

```
## Kansas            6.0     115       66 18.0
## Kentucky          9.7     109       52 16.3
## Maine             2.1      83       51  7.8
## Massachusetts     4.4     149       85 16.3
## Minnesota         2.7      72       66 14.9
## Missouri          9.0     178       70 28.2
## Montana           6.0     109       53 16.4
## Nebraska          4.3     102       62 16.5
## New Hampshire     2.1      57       56  9.5
## New Jersey        7.4     159       89 18.8
## North Dakota      0.8      45       44  7.3
## Ohio              7.3     120       75 21.4
## Oklahoma          6.6     151       68 20.0
## Oregon            4.9     159       67 29.3
## Pennsylvania      6.3     106       72 14.9
## Rhode Island      3.4     174       87  8.3
## South Dakota      3.8      86       45 12.8
## Utah              3.2     120       80 22.9
## Vermont           2.2      48       32 11.2
## Virginia          8.5     156       63 20.7
## Washington        4.0     145       73 26.2
## West Virginia     5.7      81       39  9.3
## Wisconsin         2.6      53       66 10.8
## Wyoming           6.8     161       60 15.6
```

d) What effect does scaling the variables have on the hierarchical clustering obtained? In your opinion, should the variables be scaled before the inter-observation dissimilarities are computed?

```
e) table(cutree(arrests_complete, 3), cutree(arrests_scalled_complete, 3))
f) ##
g) ##      1  2  3
h) ##   1  6  9  1
i) ##   2  2  2 10
j) ##   3  0  0 20
```

Scaling does effected clustering, before scaling variables: Assault and Urban population draw more weightage in grouping states together. After scaling all the variable were considered on relative scale.

For example, States like Arizona and California are grouped with Alabama mainly due to similar Assaults even though urban population is significantly lower than the other two states.

Scaling should be done before measuring the dissimilarities are computed as scaling after measuring dissimilarities might minimize the true distinctions between two data points thus leading to in accurate clustering.

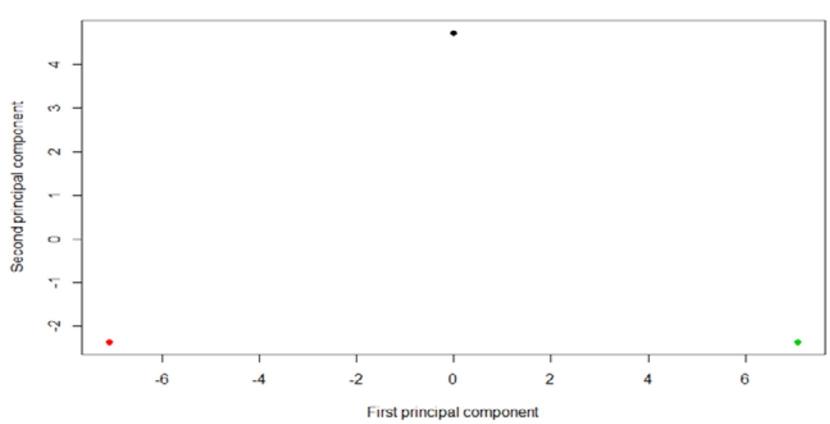
Problem 4

a) Generate a simulated data set with 20 observations in each of three classes (i.e. 60 observations total), and 50 variables.

```
set.seed(12)
groups <-  c(rep(1, 20), rep(2, 20), rep(3, 20))
data <- matrix(rnorm(60*50, mean = 0, sd = 0.001), ncol = 50)
## adding mean shifters
data[1:20,group=1]<-data[1:20,group=1]+10
data[21:40,group=2]<- data[21:40,group=2]-10
data[21:40,group=2]<- data[21:40,group=2]+10
data[41:60,group=3]<- data[41:60,group=3]-10
```

b) Perform PCA on the 60 observations and plot the first two principal component score vectors. Use a different color to indicate the observations in each of the three classes

```
data_pca =prcomp(data, scale =FALSE)
# Plot the first two principal component score vectors
plot(data_pca$x[,1:2], col=1:3, pch =19, xlab ="First principal component", ylab="Seco
nd principal component")
```



c) Perform KK-means clustering of the observations with K=3K=3. How well do the clusters that you obtained in KK-means clustering compare to the true class labels?

```
data_kmeans <- kmeans(data, 3, nstart = 20)
table(groups, data_kmeans$cluster)
##
## groups  1  2  3
##      1 20  0  0
##      2  0 20  0
##      3  0  0 20
```

The results show that clusters are formed perfectly

d) Perform KK-means clustering with K=2K=2. Describe your results.

```
## 2 Cluster
data_kmeans <- kmeans(data, 2, nstart = 20)
table(groups, data_kmeans$cluster)
##
## groups  1  2
##      1  0 20
##      2  0 20
##      3 20  0
```

All Observations from one of the cluster moved to one of the other two clusters

e) Now perform K-means clustering with K = 4, and describe your results.

```
## 4 Cluster
data_kmeans <- kmeans(data, 4, nstart = 20)
table(groups, data_kmeans$cluster)
##
## groups  1  2  3  4
##      1 20  0  0  0
##      2  0  0 20  0
##      3  0  9  0 11
```

3rd cluster broken in to two clusters now 3 and 4

f) Now perform K-means clustering with K = 3 on the first two principal component score vectors, rather than on the raw data.

```
## kmeans over PCA vectors
data_kmeans <- kmeans(data_pca$x[,1:2], 3, nstart = 20)
table(groups, data_kmeans$cluster)
##
## groups  1  2  3
##      1  0 20  0
##      2 20  0  0
##      3  0  0 20
```

All observations are perfectly clustered with PCA vectors

g) Using the scale() function, perform K-means clustering with K = 3

```
## kmeans over scaled data
data_kmeans <- kmeans(scale(data), 3, nstart = 20)
table(groups, data_kmeans$cluster)
##
## groups  1  2  3
##      1 12  1  7
##      2  5  4 11
##      3  0 15  5
```

Scaling has distorted the results in this case. Unnecessary scaling leads to inaccurate distance Euclidean between observation points.

Problem 5

Given: a data set with 100 observations, one quantitative response variable and with following possible fits:

1. Linear fit:
`Y = beta_0 + beta_1 X + beta_2 X^2 + beta_3 X^3 +epsilon`
2. Cubic fit:
`Y = beta_0 + beta_1 X +epsilon`

a) Assuming actual data is close to liner fit

As we do not have complete information about the training data, it is difficult to know which training RSS is lower between linear or cubic. But if true relationship between X and Y is linear we expect training RSS to be lower in linear model compared to cubic model

b) Answer (a) using test rather than training RSS.

Even in this case we don't have enough information about test data to comment on Test RSS. However, we may assume that cubic fit is more complex fit, can over fit the training data that can lead to higher Test RSS value compared to test RSS for liner fit

c) Suppose that the true relationship between X and Y is not linear

In general Polynomial (complex) fits has lower train RSS than the linear fit because of higher flexibility. As the actual fit is not linear it is more likely that cubit fit overt fits the training data to give lower RSS compared to a Linear fit RSS.

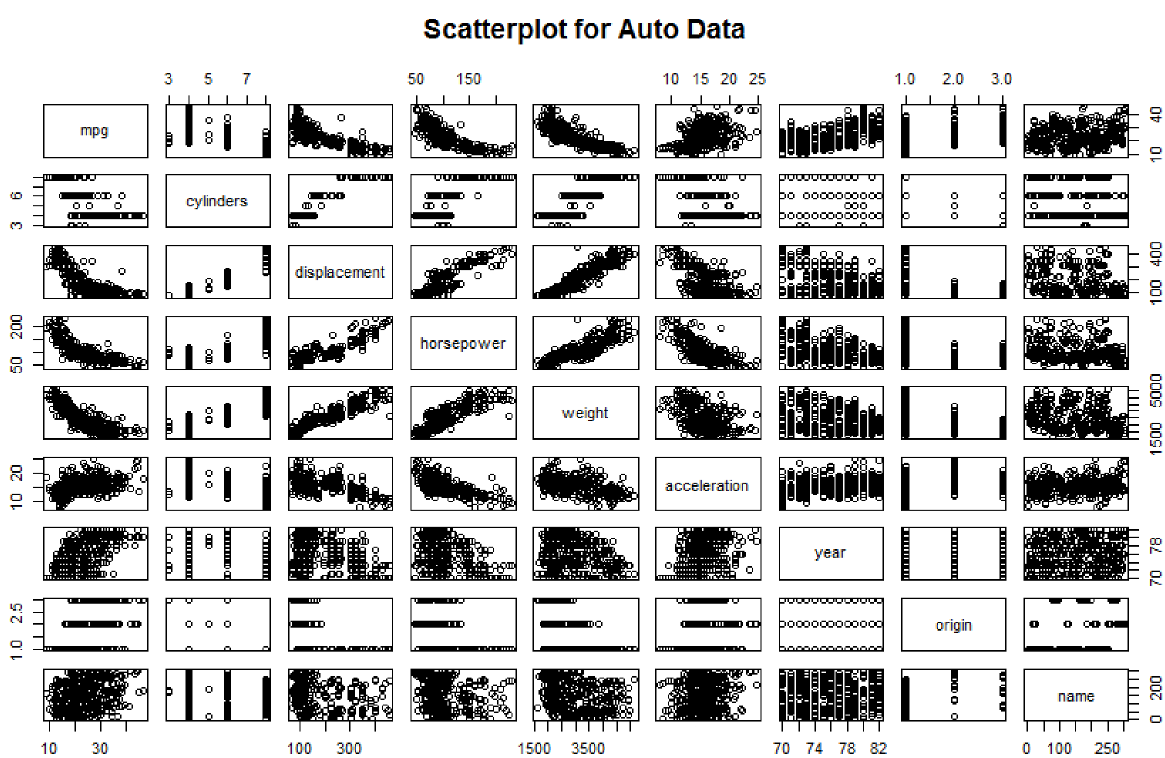
d) Answer (c) using test rather than training RSS.

As we are not aware of true nature of training and test data, it is difficult to comment on Test RSS for both the models. It all depends on true nature of the data, how linear is it, this will decide the bias variance tradeoff.

Problem 6

a) Produce a scatterplot matrix which includes all of the variables in the data set.

```
data(Auto)
pairs(Auto, main='Scatterplot for Auto Data')
```



b) Compute the matrix of correlations between the variables using the function cor().

```
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
```

c) Use the lm() function to perform a multiple linear regression with mpg as the response and all other variables except name as the predictors.

i. Is there a relationship between the predictors and the response?

```
lm_fit <- lm(mpg ~ . - name, data = Auto)
summary(lm_fit)
##
## 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
```

We can look at P value to evaluate if there is any relationship between mpg and other predictors, we can see many p values are less than 0.05 hence there are relationships between mpg and other predictors. For example: year, origin and weight. etc.

ii. Which predictors appear to have a statistically significant relationship to the response?

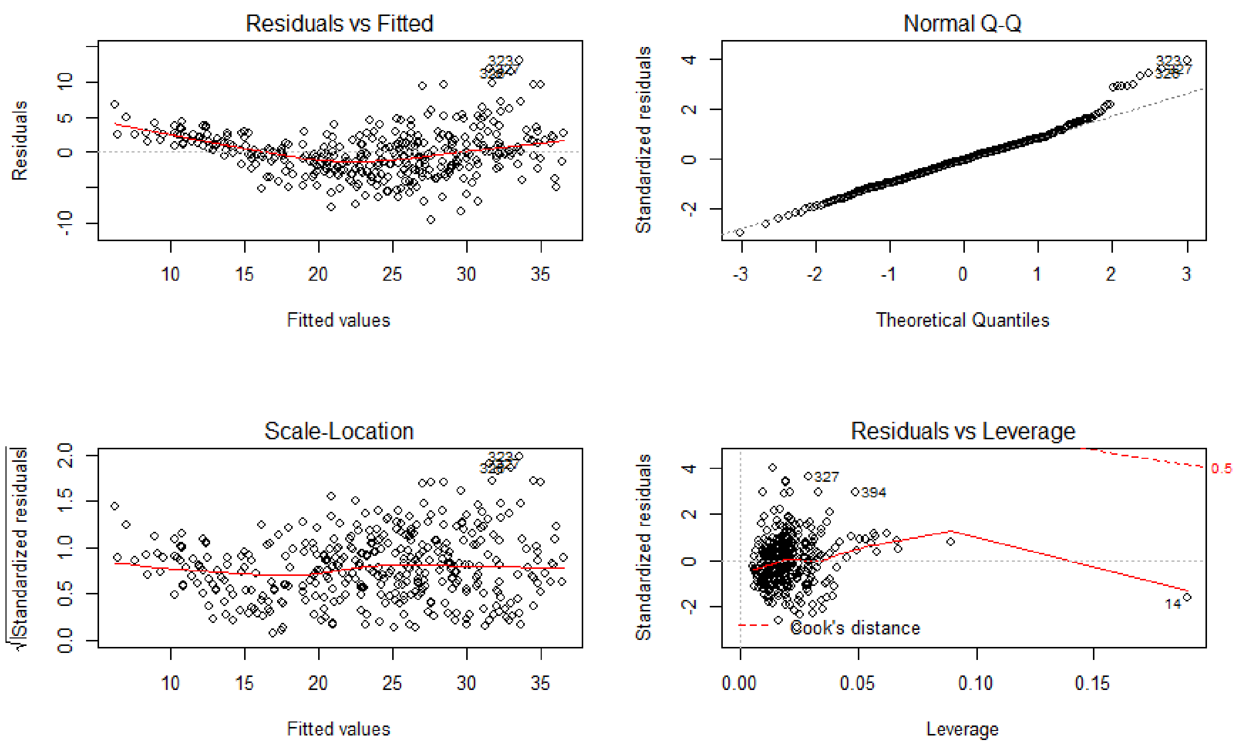
All predictors are statistically significant except cylinders, horsepower and acceleration.

iii. What does the coefficient for the "year" variable suggest?

Coefficient of year is 0.750773, this value suggests that there is positive relationship between year and mpg. Meaning Auto mpg's are improving year by year in general.

d) Use the plot() function to produce diagnostic plots of the linear regression fit. Comment on any problems you see with the fit. Do the residual plots suggest any unusually large outliers ? Does the leverage plots identify any observations with unusually high leverages ?

```
par(mfrow = c(2, 2))
plot(lm_fit)
```



- Residuals Vs Fitted plot indicates the presence of slight non linearity in the data.
- Standardized residuals Vs Leverage plot indicates the presence of a few outliers (higher than 2 or lower than -2) and one high leverage point (14)

Problem 7

Collinearity problem

a) Perform the following commands in R.

```
set.seed(1)
x1 <- runif(100)
x2 <- 0.5 * x1 + rnorm(100)/10
y <- 2 + 2 * x1 + 0.3 * x2 + rnorm(100)
```

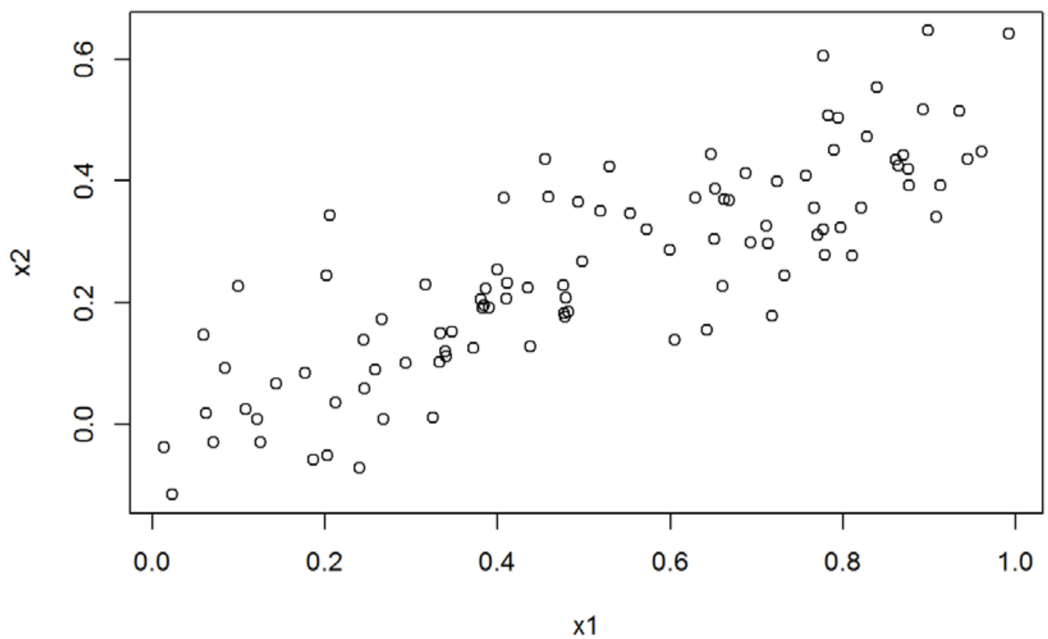
The last line corresponds to creating a linear model in which "y" is a function of "x1" and "x2". Write out the form of the linear model. What are the regression coefficients ?

```
Y = 2 + 2X_1 +0.3X_2 + epsilon
```

with ε : N(0,1) random variable. The regression coefficients are 2, 2 & 0.3 respectively

b) What is the correlation between “x1” and “x2” ? Create a scatterplot displaying the relationship between the variables

```
cor(x1, x2)
## [1] 0.8351212
plot(x1, x2)
```



X1 and X2 Highly correlated

c) Using this data, fit a least squares regression to predict “y” using “x1” and “x2”.

```
Model <- lm(y ~ x1 + x2)
summary(Model)
##
## Call:
## lm(formula = y ~ 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; $p < 0.05$ ➔ can reject the Null Hypothesis for beta_0, also this intercept is close to actual beta_0
- beta_1: 1.4396; $p < 0.05$ ➔ can reject the Null Hypothesis for beta_1
- beta_2: 1.0097; $p > 0.05$ ➔ cannot reject the Null Hypothesis for beta_2

d) Now fit a least squares regression to predict "y" using only "x1".

```
Model1 <- lm(y ~ x1)
summary(Model1)
```

```
## Call:
## lm(formula = y ~ 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
```

- beta_1: 1.9759; different from above scenario with two predictors

e) Now fit a least squares regression to predict “y” using only “x2”.

```
Model2 <- lm(y ~ x2)
summary(Model2)
##
## Call:
## lm(formula = y ~ 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
```

- beta_2: 2.8996, is different from above scenario with two predictors and X2 is significant as p values is < 0.05

f) Do the results obtained in (c)-(e) contradict each other?

- No the results are not contradicting, as X1 and X2 are highly correlated, it is difficult to measure how r=each predictors effects the response variable, this scenario is called ‘collinearity’
- With collinearity: we are unable to estimate beta values correctly also leads to high standard errors

g) Now suppose we obtain one additional observation, which was unfortunately mismeasured

```
x1 <- c(x1, 0.1)
x2 <- c(x2, 0.8)
y <- c(y, 6)
Model_new <- lm(y ~ x1 + x2)
Model1_new <- lm(y ~ x1)
Model2_new <- lm(y ~ x2)
summary(Model_new)
```

```
##
## Call:
## lm(formula = y ~ 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
```

```
summary(Model1_new)
```

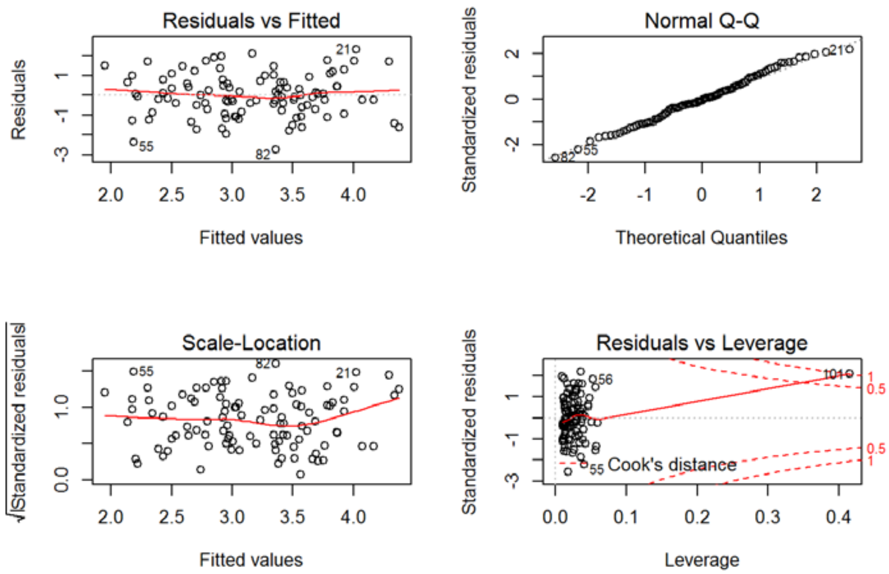
```
##
## Call:
## lm(formula = y ~ x1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8897 -0.6556 -0.0909  0.5682  3.5665
```

```
##
## 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
```

```
summary(Model2_new)
```

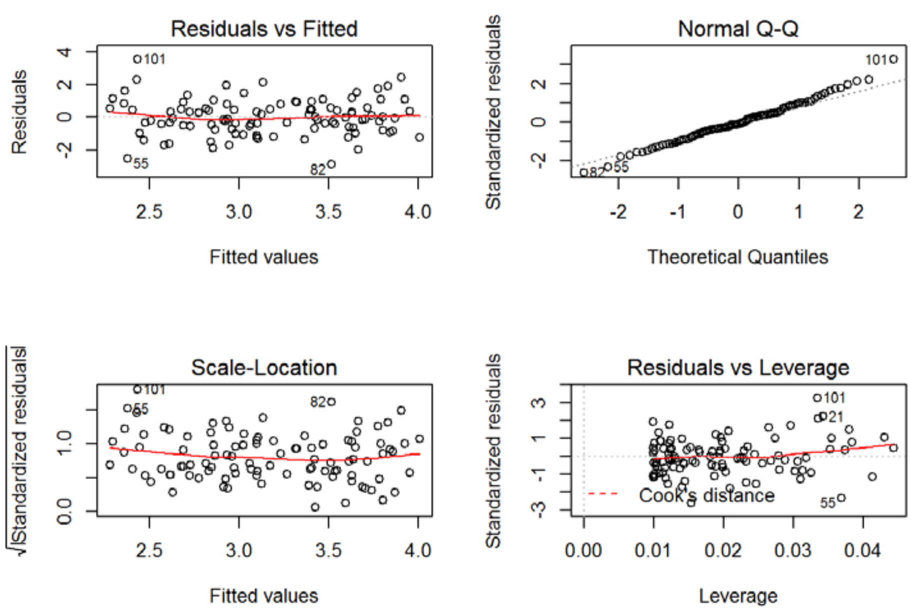
```
##
## Call:
## lm(formula = y ~ x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.64729 -0.71021 -0.06899  0.72699  2.38074
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.3451     0.1912  12.264  < 2e-16 ***
## x2            3.1190     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
```

```r
par(mfrow = c(2, 2))
plot(Model_new)
```



- last point is a high-leverage point.

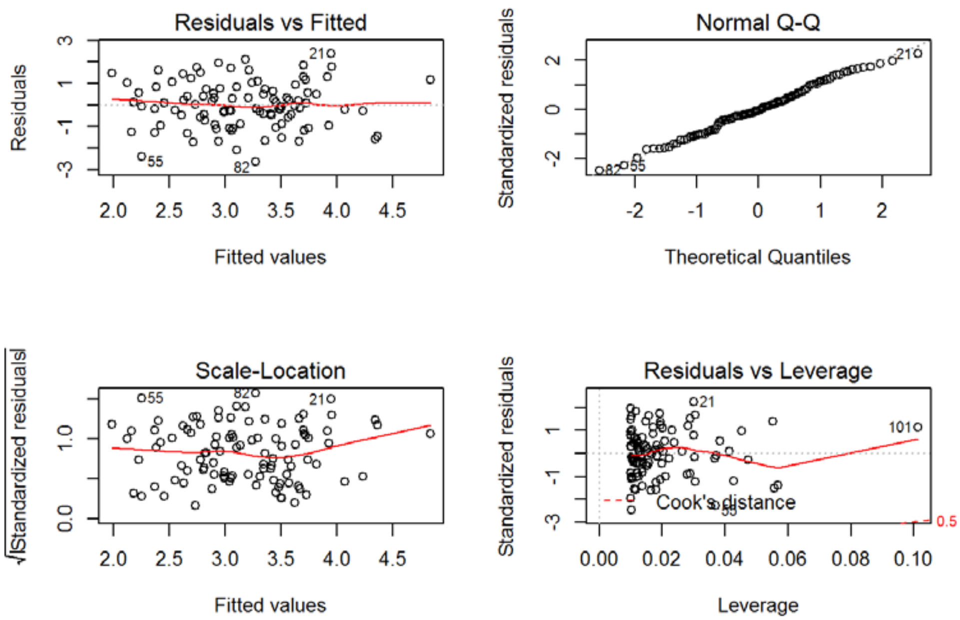
```r
par(mfrow = c(2, 2))
plot(Model1_new)
```



- The last point is an outlier and residuals & Fitted plot indicates high linearity of the model

```r
par(mfrow = c(2, 2))

plot(Model2_new)
```



- The point is again a high leverage point

Problem 8

"Boston" data set

a) For each predictor, fit a simple linear regression model to predict the response. Describe your results. In which of the models is there a statistically significant association between the predictor and the response?

```r
library(MASS)
attach(Boston)
model_zn <- lm(crim ~ zn)
summary(model_zn)

##
## Call:
## lm(formula = crim ~ zn)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.429 -4.222 -2.620  1.250 84.523
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  4.45369    0.41722  10.675  < 2e-16 ***
## zn          -0.07393    0.01609  -4.594 5.51e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.435 on 504 degrees of freedom
## Multiple R-squared:  0.04019,    Adjusted R-squared:  0.03828
## F-statistic:  21.1 on 1 and 504 DF,  p-value: 5.506e-06
```

```
model_indus <- lm(crim ~ indus)
summary(model_indus)
```

```
##
## Call:
## lm(formula = crim ~ indus)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.972  -2.698  -0.736   0.712  81.813
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.06374    0.66723  -3.093  0.00209 **
## indus        0.50978    0.05102   9.991  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.866 on 504 degrees of freedom
## Multiple R-squared:  0.1653, Adjusted R-squared:  0.1637
## F-statistic: 99.82 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
chas <- as.factor(chas)
model_chas <- lm(crim ~ chas)
summary(model_chas)
```

```
##
## Call:
## lm(formula = crim ~ chas)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.738 -3.661 -3.435  0.018 85.232
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.7444     0.3961   9.453   <2e-16 ***
## chas1        -1.8928     1.5061  -1.257    0.209
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.597 on 504 degrees of freedom
## Multiple R-squared:  0.003124,   Adjusted R-squared:  0.001146
## F-statistic: 1.579 on 1 and 504 DF,  p-value: 0.2094
```

```
model_nox <- lm(crim ~ nox)
summary(model_nox)
```

```
##
## Call:
## lm(formula = crim ~ nox)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.371  -2.738  -0.974   0.559  81.728
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -13.720      1.699  -8.073 5.08e-15 ***
## nox           31.249      2.999  10.419  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.81 on 504 degrees of freedom
## Multiple R-squared:  0.1772, Adjusted R-squared:  0.1756
## F-statistic: 108.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
model_rm <- lm(crim ~ rm)
summary(model_rm)
```

```
##
## Call:
## lm(formula = crim ~ rm)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.604 -3.952 -2.654  0.989 87.197
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   20.482      3.365   6.088 2.27e-09 ***
## rm            -2.684      0.532  -5.045 6.35e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.401 on 504 degrees of freedom
## Multiple R-squared:  0.04807,    Adjusted R-squared:  0.04618
## F-statistic: 25.45 on 1 and 504 DF,  p-value: 6.347e-07
```

```
model_age <- lm(crim ~ age)
summary(model_age)
```

```
##
## Call:
## lm(formula = crim ~ age)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.789 -4.257 -1.230  1.527 82.849
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.77791    0.94398  -4.002 7.22e-05 ***
## age          0.10779    0.01274   8.463 2.85e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.057 on 504 degrees of freedom
## Multiple R-squared:  0.1244, Adjusted R-squared:  0.1227
## F-statistic: 71.62 on 1 and 504 DF,  p-value: 2.855e-16
```

```
model_dis <- lm(crim ~ dis)
summary(model_dis)
```

```
##
## Call:
## lm(formula = crim ~ dis)
##
## Residuals:
```

```
##    Min     1Q Median     3Q    Max
## -6.708 -4.134 -1.527  1.516 81.674
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.4993     0.7304  13.006   <2e-16 ***
## dis          -1.5509     0.1683  -9.213   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.965 on 504 degrees of freedom
## Multiple R-squared:  0.1441, Adjusted R-squared:  0.1425
## F-statistic: 84.89 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
model_rad <- lm(crim ~ rad)
summary(model_rad)
```

```
##
## Call:
## lm(formula = crim ~ rad)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.164  -1.381  -0.141   0.660  76.433
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.28716    0.44348  -5.157 3.61e-07 ***
## rad          0.61791    0.03433  17.998  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.718 on 504 degrees of freedom
## Multiple R-squared:  0.3913, Adjusted R-squared:   0.39
## F-statistic: 323.9 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
model_tax <- lm(crim ~ tax)
summary(model_tax)
```

```
##
## Call:
## lm(formula = crim ~ tax)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.513  -2.738  -0.194   1.065  77.696
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.528369   0.815809  -10.45   <2e-16 ***
## tax          0.029742   0.001847   16.10   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.997 on 504 degrees of freedom
## Multiple R-squared:  0.3396, Adjusted R-squared:  0.3383
## F-statistic: 259.2 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
model_ptratio <- lm(crim ~ ptratio)
summary(model_ptratio)
```

```
##
## Call:
## lm(formula = crim ~ ptratio)
```

```
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.654 -3.985 -1.912  1.825 83.353
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.6469     3.1473  -5.607 3.40e-08 ***
## ptratio       1.1520     0.1694   6.801 2.94e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.24 on 504 degrees of freedom
## Multiple R-squared:  0.08407,    Adjusted R-squared:  0.08225
## F-statistic: 46.26 on 1 and 504 DF,  p-value: 2.943e-11
```

```
model_black <- lm(crim ~ black)
summary(model_black)
```

```
##
## Call:
## lm(formula = crim ~ black)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.756  -2.299  -2.095  -1.296  86.822
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16.553529   1.425903  11.609   <2e-16 ***
## black       -0.036280   0.003873  -9.367   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.946 on 504 degrees of freedom
## Multiple R-squared:  0.1483, Adjusted R-squared:  0.1466
## F-statistic: 87.74 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
model_lstat <- lm(crim ~ lstat)
summary(model_lstat)
```

```
##
## Call:
## lm(formula = crim ~ lstat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.925  -2.822  -0.664   1.079  82.862
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.33054    0.69376  -4.801 2.09e-06 ***
## lstat        0.54880    0.04776  11.491  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.664 on 504 degrees of freedom
## Multiple R-squared:  0.2076, Adjusted R-squared:  0.206
## F-statistic:   132 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
model_medv <- lm(crim ~ medv)
summary(model_medv)
```

```
##
## Call:
## lm(formula = crim ~ medv)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.071 -4.022 -2.343  1.298 80.957
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.79654    0.93419   12.63   <2e-16 ***
## medv        -0.36316    0.03839   -9.46   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.934 on 504 degrees of freedom
## Multiple R-squared:  0.1508, Adjusted R-squared:  0.1491
## F-statistic: 89.49 on 1 and 504 DF,  p-value: < 2.2e-16
```

- All predictors are statistically significant except for 'chas' as p values is > 0.05

b) Fit a multiple regression model to predict the response using all of the predictors.

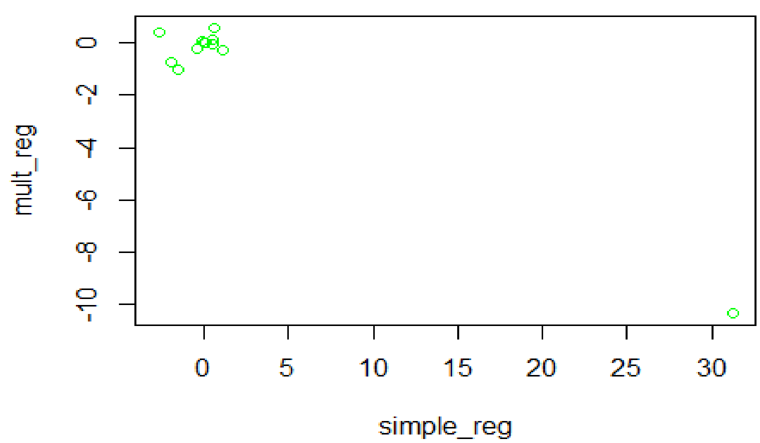
```
model_all <- lm(crim ~ ., data = Boston)
summary(model_all)

##
## Call:
## lm(formula = crim ~ ., data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.924 -2.120 -0.353  1.019 75.051
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  17.033228   7.234903   2.354 0.018949 *
## zn            0.044855   0.018734   2.394 0.017025 *
## indus        -0.063855   0.083407  -0.766 0.444294
## chas         -0.749134   1.180147  -0.635 0.525867
## nox         -10.313535   5.275536  -1.955 0.051152 .
## rm            0.430131   0.612830   0.702 0.483089
## age           0.001452   0.017925   0.081 0.935488
## dis          -0.987176   0.281817  -3.503 0.000502 ***
## rad           0.588209   0.088049   6.680 6.46e-11 ***
## tax          -0.003780   0.005156  -0.733 0.463793
## ptratio      -0.271081   0.186450  -1.454 0.146611
## black        -0.007538   0.003673  -2.052 0.040702 *
## lstat         0.126211   0.075725   1.667 0.096208 .
## medv         -0.198887   0.060516  -3.287 0.001087 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.439 on 492 degrees of freedom
## Multiple R-squared:  0.454,  Adjusted R-squared:  0.4396
## F-statistic: 31.47 on 13 and 492 DF,  p-value: < 2.2e-16
```

- Coefficients of zn, dis, rad, black and medv are significant hence we can reject the Null hypothesis for these predictors

c) How do your results from (a) compare to your results from (b)

```
simple_reg <- vector("numeric",0)
simple_reg <- c(simple_reg, model_zn$coefficient[2])
simple_reg <- c(simple_reg, model_indus$coefficient[2])
simple_reg <- c(simple_reg, model_chas$coefficient[2])
simple_reg <- c(simple_reg, model_nox$coefficient[2])
simple_reg <- c(simple_reg, model_rm$coefficient[2])
simple_reg <- c(simple_reg, model_age$coefficient[2])
simple_reg <- c(simple_reg, model_dis$coefficient[2])
simple_reg <- c(simple_reg, model_rad$coefficient[2])
simple_reg <- c(simple_reg, model_tax$coefficient[2])
simple_reg <- c(simple_reg, model_ptratio$coefficient[2])
simple_reg <- c(simple_reg, model_black$coefficient[2])
simple_reg <- c(simple_reg, model_lstat$coefficient[2])
simple_reg <- c(simple_reg, model_medv$coefficient[2])
mult_reg <- vector("numeric", 0)
mult_reg <- c(mult_reg, model_all$coefficients)
mult_reg <- mult_reg[-1]
plot(simple_reg, mult_reg, col = "green")
```



- The difference between simple and multiple regression coefficients is due to correlation among predictors
- This leads to no storing relation with multiple regression

```
cor(Boston[-c(1, 4)])
```

```
##                 zn      indus        nox         rm        age        dis
## zn       1.0000000 -0.5338282 -0.5166037  0.3119906 -0.5695373  0.6644082
## indus   -0.5338282  1.0000000  0.7636514 -0.3916759  0.6447785 -0.7080270
## nox     -0.5166037  0.7636514  1.0000000 -0.3021882  0.7314701 -0.7692301
## rm       0.3119906 -0.3916759 -0.3021882  1.0000000 -0.2402649  0.2052462
## age     -0.5695373  0.6447785  0.7314701 -0.2402649  1.0000000 -0.7478805
## dis      0.6644082 -0.7080270 -0.7692301  0.2052462 -0.7478805  1.0000000
## rad     -0.3119478  0.5951293  0.6114406 -0.2098467  0.4560225 -0.4945879
## tax     -0.3145633  0.7207602  0.6680232 -0.2920478  0.5064556 -0.5344316
## ptratio -0.3916785  0.3832476  0.1889327 -0.3555015  0.2615150 -0.2324705
## black    0.1755203 -0.3569765 -0.3800506  0.1280686 -0.2735340  0.2915117
## lstat   -0.4129946  0.6037997  0.5908789 -0.6138083  0.6023385 -0.4969958
## medv     0.3604453 -0.4837252 -0.4273208  0.6953599 -0.3769546  0.2499287
##                rad        tax    ptratio      black      lstat       medv
## zn      -0.3119478 -0.3145633 -0.3916785  0.1755203 -0.4129946  0.3604453
## indus    0.5951293  0.7207602  0.3832476 -0.3569765  0.6037997 -0.4837252
## nox      0.6114406  0.6680232  0.1889327 -0.3800506  0.5908789 -0.4273208
## rm      -0.2098467 -0.2920478 -0.3555015  0.1280686 -0.6138083  0.6953599
## age      0.4560225  0.5064556  0.2615150 -0.2735340  0.6023385 -0.3769546
## dis     -0.4945879 -0.5344316 -0.2324705  0.2915117 -0.4969958  0.2499287
## rad      1.0000000  0.9102282  0.4647412 -0.4444128  0.4886763 -0.3816262
## tax      0.9102282  1.0000000  0.4608530 -0.4418080  0.5439934 -0.4685359
## ptratio  0.4647412  0.4608530  1.0000000 -0.1773833  0.3740443 -0.5077867
## black   -0.4444128 -0.4418080 -0.1773833  1.0000000 -0.3660869  0.3334608
## lstat    0.4886763  0.5439934  0.3740443 -0.3660869  1.0000000 -0.7376627
## medv    -0.3816262 -0.4685359 -0.5077867  0.3334608 -0.7376627  1.0000000
```

d) Is there evidence of non-linear association between any of the predictors and the response?

```
library(MASS)
attach(Boston)
poly_model_zn <- lm(crim ~ poly(zn))
summary(poly_model_zn)
```

```
##
## Call:
## lm(formula = crim ~ poly(zn))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.429 -4.222 -2.620  1.250 84.523
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.614      0.375   9.636  < 2e-16 ***
## poly(zn)     -38.750      8.435  -4.594 5.51e-06 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.435 on 504 degrees of freedom
## Multiple R-squared:  0.04019,    Adjusted R-squared:  0.03828
## F-statistic:  21.1 on 1 and 504 DF,  p-value: 5.506e-06
```

```
poly_model_indus <- lm(crim ~ poly( indus))
summary(poly_model_indus)
```

```
##
## Call:
## lm(formula = crim ~ poly(indus))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.972  -2.698  -0.736   0.712  81.813
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.6135     0.3497  10.333   <2e-16 ***
## poly(indus)  78.5908     7.8663   9.991   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.866 on 504 degrees of freedom
## Multiple R-squared:  0.1653, Adjusted R-squared:  0.1637
## F-statistic: 99.82 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
poly_model_nox <- lm(crim ~ poly( nox))
summary(poly_model_nox)
```

```
##
## Call:
## lm(formula = crim ~ poly(nox))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.371  -2.738  -0.974   0.559  81.728
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.6135     0.3472   10.41   <2e-16 ***
## poly(nox)    81.3720     7.8100   10.42   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.81 on 504 degrees of freedom
## Multiple R-squared:  0.1772, Adjusted R-squared:  0.1756
## F-statistic: 108.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

```r
poly_model_rm <- lm(crim ~ poly( rm))
summary(poly_model_rm)
```

```
##
## Call:
## lm(formula = crim ~ poly(rm))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.604 -3.952 -2.654  0.989 87.197
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.6135     0.3735   9.676  < 2e-16 ***
## poly(rm)    -42.3794     8.4006  -5.045 6.35e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.401 on 504 degrees of freedom
## Multiple R-squared:  0.04807,    Adjusted R-squared:  0.04618
## F-statistic: 25.45 on 1 and 504 DF,  p-value: 6.347e-07
```

```r
poly_model_age <- lm(crim ~ poly( age))
summary(poly_model_age)
```

```
##
## Call:
## lm(formula = crim ~ poly(age))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.789 -4.257 -1.230  1.527 82.849
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.6135     0.3582  10.089  < 2e-16 ***
## poly(age)    68.1820     8.0566   8.463 2.85e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.057 on 504 degrees of freedom
## Multiple R-squared:  0.1244, Adjusted R-squared:  0.1227
## F-statistic: 71.62 on 1 and 504 DF,  p-value: 2.855e-16
```

```r
poly_model_dis <- lm(crim ~ poly( dis))
summary(poly_model_dis)
```

```
##
## Call:
## lm(formula = crim ~ poly(dis))
##
```

```
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.708 -4.134 -1.527  1.516 81.674
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.6135     0.3541  10.205   <2e-16 ***
## poly(dis)   -73.3886     7.9654  -9.213   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.965 on 504 degrees of freedom
## Multiple R-squared:  0.1441, Adjusted R-squared:  0.1425
## F-statistic: 84.89 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
poly_model_rad <- lm(crim ~ poly( rad))
summary(poly_model_rad)
```

```
##
## Call:
## lm(formula = crim ~ poly(rad))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.164  -1.381  -0.141   0.660  76.433
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.6135     0.2986    12.1   <2e-16 ***
## poly(rad)   120.9074     6.7178    18.0   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.718 on 504 degrees of freedom
## Multiple R-squared:  0.3913, Adjusted R-squared:   0.39
## F-statistic: 323.9 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
poly_model_tax <- lm(crim ~ poly( tax))
summary(poly_model_tax)
```

```
##
## Call:
## lm(formula = crim ~ poly(tax))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.513  -2.738  -0.194   1.065  77.696
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.6135     0.3111   11.62   <2e-16 ***
```

```
## poly(tax)   112.6458     6.9969   16.10   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.997 on 504 degrees of freedom
## Multiple R-squared:  0.3396, Adjusted R-squared:  0.3383
## F-statistic: 259.2 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
poly_model_ptratio <- lm(crim ~ poly( ptratio))
summary(poly_model_ptratio)
```

```
##
## Call:
## lm(formula = crim ~ poly(ptratio))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.654 -3.985 -1.912  1.825 83.353
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.6135     0.3663   9.864  < 2e-16 ***
## poly(ptratio)  56.0452     8.2402   6.801 2.94e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.24 on 504 degrees of freedom
## Multiple R-squared:  0.08407,    Adjusted R-squared:  0.08225
## F-statistic: 46.26 on 1 and 504 DF,  p-value: 2.943e-11
```

```
poly_model_black <- lm(crim ~ poly( black))
summary(poly_model_black)
```

```
##
## Call:
## lm(formula = crim ~ poly(black))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.756  -2.299  -2.095  -1.296  86.822
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.6135     0.3532  10.229   <2e-16 ***
## poly(black) -74.4312     7.9462  -9.367   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.946 on 504 degrees of freedom
## Multiple R-squared:  0.1483, Adjusted R-squared:  0.1466
## F-statistic: 87.74 on 1 and 504 DF,  p-value: < 2.2e-16
```

```r
poly_model_lstat <- lm(crim ~ poly( lstat))
summary(poly_model_lstat)

##
## Call:
## lm(formula = crim ~ poly(lstat))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.925  -2.822  -0.664   1.079  82.862
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.6135     0.3407   10.61   <2e-16 ***
## poly(lstat)  88.0697     7.6645   11.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.664 on 504 degrees of freedom
## Multiple R-squared:  0.2076, Adjusted R-squared:  0.206
## F-statistic:   132 on 1 and 504 DF,  p-value: < 2.2e-16

poly_model_medv <- lm(crim ~ poly( medv))
summary(poly_model_medv)

##
## Call:
## lm(formula = crim ~ poly(medv))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.071 -4.022 -2.343  1.298 80.957
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.6135     0.3527   10.24   <2e-16 ***
## poly(medv)  -75.0576     7.9345   -9.46   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.934 on 504 degrees of freedom
## Multiple R-squared:  0.1508, Adjusted R-squared:  0.1491
## F-statistic: 89.49 on 1 and 504 DF,  p-value: < 2.2e-16
```

- Following predictors are statistically significant as per p-value
  zn, rm, rad, tax and lstat
- Not significant predictors
  indus, nox, age, dis, ptratio and medv