CptS 475/575: Data Science, Fall 2018

Assignment 4: Linear Regression

Release Date: October 3, 2018 **Due Date:** October 12, 2018 (11:59 pm)

General instruction: This assignment has **three problems.** The first two problems are for both CptS 475 and CptS 575 students. The third problem is only for CptS 575 students. For CptS 475 students, Problem 1 and Problem 2 each carry 50% of the total weight. For CptS 575 students, Problem 1 and Problem 2 each carry 45%, and Problem 3 carries 10% of the total weight.

Your solution will be submitted as a PDF file. You are encouraged to use R Markdown to prepare your file.

1. This question involves the use of multiple linear regression on the Auto data set from the course webpage (https://scads.eecs.wsu.edu/index.php/datasets/). Ensure that you remove missing values from the dataframe, and that values are represented in the appropriate types (num or int for quantitative variables, factor, logi or str for qualitative).
    a. Produce a scatterplot matrix which includes all of the variables in the data set.
    b. Compute the matrix of correlations between the variables using the function cor(). You will need to exclude the name variable, which is qualitative.
    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. Use the summary() function to print the results. Comment on the output:
        i. Which predictors appear to have a statistically significant relationship to the response, and how do you determine this?
        ii. What does the coefficient for the cylinders variable suggest, in simple terms?
    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 plot identify any observations with unusually high leverage?
    e. Use the * and : symbols to fit linear regression models with interaction effects. Do any interactions appear to be statistically significant?
    f. Try transformations of the variables with X^3 and log(X). Comment on your findings.
2. This problem involves the Boston data set, which we saw in the lab. We will now try to predict per capita crime rate using the other variables in this data set. In other words, per capita crime rate is the response, and the other variables are the predictors.
    a. For each predictor, fit a simple linear regression model to predict the response. Include the code, but not the output for all models in your solution. In which of the models is there a statistically significant association between the predictor and the response? Considering the meaning of each variable, discuss the relationship between crim and nox, chas, medv and dis in particular. How do these relationships differ?

b. Fit a multiple regression model to predict the response using all of the predictors. Describe your results. For which predictors can we reject the null hypothesis $H_0 : \beta_j = 0$?
c. How do your results from (a) compare to your results from (b)? Create a plot displaying the univariate regression coefficients from (a) on the x-axis, and the multiple regression coefficients from (b) on the y-axis. That is, each predictor is displayed as a single point in the plot. Its coefficient in a simple linear regression model is shown on the x-axis, and its coefficient estimate in the multiple linear regression model is shown on the y-axis. What does this plot tell you about the various predictors?
d. Is there evidence of non-linear association between any of the predictors and the response? To answer this question, for each predictor X, fit a model of the form

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \varepsilon$$

Hint: use the poly() function. Again, include the code, but not the output for each model in your solution, and instead describe any non-linear trends you uncover.

3. An important assumption of the linear regression model is that the error terms are uncorrelated (independent). But error terms can sometimes be correlated, especially in time-series data.

a. What are the issues that could arise in using linear regression (via least squares estimates) when error terms are correlated? Comment in particular with respect to
    i) regression coefficients
    ii) the standard error of regression coefficients
    iii) confidence intervals

b. What methods can be applied to deal with correlated errors? Mention at least one method.

Assignment 2

Sheryl Mathew (11627236)

11 October, 2018

Question 1

1. Read the Auto csv file

```
library(kableExtra)
auto_all=read.csv("https://scads.eecs.wsu.edu/wp-content/uploads/2017/09/Auto.csv",na.strings="?")
auto=na.omit(auto_all)
kable(head(auto), format = "latex", booktabs = T,
      caption="Table containing details of Auto CSV") %>%
kable_styling(latex_options = c("striped","hold_position","scale_down"))
```

Table 1: Table containing details of Auto CSV

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

```
sapply(auto,class)
```

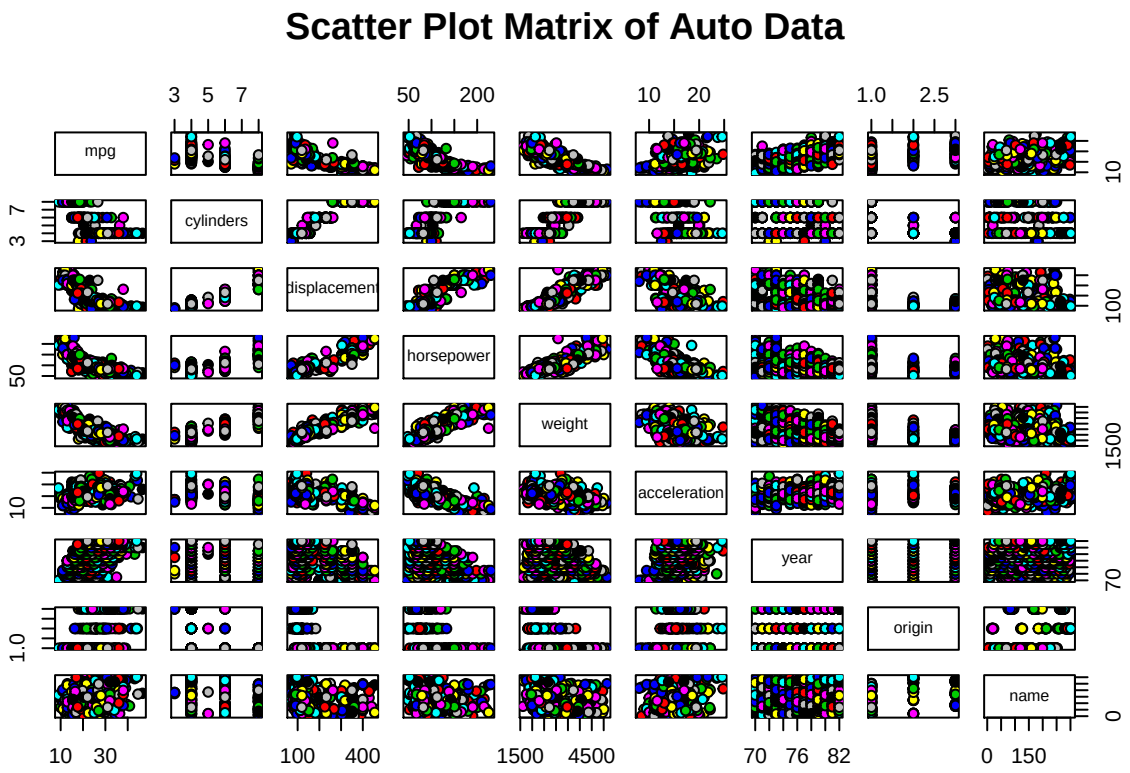
```
##          mpg    cylinders displacement   horsepower       weight
##    "numeric"    "integer"    "numeric"    "integer"    "integer"
## acceleration         year       origin         name
##    "numeric"    "integer"    "integer"     "factor"
```

```
auto$origin=as.factor(auto$origin)
sapply(auto,class)
```

```
##          mpg    cylinders displacement   horsepower       weight
##    "numeric"    "integer"    "numeric"    "integer"    "integer"
## acceleration         year       origin         name
##    "numeric"    "integer"     "factor"     "factor"
```

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

```
pairs(auto, main = "Scatter Plot Matrix of Auto Data",pch=21,bg=palette(rainbow(9)))
```



3. Compute the matrix of correlations between the variables

```
auto$origin=as.numeric(auto$origin)
cor_matrix = cor(auto[1:8])
kable(cor_matrix, format = "latex", booktabs = T,
      caption="Correlation Matrix of Auto Data") %>%
kable_styling(latex_options = c("striped","hold_position","scale_down"))
```

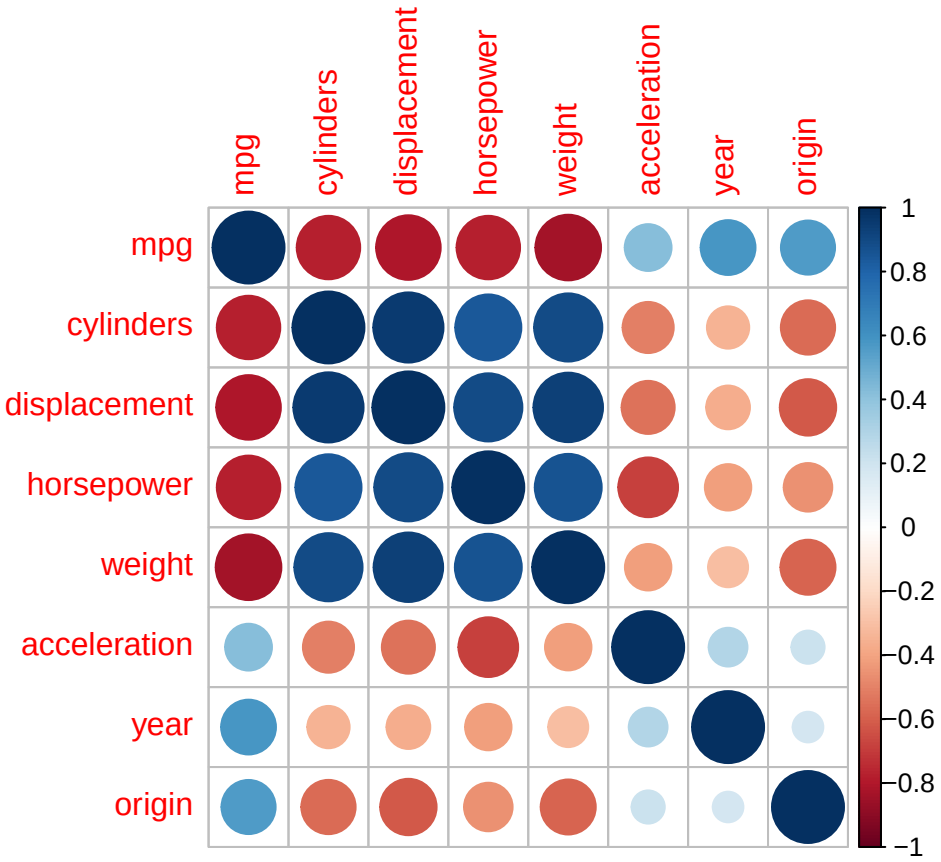
Table 2: Correlation Matrix of Auto Data

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

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
corrplot(cor_matrix)
```



4. Perform a multiple linear regression with mpg as the response and all other variables except name as the predictors

```
lr_model = lm(mpg ~ cylinders + displacement + horsepower + weight + year + origin, data = auto)
summary(lr_model)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     year + origin, data = auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.7604 -2.1791 -0.1535  1.8524 13.1209
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.556e+01  4.175e+00  -3.728 0.000222 ***
## cylinders    -5.067e-01  3.227e-01  -1.570 0.117236
## displacement  1.927e-02  7.472e-03   2.579 0.010287 *
## horsepower   -2.389e-02  1.084e-02  -2.205 0.028031 *
## weight       -6.218e-03  5.714e-04 -10.883  < 2e-16 ***
## year          7.475e-01  5.079e-02  14.717  < 2e-16 ***
## origin        1.428e+00  2.780e-01   5.138 4.43e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.326 on 385 degrees of freedom
## Multiple R-squared:  0.8212, Adjusted R-squared:  0.8184
## F-statistic: 294.6 on 6 and 385 DF,  p-value: < 2.2e-16
```

4.1 Which predictors appear to have a statistically significant relationship to the response, and how do you determine this?

Weight, Year, Origin have the highest statistically significant relationship to the response "mpg".
Horsepower, Displacement are also statistically significant relationship to the response "mpg".

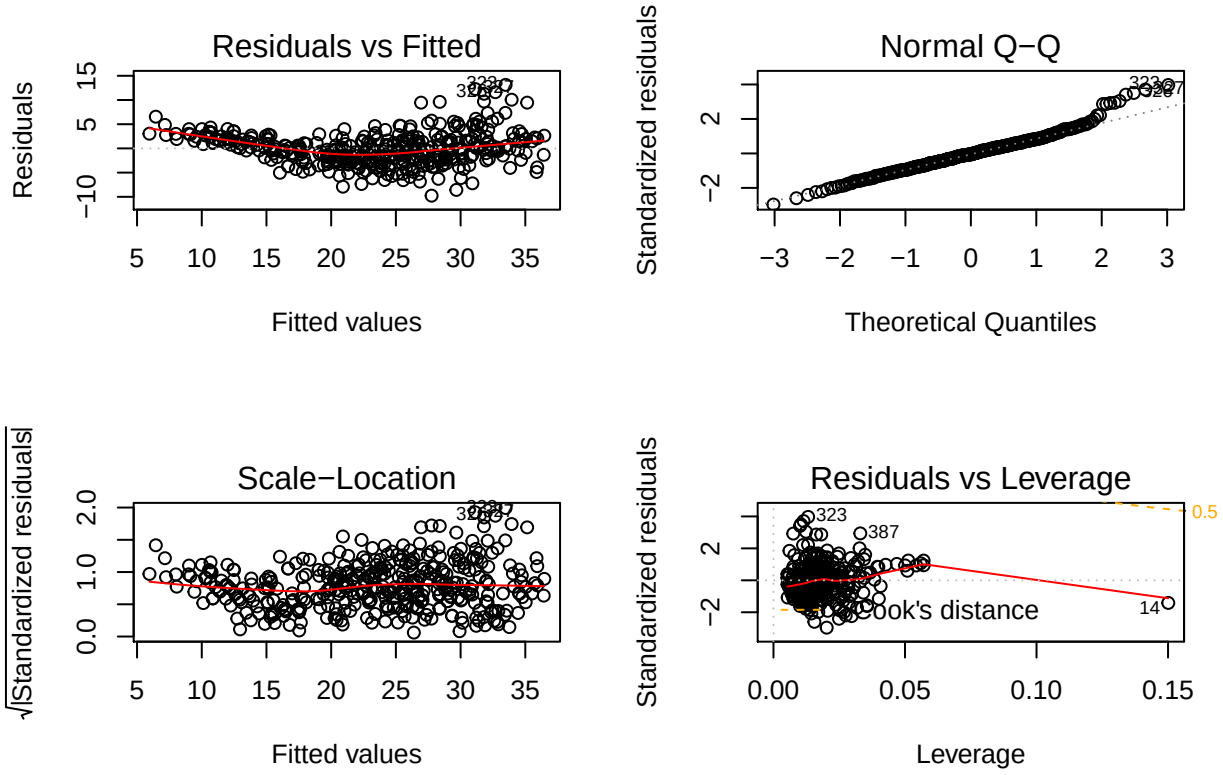
We can visually interpret the significance of the predictors by the significance stars at the end of the row ie more the number of stars beside the variable's p-Value, the more significant the variable. Therefore, Weight, Year and Origin are the most significant. However, Horsepower and Displacement are also significant because they have p-Values which are less than the pre-determined statistical significance level, which is ideally 0.05. R-squared value shows that that 82.12% of the changes in the response "mpg" can be explained by the predictors.

4.2 What does the coefficient for the cylinders variable suggest, in simple terms?

The coefficient of cylinders tell us how much the value of "mpg" will decrease (as the coefficient is negative) when the number of cylinders increases by one while keeping all the other variables like displacement, horsepower constant.

5. Produce diagnostic plots of the linear regression fit

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



Residuals vs Fitted: The red line has a U-shape which means that there is non-linear relationship between the predictor and response variables.

Normal Q-Q: Though most of the residuals follow a straight line, points 323, 327, 326 deviate from the straight line.

Scale - Location: The red line is almost horizontal which represents that the residuals are equally spread throughout the range of predictors

Residuals vs Leverage: Point 323 has a standaridized residual as 4 and Point 387 has a standaridized residual as approximately 3. Point 14 has a high leverage which means that it has a high influence ie it determines how much the predicted scores will change if the point is excluded.

6. Fit linear regression models with interaction effects.

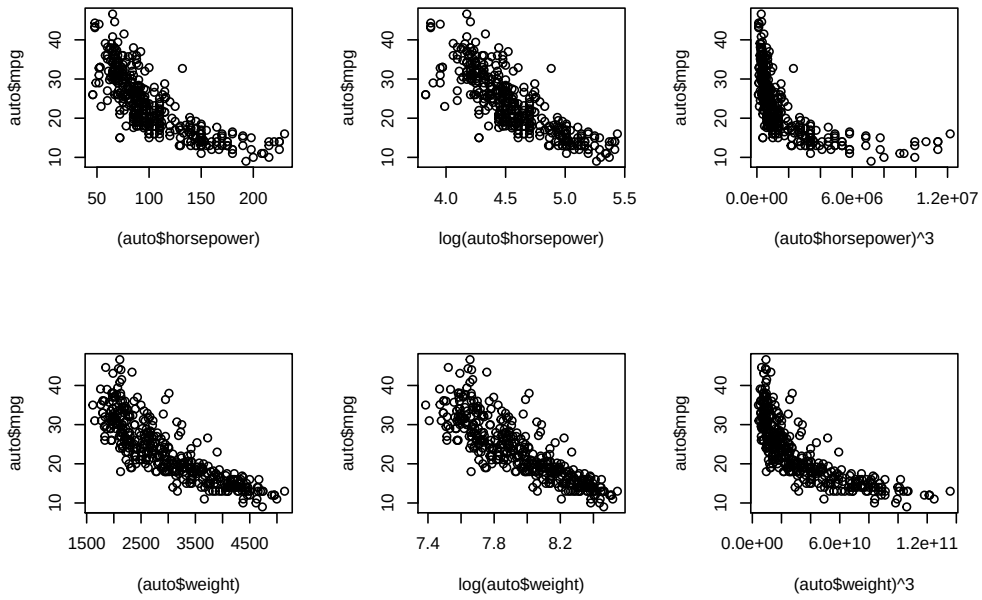
```
interaction = lm(mpg ~ cylinders + displacement + horsepower + weight + year + origin
                 + displacement:weight + displacement:cylinders
                 + acceleration*horsepower + year*origin, data = auto)
summary(interaction)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     year + origin + displacement:weight + displacement:cylinders +
##     acceleration * horsepower + year * origin, data = auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.6504 -1.6476  0.0381  1.4254 12.7893
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 5.287e+00  9.074e+00   0.583 0.560429
## cylinders                   4.249e-01  6.079e-01   0.699 0.485011
## displacement               -7.322e-02  1.334e-02  -5.490 7.38e-08 ***
## horsepower                  5.252e-02  2.586e-02   2.031 0.042913 *
## weight                     -8.689e-03  1.086e-03  -7.998 1.54e-14 ***
## year                        5.116e-01  9.976e-02   5.129 4.66e-07 ***
## origin                     -1.220e+01  4.161e+00  -2.933 0.003560 **
## acceleration                5.796e-01  1.582e-01   3.665 0.000283 ***
## displacement:weight         1.992e-05  3.608e-06   5.522 6.21e-08 ***
## cylinders:displacement     -4.368e-04  2.712e-03  -0.161 0.872156
## horsepower:acceleration    -6.735e-03  1.781e-03  -3.781 0.000181 ***
## year:origin                 1.630e-01  5.341e-02   3.051 0.002440 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.874 on 380 degrees of freedom
## Multiple R-squared:  0.8683, Adjusted R-squared:  0.8644
## F-statistic: 227.7 on 11 and 380 DF,  p-value: < 2.2e-16
```

Interaction between *displacement & weight*, *horsepower & acceleration* and *year & origin* are statistically significant.

7. Try transformations of the variables with X3 and log(X).

```
par(mfrow = c(2,3))
plot((auto$horsepower), auto$mpg)
plot(log(auto$horsepower), auto$mpg)
plot((auto$horsepower)^3, auto$mpg)
plot((auto$weight), auto$mpg)
plot(log(auto$weight), auto$mpg)
plot((auto$weight)^3, auto$mpg)
```



The log transformation of Horsepower and Weight gives the most linear looking plot.

```
lr_model1 = lm(mpg ~ log(horsepower), data = auto)
summary(lr_model1)$r.squared
```

```
## [1] 0.6683348
```

```
lr_model2 = lm(mpg ~ (horsepower)^3, data = auto)
summary(lr_model2)$r.squared
```

```
## [1] 0.6059483
```

```
lr_model3 = lm(mpg ~ (horsepower), data = auto)
summary(lr_model3)$r.squared
```

```
## [1] 0.6059483
```

The log transformation has a higher R-squared value when compared to the cube transformation

```
lr_model1 = lm(mpg ~ log(weight), data = auto)
summary(lr_model1)$r.squared
```

```
## [1] 0.7126631
```

```
lr_model2 = lm(mpg ~ (weight)^3, data = auto)
summary(lr_model2)$r.squared
```

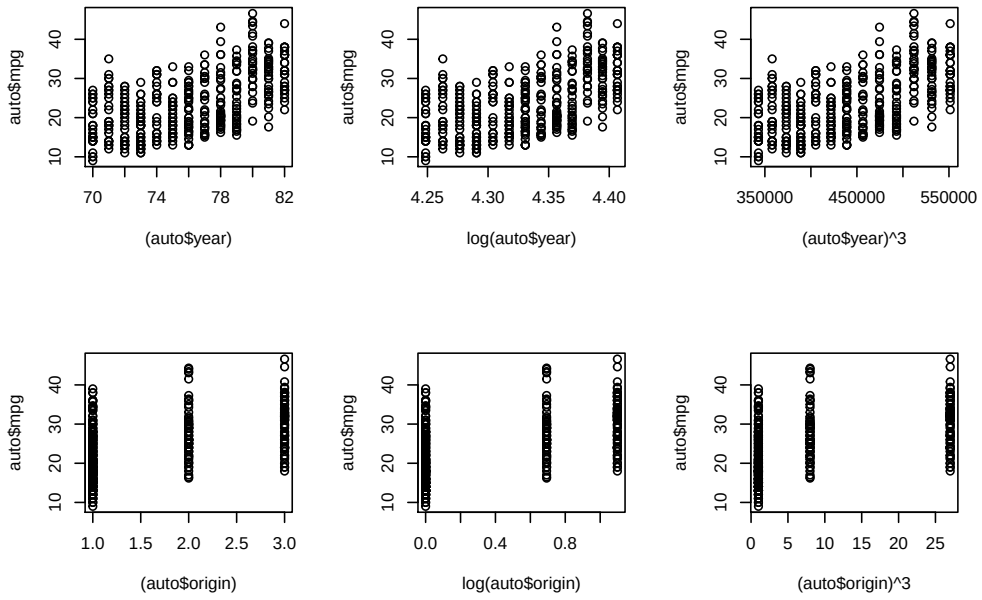
```
## [1] 0.6926304
```

```
lr_model3 = lm(mpg ~ (weight), data = auto)
summary(lr_model2)$r.squared
```

```
## [1] 0.6926304
```

The log transformation has a higher R-squared value when compared to the cube transformation

```
par(mfrow = c(2,3))
plot((auto$year), auto$mpg)
plot(log(auto$year), auto$mpg)
plot((auto$year)^3, auto$mpg)
plot((auto$origin), auto$mpg)
plot(log(auto$origin), auto$mpg)
plot((auto$origin)^3, auto$mpg)
```



The transformation on year and origin does not make any difference.

```
lr_model1 = lm(mpg ~ log(year), data = auto)
summary(lr_model1)$r.squared
```

```
## [1] 0.3323744
```

```
lr_model2 = lm(mpg ~ (year)^3, data = auto)
summary(lr_model2)$r.squared
```

```
## [1] 0.3370278
```

```
lr_model3 = lm(mpg ~ (year), data = auto)
summary(lr_model3)$r.squared
```

```
## [1] 0.3370278
```

The R-squared value of all the transformations is almost same

```
lr_model1 = lm(mpg ~ log(origin), data = auto)
summary(lr_model1)$r.squared
```

```
## [1] 0.3297927
```

```
lr_model2 = lm(mpg ~ (origin)^3, data = auto)
summary(lr_model2)$r.squared
```

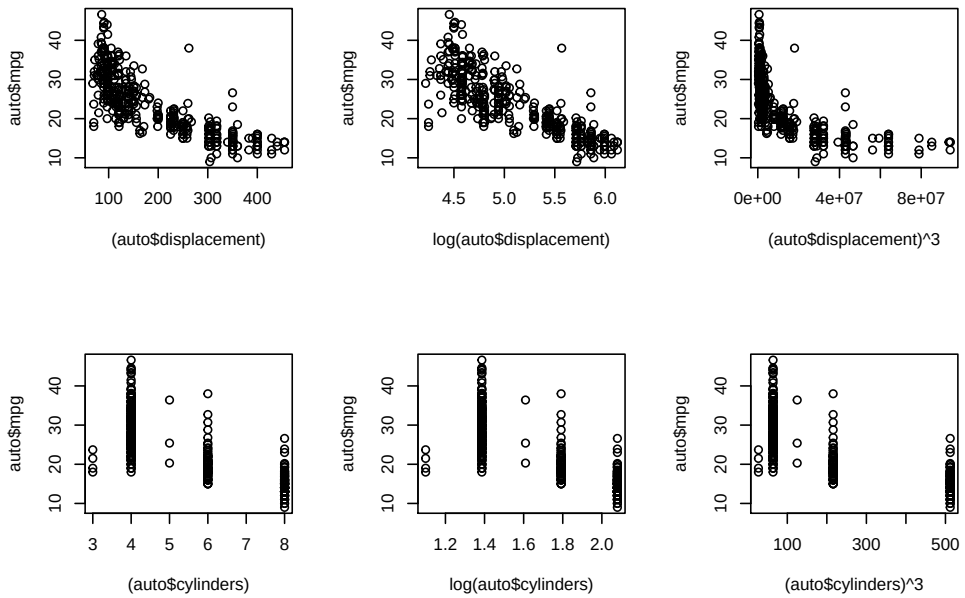
```
## [1] 0.3194609
```

```
lr_model3 = lm(mpg ~ (origin), data = auto)
summary(lr_model3)$r.squared
```

```
## [1] 0.3194609
```

The R-squared value of all the transformations is almost same

```
par(mfrow = c(2,3))
plot((auto$displacement), auto$mpg)
plot(log(auto$displacement), auto$mpg)
plot((auto$displacement)^3, auto$mpg)
plot((auto$cylinders), auto$mpg)
plot(log(auto$cylinders), auto$mpg)
plot((auto$cylinders)^3, auto$mpg)
```



The log transformation of Displacement gives the most linear looking plot. While the transformations on cylinders does not make any difference.

```
lr_model1 = lm(mpg ~ log(displacement), data = auto)
summary(lr_model1)$r.squared
```

```
## [1] 0.6863349
```

```
lr_model2 = lm(mpg ~ (displacement)^3, data = auto)
summary(lr_model2)$r.squared
```

```
## [1] 0.6482294
```

```
lr_model3 = lm(mpg ~ (displacement), data = auto)
summary(lr_model3)$r.squared
```

```
## [1] 0.6482294
```

The log transformation has a higher R-squared value when compared to the cube transformation

```
lr_model1 = lm(mpg ~ log(cylinders), data = auto)
summary(lr_model1)$r.squared
```

```
## [1] 0.6034457
```

```
lr_model2 = lm(mpg ~ (cylinders)^3, data = auto)
summary(lr_model2)$r.squared
```

```
## [1] 0.604689
```

```
lr_model3 = lm(mpg ~ (cylinders), data = auto)
summary(lr_model3)$r.squared
```

```
## [1] 0.604689
```

The R-squared value of all the transformations is almost same

Question 2

1. Get Boston file

```
library(MASS)
boston=Boston
kable(head(boston), format = "latex", booktabs = T,
      caption="Table containing details of Boston data") %>%
kable_styling(latex_options = c("striped","hold_position","scale_down"))
```

Table 3: Table containing details of Boston data

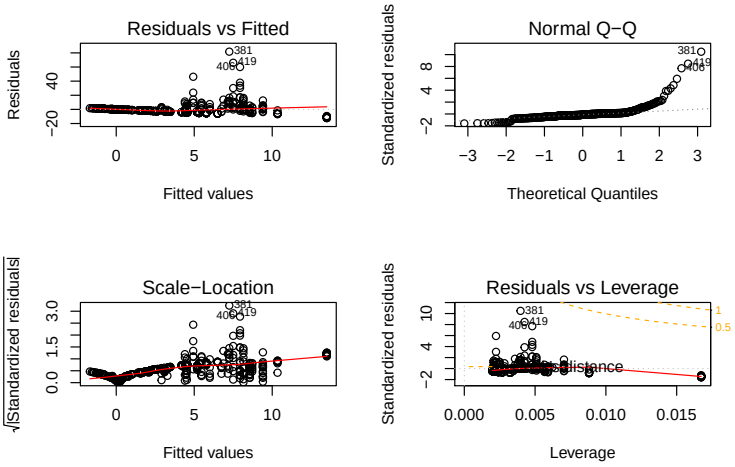
crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
0.00632	18	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0
0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6
0.02729	0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2
0.02985	0	2.18	0	0.458	6.430	58.7	6.0622	3	222	18.7	394.12	5.21	28.7

2. Fit a simple linear regression model to predict the response

```
lr_zn = lm(crim ~ zn , data = boston)
#summary(lr_zn)
lr_indus = lm(crim ~ indus , data = boston)
#summary(lr_indus)
lr_chas = lm(crim ~ chas , data = boston)
#ummary(lr_chas)
lr_nox = lm(crim ~ nox , data = boston)
#summary(lr_nox)
lr_rm = lm(crim ~ rm , data = boston)
#summary(lr_rm)
lr_age = lm(crim ~ age , data = boston)
#summary(lr6)
lr_dis = lm(crim ~ dis , data = boston)
#summary(lr_dis)
lr_rad = lm(crim ~ rad , data = boston)
#summary(lr_rad)
lr_tax = lm(crim ~ tax , data = boston)
#summary(lr_tax)
lr_ptratio = lm(crim ~ ptratio , data = boston)
#summary(lr_ptratio)
lr_black = lm(crim ~ black , data = boston)
#summary(lr_black)
lr_lstat = lm(crim ~ lstat , data = boston)
#summary(lr_lstat)
lr_medv = lm(crim ~ medv , data = boston)
#summary(lr_medv)
```

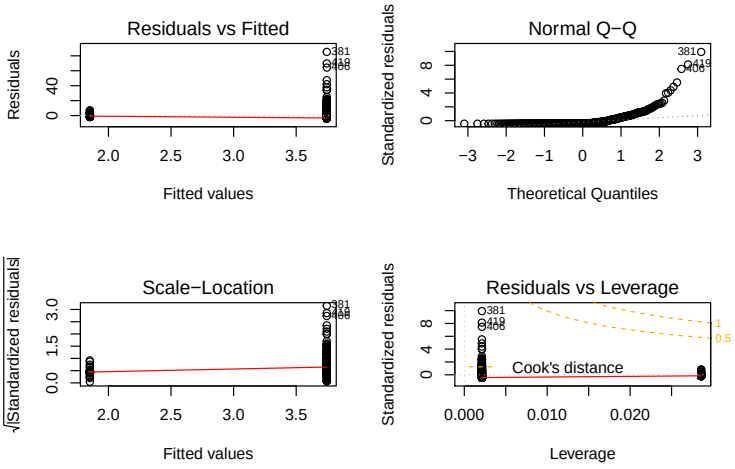
There is a statistically significant association between the predictor and the response for all variables except chas (Charles River dummy variable). All the variables have a low R-squared value therefore the changes in the response "crim" can only be explained by these predictors by a small value.

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



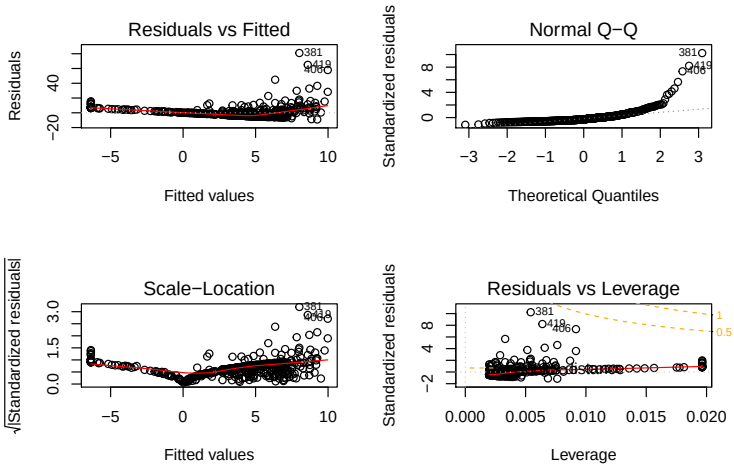
Residuals vs Fitted: The red line is horizontal which means that there is linear relationship between the predictor and response variables.
Normal Q-Q: Though most of the residuals follow a straight line, some points deviate from the straight line.
Scale - Location: The red line is almost horizontal which represents that the residuals are equally spread throughout the range of predictors
Residuals vs Leverage: There is no influential point

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



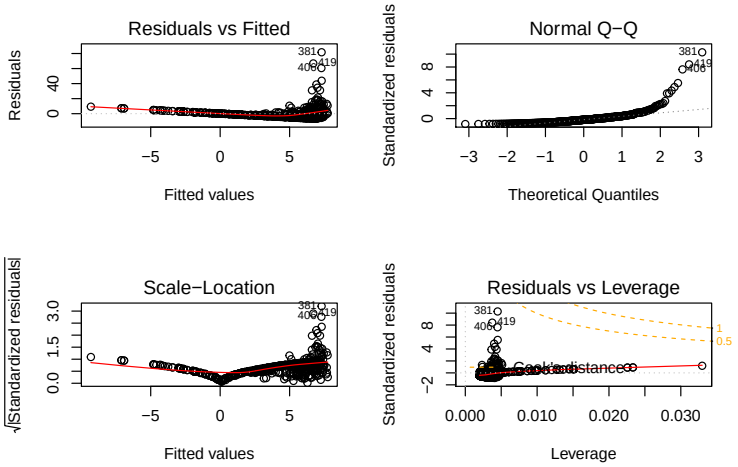
Residuals vs Fitted: The points are present at 2 ends which does not give any meaningful information.
Normal Q-Q: Though most of the residuals follow a straight line, some points deviate from the straight line.
Scale - Location: The points are present at 2 ends which does not give any meaningful information
Residuals vs Leverage: The points are present at 2 ends which does not give any meaningful information

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



Residuals vs Fitted: The red line is horizontal which means that there is linear relationship between the predictor and response variables.
Normal Q-Q: Though most of the residuals follow a straight line, some points deviate from the straight line.
Scale - Location: The red line is almost horizontal which represents that the residuals are equally spread throughout the range of predictors
Residuals vs Leverage: There is no influential point

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



Residuals vs Fitted: The red line is horizontal which means that there is linear relationship between the predictor and response variables.
Normal Q-Q: Though most of the residuals follow a straight line, some points deviate from the straight line.
Scale - Location: The red line is almost horizontal which represents that the residuals are equally spread throughout the range of predictors
Residuals vs Leverage: There is no influential point

```r
summary(lr_nox)$r.squared
```

```
## [1] 0.1772172
```

```r
summary(lr_chas)$r.squared
```

```
## [1] 0.003123869
```

```r
summary(lr_medv)$r.squared
```

```
## [1] 0.1507805
```

```r
summary(lr_dis)$r.squared
```

```
## [1] 0.1441494
```

The R-squared value for *nox* is higher when compared to *chas*, *medv*, *dis*

3. Fit a multiple regression model to predict the response using all of the predictor

```r
lr_all = lm(crim ~ . - crim, data = boston)
summary(lr_all)
```

```
##
## Call:
## lm(formula = crim ~ . - 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
```

Predictors dis and rad have high statistical significance at 0.001 level. Predictors medv has statistical significance at 0.01 level. Predictors black, zn have statistical significance at 0.05 level. Predictors nox, lstat have statistical significance at 0.1 level. R-squared value is higher for multiple regression when compared to the simple regression of individual predictors.

A small p-value (< 0.05) indicates strong evidence against the null hypothesis, so we can reject *dis, rad, medv, black, zn*. A large p-value (> 0.05) indicates weak evidence against the null hypothesis, so we do not reject those predictors. A p-value very close to the cutoff (0.05) is considered to be marginal and can go either way so we do not reject those predictors.

4. How do your results from (2) compare to your results from (3)? Create a plot displaying the univariate regression coefficients from (2) on the x-axis, and the multiple regression coefficients from (3) on the y-axis. That is, each predictor is displayed as a single point in the plot. Its coefficient in a simple linear regression model is shown on the x-axis, and its coefficient estimate in the multiple linear regression model is shown on the y-axis. What does this plot tell you about the various predictors?

```
lr_simple =  vector("numeric",0)
lr_simple = c(lr_simple, lr_zn$coefficient[2])
lr_simple = c(lr_simple, lr_indus$coefficient[2])
lr_simple = c(lr_simple, lr_chas$coefficient[2])
lr_simple = c(lr_simple, lr_nox$coefficient[2])
lr_simple = c(lr_simple, lr_rm$coefficient[2])
lr_simple = c(lr_simple, lr_age$coefficient[2])
lr_simple = c(lr_simple, lr_dis$coefficient[2])
lr_simple = c(lr_simple, lr_rad$coefficient[2])
lr_simple = c(lr_simple, lr_tax$coefficient[2])
lr_simple = c(lr_simple, lr_ptratio$coefficient[2])
lr_simple = c(lr_simple, lr_black$coefficient[2])
lr_simple = c(lr_simple, lr_lstat$coefficient[2])
lr_simple = c(lr_simple, lr_medv$coefficient[2])
lr_simple
```

```
##          zn       indus        chas         nox          rm         age
## -0.07393498  0.50977633 -1.89277655 31.24853120 -2.68405122  0.10778623
##         dis         rad         tax     ptratio       black       lstat
## -1.55090168  0.61791093  0.02974225  1.15198279 -0.03627964  0.54880478
##        medv
## -0.36315992
```

```
lr_multiple = vector("numeric", 0)
lr_multiple = c(lr_multiple, lr_all$coefficients)
lr_multiple
```

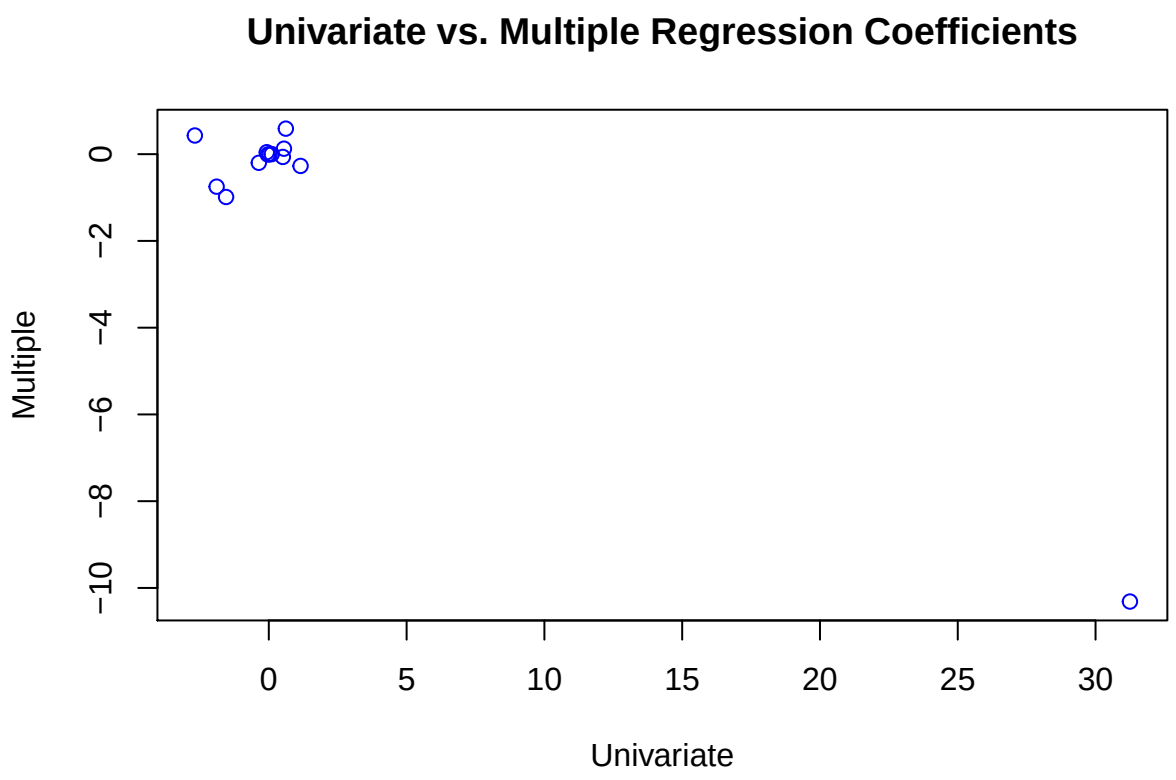
```
##   (Intercept)            zn         indus          chas           nox
##  17.033227523   0.044855215  -0.063854824  -0.749133611 -10.313534912
##            rm           age           dis           rad           tax
##   0.430130506   0.001451643  -0.987175726   0.588208591  -0.003780016
##       ptratio         black         lstat          medv
##  -0.271080558  -0.007537505   0.126211376  -0.198886821
```

```
lr_multiple = lr_multiple[-1] #To discard the intercept vale
lr_multiple
```

```
##            zn         indus          chas           nox            rm
##   0.044855215  -0.063854824  -0.749133611 -10.313534912   0.430130506
##           age           dis           rad           tax       ptratio
##   0.001451643  -0.987175726   0.588208591  -0.003780016  -0.271080558
```

```
##         black         lstat          medv
##  -0.007537505   0.126211376  -0.198886821
```

```
plot(lr_simple, lr_multiple, main = "Univariate vs. Multiple Regression Coefficients",
     xlab = "Univariate", ylab = "Multiple", col='blue')
```



When comparing the coefficients of simple regression and multiple regression we see a significant difference. This is because when we do simple regression we only consider the effect of increase of that predictor on the response "crim" while we ignore all the other predictors. But in multiple regression, we consider the average effect of increase of that predictor on the response "crim" while we keep all the other predictors as fixed values.

The plot shows that most of the multiple regression coefficients are 0 except one while the simple regression coefficients are more widely spread. This means that multiple regression coefficients show no relationship between most of the predictors and the response "crim" while simple regression coefficients show a stronger relationship between the predictors and the response "crim"

5. Is there evidence of non-linear association between any of the predictors and the response?

```
poly_zn = lm(crim ~ poly(zn, 3), data = boston)
#summary(poly_zn)
poly_indus = lm(crim ~ poly(indus, 3), data = boston)
#summary(poly_indus)
poly_chas = lm(crim ~ chas + I(chas^2) + I(chas^3), data = boston)
#summary(poly_chas)
poly_nox = lm(crim ~ poly(nox, 3), data = boston)
#summary(poly_nox)
poly_rm = lm(crim ~ poly(rm, 3), data = boston)
#summary(poly_rm)
poly_age = lm(crim ~ poly(age, 3), data = boston)
#summary(poly_age)
poly_dis = lm(crim ~ poly(dis, 3), data = boston)
#summary(poly_dis)
poly_rad = lm(crim ~ poly(rad, 3), data = boston)
#summary(poly_rad)
poly_tax = lm(crim ~ poly(tax, 3), data = boston)
#summary(poly_tax)
poly_ptratio = lm(crim ~ poly(ptratio, 3), data = boston)
#summary(poly_ptratio)
poly_black = lm(crim ~ poly(black, 3), data = boston)
#summary(poly_black)
poly_lstat = lm(crim ~ poly(lstat, 3), data = boston)
#summary(poly_lstat)
poly_medv = lm(crim ~ poly(medv, 3), data = boston)
#summary(poly_medv)
```

For chas predictor, squared and cubed terms contain NA. This is because chas only contains 0 and 1 which when we square or cube does not make any difference.

Predictors *nox, age, dis, tax, medv* have high statistical significance of 0.001 level for Quadratic terms. Predictors *zn, indus, rm, rad, ptratio, black* have statistical significance of 0.01 level for Quadratic terms. This is indicative of non-linear association between the above predictors and response "crim".

Predictors *indus, nox, dis, medv* have high statistical significance of 0.001 level for Cubic terms. Predictors *age, ptratio* have statistical significance of 0.01 level for Cubic terms. This is indicative of non-linear association between the above predictors and response "crim".

Question 3

1. What are the issues that could arise in using linear regression (via least squares estimates) when error terms are correlated? Comment in particular with respect to regression coefficients

Regression Coefficients gives the relation between the predictors and the response ie how much a response variable will increase/decrease when the predictor variable is increased/decreased. The coefficients of least square estimates can be found by finding numerical values that minimize the sum of the squared deviations between the observed responses and the functional portion of the linear regression model. When correlated errors are present then the regression coefficients become inefficient since they no longer have minimum variance.

2. What are the issues that could arise in using linear regression (via least squares estimates) when error terms are correlated? Comment in particular with respect to the standard error of regression coefficients

Standard error of regression coefficient is used to measure the accuracy of prediction ie it tells how precisely the regression model is able to estimate the unknown value of a coefficient. It is calculated as the square root of the average squared error of prediction. When correlated errors are present then the standard error of regression coefficients will be less when compared to the actual error. An incorrect inferenece on the accuracy will be made ie a low accuracy model will be projected as a high accuracy model.

3. What are the issues that could arise in using linear regression (via least squares estimates) when error terms are correlated? Comment in particular with respect to confidence intervals

The width of the confidence interval is used to measure the overall quality of the regression. It is the interval estimate between the independent variable and the mean of the dependent variable. When correlated errors are present then the width of the confidence interval will become narrower and this test of significance will become invalid since we will assume the model to be more confident than it actually is.

4. What methods can be applied to deal with correlated errors?

Correlation in the error terms means that there is extra information in the data that we have not included in the current regression model ie the errors follow a pattern. The assumption in least squares estimates is that the error term observations are independent of each other ie the error term observation must not be correlated with the error term observation that comes after it. If this is violated then it is called autocorrelation.

We can overcome this using the generalized least squares (GLS) estimates method in linear regression. Here we apply ordinary least squares to a linearly transformed version of the data which makes it more efficient and accurate when compared to an ordinary least square method. Another way to overcome correlated errors is to use lme() function (Linear Mixed-Effects Models) in R. It allows within-group errors to be correlated and/or have unequal variances.