CptS 575 Data Science: Assignment 5

Md Muhtasim Billah

10/27/2020

Question 1.

First, we load the "Auto" dataset from the class website and remove the missing values (if any).

```
#load the whole dataset as dataframe
Auto_full = read.csv(url("https://scads.eecs.wsu.edu/wp-content/uploads/2017/09/Auto.csv"),
                     stringsAsFactors = FALSE, na.strings = "?")
#check the dimensions
dim(Auto_full)
```

```
[1] 397   9
```

```
#remove the rows with missing values
Auto = na.omit(Auto_full)
#check the dimensions again
dim(Auto)
```

```
[1] 392   9
```

It was found that some values for the variable `horsepower` were `'?'` in the dataset which were removed. Checking the dimensions, it seems that there were `5` missing values in the dataset.

Now, we check the variable types to see if they are properly labelled. We notice

```
#check the varoable types/class
sapply(Auto, class)
```

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

```
#convert the origin variable from integer to factor
Auto$origin = as.factor(Auto$origin)
#check the variable types again
sapply(Auto, class)
```

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

1(a)

Multiple linear regression is performed with `mpg` as the response and all other variables (except `name`) as the predictors. A printout of the result is shown that includes coefficients, error and t-values for each predictor.

```
model_1a = lm(mpg ~ . - name, data = Auto)
summary(model_1a)
```

```
Call:
lm(formula = mpg ~ . - name, data = Auto)

Residuals:
    Min      1Q  Median      3Q     Max
-9.0095 -2.0785 -0.0982  1.9856 13.3608

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.795e+01  4.677e+00  -3.839 0.000145 ***
cylinders    -4.897e-01  3.212e-01  -1.524 0.128215
displacement  2.398e-02  7.653e-03   3.133 0.001863 **
horsepower   -1.818e-02  1.371e-02  -1.326 0.185488
weight       -6.710e-03  6.551e-04 -10.243  < 2e-16 ***
acceleration  7.910e-02  9.822e-02   0.805 0.421101
year          7.770e-01  5.178e-02  15.005  < 2e-16 ***
origin2       2.630e+00  5.664e-01   4.643 4.72e-06 ***
origin3       2.853e+00  5.527e-01   5.162 3.93e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.307 on 383 degrees of freedom
Multiple R-squared:  0.8242,    Adjusted R-squared:  0.8205
F-statistic: 224.5 on 8 and 383 DF,  p-value: < 2.2e-16
```

i)

Based on a significance level $\alpha = 0.05$, the intercept as well as some of the variables such as `displacement`, `weight`, `year` and `origin` seem to have significant relationship to the response `mpg` as designated by their p-value. It is also visible that, the variables `weight` and `year` have the most significant relationship since they have extremely small ($< 2e^{-16}$) p-value.

ii)

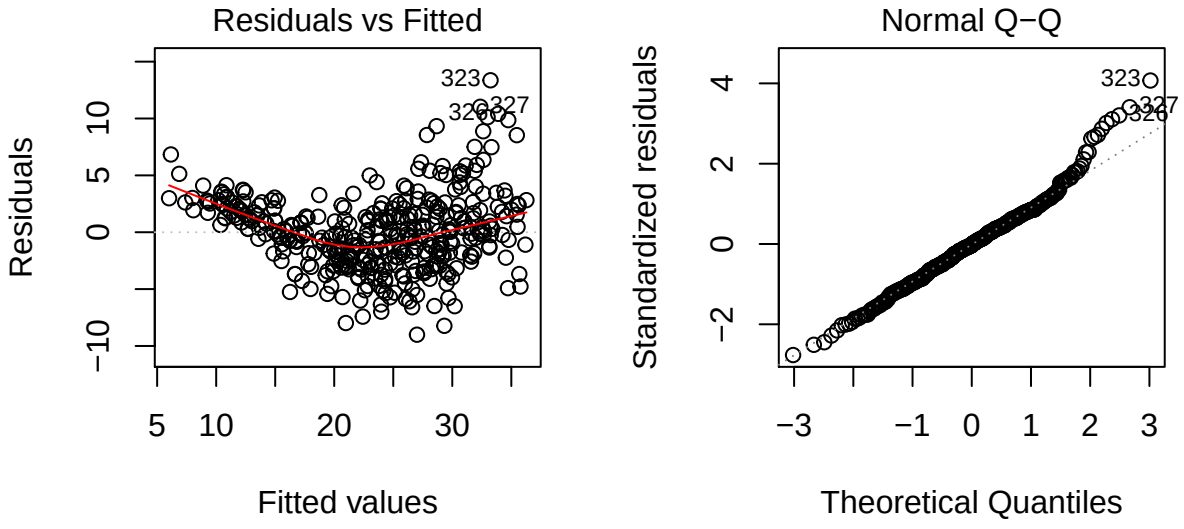
The coefficient for the `displacement` variable is found to be $2.398e^{-02}$ which indicates a positive relationship with the response. This value indicates that for a unit increase of `displacement`, the response `mpg` will be increased by the amount of $2.398e^{-02}$.

1(b)

Diagnostics plots indicate the validity of the assumptions essential for performing linear regression. Two primary assumptions are the normality of the residuals and the constant variance (homoschedasticity) of the

residuals.

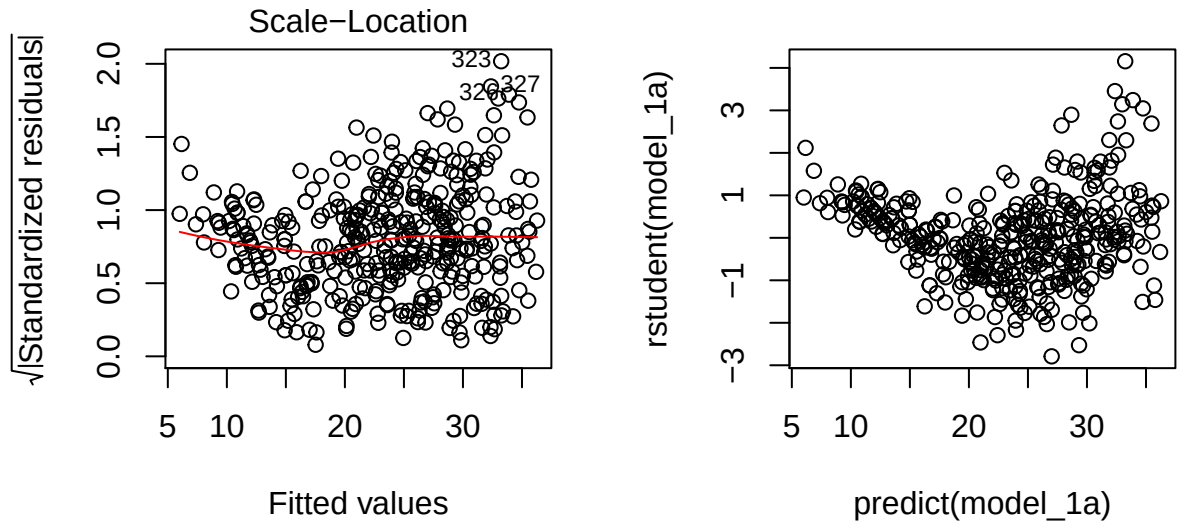
```
par(mfrow=c(1,2))
plot(model_1a,1:2)
```



From the residual vs fitted plot (left), there seems to be a slight nonlinear pattern (quadratic) among the data points rather than random, scattered distribution which indicates that the assumption of constant variance might be violated. From the normal Q-Q plot, it seems that the residuals tend to deviate from the line at the upper tail which indicates that the distribution of the residuals might not be normal. Since diagnostic plots sometimes can be misleading, for certainty, more robust and reliable statistical tests can be further performed.

To check if there are any unusually large outliers, we can further plot the standardized and studentized residuals against the predicted (fitted values) as below.

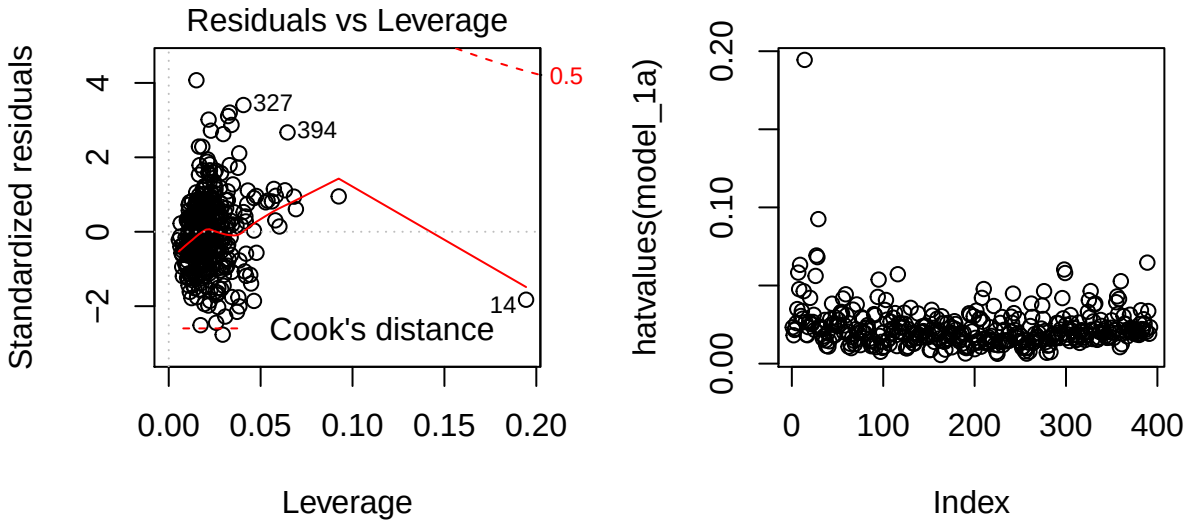
```
par(mfrow=c(1,2))
plot(model_1a,3)
plot(predict(model_1a), rstudent(model_1a))
```



From the above plots, it seems that some of the residuals are have pretty large values. However, none of them are unusually high and thus can not be considered as huge outliers.

To identify any observations with unusually high leverage, the residuals vs leverage plot and the hat values of the observations can be plotted as below.

```
par(mfrow=c(1,2))
plot(model_1a,5)
plot(hatvalues(model_1a))
```



From the above plot, it seems that observation number 14 has unusually high leverage. The hat value indicates how much the predicted scores will change if this observation is excluded from the dataset. This can also be verified as below.

```
which.max(hatvalues(model_1a))
```

```
14
14
```

1(c)

The multiple linear regression model is fit for all the predictors (except `name`) with interaction effects (only pairwise interactions). The results are summarized below.

```
model_1c = lm(mpg ~ (. - name)^2, data = Auto)
summary(model_1c)
```

```
Call:
lm(formula = mpg ~ (. - name)^2, data = Auto)

Residuals:
    Min      1Q  Median      3Q     Max
-7.6008 -1.2863  0.0813  1.2082 12.0382

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)                4.401e+01  5.147e+01   0.855 0.393048
cylinders                  3.302e+00  8.187e+00   0.403 0.686976
displacement              -3.529e-01  1.974e-01  -1.788 0.074638 .
horsepower                 5.312e-01  3.390e-01   1.567 0.117970
weight                    -3.259e-03  1.820e-02  -0.179 0.857980
acceleration              -6.048e+00  2.147e+00  -2.818 0.005109 **
year                       4.833e-01  5.923e-01   0.816 0.415119
origin2                   -3.517e+01  1.260e+01  -2.790 0.005547 **
origin3                   -3.765e+01  1.426e+01  -2.640 0.008661 **
cylinders:displacement    -6.316e-03  7.106e-03  -0.889 0.374707
cylinders:horsepower       1.452e-02  2.457e-02   0.591 0.555109
cylinders:weight           5.703e-04  9.044e-04   0.631 0.528709
cylinders:acceleration     3.658e-01  1.671e-01   2.189 0.029261 *
cylinders:year            -1.447e-01  9.652e-02  -1.499 0.134846
cylinders:origin2         -7.210e-01  1.088e+00  -0.662 0.508100
cylinders:origin3          1.226e+00  1.007e+00   1.217 0.224379
displacement:horsepower   -5.407e-05  2.861e-04  -0.189 0.850212
displacement:weight        2.659e-05  1.455e-05   1.828 0.068435 .
displacement:acceleration -2.547e-03  3.356e-03  -0.759 0.448415
displacement:year          4.547e-03  2.446e-03   1.859 0.063842 .
displacement:origin2      -3.364e-02  4.220e-02  -0.797 0.425902
displacement:origin3       5.375e-02  4.145e-02   1.297 0.195527
horsepower:weight         -3.407e-05  2.955e-05  -1.153 0.249743
horsepower:acceleration   -3.445e-03  3.937e-03  -0.875 0.382122
horsepower:year           -6.427e-03  3.891e-03  -1.652 0.099487 .
horsepower:origin2        -4.869e-03  5.061e-02  -0.096 0.923408
horsepower:origin3         2.289e-02  6.252e-02   0.366 0.714533
weight:acceleration       -6.851e-05  2.385e-04  -0.287 0.774061
weight:year               -8.065e-05  2.184e-04  -0.369 0.712223
weight:origin2             2.277e-03  2.685e-03   0.848 0.397037
weight:origin3            -4.498e-03  3.481e-03  -1.292 0.197101
acceleration:year          6.141e-02  2.547e-02   2.412 0.016390 *
acceleration:origin2       9.234e-01  2.641e-01   3.496 0.000531 ***
acceleration:origin3       7.159e-01  3.258e-01   2.198 0.028614 *
year:origin2               2.932e-01  1.444e-01   2.031 0.043005 *
year:origin3               3.139e-01  1.483e-01   2.116 0.035034 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.628 on 356 degrees of freedom
Multiple R-squared:  0.8967,    Adjusted R-squared:  0.8866
F-statistic: 88.34 on 35 and 356 DF,  p-value: < 2.2e-16
```

From the model summary, some of the interactions appear to be statistically significant (at a significance level $\alpha = 0.05$) such as `cylinders:acceleration`, `acceleration:year`, `acceleration:origin2`, `acceleration:origin3`, `year:origin2` and `year:origin3`.

Question 2.

2(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.

First, load the `Boston` dataset and attach it to the kernel.

```
#load the library
library(MASS)
#attach Boston data
attach(Boston)
```

Now, for each predictor, a simple linear regression model is fitted to predict the response.

```
lr_zn = lm(crim ~ zn, data = Boston)
#summary(lr_zn)
lr_indus = lm(crim ~ indus, data = Boston)
#summary(lr_indus)
#the type for the variable chas should be a factor
Boston$chas = as.factor(Boston$chas)
lr_chas = lm(crim ~ chas, data = Boston)
#summary(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(lr_age)
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)
```

2(b)

Looking at the summary output from all the models, it is found that, apart from the predictor `chas`, all the other predictors have statistically significant association with the response `crim`.

The meaning of the response variable `crim` and predictors `nox`, `chas`, `medv` and `dis` are as below.

crim: Per capita crime rate by town.
nox: Nitrogen oxides concentration (parts per 10 million).
chas: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise).
medv: Median value of owner-occupied homes in $1000s.
dis: Weighted mean of distances to five Boston employment centres.

From the summary output of the individual models, the relationship between the response and the aforementioned predictors are discussed below.

crim vs nox:

The predictor `nox` has statistically significant relationship (`p-value<2e-16`) with the response variable `crim` and the relation is positive as seen from its coefficient value `31.249`. This indicates that, for a unit increase in the Nitrogen Oxides concentration (parts per 10 million), the per capita crime rate will increase by a value of `31.249` which seems very unreasonable since there is no plausible explanation for this. This model is a classic example of the phrase "correlation doesn't necessarily mean causation". Also, the R-squared values for this model is roughly `17%` which means that very little variance in the response was explained by this model and more predictors are required for a better prediction.

crim vs chas:

The factor predictor `chas` doesn't have a statistically significant relationship (`p-value=0.209`) with the response variable `crim` and the relation is negative as seen from its coefficient value `-1.8928`. This indicates that, if tract bounds river, the per capita crime rate will decrease by a value of `1.8928`. Though there is a significant relation, there is no apparent explanation for this to happen. This is not a very reliable estimate of the response also because the R-squared values for this model is extremely small (roughly `0.31%`) which means that very little variance in the response was explained by this model and more predictors are required for a better prediction.

crim vs medv:

The predictor `medv` has statistically significant relationship (`p-value<2e-16`) with the response variable `crim` and the relation is negative as seen from its coefficient value `-0.36316`. This indicates that, for a unit increase in the median value of owner-occupied homes (in $1000s), the per capita crime rate will decrease by a value of `0.36316`. While this makes sense since increased housing price in a locality should indicate to a better lifestyle, for better prediction of the response, more predictors are necessary. This is also seen from the R-squared values for this model which is very small (roughly `15%`).

crim vs dis:

The predictor `dis` has statistically significant relationship (`p-value<2e-16`) with the response variable `crim` and the relation is negative as seen from its coefficient value `-1.5509`. This indicates that, for a unit increase in the weighted mean of distances to five Boston employment centres, the per capita crime rate will decrease by a value of `1.5509`. This might be reasonable in the sense that, with longer commute to the workplace, there are more chances for crimes to happen. Nonetheless, for better prediction of the response, more predictors are necessary as indicated by the R-squared values for this model which is very small (roughly `14%`).

How these relationships differ from one another is that first, only the first predictor seem to increase the per capita crime rate where the rest seem to cause a decrease. Second, apart from `chas`, the rest of the predictors has statistically significant relationship with the response. However, the high correlations of the predictor `nox` cannot explain the causation while `medv` and `dis` can do that to some extent. However, for all the models, the R-squared and adjusted R-squared values are very small which indicates to the necessity of a multiple linear regression model for explaining the response.

2(c)

A multiple linear regression is performed on the Boston dataset which includes all the predictors altogether.

```
#regression with all the predictors
mlr = lm(crim ~ . , data=Boston)
summary(mlr)
```

```
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
chas1        -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
```

The summary output indicates towards the model's high significance given by its large F-statistic value and an extremely low p-value. Also, from the model R-squared values, it is evident that this model can explain approximately `45%` of the variance of the response. This value is significantly higher from the individual simple linear regression models, however is not high enough since most of the variance remains unexplained.

From the model summary, it is evident that at level of significance $\alpha = 0.05$, the predictors `zn`, `dis`, `rad`, `black` and `medv` have strong relationship with the response variable `crim`. So, for these predictors, we can reject the null hypothesis ($\beta_0 = 0$). The predictor `nox` might also be considered to be significant since it has a marginal p-value (`0.051152`). However, the rest of the predictor doesn's seem to be statistically significant.

2(d)

First, a vector is created taking all the coefficients from the individual models.

```
slr_coef = c(lr_zn$coefficients[2],lr_indus$coefficients[2],lr_chas$coefficients[2],
             lr_nox$coefficients[2],lr_rm$coefficients[2],
             lr_age$coefficients[2],lr_dis$coefficients[2],
             lr_rad$coefficients[2],lr_tax$coefficients[2],
             lr_ptratio$coefficients[2],lr_black$coefficients[2],
             lr_lstat$coefficients[2],lr_medv$coefficients[2])
round(slr_coef,3)
```

```
     zn   indus    chas1     nox      rm     age     dis     rad     tax ptratio
 -0.074   0.510  -1.893  31.249  -2.684   0.108  -1.551   0.618   0.030   1.152
  black   lstat    medv
 -0.036   0.549  -0.363
```

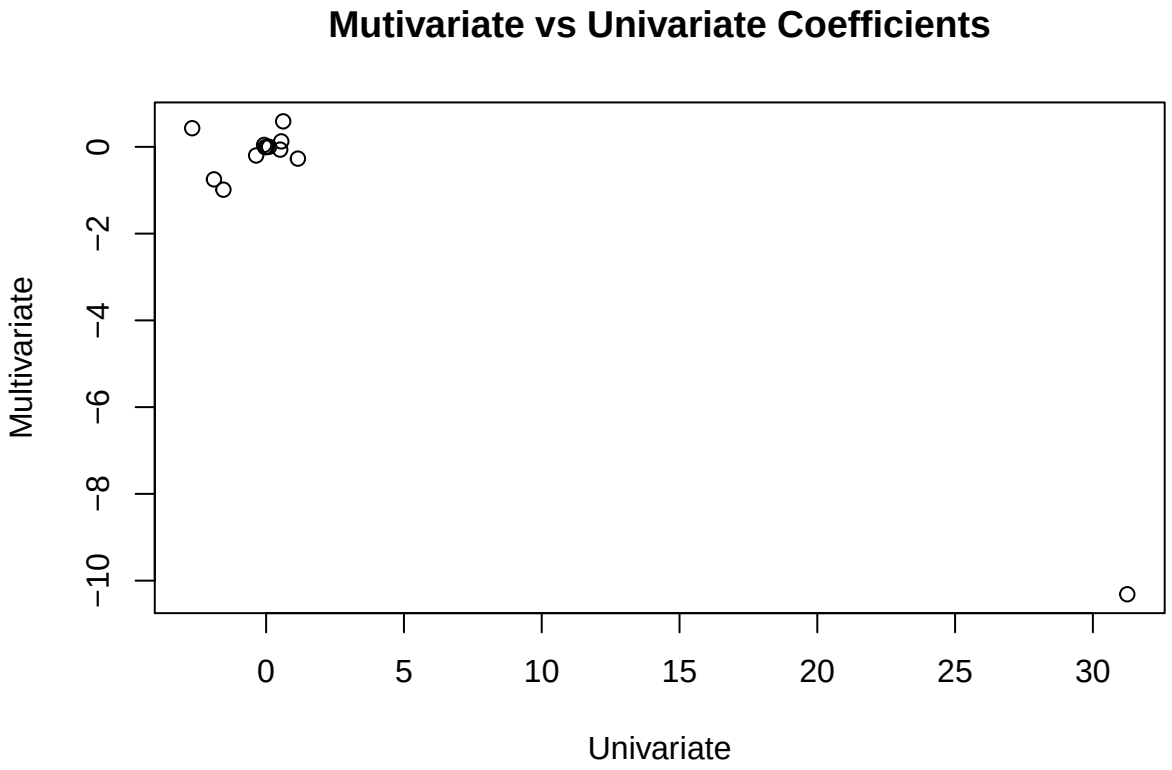
Next, another vector is created taking the coefficients (except the intercept) from the multiple linear regression model.

```
mlr_coef = c(mlr$coefficients)
mlr_coef = mlr_coef[-1]
round(mlr_coef,3)
```

```
     zn   indus    chas1     nox      rm     age     dis     rad     tax ptratio
  0.045  -0.064  -0.749 -10.314   0.430   0.001  -0.987   0.588  -0.004  -0.271
  black   lstat    medv
 -0.008   0.126  -0.199
```

Now, a plot is created displaying the univariate regression coefficients from (a) on the x-axis, and the multiple regression coefficients from (c) on the y-axis as below.

```
plot(slr_coef,mlr_coef,main = "Mutivariate vs Univariate Coefficients",
     xlab = "Univariate", ylab = "Multivariate")
```



Looking at the values of the coefficients as well the graph plotted above, we can see significant difference between the results from (a) and (c). The coefficients for the individual models span over a very wide range (starting from `-2.684` to `31.249`) while the coefficients from the multiple regression model span over a much shorter range (starting from `-0.987` to `0.588`). This indicates that the univariate coefficients have much stronger effect on the response while considered individually but together, the effect is minimized.

2(e)

```
nlr_zn = lm(crim ~ poly(zn,3), data = Boston)
summary(nlr_zn)
nlr_indus = lm(crim ~ poly(indus,3), data = Boston)
summary(nlr_indus)
#the type for the variable chas should be a factor
Boston$chas = as.factor(Boston$chas)
nlr_chas = lm(crim ~ chas + I(chas^2) + I(chas^3), data = Boston)
summary(nlr_chas)
nlr_nox = lm(crim ~ poly(nox,3), data = Boston)
summary(nlr_nox)
nlr_rm = lm(crim ~ poly(rm,3), data = Boston)
summary(nlr_rm)
nlr_age = lm(crim ~ poly(age,3), data = Boston)
summary(nlr_age)
nlr_dis = lm(crim ~ poly(dis,3), data = Boston)
summary(nlr_dis)
nlr_rad = lm(crim ~ poly(rad,3), data = Boston)
summary(nlr_rad)
nlr_tax = lm(crim ~ poly(tax,3), data = Boston)
```

```
summary(nlr_tax)
nlr_ptratio = lm(crim ~ poly(ptratio,3), data = Boston)
summary(nlr_ptratio)
nlr_black = lm(crim ~ poly(black,3), data = Boston)
summary(nlr_black)
nlr_lstat = lm(crim ~ poly(lstat,3), data = Boston)
summary(nlr_lstat)
nlr_medv = lm(crim ~ poly(medv,3), data = Boston)
summary(nlr_medv)
```

For the dummy variable `char`, higher order non-linearity of the given form doesn't mean anything since the squared and cubic values for 0 and 1 contains NA values.

At a level of significance, $\alpha = 0.05$, the **quadratic term** of all the predictors apart from `black` has statistical significane which indicates that a non-linear, quadratic model can be considered for making better prediction of the response.

At a level of significance, $\alpha = 0.05$, the **cubic term** of the predictors `indus`, `nox`, `age`, `dis`, `ptratio` and `medv` have statistical significane which indicates that a non-linear, cubic model can be considered for making better prediction of the response.

Question 3.

3(a)

Given, the variables,

$X1$ = Hours studied,
$X2$ = Undergrad GPA,
$X3$ = PSQI score (a sleep quality index), and
Y = Receive an A.

And, the estimated coefficient from the logistic regression are,

$\beta_0 = -7$
$\beta_1 = 0.1$
$\beta_2 = 1$
$\beta_3 = -0.04$

From logistic regression, the probability for a student to receive an A can be expressed as follows.

$$\hat{P}(Y = 1 \mid X_1, X_2, X_3) = \hat{P}(X) = \frac{e^{\hat{\beta_0}+\hat{\beta_1}X_1+\hat{\beta_2}X_2+\hat{\beta_3}X_3}}{1 + e^{\hat{\beta_0}+\hat{\beta_1}X_1+\hat{\beta_2}X_2+\hat{\beta_3}X_3}}$$

From the above equation, the probability, $Y = \hat{P}(X)$ that a student who studies for $X_1 = 32$ h, has a PSQI score of $X_3 = 12$ and has an undergrad GPA of $X_2 = 3.0$ gets an A in the class can be calculated.

$$Y = \hat{P}(X) = \frac{e^{-7 \ + \ 0.1\times32 \ + \ 1\times3.0 \ + \ (-0.04)\times12}}{1 + e^{-7 \ + \ 0.1\times32 \ + \ 1\times3.0 \ + \ (-0.04)\times12}}$$

Here,

$$-7 \ + \ 0.1 \times 32 \ + \ 1 \times 3.0 \ + \ (-0.04) \times 12 = -1.28$$

Thus, the probablity of getting an A is,

$$Y = \frac{e^{-1.28}}{1 + e^{-1.28}} = \frac{0.278}{1 + 0.278} = 0.2175$$

3(b)

Another form of the logistic regression equation is as follows,

$$\log \frac{\hat{P}(X)}{1 - \hat{P}(X)} = \hat{\beta_0} + \hat{\beta_1}X_1 + \hat{\beta_2}X_2 + \hat{\beta_3}X_3$$

Here, the chance for getting an A,

$$\hat{P}(X) = Y = 0.5$$

And,

$$\hat{\beta_0} + \hat{\beta_1} X_1 + \hat{\beta_2} X_2 + \hat{\beta_3} X_3 = -7 \ + \ 0.1 \times X_1 \ + \ 1 \times 3.0 \ + \ (-0.04) \times 12 = 0.1 X_1 - 4.48$$

Thus,

$$\log \frac{0.5}{1 - 0.5} = 0.1 X_1 - 4.48$$

or,

$$\log(1) = 0.1 X_1 - 4.48$$

or,

$$X_1 = 44.8$$

Thus, the student in part (a) needs to study for 44.8 hours to have a 50% chance of getting an A in the class.

3(c)

Here, the chance for getting an A,

$$\hat{P}(X) = Y = 0.5$$

And,

$$\hat{\beta_0} + \hat{\beta_1} X_1 + \hat{\beta_2} X_2 + \hat{\beta_3} X_3 = -7 \ + \ 0.1 \times X_1 \ + \ 1 \times 3.0 \ + \ (-0.04) \times 3 = 0.1 X_1 - 4.12$$

Thus,

$$\log \frac{0.5}{1 - 0.5} = 0.1 X_1 - 4.12$$

or,

$$\log(1) = 0.1 X_1 - 4.12$$

or,

$$X_1 = 41.2$$

Thus, a student with a 3.0 GPA and a PSQI score of 3 will need to study for 41.2 hours to have a 50% chance of getting an A in the class.

Question 4.

First, we load the data and the necessary packages.

```
#load necessary libraries
library(GuardianR)
library(plyr)
library(dplyr)
library(curl)
#library(RCurl)
library(quanteda)
#load the data
articles=read.csv("/Users/muhtasim/Desktop/GuardianArticles.csv",
                  stringsAsFactors=FALSE, fileEncoding="latin1")
```

We remove any articles with zero word count and then find the counts for different types of articles.

```
articles = na.omit(articles)
count = articles %>%
        group_by(section) %>%
        tally
count
```

```
# A tibble: 6 x 2
  section          n
  <chr>        <int>
1 artanddesign  1998
2 business      1849
3 culture       1884
4 sport         1771
5 technology    1985
6 world         1971
```

Then, we check the `body`, `section` and `title` of the news for anomalies.

```
library(stringi)
#check the body, section and title of the news for anomalies
#title
print(articles$title[1], max.levels = 0)
```

```
[1] "Norman Foster's Bloomberg office in London wins Stirling prize"
```

```
#section
print(articles$section[1], max.levels = 0)
```

```
[1] "artanddesign"
```

```
#body
cat(stri_pad(stri_wrap(articles$body[1]),side = 'right'),sep = '\n')
```

one of the most environmentally friendly office buildings ever conceived has been named the winner of the 2018 stirling prize beating off competition from a quirky brick nursery a mudwalled cemetery and the extension of the tate st ives gallery the 1bn european headquarters for bloomberg designed by foster partners stands in the city of london as a gargantuan temple to gadgetry its every detail subjected to months of research and development as the ribas president ben derbyshire put it the architects have not just raised the bar for office design and city planning but smashed the ceiling the 1 million sq ft complex is kitted out with vacuumflush toilets breathing gills in the facade a petalstudded ceiling that provides lighting cooling and acoustic attenuation and a host of other features that mean it should use 70 less water and 40 less energy than a typical office block it is the kind of noexpensespared statement you might expect from its billionaire client michael bloomberg the former mayor of new york the author of climate of hope and the boss of this financial software data and media empire it is an elaborate monument to his belief that businesses have a role to play in saving the planet as grand corporate gestures go it does its best to be demure many companies of our size would have opted for a glass skyscraper bloomberg said at the buildings opening but we place value on being good neighbours we are conscious of the fact we are guests in london an earlier scheme for the site designed by foster for a different client imagined a dark glass tower rising to 22 storeys capped with a domed hat earning it the nickname darth vaders helmet bloombergs vision is a squatter affair of two 10storey blocks either side of a new arcade lined with food outlets selected by bloombergs food critic connected by a bridge and dressed in a polite costume of sandstone and bronze standing across from fosters bulbous glass walbrook building it shows just how willing the architect is to turn his hand to whatever the client wants the exterior of the bloomberg headquarters in london photograph alamy the visitor experience is meticulously choreographed like few other corporate hqs you arrive into the vortex a lobby of swirling timber shells with a feeling of walking into a richard serra sculpture from where cantilevered glass lifts whisk you up to the sixthfloor pantry a staff breakout area conceived on the scale of an airport terminal this voluminous greeting relaxing collaborating space is flanked by fish tanks and vast windows framing views of st pauls cathedral as if captured in a vitrine for the 4000 bloomberg staff who work here 700 to a floor employees fields of flexible workstations are connected by a great bronze ramp that coils through the floors providing a possibility for chance encounter where workers might exchange bon mots of financial reporting on their way to the wellness centre it is a hive of collaboration and teamwork illuminated by a dazzling ceiling of 25m folded aluminium petals that twinkle with led lights adding an air of bloombergian razzmatazz to the corporate scene secreted voice lift microphones in the auditorium and meeting rooms isolate and amplify speakers voices so that everyone may be heard while the oak floorboards are fixed to magnetic plates so they can be lifted to access the services below the big gills serve to draw in natural air with flaps that open and close automatically allowing the building to breathe while softening traffic noise from outside a great bronze ramp coils through the floors providing a possibility for chance encounters photograph alamy all of these innovations underwent relentless refinement and optimisation at a dedicated test facility in battersea where the natural

```
ventilation system was modelled with water by fluiddynamics experts and
11 mockups regularly examined by bloomberg himself in a rare level of
handson involvement some people say the reason it took almost a decade
to build this is because we had a billionaire who wanted to be an
architect quipped bloomberg working with an architect who wanted to be
a billionaire but for all their trailblazing innovations it is hard to
escape the feeling of being trapped in a very deepplan office building
very far from a window with views to the outside world often obscured by
the big bronze baffles it is a very inwardlooking place lest employees
be distracted from their duties to planet bloomberg the incorporation
of the roman temple of mithras in the basement might make some uneasy
about the companys ambitious level of stewardship while the 600 tonnes
of bronze imported from japan and the quarryfull of granite from india
also make you wonder about the projects true environmental credentials
related reconstructed roman temple of mithras opens to public in
london it marks the third time that norman fosters practice has won the
stirling prize having previously taken the accolade for the elliptical
hangar of the imperial war museum duxford in 1998 and the magnificent
gherkin in 2004 bloomberg was the favourite to win at 31 odds and it
shouldnt be a surprise the prize is explicitly awarded to the building
that makes the biggest contribution to the evolution of architecture
and fosters hightech innovations will no doubt influence the future of
office design the bloomberg european headquarters triumphed over the
tates magical new gallery in cornwall busheys jewish cemetery some fine
student accommodation for the university of roehampton a jewellike arts
building for worcester college oxford and my favourite an inventive
nursery and community hall in cambridge by muma that does a lot with a
modest budget it couldnt be further than last years winner hastings pier
a communityled project that tragically went into administration shortly
after the prize was awarded and was sold to a private hotelier for a
fraction of the cost of its renovation
```

From the randomly selected article body, it seems like there is no need for further cleaning the data. Now that we are confident about the cleanliness of the data, we will do the rest of the procedures.

4(a)

First, we will create the word corpus from all the articles and then perform tokenization and stemming on the corpus. Then we creathe the document-feature matrix.

```
#create corpus
corpus = corpus(articles$body)
#create token
tokens = tokens(corpus)
#removing punctuations and numbers from the tokens
tokens = tokens(tokens, remove_punct = TRUE, remove_numbers = TRUE)
#stemming the token
tokens = tokens_wordstem(tokens)
#creating the document-features matrix
dfm = dfm(tokens, remove = stopwords())
```

Now, we will keep only the words which appear in more than 10% of the documents by trimming the dfm.

```
# we keep only words occurring frequently (in more than 10% fo the documents)
feature_matrix = dfm_trim(dfm, min_docfreq = 100, min_termfreq = 0.1, termfreq_type = "quantile")
feature_matrix
```

```
Document-feature matrix of: 11,458 documents, 4,503 features (94.7% sparse).
       features
docs    one environment friend offic build ever conceiv name winner prize
  text1   1           2      1     5     8    1       2    1      2     4
  text2   2           0      1     0     0    0       0    0      0     0
  text3   3           0      0     0     0    0       1    0      0     0
  text4   2           0      0     0     0    0       0    1      0     0
  text5   4           0      0     0     1    1       0    0      0     0
  text6   5           0      0     0     0    0       0    0      0     0
[ reached max_ndoc ... 11,452 more documents, reached max_nfeat ... 4,493 more features ]
```

Now, we will print the features of a random article along with the non-zero entries of its feature vector. We select article 1375.

```
#selecting a random article, article#1375
random_doc = as.data.frame(feature_matrix[1375,])
randoc = random_doc[, random_doc != 0]
nonzero = t(randoc)
nonzero
```

```
            [,1]
doc_id      "text1375"
competit    "2"
european    "2"
everi       "1"
just        "1"
plan        "1"
million     "3"
provid      "2"
mean        "1"
boss        "1"
busi        "2"
go          "1"
doe         "1"
compani     "1"
size        "1"
said        "5"
place       "1"
site        "1"
rise        "1"
like        "1"
staff       "2"
view        "1"
way         "1"
well        "1"
ad          "1"
may         "1"
can         "1"
peopl       "1"
```

```
say          "1"
far          "1"
world        "1"
make         "1"
also         "1"
relat        "1"
time         "1"
taken        "1"
last         "1"
administr    "1"
cost         "4"
produc       "2"
ship         "2"
challeng     "1"
form         "1"
onli         "2"
look         "2"
befor        "1"
told         "1"
certain      "2"
someth       "1"
back         "1"
move         "3"
huge         "1"
part         "5"
right        "1"
come         "4"
bit          "1"
get          "1"
live         "1"
theyr        "1"
death        "1"
call         "1"
cours        "1"
instead      "1"
point        "1"
week         "3"
job          "1"
strike       "1"
day          "2"
largest      "1"
deal         "1"
term         "2"
pass         "1"
product      "8"
pose         "1"
behind       "1"
land         "3"
negat        "1"
reli         "1"
appli        "1"
despit       "1"
end          "1"
run          "1"
```

```
global      "1"
guardian    "1"
follow      "1"
four        "1"
flow        "1"
countri     "2"
govern      "2"
announc     "1"
europ       "3"
employ      "1"
continu     "1"
compon      "1"
repeat      "1"
argu        "1"
sometim     "1"
bbc         "1"
support     "1"
industri    "2"
compet      "1"
sign        "1"
probabl     "1"
organis     "1"
keep        "1"
away        "1"
today       "1"
brexit      "6"
concern     "1"
uk          "8"
eu          "3"
document    "1"
plant       "3"
wed         "3"
cut         "1"
built       "1"
car         "6"
potenti     "3"
daili       "1"
factori     "2"
reach       "2"
sale        "1"
lost        "1"
reduc       "1"
invest      "1"
pound       "1"
higher      "1"
oper        "1"
difficult   "1"
sector      "2"
agreement   "1"
vehicl      "1"
choos       "1"
thousand    "2"
condit      "1"
japanes     "2"
```

```
network      "1"
requir       "1"
damag        "2"
major        "1"
deliveri     "1"
fail         "2"
west         "1"
commit       "1"
ten          "4"
britain      "1"
grappl       "1"
decis        "1"
theresa      "1"
nodeal       "4"
warn         "3"
impact       "2"
manufactur   "4"
civic        "1"
averag       "1"
hit          "1"
castl        "1"
addit        "4"
negoti       "1"
email        "1"
owner        "1"
trade        "1"
caus         "1"
twitter      "1"
midland      "1"
prospect     "1"
truck        "1"
radio        "1"
brussel      "1"
risk         "1"
basi         "1"
ian          "1"
slump        "1"
businessdesk "1"
tariff       "6"
multin       "1"
```

4(b)

Now, we will apply Naive Bayes for classification of the articles into 6 classes or categories.

```
#load required packages
library(caret)
library(e1071)
#finding the correlation matrix and
#removing highly correlated features from the matrix
matrix = as.matrix(feature_matrix)
cor_Matrix = cor(matrix)
```

```
#store the number of rows of the matrix
matrix_rows = nrow(matrix)
matrix_rows
```

```
[1] 11458
```

```
cor_indices = findCorrelation(cor_Matrix, cutoff = 0.90)
matrix = matrix[, -c(cor_indices)]
```

```
#Splitting data into train and test set
num_train = (80/100)*matrix_rows
train_data = matrix[1:num_train,]
test_data = matrix[(num_train+1):matrix_rows,]
#labels
train_label = articles[1:num_train,]$section
test_label = articles[(num_train+1):matrix_rows,]$section
#classify using Naive Bayes
classifier = naiveBayes(train_data, as.factor(train_label) )
#make predictions
prediction = predict(classifier, test_data)
#generate the confusion matrix
confusion_matrix = confusionMatrix(prediction,as.factor(test_label))
confusion_matrix
```

```
Confusion Matrix and Statistics

              Reference
Prediction     artanddesign business culture sport technology world
  artanddesign          327        3     108     5          7    17
  business               15      330      18     6         68    56
  culture                27        2     200    10         10    13
  sport                  13        4      38   320          8    22
  technology              9       21       8     0        292    25
  world                   9        7      10     5         14   264

Overall Statistics

               Accuracy : 0.7564
                 95% CI : (0.7383, 0.7739)
    No Information Rate : 0.1746
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7079

 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: artanddesign Class: business Class: culture
Sensitivity                       0.8175          0.8992         0.5236
Specificity                       0.9260          0.9153         0.9675
Pos Pred Value                    0.7002          0.6694         0.7634
```

```
Neg Pred Value                    0.9600          0.9794         0.9103
Prevalence                        0.1746          0.1602         0.1667
Detection Rate                    0.1427          0.1440         0.0873
Detection Prevalence              0.2038          0.2152         0.1144
Balanced Accuracy                 0.8717          0.9072         0.7455
                     Class: sport Class: technology Class: world
Sensitivity                0.9249            0.7318       0.6650
Specificity                0.9563            0.9667       0.9762
Pos Pred Value             0.7901            0.8225       0.8544
Neg Pred Value             0.9862            0.9447       0.9329
Prevalence                 0.1510            0.1742       0.1733
Detection Rate             0.1397            0.1275       0.1152
Detection Prevalence       0.1768            0.1550       0.1349
Balanced Accuracy          0.9406            0.8493       0.8206
```

Now, calculate the precision of our prediction.

```
 #precision for each class
precision = confusion_matrix$byClass[,5]
precision
```

```
Class: artanddesign     Class: business      Class: culture        Class: sport
          0.7002141           0.6693712           0.7633588           0.7901235
  Class: technology        Class: world
          0.8225352           0.8543689
```

Finally, we calculate the recall of the prediction.

```
#recall
recall = confusion_matrix$byClass[, 6]
recall
```

```
Class: artanddesign     Class: business      Class: culture        Class: sport
          0.8175000           0.8991826           0.5235602           0.9248555
  Class: technology        Class: world
          0.7318296           0.6649874
```