Assignment2

1.Recitation Problem:

1.1 Chapter 3

1. **Describe the null hypotheses to which the p-values given in Table 3.4 correspond. Explain what conclusions you can draw based on these p-values. Your explanation should be phrased in terms of sales, TV, radio, and newspaper, rather than in terms of the coefficients of the linear model.**

	Coefficient	Std. error	t-statistic	p-value
Intercept	2.939	0.3119	9.42	< 0.0001
TV	0.046	0.0014	32.81	< 0.0001
radio	0.189	0.0086	21.89	< 0.0001
newspaper	−0.001	0.0059	−0.18	0.8599

➔

The Null hypothesis measures the contribution of a variable keeping the remaining variables already present in the model. For the given model

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2 + \hat{\beta}_3 X_3$$

if the test is carried out for $\hat{\beta}_1$ then the test will check the significance of including the variable X_1 when the variables X_2 and X_3 are already included in the model. Based on the p-values, we can conclude that if the p-value is small enough (typically ≤ 0.05) then we can reject the null hypothesis otherwise it holds true for an attribute.

- The p-value for the TV advertising attribute is < 0.0001, which shows that null-hypothesis can be rejected, and we can conclude that TV has an impact on Sales when other attributes (Radio and Newspaper advertising) were kept constant.
- The p-value for Radio advertising attribute is < 0.0001, which shows that null-hypothesis can be rejected, and we can conclude that Radio has an impact on Sales when other attributes (TV and Newspaper advertising) were kept constant.
- The p-value for Newspaper advertising attribute is 0.8599, which shows that null-hypothesis holds true here and we can conclude that Newspaper does not have an impact on Sales when other attributes (TV and Radio advertising) were kept constant.

3. **Suppose we have a data set with five predictors, $X1$ =GPA, $X2$ = IQ, $X3$ = Gender (1 for Female and for Male), $X4$ = Interaction between GPA and IQ, and $X5$ = Interaction between GPA and Gender. The response is starting salary after graduation (in thousands of dollars). Suppose we use least squares to fit the model, and get $\hat{\beta}0 = 50$, $\hat{\beta}1 =20$, $\hat{\beta}2 = 0.07$, $\hat{\beta}3 = 35$, $\hat{\beta}4 = 0.01$, $\hat{\beta}5 = -10$.**

➔

(a) **Which answer is correct, and why?**

i. **For a fixed value of IQ and GPA, males earn more on average than females.**
ii. **For a fixed value of IQ and GPA, females earn more on average than males.**
iii. **For a fixed value of IQ and GPA, males earn more on average than females provided that the GPA is high enough.**
iv. **For a fixed value of IQ and GPA, females earn more on average than males provided that the GPA is high enough.**

➢ Least square equation is given by:

$$\hat{y} = 50 + 20*GPA + 0.07*IQ + 35*Gender + 0.01*GPA*IQ - 10*GPA*Gender$$

For Male:

$$\hat{y} = 50 + 20*GPA + 0.07*IQ + 35*0 + 0.01*GPA*IQ - 10*GPA*0$$

⇨ $\hat{y} = 50 + 20*GPA + 0.07*IQ + 0.01*GPA*IQ$

For Female:

$\hat{y}$ = 50 + 20*GPA + 0.07*IQ + 35*1 + 0.01* GPA*IQ – 10*GPA*1

⇨ $\hat{y}$ = 85 + 10*GPA + 0.07*IQ + 0.01*GPA*IQ

For a fixed value of IQ and GPA, males will have a higher starting salary than females when

$50 + 20*GPA \geq 85 + 10*GPA$

⇨ $GPA \geq 3.5$

⇨ Option (iii) is correct.

(b) Predict the salary of a female with an IQ of 110 and a GPA of 4.0.

- For females least square equation is given by:

  $\hat{y}$ = 85 + 10*GPA + 0.07*IQ + 0.01*GPA*IQ

  Here IQ = 110, GPA = 4.0

  ⇨ $\hat{y}$ = 85 + 10*4 + 0.07*110 + 0.01*4*110

  ⇨ $\hat{y}$ = 137.1 thousand of dollars

  ⇨ $\hat{y}$ = 137.1 * 1000 = $137100

  Thus, predicted salary for female will be $137100.

(c) True or false: Since the coefficient for the GPA/IQ interaction team is very small, there is very little evidence of an interaction effect. Justify your answer.

- FALSE

  Based on the coefficient of GPA/IQ we cannot comment on the interaction effect. We need to test the hypothesis and for that, we will require p-value to decide whether the interaction is statistically significant or not.

4. I collect a set of data (*n* = 100 observations) containing a single predictor and a quantitative response. I then fit a linear regression model to the data, as well as a separate cubic regression, i.e. $Y = \beta 0 + \beta 1X + \beta 2X2 + \beta 3X3 + \varepsilon$.

➔

(a) Suppose that the true relationship between XX and YY is linear, i.e. Y=β0+β1X+ε. Consider the training residual sum of squares (RSS) for the linear regression and the training RSS for the cubic regression. Would we expect one to be lower than the other, would we expect them to be the same, or is there not enough information to tell? Justify your answer.

- Although the true relationship between XX and YY is linear, we can expect that the training Residual Sum of Squares(RSS) for Cubic Regression will be lower than that for Linear Regression because as the degree of polynomial increases it will become more flexible to fit the data points more closely and accurately which will result in less training errors.

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

- For the test data, the Residual Sum of Squares (RSS) for Linear Regression will be lower than that for Cubic Regression because the high degree polynomial will overfit the training data that will result in more testing errors.

(c) Suppose that the true relationship between X and Y is not linear, but we don't know how far it is from linear. Consider the training RSS for the linear regression and the training RSS for the cubic regression. Would we expect one to be lower than the other, would we expect them to be the same, or is there not enough information to tell? Justify your answer.

- Although we don't know how far the true relationship is from the linear, high order polynomial will have less training Residual Sum of Squares (RSS) than for linear regression because high degree polynomial model is more flexible to fit the data more closely and accurately which will result in less training errors. So, in our case training Residual Sum of Squares (RSS) for Cubic regression will be less.

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

➢ There is no enough information to tell whether a Linear or Cubic regression will have a lower Residual Sum of Squares (RSS) in the test data. We don’t exactly know how far it is from linear, it totally depends on the true relationship between X and Y. If the true relationship is close to linear then the Linear regression will have a lower Residual Sum of Squares (RSS) for testing data and if the true relationship is non-linear then the Cubic regression will have lower Residual Sum of Squares (RSS).

1.2 Chapter 4

4. When the number of features *p* is large, there tends to be a deterioration in the performance of KNN and other *local* approaches that perform prediction using only observations that are *near* the test observation for which a prediction must be made. This phenomenon is known as the *curse of dimensionality*, and it ties into the fact that non-parametric approaches often perform poorly when *p* is large. We will now investigate this curse.

➔

(a) Here we are given with the set of observations with P= 1 feature X, which is uniformly distributed over the range between 0 and 1 and as we are to use only observations that are within 10% of the range of X. Hence, the fraction of available observations that we will use to make prediction will be

For X=0.6 X € [0.55, 0.65]

$$= \frac{(0.65 - 0.55)}{(1-0)} * 100$$

$$= 10\%$$

(b) Here we are given with the set of observations with P= 2 features X_1 and X_2 that are also uniformly distributed over the same range such that (X_1, X_2) € [0,1] X [0, 1].

So, the fraction of available observations that we will use to make prediction is given by:

$$= (10\%) * (10\%)$$

$$= 1\%$$

(c) Here we are given with the set of observations with P= 100 features and all of which are uniformly distributed over the range between 0 and 1.

So, the fraction of available observations that we will use to make prediction is given by:

$$= (0.1)^{100} * 100$$

$$= (10)^{-98}\ \%$$

(d) From the above discussion we can conclude that as the number of features increases then the percentage of observations that are used to predict KNN becomes very small. Hence, more features leads to fewer the neighbors (Curse of dimensionality).

(e) According to the question:

For P= 1 => length = (0.10)

For P= 2 => length = $(0.10)^{1/2}$ = 0.316

For P= 100 => length = $(0.10)^{1/100}$ = 0.977

From the above, we can conclude that when we wish to make predictions for the test observations that contains on average 10% of the training observations then If we have large number of features, it will be better to include all the features.

Assignment2

6. **Suppose we collect data for a group of students in a statistics class with variables *X*1 =hours studied, *X*2 =undergrad GPA, and *Y* =receive an A. We fit a logistic regression and produce estimated coefficient, ˆ *β*0 = −6, ˆ*β*1 = 0.05, ˆ*β*2 = 1.**

   ➔

   Logit function is given by:

$$\mathbf{P(x)=} \frac{e^{\hat{\beta}0\ +\ \hat{\beta}1X1\ +\ \hat{\beta}2X2}}{1+\ e^{\hat{\beta}0\ +\ \hat{\beta}1X1\ +\ \hat{\beta}2X2}}$$

Here, $\hat{\beta}_0$ = -6, $\hat{\beta}_1$ = 0.05, $\hat{\beta}_2$ = 1

And *X*1 =hours studied, *X*2 =undergrad GPA

(a) Estimate the probability that a student who studies for 40 h and has an undergrad GPA of 3.5 gets an A in the class.

- Here X1 = 40hrs, X2 = 3.5
- P(x)= $\frac{e^{-6\ +\ 0.05*40\ +\ 1*3.5}}{1+\ e^{-6+0.05*40+1*3.5}}$

  $= \frac{e^{-0.5}}{1+\ e^{-0..5}}$

  = 37.75 %

(b) How many hours would the student in part (a) need to study to have a 50% chance of getting an A in the class?

- Here P(x) = 0.5, X2 = 3.5
- $0.5 = \frac{e^{-6\ +\ 0.05*X1\ +\ 1*3.5}}{1+\ e^{-6+0.05*X1+1*3.5}}$
- $0.5 + 0.5 * e^{-6+0.05*X1+1*3.5} = e^{-6\ +\ 0.05*X1\ +\ 1*3.5}$
- $0.5 * e^{0.05*X1-2.5} = 0.5$
- $e^{0.05*X1-\ 2.5} = 1$
- 0.05*X1 – 2.5 = $\log_e(1)$
- 0.05 * X1 = 2.5
- X1 = 50 hrs.

7. **Suppose that we wish to predict whether a given stock will issue a dividend this year ("Yes" or "No") based on *X*, last year's percent profit. We examine a large number of companies and discover that the mean value of *X* for companies that issued a dividend was ¯*X* = 10, while the mean for those that didn't was ¯*X* = 0. In addition, the variance of *X* for these two sets of companies was ˆ*σ*2 = 36. Finally, 80% of companies issued dividends. Assuming that *X* follows a normal distribution, predict the probability that a company will issue a dividend this year given that its percentage profit was *X* = 4 last year.**

➔ According to Bayes theorem:

$$\boldsymbol{Pk(x)} = \frac{\pi_k f_k(x)}{\sum_{l=1}^{k} \pi_l f_l(x)}$$

Where $f_k(x) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{(\frac{-1}{2\sigma^2}(x-\mu_k)^2)}$

- $\mathbf{P_k(x)=} \frac{\pi_k \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{(\frac{-1}{2\sigma^2}(x-\mu_k)^2)}}{\sum_{l=1}^{k} \pi_l \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{(\frac{-1}{2\sigma^2}(x-\mu_l)^2)}}$

Here π_{yes} = 0.8, π_{no} = 0.2, μ_{yes}= 10, μ_{no}= 0, x=4, σ^2 =36

- $P_{yes}(x)= \frac{0.8*\frac{1}{\sqrt{2\pi(36)}} e^{(\frac{-1}{2*(36)}(4-10)^2)}}{0.8*\frac{1}{\sqrt{2\pi(36)}} e^{(\frac{-1}{2*(36)}(4-10)^2)} + 0.2*\frac{1}{\sqrt{2\pi(36)}} e^{(\frac{-1}{2*(36)}(4-0)^2)}}$

- $P_{yes}(x)= \frac{0.8*e^{(\frac{-1}{2*(36)}(36))}}{0.8*e^{(\frac{-1}{2*(36)}(36)}+0.2*e^{(\frac{-1}{2*(36)}(16))}}$
- $P_{yes}(x)= \frac{0.8*e^{(\frac{-1}{2})}}{0.8*e^{(\frac{-1}{2})}+0.2*e^{(\frac{-2}{9})}}$
- $P_{yes}(x)$=0.752

9. This problem has to do with *odds*.

➔

(a) On average, what fraction of people with an odds of 0.37 of defaulting on their credit card payment will in fact default?

- **We know that**

$$\textbf{Odds} = \frac{P(x)}{1-P(x)}$$

Here odds = 0.37

- 0.37 = $\frac{P(x)}{1-P(x)}$
- 0.37 * (1 – P(x)) = P(x)
- P(x) + 0.37 * P (x) = 0.37
- P(x) = 0.27
- P(x) = 27%

(b) Suppose that an individual has a 16% chance of defaulting on her credit card payment. What are the odds that she will default?

- Here P(x) = 0.16
- 1 – P(x) = 1 – 0.16 = 0.84
- Odds = $\frac{0.16}{0.84}$
- Odds= 0.19

==

2. Practicum problem:

2.1 Problem 1

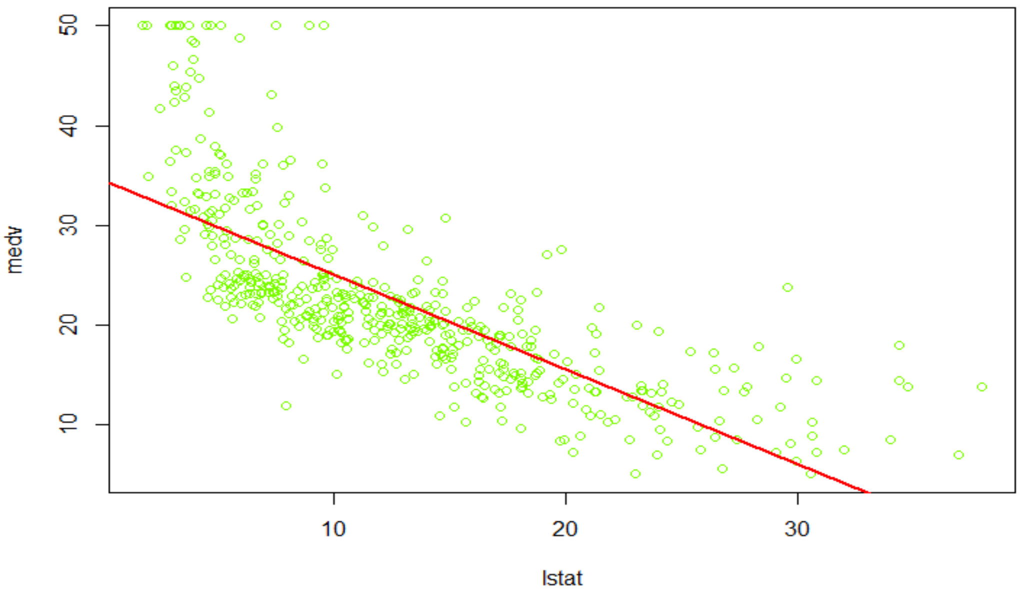
```
 #Load Libraries:
library(MASS)
library(ggplot2)
library(gridExtra)

#Load dataset:
boston<-data.frame(Boston)
attach(Boston)

#Check number of rows:
nrow(boston)

## [1] 506

#Check number of column:
ncol(boston)

## [1] 14
```

Assignment2

```
#view top 6 rows of dataset:
head(boston)

##      crim zn indus chas   nox    rm  age    dis rad tax ptratio  black
## 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3 396.90
## 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8 396.90
## 3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8 392.83
## 4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7 394.63
## 5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7 396.90
## 6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7 394.12
##   lstat medv
## 1  4.98 24.0
## 2  9.14 21.6
## 3  4.03 34.7
## 4  2.94 33.4
## 5  5.33 36.2
## 6  5.21 28.7

#Check the datatypes of dataset:
str(boston)

## 'data.frame':    506 obs. of  14 variables:
##  $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
##  $ chas   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...
##  $ rm     : num  6.58 6.42 7.18 7 7.15 ...
##  $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
##  $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ rad    : int  1 2 2 3 3 3 5 5 5 5 ...
##  $ tax    : num  296 242 242 222 222 222 311 311 311 311 ...
##  $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
##  $ black  : num  397 397 393 395 397 ...
##  $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
##  $ medv   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...

#fit a regression model:
res<-lm(medv~lstat,data=boston)


#Display Summary:
summary(res)

##
## Call:
## lm(formula = medv ~ lstat, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.168  -3.990  -1.318   2.034  24.500
```

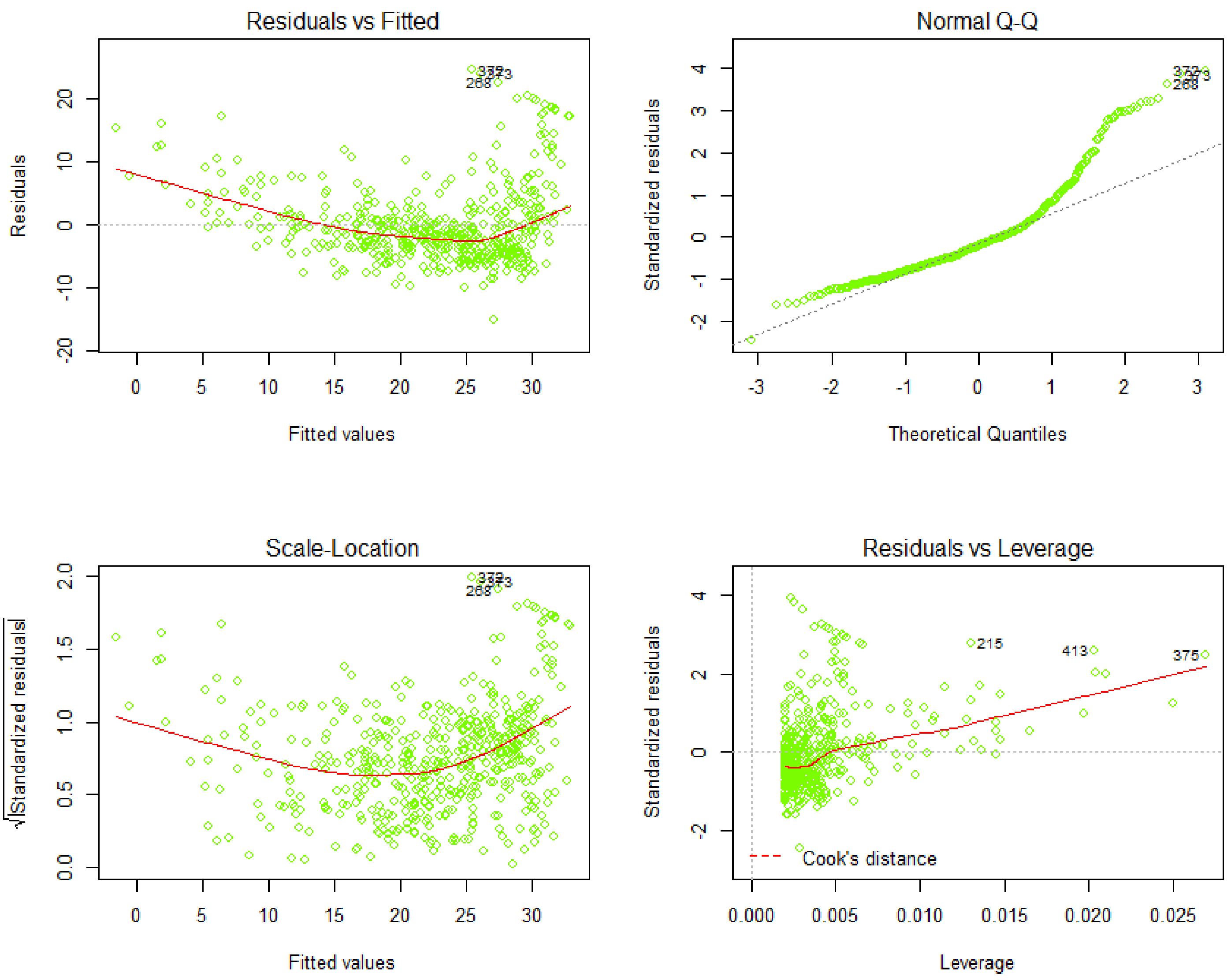
```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.55384    0.56263   61.41   <2e-16 ***
## lstat       -0.95005    0.03873  -24.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.216 on 504 degrees of freedom
## Multiple R-squared:  0.5441, Adjusted R-squared:  0.5432
## F-statistic: 601.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
#Plot for Resulting fit:
plot(lstat,medv, col="lawngreen")
abline(res,col="red",lwd=2)
```

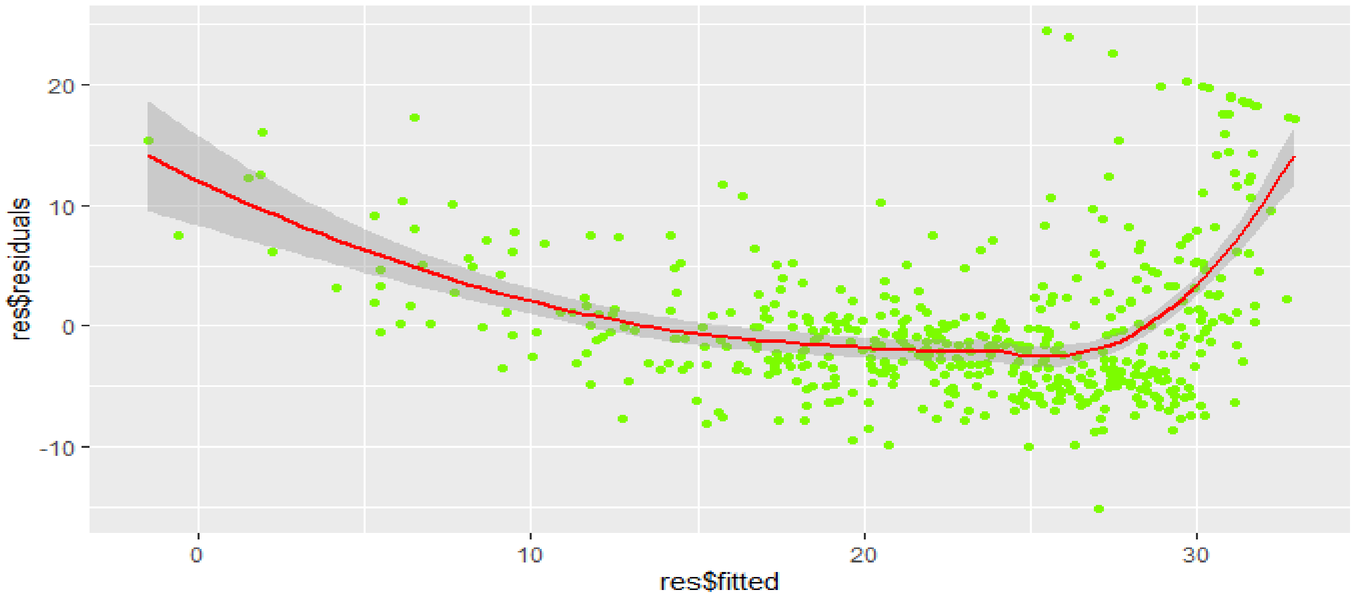


```
par(mfrow=c(2,2))
plot(res, col="lawngreen")
```

Assignment2



```
#Plot for Fitted values Vs. Residuals
p1 <- ggplot(data=boston,aes(x=res$fitted,y=res$residual)) +
  geom_point(col="lawngreen")+
  stat_smooth(col="red")+
p1
```



In the above graph we can see that there is a non-linear relationship between predictor and response variables.

```
#predictions for lstat values of 5, 10 and 15
test=data.frame(lstat=c(5,10,15))
predict(res,test,interval = "confidence")

##        fit      lwr      upr
## 1 29.80359 29.00741 30.59978
## 2 25.05335 24.47413 25.63256
## 3 20.30310 19.73159 20.87461

predict(res,test,interval = "predict")

##        fit       lwr      upr
## 1 29.80359 17.565675 42.04151
## 2 25.05335 12.827626 37.27907
## 3 20.30310  8.077742 32.52846
```

The confidence and the prediction intervals are not same for the response values. Although we can see that the fitted values are same for both the intervals however the range is wider in case of prediction interval. This is due to the additional term in the standard error of predication interval. Prediction interval shows the uncertainty around a single value, while confidence interval shows the uncertainty around the mean prediction, that is why prediction interval has wider range.

```
#modified regression model:
newres <- lm(medv~lstat+I(lstat^2),data=boston)


#summary for modified model:
summary(newres)

##
## Call:
```

```
## lm(formula = medv ~ lstat + I(lstat^2), data = boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.2834  -3.8313  -0.5295   2.3095  25.4148
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 42.862007   0.872084   49.15   <2e-16 ***
## lstat       -2.332821   0.123803  -18.84   <2e-16 ***
## I(lstat^2)   0.043547   0.003745   11.63   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.524 on 503 degrees of freedom
## Multiple R-squared:  0.6407, Adjusted R-squared:  0.6393
## F-statistic: 448.5 on 2 and 503 DF,  p-value: < 2.2e-16
```

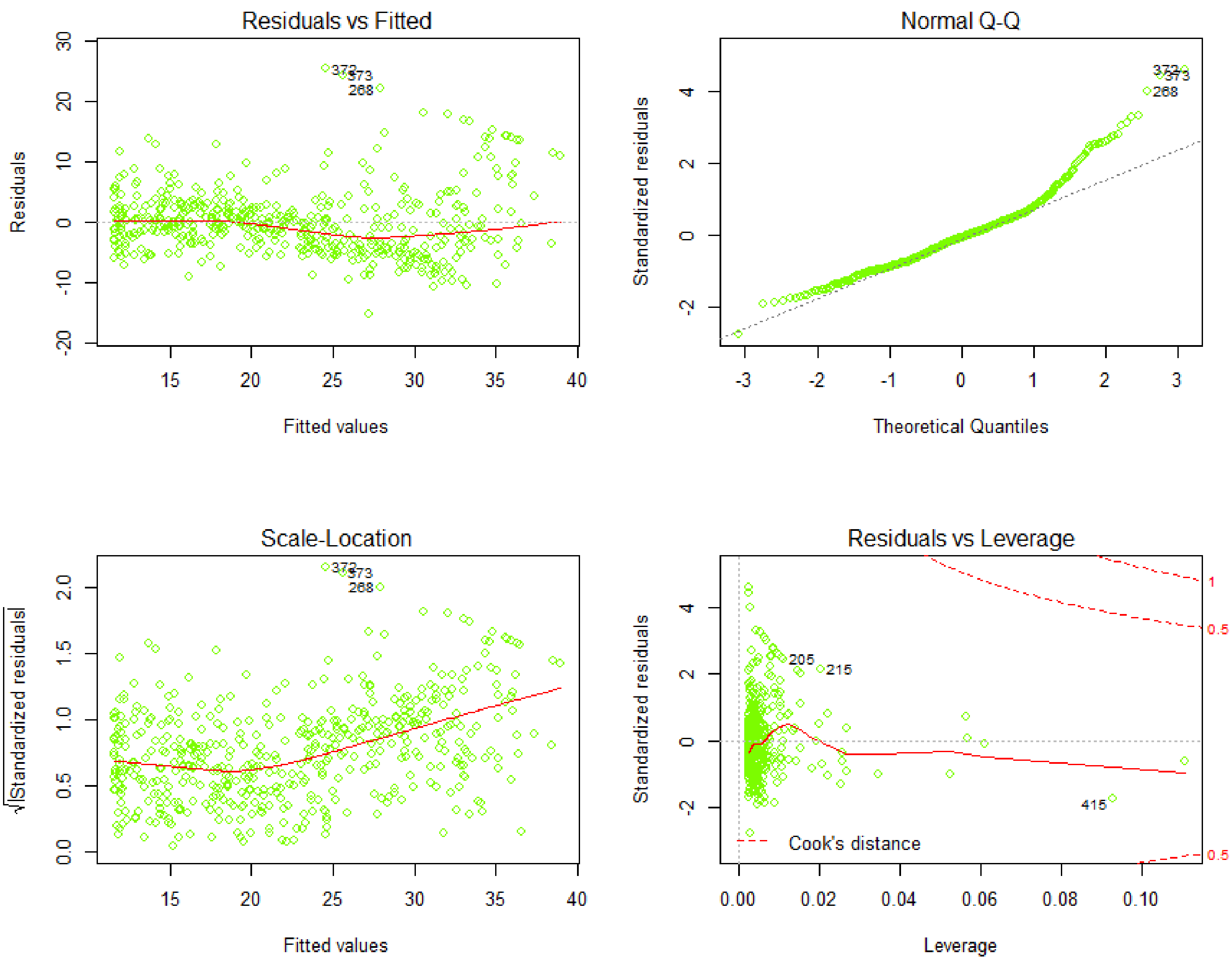
```
#comparison of two models
anova(res,newres)
```

```
## Analysis of Variance Table
##
## Model 1: medv ~ lstat
## Model 2: medv ~ lstat + I(lstat^2)
##   Res.Df   RSS Df Sum of Sq     F    Pr(>F)
## 1    504 19472
## 2    503 15347  1    4125.1 135.2 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

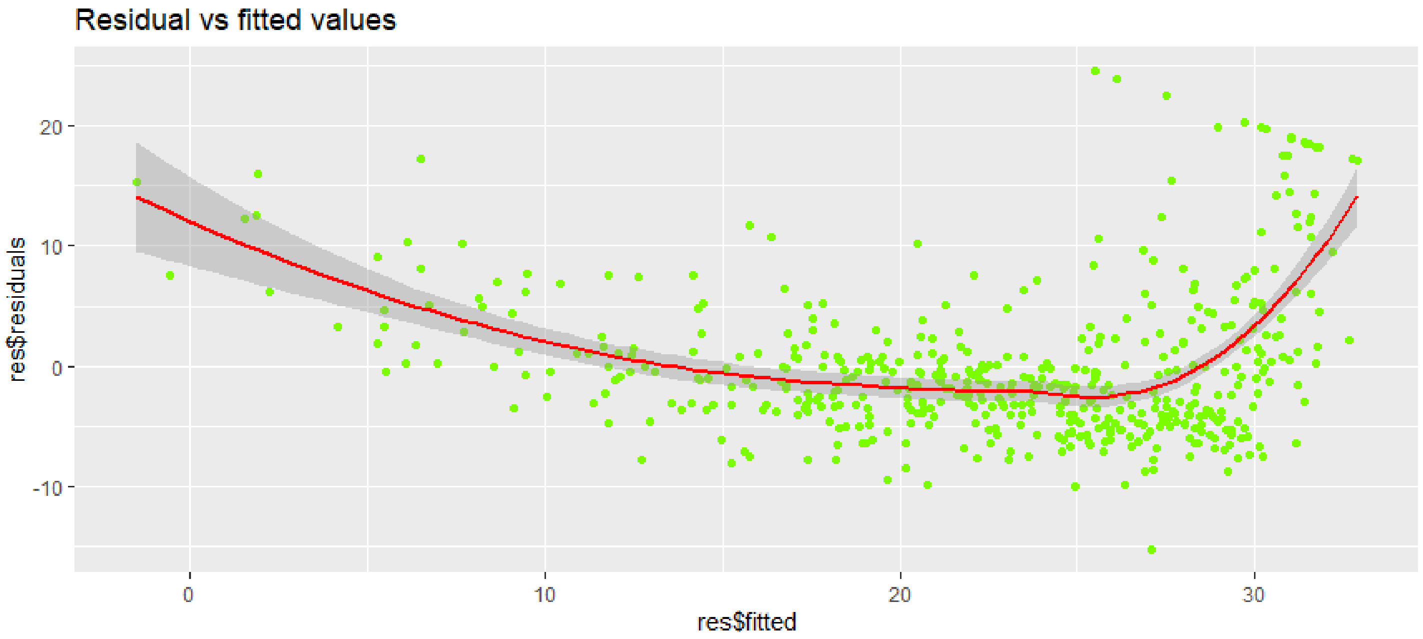
From the summary of both linear and non-Linear models, we can see that the value of adjusted R^2 has increased from 54% to 64%. Which suggest that almost 10% more variance can be explained by the model. Hence, we can say that the performance of model has improved as we have introduced higher degree polynomial in our model.

```
par(mfrow=c(2,2))
plot(newres, col="lawngreen")
```

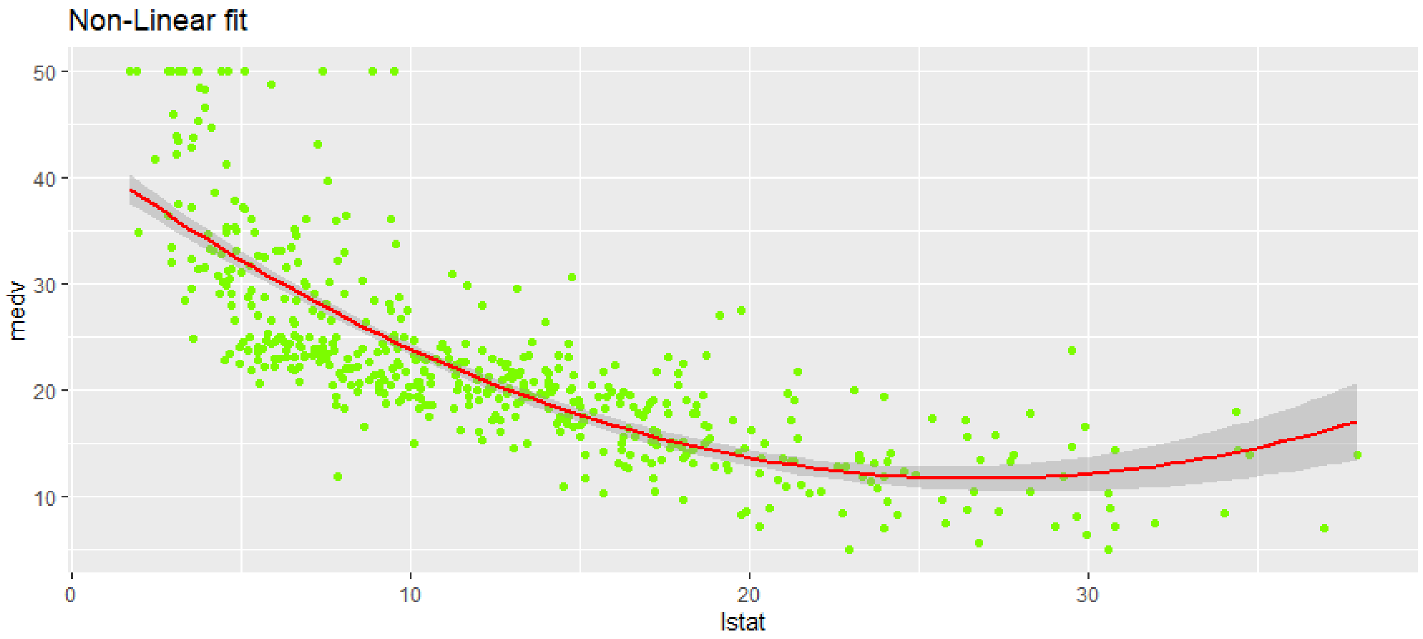
Assignment2



```
#plot for fitted values and residual for modified model:
p1 <- ggplot(data=boston,aes(x=res$fitted,y=res$residuals)) +
  geom_point(col="lawngreen")+
  stat_smooth(col="red")+
  ggtitle("Non-Linear fit")
p1
```



```
#plot for non-linear fit:
p1 <- ggplot(data=boston,aes(x=lstat,y=medv)) +
  geom_point(col="lawngreen")+
  stat_smooth(formula = y ~ x + I(x^2),method="lm",col="red")+
  ggtitle("Non-Linear fit")
p1
```



==

2.2 Problem 2

```
#chapter2
#Load Libraries:
library(ggplot2)
library(lattice)
library(caret)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(ROCR)
```

```
## Loading required package: gplots
```

```
## 
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
## 
##     lowess
```

```
## corrplot 0.84 loaded
#Load dataset:
URL<- "https://archive.ics.uci.edu/ml/machine-learning-databases/abalone/abalone.data"
data <- read.table(URL,stringsAsFactors = FALSE, sep = ",")
colnames(data) <- c("Sex","Length","Diameter","Height","Whole_Weigth","Shucked_weight","Viscera_weigh
t","Shell_weight","Rings")

#Display Summary:
summary(data)
```

```
##      Sex                Length         Diameter          Height      
##  Length:4177        Min.   :0.075   Min.   :0.0550   Min.   :0.0000  
##  Class :character   1st Qu.:0.450   1st Qu.:0.3500   1st Qu.:0.1150  
##  Mode  :character   Median :0.545   Median :0.4250   Median :0.1400  
##                     Mean   :0.524   Mean   :0.4079   Mean   :0.1395  
##                     3rd Qu.:0.615   3rd Qu.:0.4800   3rd Qu.:0.1650  
##                     Max.   :0.815   Max.   :0.6500   Max.   :1.1300  
##   Whole_Weigth    Shucked_weight   Viscera_weight    Shell_weight   
##  Min.   :0.0020   Min.   :0.0010   Min.   :0.0005   Min.   :0.0015  
##  1st Qu.:0.4415   1st Qu.:0.1860   1st Qu.:0.0935   1st Qu.:0.1300  
##  Median :0.7995   Median :0.3360   Median :0.1710   Median :0.2340  
##  Mean   :0.8287   Mean   :0.3594   Mean   :0.1806   Mean   :0.2388  
##  3rd Qu.:1.1530   3rd Qu.:0.5020   3rd Qu.:0.2530   3rd Qu.:0.3290  
##  Max.   :2.8255   Max.   :1.4880   Max.   :0.7600   Max.   :1.0050  
##      Rings       
##  Min.   : 1.000  
##  1st Qu.: 8.000  
##  Median : 9.000  
##  Mean   : 9.934  
##  3rd Qu.:11.000  
##  Max.   :29.000
```

```
#Check number of rows:
nrow(data)
```

```
## [1] 4177
```

```
#Check number of column:
ncol(data)

## [1] 9

#check for all available sex categories in dataset:
unique(data$Sex)

## [1] "M" "F" "I"

# Remove Infant category:
newdata <- data[ which(data$Sex!='I'), ]

#Check number of rows of new dataset:
nrow(newdata)

## [1] 2835

#Check number of columns of new dataset:
ncol(newdata)

## [1] 9

#check for all sex categories in new dataset:
unique(newdata$Sex)

## [1] "M" "F"

#Check for datatypes of new dataset:
str(newdata)

## 'data.frame':    2835 obs. of  9 variables:
##  $ Sex           : chr  "M" "M" "F" "M" ...
##  $ Length        : num  0.455 0.35 0.53 0.44 0.53 0.545 0.475 0.55 0.525 0.43 ...
##  $ Diameter      : num  0.365 0.265 0.42 0.365 0.415 0.425 0.37 0.44 0.38 0.35 ...
##  $ Height        : num  0.095 0.09 0.135 0.125 0.15 0.125 0.125 0.15 0.14 0.11 ...
##  $ Whole_Weigth  : num  0.514 0.226 0.677 0.516 0.777 ...
##  $ Shucked_weight: num  0.2245 0.0995 0.2565 0.2155 0.237 ...
##  $ Viscera_weight: num  0.101 0.0485 0.1415 0.114 0.1415 ...
##  $ Shell_weight  : num  0.15 0.07 0.21 0.155 0.33 0.26 0.165 0.32 0.21 0.135 ...
##  $ Rings         : int  15 7 9 10 20 16 9 19 14 10 ...

#transform the datatype of Sex attribute to factor type:
newdata$Sex<-factor(newdata$Sex)

#check for the change:
str(newdata)

## 'data.frame':    2835 obs. of  9 variables:
##  $ Sex           : Factor w/ 2 levels "F","M": 2 2 1 2 1 1 2 1 1 2 ...
##  $ Length        : num  0.455 0.35 0.53 0.44 0.53 0.545 0.475 0.55 0.525 0.43 ...
##  $ Diameter      : num  0.365 0.265 0.42 0.365 0.415 0.425 0.37 0.44 0.38 0.35 ...
##  $ Height        : num  0.095 0.09 0.135 0.125 0.15 0.125 0.125 0.15 0.14 0.11 ...
##  $ Whole_Weigth  : num  0.514 0.226 0.677 0.516 0.777 ...
##  $ Shucked_weight: num  0.2245 0.0995 0.2565 0.2155 0.237 ...
##  $ Viscera_weight: num  0.101 0.0485 0.1415 0.114 0.1415 ...
##  $ Shell_weight  : num  0.15 0.07 0.21 0.155 0.33 0.26 0.165 0.32 0.21 0.135 ...
##  $ Rings         : int  15 7 9 10 20 16 9 19 14 10 ...
```

```
#partition the dataset (80% for training and 20% for testing)
partition <- createDataPartition(newdata$Sex, p = .8,list = FALSE)
Train <- newdata[ partition,]
Test  <- newdata[-partition,]

#fit the logistic regression model:
model  <- glm(Sex~.,data=Train,family=binomial)
model
```

```
## 
## Call:  glm(formula = Sex ~ ., family = binomial, data = Train)
## 
## Coefficients:
##    (Intercept)          Length        Diameter          Height  
##       2.915696       -2.562642       -3.453171       -4.795840  
##   Whole_Weigth  Shucked_weight  Viscera_weight    Shell_weight  
##      -0.371209        3.386326       -1.782936        1.031882  
##          Rings  
##      -0.001936  
## 
## Degrees of Freedom: 2268 Total (i.e. Null);  2260 Residual
## Null Deviance:       3132 
## Residual Deviance: 3069  AIC: 3087
```

```
#display summary of fitted model:
summary(model)
```

```
## 
## Call:
## glm(formula = Sex ~ ., family = binomial, data = Train)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -1.8258  -1.2039   0.8713   1.1173   1.5141  
## 
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)    
## (Intercept)     2.915696   0.514181   5.671 1.42e-08 ***
## Length         -2.562642   2.283101  -1.122 0.261676    
## Diameter       -3.453171   2.695733  -1.281 0.200202    
## Height         -4.795840   2.396615  -2.001 0.045383 *  
## Whole_Weigth   -0.371209   0.817163  -0.454 0.649638    
## Shucked_weight  3.386326   0.995829   3.401 0.000673 ***
## Viscera_weight -1.782936   1.423468  -1.253 0.210377    
## Shell_weight    1.031882   1.267792   0.814 0.415691    
## Rings         -0.001936   0.017771  -0.109 0.913254
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3131.7  on 2268  degrees of freedom
## Residual deviance: 3069.4  on 2260  degrees of freedom
## AIC: 3087.4
##
## Number of Fisher Scoring iterations: 4
```

Considering the P-value of all the attributes, the attributes with p-value typically <0.05 are statistically significant and are relevant to response attribute and the null hypothesis can be rejected for such attributes. Here in this case we can see that only attribute Shucked_weight has p-value 0.000673. Hence, it is strongly relevant to response variable. However, all the other attributes have very high p-value which means that they are not strongly relevant to response attributes and null hypothesis can hold true for them.

```
#calculation of confidence interval for predictors:
confint(model)

## Waiting for profiling to be done...

##                       2.5 %      97.5 %
## (Intercept)       1.92293095  3.93945706
## Length           -7.04288243  1.91332447
## Diameter         -8.74967906  1.82628262
## Height           -9.73613145 -0.74960347
## Whole_Weigth     -1.98630510  1.22894780
## Shucked_weight    1.44554636  5.35724940
## Viscera_weight   -4.57885721  1.00749966
## Shell_weight     -1.45203261  3.52953903
## Rings            -0.03680069  0.03291493
```

An effect of an attribute is significant if all the values in the confidence interval belongs to the same side of zero. And we know that the null hypothesis can be rejected if the attribute is significant enough. Here in this case we can see that all the attributes except shucked_weight contains zero within their range. Hence null hypothesis holds true for all those attributes except shucked_weight. That means these attributes have very less effect on response attribute. Also, as discussed above, p-value supports the null hypothesis assumption.

```
#predict the regression value for test data
t<-predict(model,Test,type= "response")

#use 50% cutoff to tag a male/female:
new_Temp=ifelse(t >0.5,"M","F")

#check the datatype of predicted result:
str(new_Temp)

##  Named chr [1:566] "M" "M" "M" "M" "F" "F" "F" "M" "M" "M" "M" "F" "F" ...
##  - attr(*, "names")= chr [1:566] "9" "18" "19" "21" ...
```

```
#convert the datatype to factor:
new_Temp<-factor(new_Temp)

#check for the change in datatype:
str(new_Temp)
```

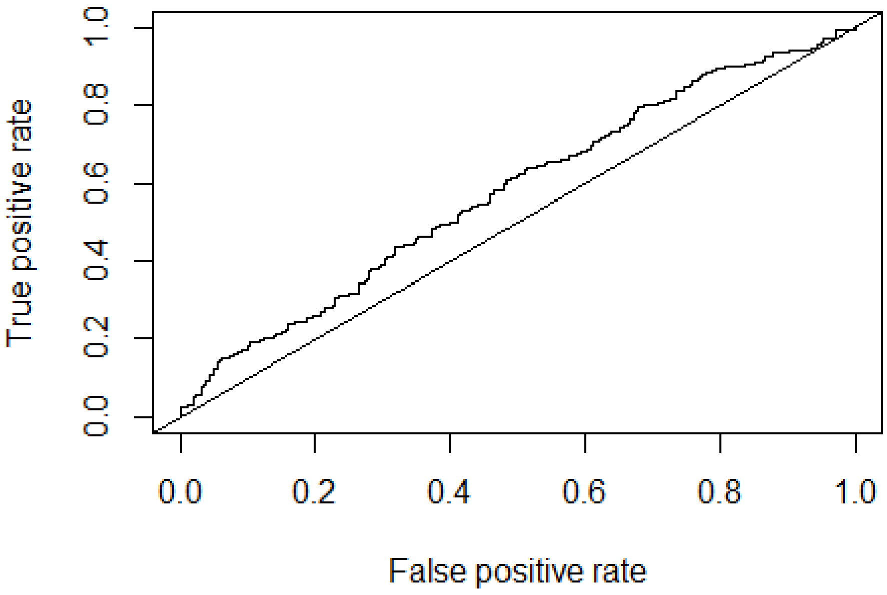
```
##  Factor w/ 2 levels "F","M": 2 2 2 2 1 1 1 2 2 2 ...
##  - attr(*, "names")= chr [1:566] "9" "18" "19" "21" ...
```

```
#Display confusion matrix:
confusionMatrix(Test$Sex,new_Temp)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   F   M
##          F  93 168
##          M  81 224
## 
##                Accuracy : 0.5601
##                  95% CI : (0.5181, 0.6014)
##     No Information Rate : 0.6926
##     P-Value [Acc > NIR] : 1
## 
##                   Kappa : 0.093
## 
##  Mcnemar's Test P-Value : 5.036e-08
## 
##             Sensitivity : 0.5345
##             Specificity : 0.5714
##          Pos Pred Value : 0.3563
##          Neg Pred Value : 0.7344
##              Prevalence : 0.3074
##          Detection Rate : 0.1643
##    Detection Prevalence : 0.4611
##       Balanced Accuracy : 0.5530
## 
##        'Positive' Class : F
## 


## 
```

Accuracy of this model is 56.01%

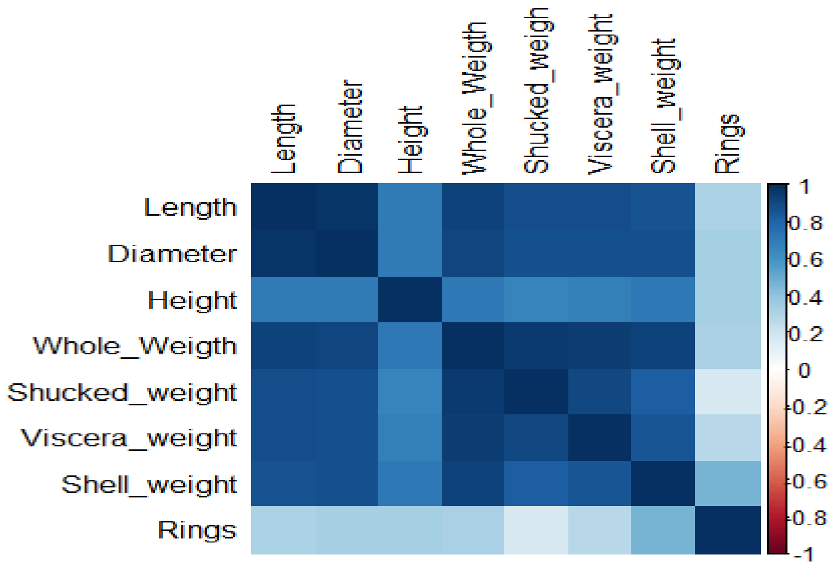
```
#ROC curve
pred=prediction(t,Test$Sex)
perf=performance(pred,measure="tpr",x.measure="fpr")
plot(perf)
abline(0,1)
```



```
aucperf=performance(pred,measure="auc")
cat("Area under the curve:")

## Area under the curve:

aucperf@y.values

## [[1]]
## [1] 0.5785441
```

Area under the curve shows the accuracy of 57.85%

```
#Display coorelation between predictors:
M <- cor(newdata[-1])
corrplot(M, method = "color", tl.col = "black", tl.srt = 90)
```



From the above figure we can see that there is a positive correlation between the predictors that indicates multicollinearity among predictors, and which may result in reducing the performance of model. Hence we can say that not all the predictors are relevant in predicting the response.

==

2.3 Problem 3

```
#load libraries:
library(caTools)
library(e1071)

#Load dataset:
URL <- "https://archive.ics.uci.edu/ml/machine-learning-databases/mushroom/agaricus-lepiota.data"
m_dataset <- read.table(URL,stringsAsFactors = FALSE, sep = ",")
colnames(m_dataset) <- c("class","cap-shape","cap-surface","cap-color","bruises","odor","gill-attachment","gil
l-spacing","gill-size","gill-color","stalk-shape","stalk-root","stalk-surface-above-ring","stalk-surface-below-ring
","stalk-color-above-ring","stalk-color-below-ring","veil-type","veil-color","ring-number","ring-type","spore-pr
int-color","population","habitat")

#Display Summary:
summary(m_dataset)
```

```
##     class            cap-shape          cap-surface       
##  Length:8124        Length:8124        Length:8124       
##  Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character  
##   cap-color           bruises              odor          
##  Length:8124        Length:8124        Length:8124       
##  Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character  
##  gill-attachment    gill-spacing        gill-size        
##  Length:8124        Length:8124        Length:8124       
##  Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character  
##   gill-color        stalk-shape         stalk-root       
##  Length:8124        Length:8124        Length:8124       
##  Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character  
##  stalk-surface-above-ring stalk-surface-below-ring stalk-color-above-ring
##  Length:8124              Length:8124              Length:8124           
##  Class :character         Class :character         Class :character      
##  Mode  :character         Mode  :character         Mode  :character      
##  stalk-color-below-ring  veil-type          veil-color       
##  Length:8124            Length:8124        Length:8124       
##  Class :character       Class :character   Class :character  
##  Mode  :character       Mode  :character   Mode  :character  
##  ring-number         ring-type         spore-print-color 
##  Length:8124        Length:8124        Length:8124       
##  Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character  
##   population          habitat         
##  Length:8124        Length:8124       
##  Class :character   Class :character  
##  Mode  :character   Mode  :character
```

```
#Check the datatypes of dataset:
str(m_dataset)
```

```
## 'data.frame':    8124 obs. of  23 variables:
##  $ class                   : chr  "p" "e" "e" "p" ...
##  $ cap-shape               : chr  "x" "x" "b" "x" ...
##  $ cap-surface             : chr  "s" "s" "s" "y" ...
##  $ cap-color               : chr  "n" "y" "w" "w" ...
##  $ bruises                 : chr  "t" "t" "t" "t" ...
##  $ odor                    : chr  "p" "a" "l" "p" ...
##  $ gill-attachment         : chr  "f" "f" "f" "f" ...
##  $ gill-spacing            : chr  "c" "c" "c" "c" ...
##  $ gill-size               : chr  "n" "b" "b" "n" ...
##  $ gill-color              : chr  "k" "k" "n" "n" ...
##  $ stalk-shape             : chr  "e" "e" "e" "e" ...
##  $ stalk-root              : chr  "e" "c" "c" "e" ...
##  $ stalk-surface-above-ring: chr  "s" "s" "s" "s" ...
##  $ stalk-surface-below-ring: chr  "s" "s" "s" "s" ...
##  $ stalk-color-above-ring  : chr  "w" "w" "w" "w" ...
##  $ stalk-color-below-ring  : chr  "w" "w" "w" "w" ...
##  $ veil-type               : chr  "p" "p" "p" "p" ...
##  $ veil-color              : chr  "w" "w" "w" "w" ...
##  $ ring-number             : chr  "o" "o" "o" "o" ...
##  $ ring-type               : chr  "p" "p" "p" "p" ...
##  $ spore-print-color       : chr  "k" "n" "n" "k" ...
##  $ population              : chr  "s" "n" "n" "s" ...
##  $ habitat                 : chr  "u" "g" "m" "u" ...
```

```
#check for different categories for attribute stalk-root
unique(m_dataset$`stalk-root`)
```

```
## [1] "e" "c" "b" "r" "?"
```

We can see that the attribute stalk-root contains some missing values. So, we need to handle these missing values by data cleaning process before we apply Naïve Bayes classifier to build the model.

```
#calculate the total number of rows containing missing values:
cnt <- sum(m_dataset$`stalk-root`=='?')
cnt

## [1] 2480

#remove all the rows containing missing values:
m_newdataset <- m_dataset[ which(m_dataset$'stalk-root'!='?'), ]

#display total number of rows after removing missing values:
nrow(m_newdataset)

## [1] 5644

#calculate number of rows for each target class(poisonous and edible)
table(m_newdataset$class)

## 
##    e    p 
## 3488 2156
```

We have performed data cleaning by removing all the rows containing missing values. As we are given with response attribute containing only two target classes (edible and poisonous) and after removing all the rows containing missing values, we have 3488 number of rows for edible class and 2156 number of rows for poisonous class, which are sufficient enough to develop high performance classification model. Hence, we can remove all the rows containing missing values.

```
#display the datatype of attributes:
str(m_newdataset)

## 'data.frame':    5644 obs. of  23 variables:
##  $ class                   : chr  "p" "e" "e" "p" ...
##  $ cap-shape               : chr  "x" "x" "b" "x" ...
##  $ cap-surface             : chr  "s" "s" "s" "y" ...
##  $ cap-color               : chr  "n" "y" "w" "w" ...
##  $ bruises                 : chr  "t" "t" "t" "t" ...
##  $ odor                    : chr  "p" "a" "l" "p" ...
##  $ gill-attachment         : chr  "f" "f" "f" "f" ...
##  $ gill-spacing            : chr  "c" "c" "c" "c" ...
##  $ gill-size               : chr  "n" "b" "b" "n" ...
##  $ gill-color              : chr  "k" "k" "n" "n" ...
##  $ stalk-shape             : chr  "e" "e" "e" "e" ...
##  $ stalk-root              : chr  "e" "c" "c" "e" ...
##  $ stalk-surface-above-ring: chr  "s" "s" "s" "s" ...
##  $ stalk-surface-below-ring: chr  "s" "s" "s" "s" ...
##  $ stalk-color-above-ring  : chr  "w" "w" "w" "w" ...
##  $ stalk-color-below-ring  : chr  "w" "w" "w" "w" ...
##  $ veil-type               : chr  "p" "p" "p" "p" ...
##  $ veil-color              : chr  "w" "w" "w" "w" ...
##  $ ring-number             : chr  "o" "o" "o" "o" ...
##  $ ring-type               : chr  "p" "p" "p" "p" ...
##  $ spore-print-color       : chr  "k" "n" "n" "k" ...
##  $ population              : chr  "s" "n" "n" "s" ...
##  $ habitat                 : chr  "u" "g" "m" "u" ...

#convert the datatype of response attribute to factor:
m_newdataset$class=factor(m_newdataset$class)

#split the dataset(80% for training and 20% for testing)
set.seed(123)
sample = sample.split(m_newdataset,SplitRatio = 0.80)
train1 =subset(m_newdataset,sample ==TRUE)
test1=subset(m_newdataset, sample==FALSE)

#fitting the model using Naive Bayes classifier:
nb_model<-naiveBayes(train1[-1],train1$class)

#make prediction on testing data:
nb_test_predict<-predict(nb_model,test1[,-1])

#make prediction on training data:
nb_train_predict<-predict(nb_model,train1[,-1])

#Accuracy calculation for training data:
mean(nb_train_predict==train1$class)

## [1] 0.954494

#Accuracy calculation for testing data:
mean(nb_test_predict==test1$class)

## [1] 0.9494703

#confusion matrix
table(nb_test_predict,test1$class)

## 
## nb_test_predict   e   p
##               e 743  55
##               p   7 422
```

This model has produced total of 55 False Positives. i.e, this model has wrongly classified55 edible class tuples to poisonous class.