# PES University, Bengaluru

## UE18CS312 - Data Analytics

## Session: Aug – Dec 2020

## Weeks 3-4 – Code Snippets for Worksheet 2(a) (for Unit 2)

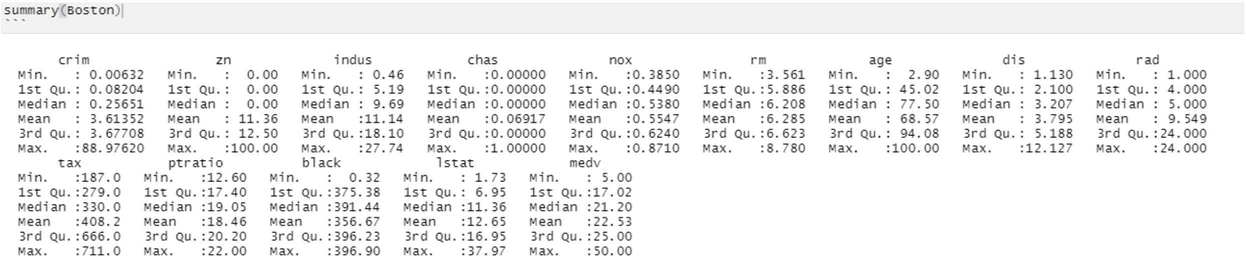
**Compiled by**: Ms. Richa and Ms. Mainaki Saraf
VII CSE, PES University RR Campus

## WORKSHEET 1

### 1.0 Getting started

For this worksheet, we will be using the Boston housing dataset from the ISLR package in R. The dataset can also be found here:
https://www.kaggle.com/altavish/boston-housing-dataset
The description of the dataset can be found here:
https://www.kaggle.com/c/boston-housing

1. Read the dataset
    - Go through the meta-data and understand what each column is
    - Try different summarisation techniques from https://dabblingwithdata.wordpress.com/2018/01/02/my-favourite-r-package-for-summarising-data/
      Here is a result of our favourite:



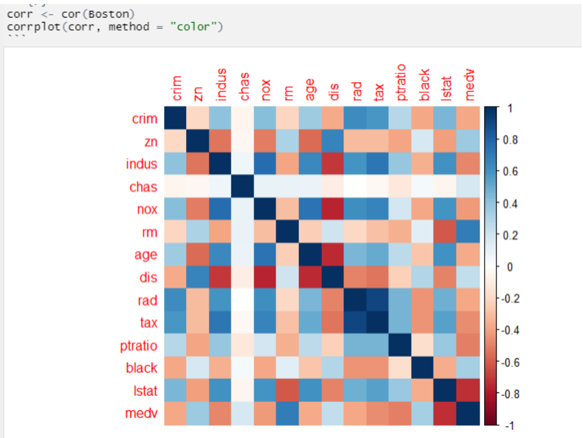
```
summary(Boston)
```

```
      crim                zn             indus            chas              nox               rm             age              dis              rad        
 Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000   Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130   Min.   : 1.000  
 1st Qu.: 0.08204   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000   1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100   1st Qu.: 4.000  
 Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000   Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207   Median : 5.000  
 Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917   Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795   Mean   : 9.549  
 3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000   3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188   3rd Qu.:24.000  
 Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000   Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127   Max.   :24.000  
      tax           ptratio          black            lstat            medv      
 Min.   :187.0   Min.   :12.60   Min.   :  0.32   Min.   : 1.73   Min.   : 5.00  
 1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38   1st Qu.: 6.95   1st Qu.:17.02  
 Median :330.0   Median :19.05   Median :391.44   Median :11.36   Median :21.20  
 Mean   :408.2   Mean   :18.46   Mean   :356.67   Mean   :12.65   Mean   :22.53  
 3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23   3rd Qu.:16.95   3rd Qu.:25.00  
 Max.   :711.0   Max.   :22.00   Max.   :396.90   Max.   :37.97   Max.   :50.00  
```

2. Check for missing values and use an appropriate technique to fill it in.
   The dataset has no missing values

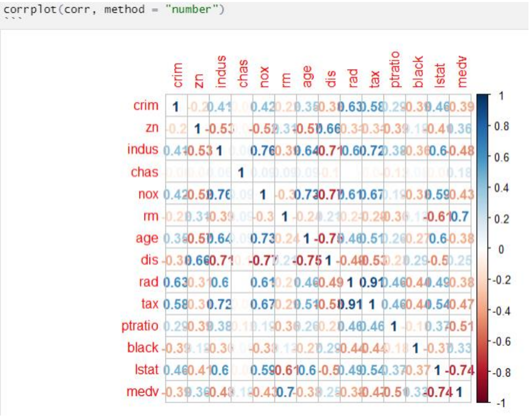
### 1.1 Correlation

1. Plot a correlogram and check the correlation between the variables. Check out the different parameters and types of plot.



```
corr <- cor(Boston)
corrplot(corr, method = "color")
```

We used the method “color” to print this plot which gives us the tentative correlation based on the shade



```
corrplot(corr, method = "number")
```

We used the method “number” here to visualise the Pearson correlation between the columns

You can check out the different methods and other parameters of corrplot here: https://cran.r-project.org/web/packages/corrplot/vignettes/corrplot-intro.html#:~:text=to%20corrplot%20Package-,Introduction,color%20labels%2C%20layout%2C%20etc.

2. What inferences can you draw from the correlogram?
   We can infer the direction and strength of the relationship between two random variables in the Boston housing dataset using the correlograms given above.

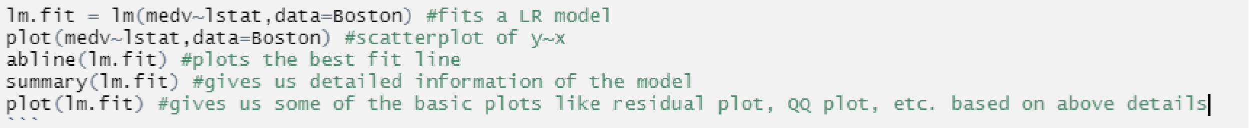
| Size of Correlation | Interpretation |
|---|---|
| .90 to 1.00 (−.90 to −1.00) | Very high positive (negative) correlation |
| .70 to .90 (−.70 to −.90) | High positive (negative) correlation |
| .50 to .70 (−.50 to −.70) | Moderate positive (negative) correlation |
| .30 to .50 (−.30 to −.50) | Low positive (negative) correlation |
| .00 to .30 (.00 to −.30) | negligible correlation |

3. Try finding the different correlation (spearman rank, point biserial, phi coefficient) from the data by dividing them into ordinal or binary data based on a certain range. What can you see from the output?
   Hint: Explore **phi, biserial.cor** and the method parameter of **cor.test**

## 1.2 Simple Linear Regression (SLR)

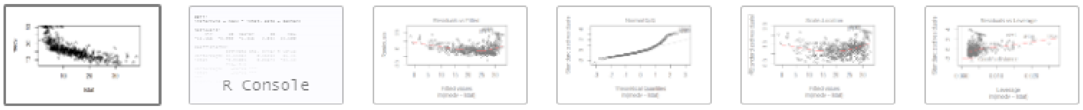
1. Explore the **lm** package in R for this section.
   https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/lm

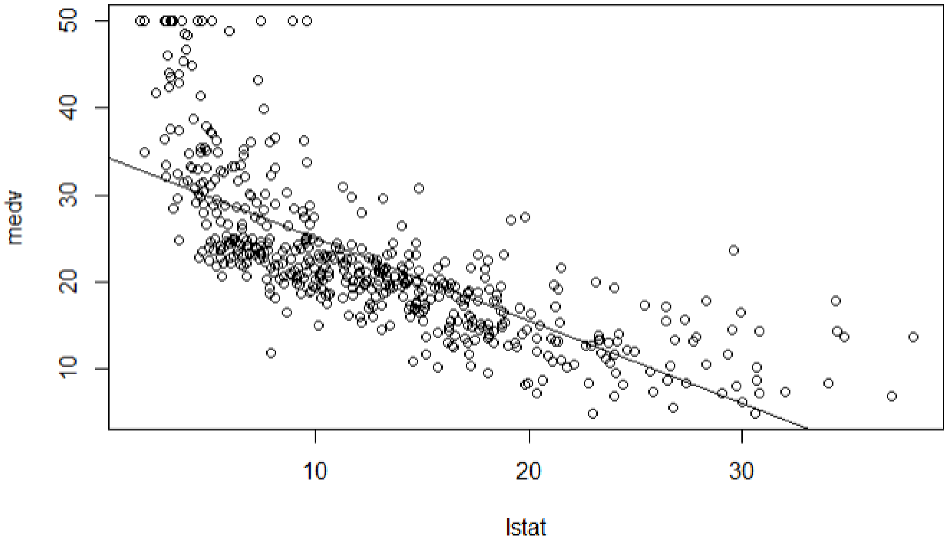
2. Plot the best fit line for all the variables with the target variable **medv.** Play around with the variables and map them to the concepts studied in theory.



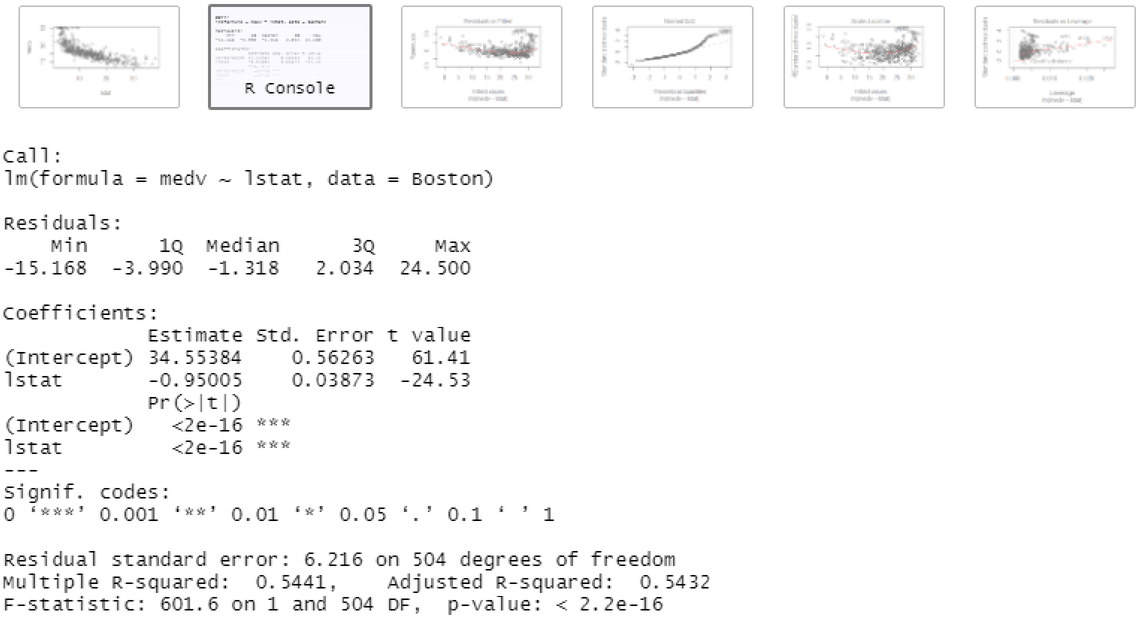
```
lm.fit = lm(medv~lstat,data=Boston) #fits a LR model
plot(medv~lstat,data=Boston) #scatterplot of y~x
abline(lm.fit) #plots the best fit line
summary(lm.fit) #gives us detailed information of the model
plot(lm.fit) #gives us some of the basic plots like residual plot, QQ plot, etc. based on above details
```

**Best-fit line:**



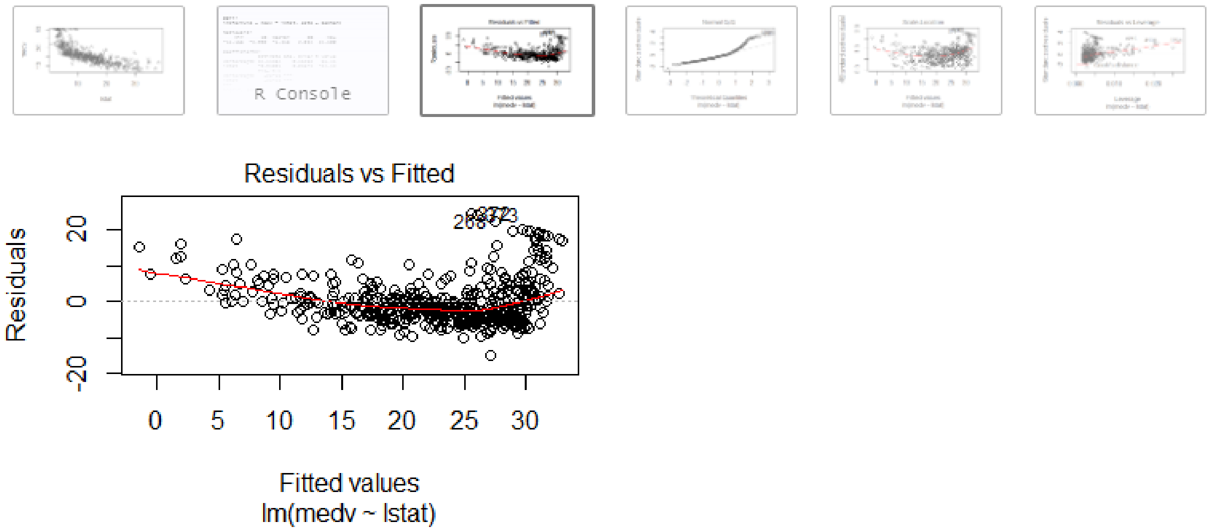


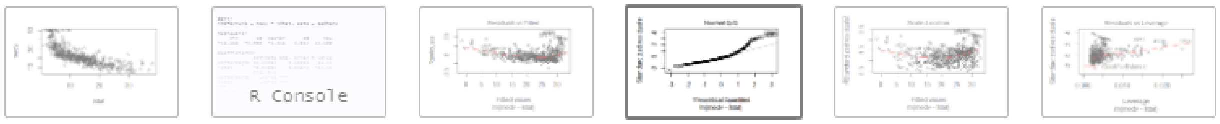
## Model Summary:

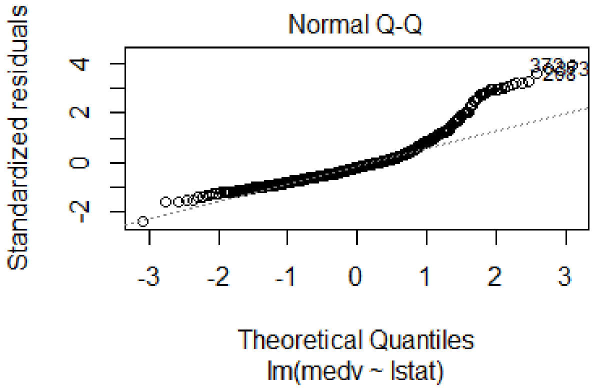


```
Call:
lm(formula = medv ~ lstat, data = Boston)

Residuals:
    Min      1Q  Median      3Q     Max 
-15.168  -3.990  -1.318   2.034  24.500 

Coefficients:
            Estimate Std. Error t value
(Intercept) 34.55384    0.56263   61.41
lstat       -0.95005    0.03873  -24.53
            Pr(>|t|)    
(Intercept)   <2e-16 ***
lstat         <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(lm.fit):







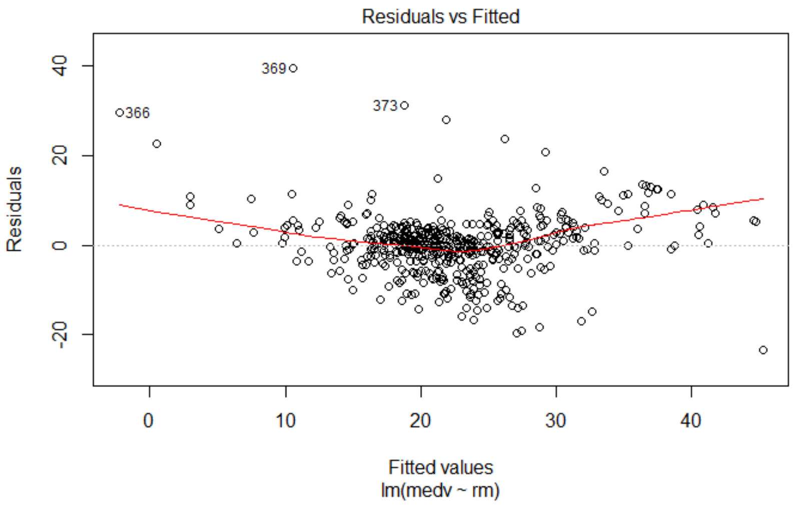
The other two plots are "Scale-Location" and "Residuals vs Leverage"
These plots are for lsat considering medv as target variable. Similarly, you can plot the graphs for the other variables as well.
You can also find out more about the parameters used here:

a. plot: https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/plot
b. abline: https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/abline

3. Plot the residual plots for all the variables as well. Which plot satisfies all the assumptions of linear regression?
   Hint: Use the **plot** function in R with appropriate parameters
   The above plot for lsat vs medv satisfies all assumptions. We can see that rm vs medv also satisfies the assumptions except we can see slight influence of outliers.
   We can also plot the residual plot using the function **plot(lm.fit, where = c(1))**

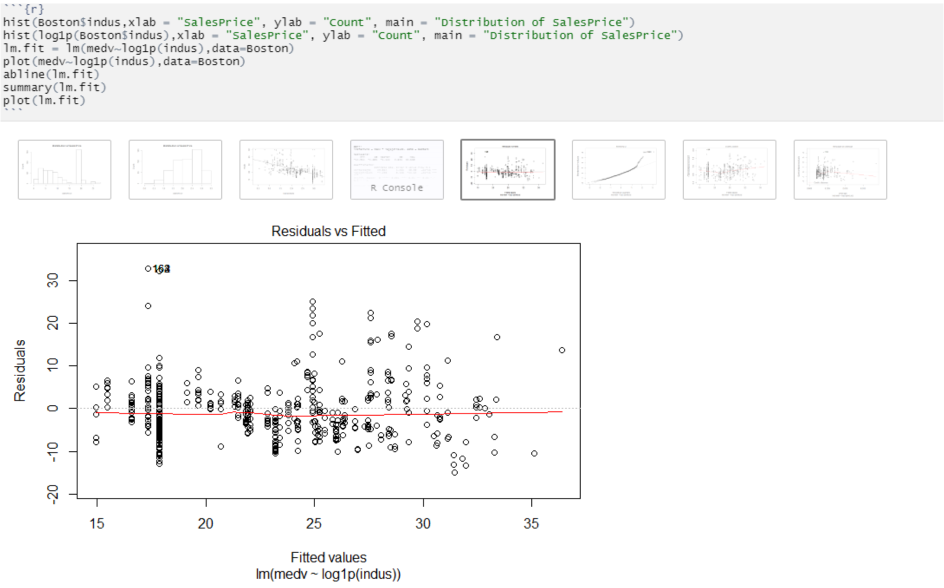
4. Find the best predictor for the target variable **medv** using the following methods:
   a. Correlation: Which is better - high correlation or low correlation?
      A higher correlation
      As seen above, lstat vs medv has a high -ve correlation and thus, it is a better predictor

   b. Residual plots
      We can see that the residual plots for rm vs medv is more randomly distributed than lstat vs medv even though lstat vs med has a higher correlation value



5. Is there any variable (apart from the best predictor) that can be transformed to fit the SLR model while satisfying all assumptions? If yes, perform the valid transformation and try to fit the SLR model again.
   Hint: Plot a histogram to draw inferences
   Plot histogram using the line of code:
   **hist(Boston$lstat,xlab = "Lower Status of Population", ylab = "Count", main = "Distribution of Lower Status of Population")**
   Performing transformations on the column like indus gives us a better fitted model by normalising the residuals which can be seen by the reduction in pattern in the residual plot. However, the result is not very significant.

Use the following lines of code to check:



~~~
```{r}
hist(Boston$indus,xlab = "SalesPrice", ylab = "Count", main = "Distribution of SalesPrice")
hist(log1p(Boston$indus),xlab = "SalesPrice", ylab = "Count", main = "Distribution of SalesPrice")
lm.fit = lm(medv~log1p(indus),data=Boston)
plot(medv~log1p(indus),data=Boston)
abline(lm.fit)
summary(lm.fit)
plot(lm.fit)
```
~~~

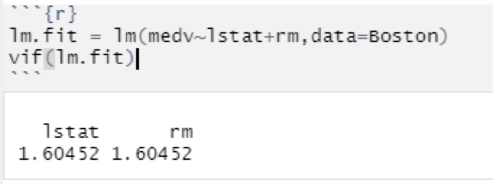
## 1.3 Multiple Linear Regression (MLR)

1. For this section also we will be using the **lm** package. Explore how we can use it for MLR models.
   lm(y~x1+x2+x3)
   https://www.tutorialspoint.com/r/r_multiple_regression.htm

2. Using correlation and residual plots from Section 1.2, decide the variables you want to keep.
   We can keep the variables lstat and rm as they have high correlation with medv and relatively low correlation with each other.
   We can also see that their respective residual plots are normally distributed.

3. Is there any multicollinearity?
   Hint: Use VIF to verify
   https://www.rdocumentation.org/packages/regclass/versions/1.6/topics/VIF
   When we take all the variables with medv, we see that the VIF value of tax is very high thus showing high multicollinearity.
   However, when we use lstat and rm with medv, the vif value is lower than 2 (1.6) which shows very less multicollinearity.

   Use the following lines of code to verify the above results:

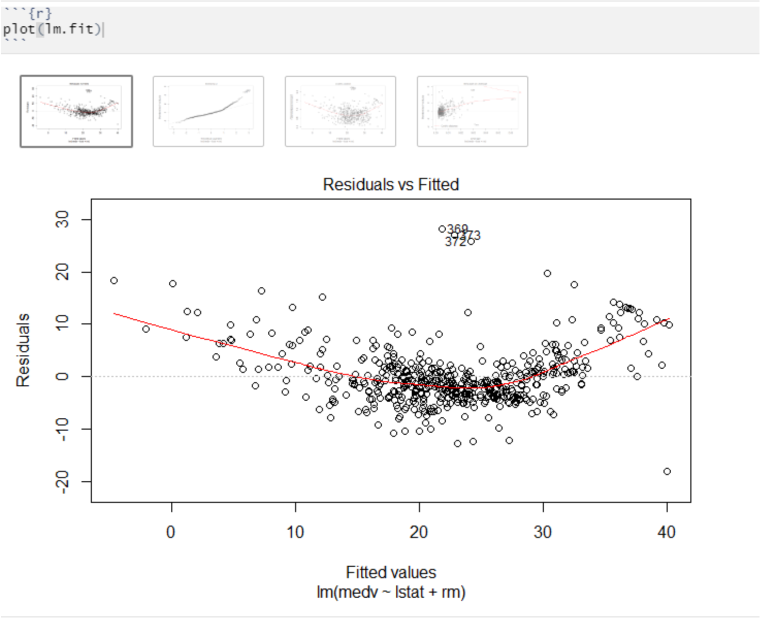


~~~
```{r}
lm.fit = lm(medv~lstat+rm,data=Boston)
vif(lm.fit)
```
~~~

```
   lstat      rm 
1.60452 1.60452 
```

4. Analyse the residual plot to ensure that all the assumptions of MLR are fulfilled.

As we can observe from the residual plot given below, there is a pattern in the distribution however they are randomly/ normally distributed towards the middle.
However, this is the best combination of variables and it gives us the most apt model.



~~~
```{r}
plot(lm.fit)
```
~~~

5. Plot the best fit line and compare it to SLR.
   Trick question: The best fit line for MLR will be multi-dimensional giving us no basis for comparison unless we reduce it to two dimensions. However, you can try it for your own learning but it is out of the scope of this course.

## WORKSHEET 2

### 2.0 Getting started

For this worksheet, we will be using the dataset and models from the previous worksheet (both SLR and MLR model).
The dataset can also be found here:
https://www.kaggle.com/altavish/boston-housing-dataset
The description of the dataset can be found here:
https://www.kaggle.com/c/boston-housing

### 2.1 R-square analysis

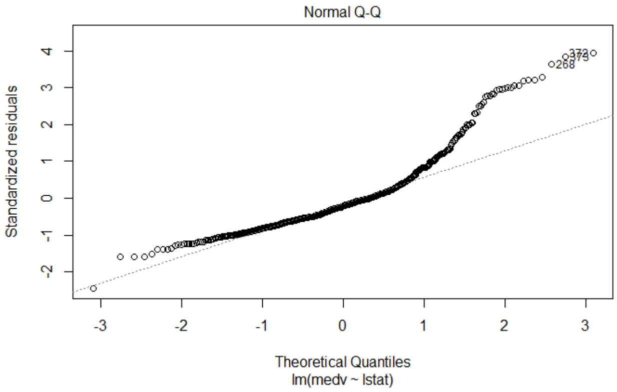
1. In section 1. 2 and 1.3 we calculated the best fit SLR and MLR models for the Boston Housing dataset. Calculate the R-square value of both the models and compare. What do you infer?
   We can use the function:
   **summary(lm.fit)**
   The R-square values are -

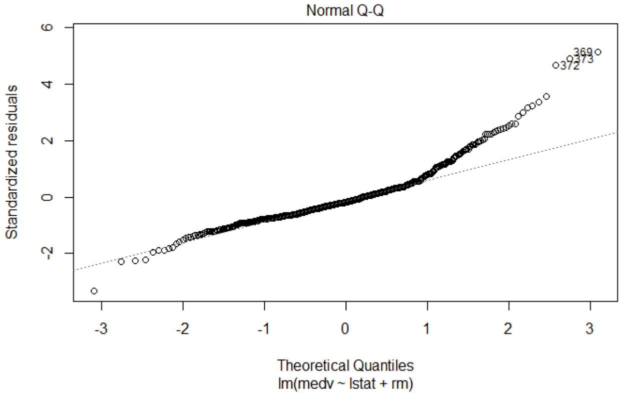
SLR: 0.5441
MLR: 0.6386
It tells us that the variability in the dependent variables that is predictable by the independent variables is explained better in the MLR model (higher R-square).

2. Calculate the adjusted R-square value too. Is there a large variation from R-square? If yes, why?
   The adjusted R-square values are -
   SLR: 0.5432
   MLR: 0.6371
   The variation of adjusted R-square from R-square is not too large for both models. This indicates that the new variable (rm) that is added is statistically insignificant.

## 2.2 Q-Q plot analysis

1. Analyse the residuals from the two models above and check if they are normally distributed or not.
   plot(lm.fit) also plots the Q-Q plot





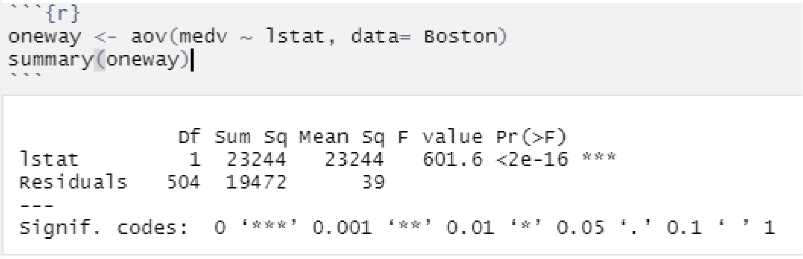
We can see that the Q-Q plot for the MLR model is more normally distributed as compared to that of the SLR model. Thus, the adjusted R-square value was slightly misleading and Q-Q plot gives us a clearer picture.

2. Plot the summary of both the models.

We can simply use **summary(lm.fit)** to see the summary of the model and **plot(lm.fit)** plots some of the basic models.

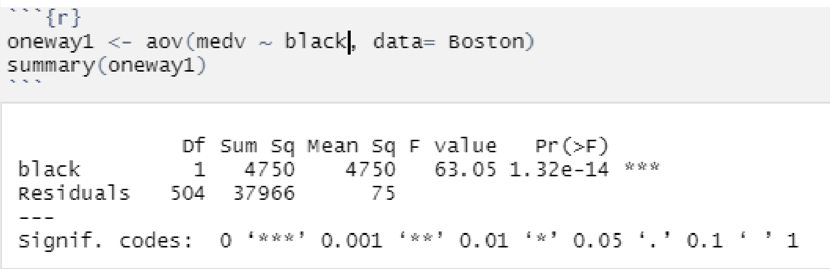
## 2.3 ANOVA

1. Find the one-way ANOVA of the dataset with the best predictor of SLR and another variable.
    a. Print the summary of both and compare them. List down the differences.



~~~
```{r}
oneway <- aov(medv ~ lstat, data= Boston)
summary(oneway)
```
~~~

```
              Df Sum Sq Mean Sq F value Pr(>F)    
lstat          1  23244   23244   601.6 <2e-16 ***
Residuals    504  19472      39                   
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```



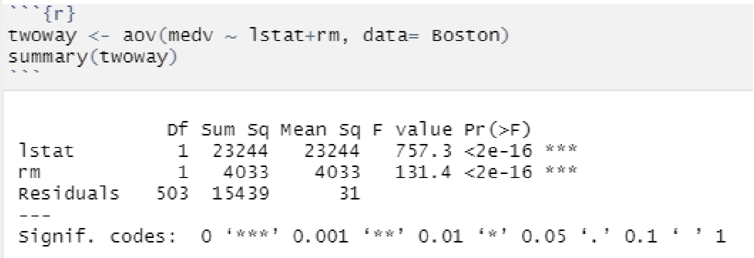
~~~
```{r}
oneway1 <- aov(medv ~ black, data= Boston)
summary(oneway1)
```
~~~

```
              Df Sum Sq Mean Sq F value   Pr(>F)    
black          1   4750    4750   63.05 1.32e-14 ***
Residuals    504  37966      75                     
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

We can see that in both the models the pvalue is less than 0.001 which implies that lstat and black have an impact on medv.
However, the other factors come into play then. The larger the F value, the more likely it is that the variation caused by the independent variable is real and not due to chance. Since, lstat has a higher F value than black, it is more significant.

    b. Play around with the parameters and see how it influences the output.
    You can learn more about the parameters in aov using:
    https://www.scribbr.com/statistics/anova-in-r/
    https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/aov

2. Find the two-way ANOVA of the dataset for the variables you think are most significant based on correlation.
Print the summary of two of them and compare the results.



~~~
```{r}
twoway <- aov(medv ~ lstat+rm, data= Boston)
summary(twoway)
```
~~~

```
              Df Sum Sq Mean Sq F value Pr(>F)    
lstat          1  23244   23244   757.3 <2e-16 ***
rm             1   4033    4033   131.4 <2e-16 ***
Residuals    503  15439      31                   
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

We can see that in both the variables the p-value is less than 0.001 which implies that lstat and rm have an impact on medv.

## 2.4 SLR with Gradient Descent

1. Set the SSE as the loss function and using the concept of SLR with Gradient Descent find the optimal values of m and c.
   You can use the following tutorials as reference for this question:
   https://rstudio-pubs-static.s3.amazonaws.com/159940_4a7d620cb4e0460486c364281cdf5780.html
   https://www.r-bloggers.com/linear-regression-by-gradient-descent/
   You can use the code snippet given below for 100 epochs and 0.0001 learning rate.
   a. Start with m = 0 and c = 0, learning rate = 0.01, 0.001
   b. Run it for 250, 500 and 1000 epochs, check the R-square at each point.
   c. Draw inferences from the R-square value and reason out as to why we get those values.