In this project, we performed multiple exploratory and pre-processing techniques:

We loaded the data using the command read.table. Then we looked at the summary statistics our data. The summary statistics provide values such as the mean, standard deviation, the maximum, and the minimum value of each column in our data set. For instance, the minimum value for UV is 0.0 and the maximum value for log10APC is 5.760:

> D <- read.table("C:/Users/BFOUA/Desktop/AssignmentdataforEDA.txt", header=TRUE)

> summary(D)

UV Log10APC

Min. :0.0000 Min. :1.000

1st Qu.:0.1159 1st Qu.:2.534

Median :0.1543 Median :2.929

Mean :0.1613 Mean :3.111

3rd Qu.:0.2108 3rd Qu.:3.582

Max. :0.4444 Max. :5.760

We also find the number of rows (100) in our data set to see how large our data set is. The mean for each column is respectively 0.16 (for UV) and 3.11 (for Log10APC). The standard deviation for each column is respectively 0.08 (for UV) and 0.98 ( for Log10APC). As a reminder the mean calculate the average value in each column and the standard deviation is the amount of dispersion each column:

> nrow(D)

[1] 100

> mean(D$UV)

[1] 0.161298

> sd(D$UV)

[1] 0.07652462

> mean(D$Log10APC)

[1] 3.111044

> sd(D$Log10APC)

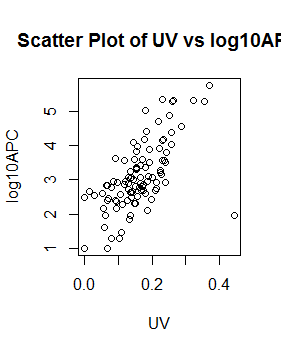
[1] 0.9772074

We can look at relations between values by plotting them. The following scatter plot will show us the correlation between the UV and Log10APC. To find the correlation coefficient we use cor() :

> plot(D$UV,D$Log10APC, main = "Scatter Plot of UV vs log10APC", xlab="UV",ylab = "log10APC")

> cor(D$UV,D$Log10APC)

[1] 0.6352417



The correlation coefficient is + 0.63. It shows a positive relationship between the two variables. Meaning that a change in one variable will likely impact the other variable. As we can see on the graph, as UV increases Log10APC also increases.

Now we look at a run chart of each column. A run chart will provide us with change of each column overtime. To plot each column run chart, we pick a few values in our data set:

> smallD <- D[D$UV<0.11,]

> plot(smallD$Log10APC, type="b")

> abline(h=mean(smallD$Log10APC))

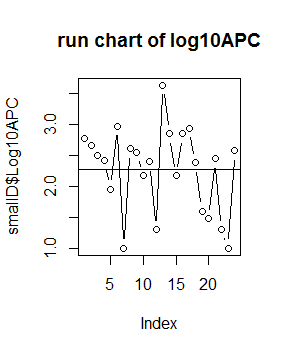
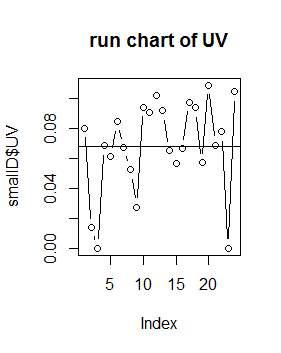
> title(main="run chart of log10APC")

> smallD2 <- D[D$Log10APC<3,]

> plot(smallD$UV, type="b")

> abline(h=mean(smallD$UV))

> title(main="run chart of UV")



The run shows that for UV are in between 0.04 and 0.12 and most values in the Log10APC columns are in between 2.0 and 3.0.

The next analysis will show us the histogram plots of each columns. The histogram charts create groups within our data:

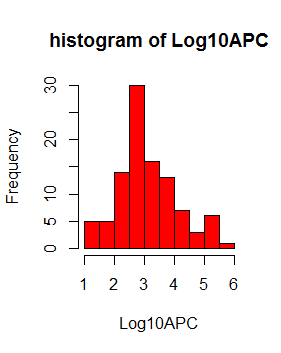
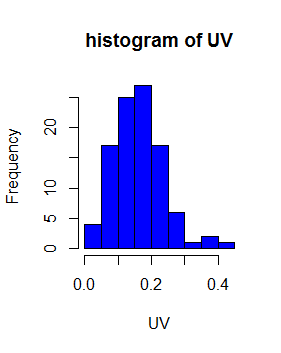
> hist(D$UV, main="histogram of UV", xlab = "UV", ylab = "Frequency", col="blue")

> hist(D$Log10APC, main="histogram of Log10APC", xlab = "Log10APC", ylab = "Frequency", col="red")

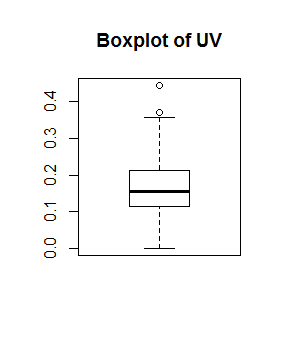
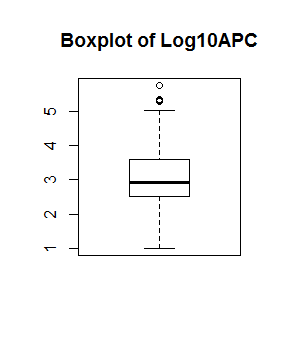
> plot(D$UV,D$Log10APC)

> plot(D$UV,D$Log10APC, main = "Scatter Plot of UV vs log10APC", xlab="UV",ylab = "log10APC")

The histogram charts show better visualization categories in our data then the run charts we plotted earlier. As we can see the most UV values are in between 0.05 and 0.25 (not 0.04 and 0.08!) and values for Log10APC are between 2 and 4 (not 2 and 3!). Although, to be fair we only plot a few values in the run charts, it will likely get all these raring correct if we plot all the values in the run chart. We should however note that the distributions of values in our histogram show that both histograms are skewed right, because their distribution considerably drawn out on the right side.



When analyzing all the graphs, some of our values seems to be outside our trends and patterns. Our guess is there are outliers. We will use boxplot to locate our outliers:



Our doubts are there are outliers in both columns. The outliers in Log10APc are above 5, and the outliers in Uv are above 0.35(or almost!).

Now we can split the data set into a test set and train. We will use 60 % of the data to train and 40% to test:

> set.seed(17)

> R=runif(100)

> D$R = R

> D$type<-ifelse(D$R<=.6,"train","test")

> train <- D[D$type=="train",]

> test <- D[D$type=="test",]

View(D)

**Full code R code**

R version 3.6.1 (2019-07-05) -- "Action of the Toes"

Copyright (C) 2019 The R Foundation for Statistical Computing

Platform: x86\_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.

You are welcome to redistribute it under certain conditions.

Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.

Type 'contributors()' for more information and

'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or

'help.start()' for an HTML browser interface to help.

Type 'q()' to quit R.

> D <- read.table("C:/Users/BFOUA/Desktop/AssignmentdataforEDA.txt", header=TRUE)

> summary(D)

UV Log10APC

Min. :0.0000 Min. :1.000

1st Qu.:0.1159 1st Qu.:2.534

Median :0.1543 Median :2.929

Mean :0.1613 Mean :3.111

3rd Qu.:0.2108 3rd Qu.:3.582

Max. :0.4444 Max. :5.760

> nrow(D)

[1] 100

> mean(D$UV)

[1] 0.161298

> sd(D$UV)

[1] 0.07652462

> mean(D$Log10APC)

[1] 3.111044

> sd(D$Log10APC)

[1] 0.9772074

> plot(D$UV,D$Log10APC)

> plot(D$UV,D$Log10APC, main = "Scatter Plot of UV vs log10APC", xlab="UV",ylab = "log10APC")

> cor(D$UV,D$Log10APC)

[1] 0.6352417

> smallD <- D[D$UV<0.11,]

> plot(smallD$Log10APC, type="b")

> abline(h=mean(smallD$Log10APC))

> title(main="run chart of log10APC")

> smallD2 <- D[D$Log10APC<3,]

> plot(smallD$UV, type="b")

> abline(h=mean(smallD$UV))

> title(main="run chart of UV")

> hist(D$UV, main="histogram of UV", xlab = "UV", ylab = "Frequency", col="blue")

> hist(D$Log10APC, main="histogram of Log10APC", xlab = "Log10APC", ylab = "Frequency", col="red")

> plot(D$UV,D$Log10APC)

> plot(D$UV,D$Log10APC, main = "Scatter Plot of UV vs log10APC", xlab="UV",ylab = "log10APC")

> boxplot(D$UV, main = "Boxplot of UV")

> boxplot(D$Log10APC, main = "Boxplot of Log10APC")

> boxplot(D$Log10APC~D$UV, main = "Boxplot of attributes")

> boxplot(D$Log10APC,D$UV, main = "two boxplots for comparison")

> set.seed(17)

> R=runif(100)

> D$R = R

> D$type<-ifelse(D$R<=.6,"train","test")

> train <- D[D$type=="train",]

> test <- D[D$type=="test",]

View(D)

**Graphs**

