1. EDA Using R , Minitab and Excel
2. Hypothesis Testing documentation
3. Simple Linear Regression Document using R
4. Multiple Linear Regression Document using R
5. Logistic Regression
6. Machine Learning –H-Clustering
7. **EDA Using R:**

Taking mba dataset for EDA.

1. **First Business Moment Decision**:
2. Calculate Mean:

**mean(mba$gmat)**

1. Calculate Median:

**median(mba$gmat)**

1. Calculate Mode:

**temp<-table(as.vector(mba$gmat))**

**temp**

**names(temp)[temp == max(temp)]**

1. **Second Business Moment Decision:**
2. Calculate Variance:

**var(mba$gmat)**

1. Calculate Standard Deviation.

**sd(mba$gmat)**

1. Calculate Range =Max-Min:

**X<-max(mba$gmat)**

**x**

**y<-min(mba$gmat)**

**y**

**r1<-x-y**

**r1**

**(Or)**

**range(max(mba$gmat) - min(mba$gmat))**

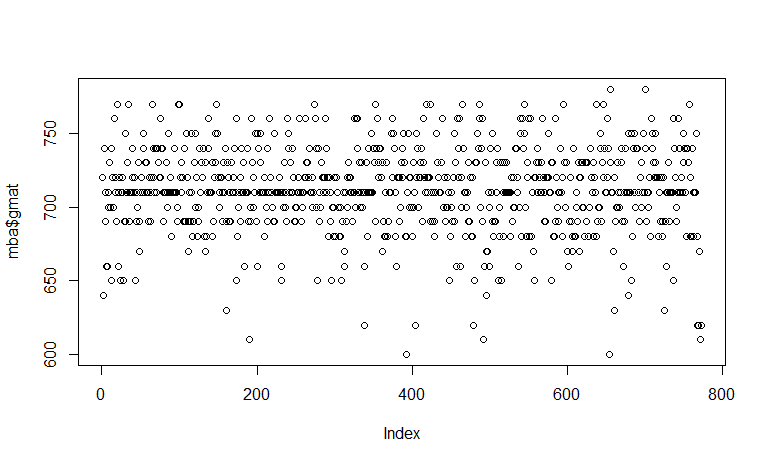
1. **Third & Fourth Business Moment Decision:**
2. Calculate Skewness & Kurtosis: For this ,we have to install the package e1071

**skewness(mba$gmat)🡪 Negative Skewness**

**kurtosis(mba$gmat)🡪 Positive Kurtosis**

1. **Various Graphical Representations:**
2. Plot

**plot(mba$gmat)**

****

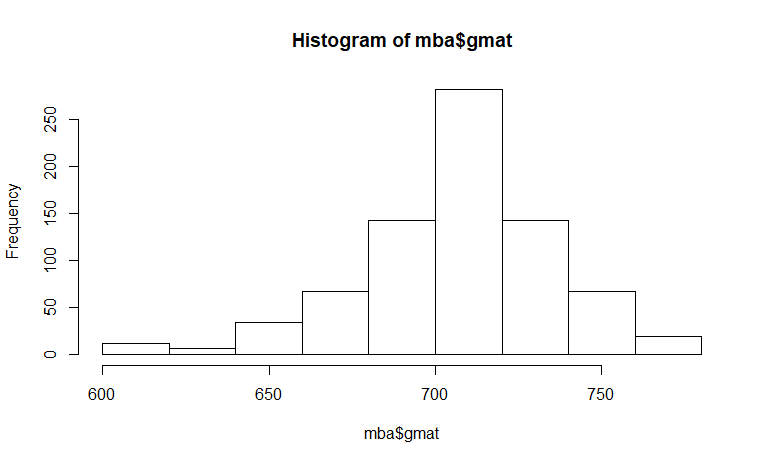
1. Bar Plot

**barplot(mba$gmat)**



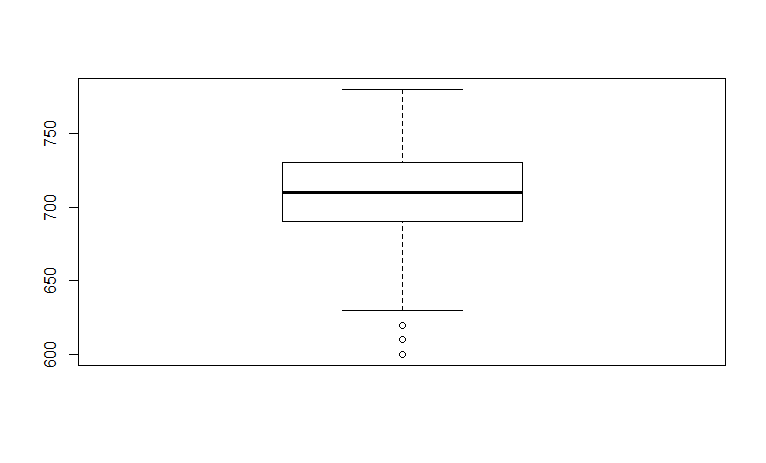
1. Histogram

**hist(mba$gmat)**



1. BoxPlot

**boxplot(mba$gmat)**



1. **Calculate Outliers Lying between:**

**summary(mba$gmat)**

Min. 1st Qu. Median Mean 3rd Qu. Max.

600.0 690.0 710.0 711.2 730.0 780.0

**Q1<-690**

**Q1**

**Q3<-730**

**Q3**

**IQR <- Q3-Q1**

**IQR =**40

**lowvalue <- Q1-1.5\*IQR**

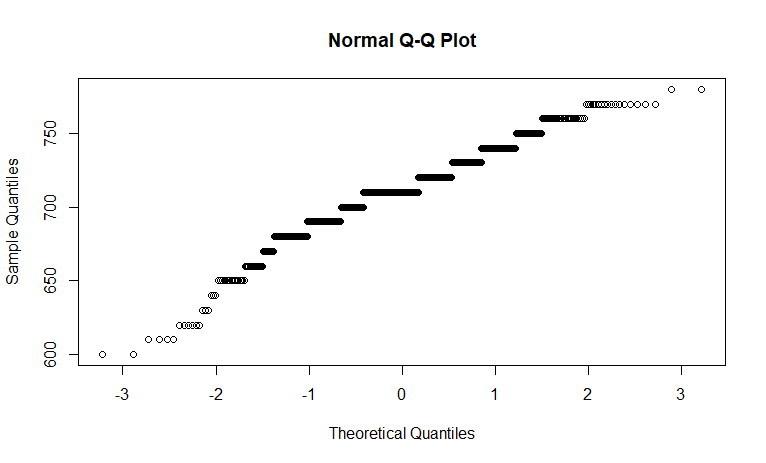
**lowvalue = 630**

**Highvalue <- Q3+1.5\*IQR**

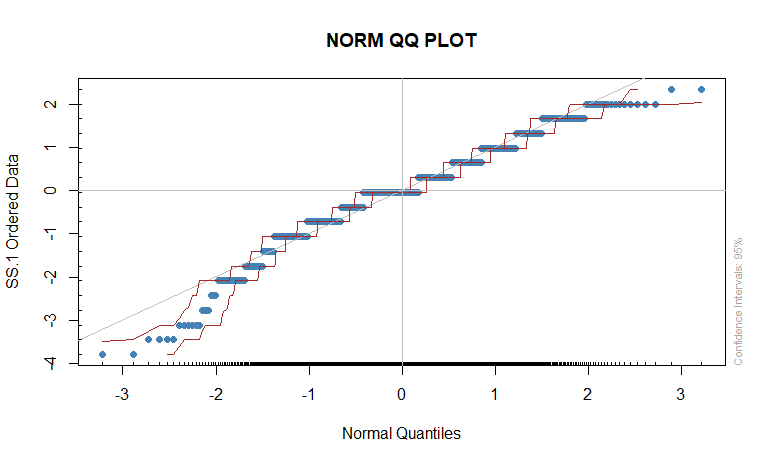
**Highvalue = 790**

**Note: Outliers lying between 630 – 790**

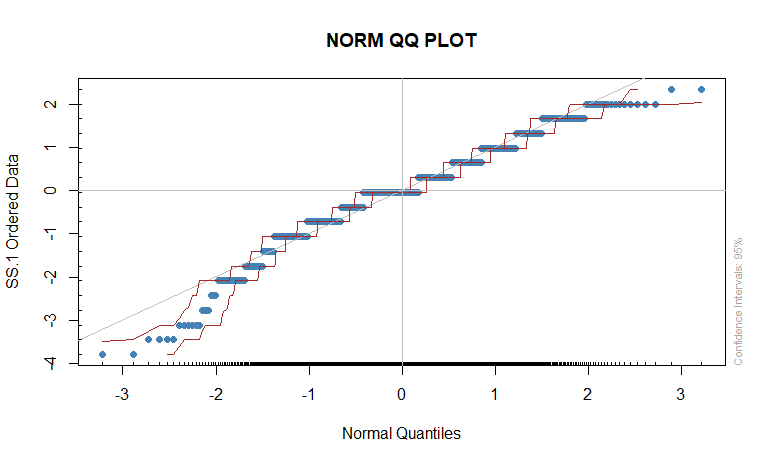
1. **Calculate Normalization:**
2. **qqnorm(mba$gmat)**



1. **qqnormPlot(mba$gmat) # calculate the plot area of Normalization**



1. **qqline(mba$gmat)**



1. **Calculate Confident Level.**

**qnorm(.90) =**  1.281552

**qnorm(.95)** = 1.644854

**qnorm(.99)** = 2.326348

**qt(.90,711)** = 1.282743

**qt(.95,711)** = 1.647

**qt(.99,711)** = 2.331604

**Simple Linear Regression**

1. **Performing Explanatory Data Analysis (EDA)**

**basicStats(wc\_at)**

basicStats(wc\_at)

Waist AT

nobs 109.000000 109.000000

NAs 0.000000 0.000000

Minimum 63.500000 11.440000

Maximum 121.000000 253.000000

1. Quartile 80.000000 50.880000

3. Quartile 104.000000 137.000000

Mean 91.901835 101.894037

Median 90.800000 96.540000

Sum 10017.300000 11106.450000

SE Mean 1.298728 5.487843

LCL Mean 89.327531 91.016180

UCL Mean 94.476139 112.771894

Variance 183.849626 3282.689835

Stdev 13.559116 57.294763

Skewness 0.130389 0.568870

Kurtosis -1.141846 -0.376006

**(Or)**

**summary(wc\_at)**

summary(wc\_at)

Waist AT

Min. : 63.5 Min. : 11.44

1st Qu.: 80.0 1st Qu.: 50.88

Median : 90.8 Median : 96.54

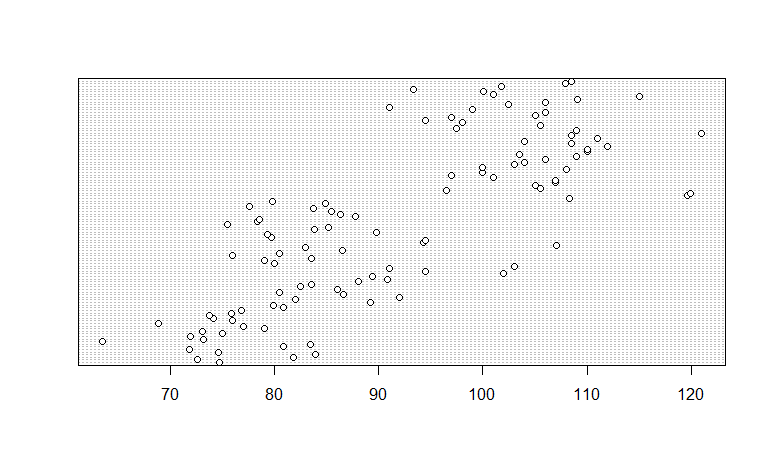
Mean : 91.9 Mean :101.89

3rd Qu.:104.0 3rd Qu.:137.00

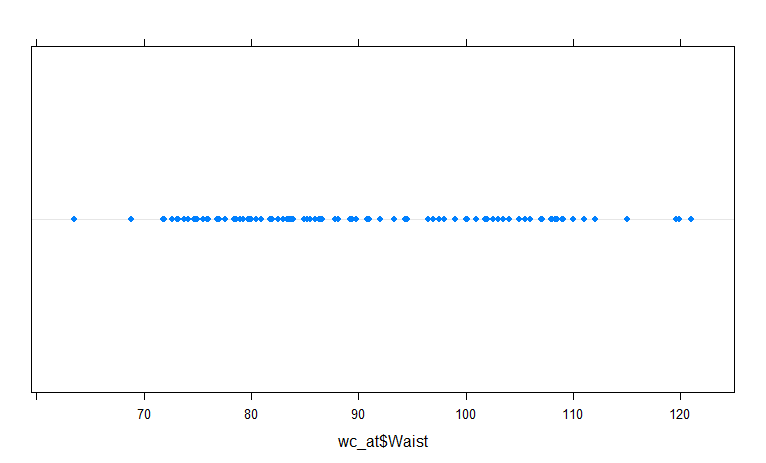
Max. :121.0 Max. :253.00

1. **Graphical Exploration**:

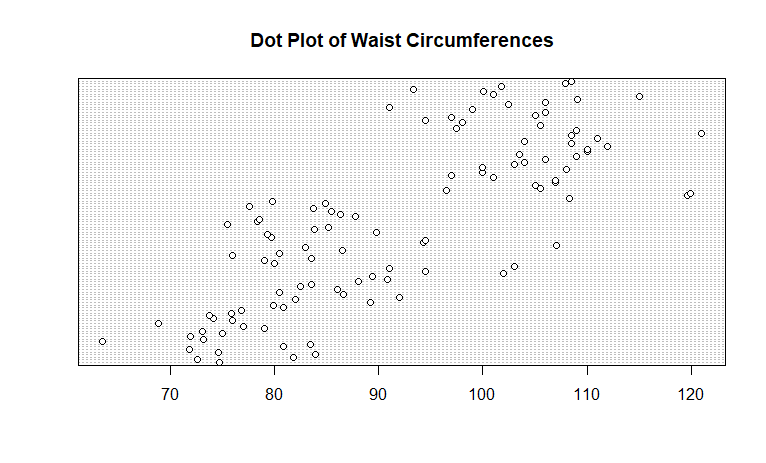
**dotchart(wc\_at$Waist)**



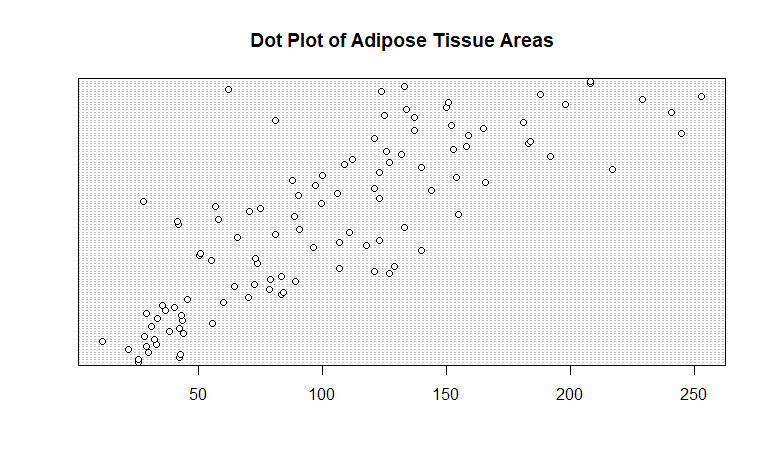
**dotplot(wc\_at$Waist)**



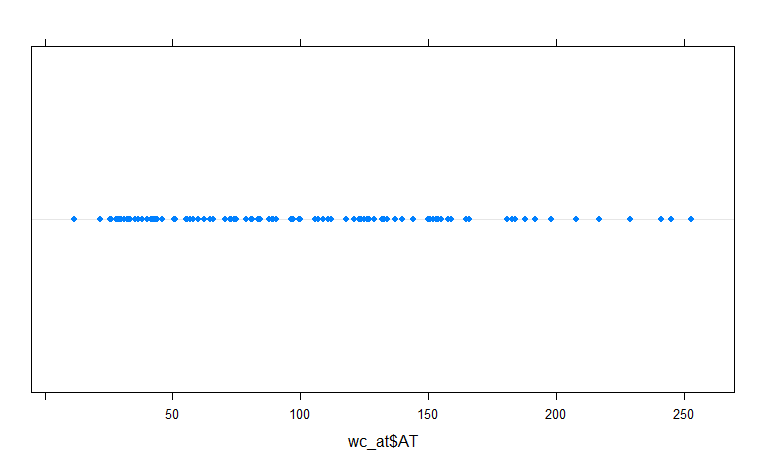
**dotchart(wc\_at$Waist, main="Dot Plot of Waist Circumferences")**



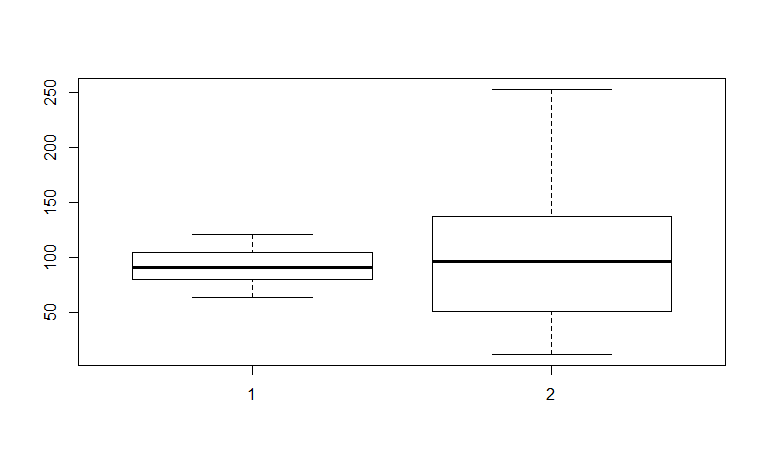
**dotchart(wc\_at$AT, main="Dot Plot of Adipose Tissue Areas")**



**dotplot(wc\_at$AT)**



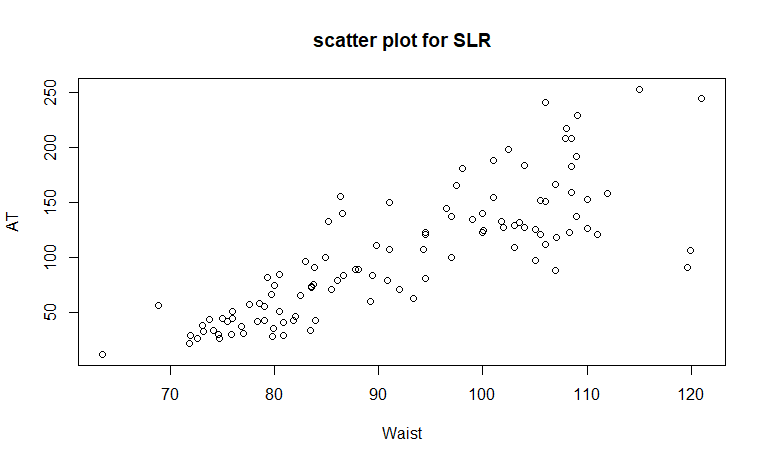
**boxplot(wc\_at$Waist , wc\_at$AT)**



1. **Scatter Plot:**

**attach(wc\_at)**

**plot(Waist,AT,main = "scatter plot for SLR")**



1. **Calculate Correlation Coefficient**

**cor(wc\_at$Waist , wc\_at$AT)**

cor(wc\_at$Waist , wc\_at$AT) = 0.8185578

Now R-value is greater than 0.85 then both columns are in strong relationship

1. **Calculate Linear Regression:**

**reg <- lm(AT ~ Waist , data = wc\_at)**

**summary(reg)**

**OutPut:**

summary(reg)

Call:

lm(formula = AT ~ Waist, data = wc\_at)

Residuals:

Min 1Q Median 3Q Max

-107.288 -19.143 -2.939 16.376 90.342

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -215.9815 21.7963 -9.909 <2e-16 \*\*\*

Waist 3.4589 0.2347 14.740 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 33.06 on 107 degrees of freedom

Multiple R-squared: 0.67, Adjusted R-squared: 0.667

F-statistic: 217.3 on 1 and 107 DF, p-value: < 2.2e-16

**confint(reg,level = 0.95)**

confint(reg,level = .95)

2.5 % 97.5 %

(Intercept) -259.190053 -172.77292

Waist 2.993689 3.92403

**predict(reg,level = "predict")**

predict(reg,level = "predict")

1 2 3 4 5 6 7 8 9

42.568252 35.131704 66.953210 74.389758 42.222366 32.537559 63.840237 72.487385 3.656083

10 11 12 13 14 15 16 17 18

37.207020 32.710502 43.432966 36.861134 57.268404 50.350685 22.160981 46.718883 40.492936

19 20 21 22 23 24 25 26 27

39.282335 46.545940 49.831856 63.840237 60.381377 92.548770 67.644982 102.233576 83.555735

28 29 30 31 32 33 34 35 36

62.456693 81.480420 69.374412 72.833271 88.744024 98.082945 93.240542 136.822170 110.880725

37 38 39 40 41 42 43 44 45

98.774717 140.281029 60.727263 57.268404 72.833271 46.891826 62.456693 83.209849 71.103842

46 47 48 49 50 51 52 53 54

154.462353 110.188953 110.880725 59.689606 58.306062 94.624085 73.870929 78.713332 45.162396

55 56 57 58 59 60 61 62 63

55.193088 55.884860 87.706367 82.518078 79.750990 73.525043 52.426001 77.675674 60.035492

64 65 66 67 68 69 70 71 72

158.612984 197.698095 198.735753 117.798443 148.928178 147.198748 154.116467 154.116467 133.363311

73 74 75 76 77 78 79 80 81

119.527873 129.904451 157.575326 129.904451 140.281029 143.739889 150.657608 161.034186 142.010459

82 83 84 85 86 87 88 89 90

164.493045 164.493045 171.410764 159.304756 143.739889 167.951905 159.304756 202.540498 161.034186

91 92 93 94 95 96 97 98 99

121.257303 148.928178 122.986732 110.880725 119.527873 147.198748 150.657608 126.445592 98.774717

100 101 102 103 104 105 106 107 108

138.551600 150.657608 161.380072 181.787342 133.363311 130.250337 106.730093 136.130398 157.229440

109

159.304756

**reg\_log<-lm(AT ~ log(Waist), data = wc\_at)**

**summary(reg\_log)**

**confint(reg\_log,level = 0.95)**

**predict(reg\_log , level = "predict")**

**Output:**

summary(reg\_log)

Call:

lm(formula = AT ~ log(Waist), data = wc\_at)

Residuals:

Min 1Q Median 3Q Max

-98.473 -18.273 -2.374 14.538 90.400

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -1328.34 95.92 -13.85 <2e-16 \*\*\*

log(Waist) 317.14 21.26 14.92 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 32.8 on 107 degrees of freedom

Multiple R-squared: 0.6753, Adjusted R-squared: 0.6723

F-statistic: 222.6 on 1 and 107 DF, p-value: < 2.2e-16

**confint(reg\_log,level = 0.95)**

confint(reg\_log,level = 0.95)

2.5 % 97.5 %

(Intercept) -1518.4980 -1138.1860

log(Waist) 274.9936 359.2775

**reg\_exp<-lm(log(AT) ~ Waist , data = wc\_at)**

**summary(reg\_exp)**

**confint(reg\_exp , level = 0.95)**

**predict(reg\_exp , level = "predict")**

**Multiple Linear Regression**

1. **Calculate EDA**

**basicStats(Cars)**

basicStats(Cars)

HP MPG VOL SP WT

nobs 81.000000 81.000000 81.000000 81.000000 81.000000

NAs 0.000000 0.000000 0.000000 0.000000 0.000000

Minimum 49.000000 12.101263 50.000000 99.564907 15.712859

Maximum 322.000000 53.700681 160.000000 169.598513 52.997752

1. Quartile 84.000000 27.856252 89.000000 113.829145 29.591768

3. Quartile 140.000000 39.531633 113.000000 126.404312 37.392524

Mean 117.469136 34.422076 98.765432 121.540272 32.412577

Median 100.000000 35.152727 101.000000 118.208698 32.734518

Sum 9515.000000 2788.188134 8000.000000 9844.762047 2625.418730

SE Mean 6.345945 1.014605 2.477944 1.575715 0.832535

LCL Mean 104.840303 32.402947 93.834166 118.404500 30.755780

UCL Mean 130.097968 36.441204 103.696698 124.676044 34.069374

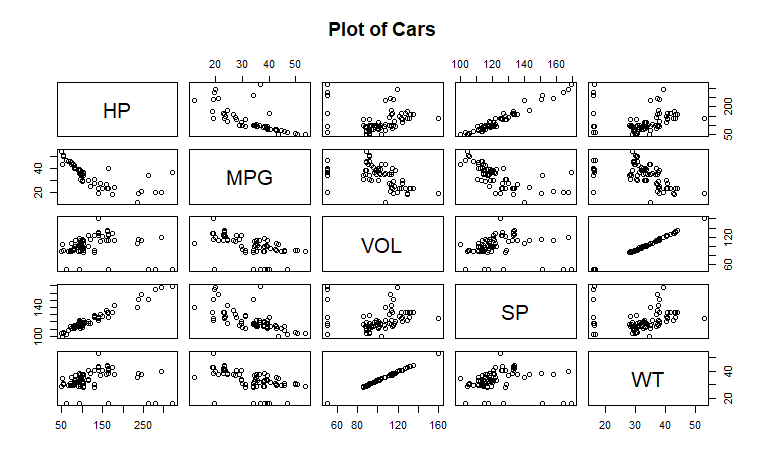
Variance 3261.952160 83.383283 497.356790 201.113002 56.142247

Stdev 57.113502 9.131445 22.301497 14.181432 7.492813

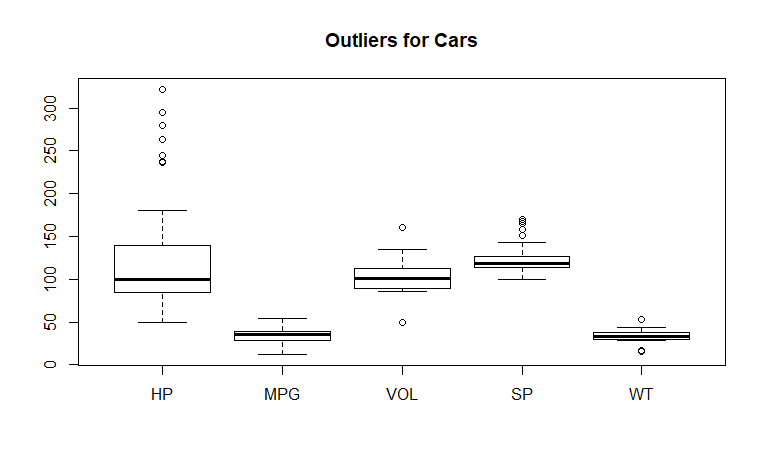
Skewness 1.653176 -0.171410 -0.568518 1.552258 -0.592172

Kurtosis 2.567217 -0.705460 0.698194 2.583072 0.725740

1. **Graphical Exploration:**

**plot(Cars , main = "Plot of Cars")**

**boxplot(Cars , main = "Outliers for Cars")**



1. **Calculate Correlation Coefficient:**

**cor(Cars$HP , Cars$VOL)**

**cor(Cars$HP , Cars$SP)**

**cor(Cars$HP , Cars$WT)**

**cor(Cars$VOL ,Cars$SP)**

**cor(Cars$VOL , Cars$WT)**

**cor(Cars$MPG , Cars$HP+Cars$VOL+Cars$SP+Cars$WT)**

**cor(Cars)**

**Output:**

cor(Cars$HP , Cars$VOL)

[1] 0.07745947

>cor(Cars$HP , Cars$SP)

[1] 0.9738481

>cor(Cars$HP , Cars$WT)

[1] 0.07651307

>cor(Cars$VOL , Cars$SP)

[1] 0.10217

>cor(Cars$VOL , Cars$WT)

[1] 0.9992031

>cor(Cars$MPG , Cars$HP+Cars$VOL+Cars$SP+Cars$WT)

[1] -0.8443815

>cor(Cars)

HP MPG VOL SP WT

HP 1.00000000 -0.7250383 0.07745947 0.9738481 0.07651307

MPG -0.72503835 1.0000000 -0.52905658 -0.6871246 -0.52675909

VOL 0.07745947 -0.5290566 1.00000000 0.1021700 0.99920308

SP 0.97384807 -0.6871246 0.10217001 1.0000000 0.10243919

WT 0.07651307 -0.5267591 0.99920308 0.1024392 1.00000000

1. **Calculate Regression :**

**attach(Cars)**

**model.car <- lm(MPG ~ SP+VOL+WT+HP)**

**summary(model.car)**

**confint(model.car ,level = 0.95)**

**predict(model.car , level = "predict")**

**model.carv <- lm(MPG ~ VOL)**

**summary(model.carv)**

**confint(model.carv ,level = 0.95)**

**predict(model.carv , level = "predict")**

**model.cars <-lm(MPG ~ SP)**

**summary(model.cars)**

**confint(model.cars ,level = 0.95)**

**predict(model.cars , level = "predict")**

**model.carw <- lm(MPG ~ WT)**

**summary(model.carw)**

**confint(model.carw ,level = 0.95)**

**predict(model.carw , level = "predict")**

**model.carh <- lm(MPG ~ HP)**

**summary(model.carh)**

**confint(model.carh ,level = 0.95)**

**predict(model.carh , level = "predict")**

**model.carvw <- lm(MPG ~ VOL+WT)**

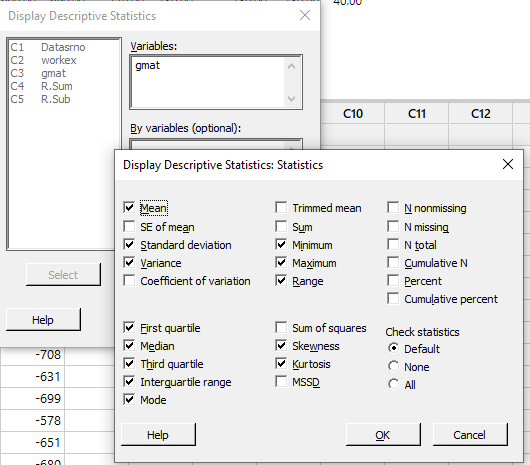
**summary(model.carvw)**

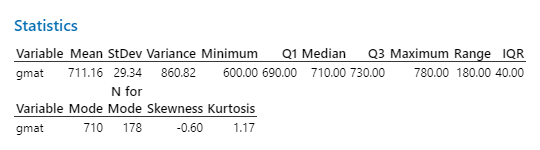
**influence.measures(model.car)**

**Explanatory Data Analysis (EDA) Using Minitab**

1. **Basic Statistics from Minitab:**

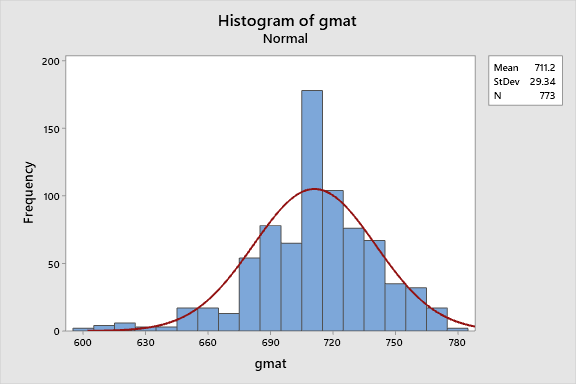
Go to Minitab 🡪 select “Stat” 🡪select “Basic Statistics” 🡪 select “Display Descriptive statistics” 🡪 select all basic statistics



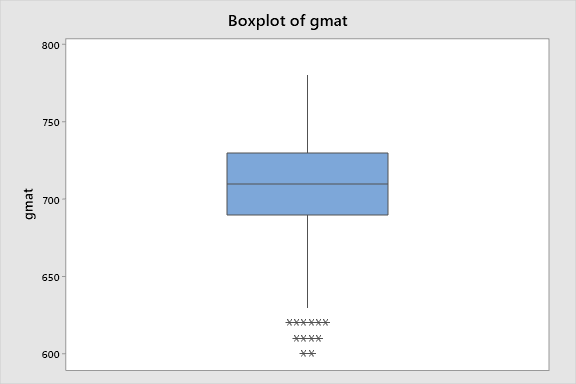


1. **Graphical Representation:**

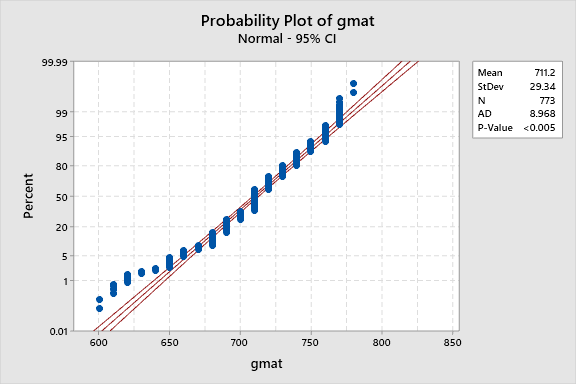
Go to Minitab 🡪 select “Graph” 🡪 select “Histogram”



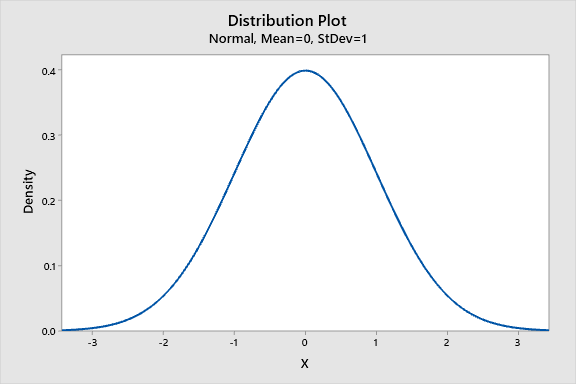
Go to Minitab 🡪 select “Graph” 🡪 select “Box plot”



Go to Minitab 🡪 select “Graph” 🡪 select “Probability Plot”

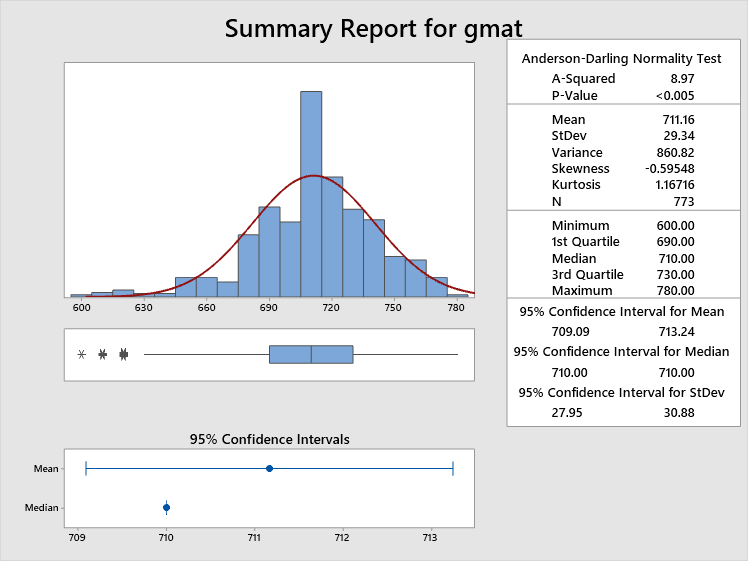


Go to Minitab 🡪 select “Graph” 🡪 select “Probability distribution Plot”



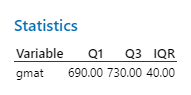
**(OR)**

**We will see all graphical representations using “Go to Minitab 🡪 select “stat”🡪 select “Basic statistics” 🡪 select “ Graphical Summary”**

****

1. **Calculate Outliers lying between:**

**IQR :**

****

**Low Value = Q1-1.5\*IQR**

**High Value = Q3+1.5\*IQR**

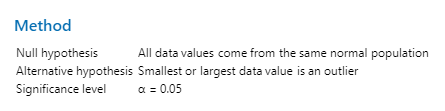
**r1 =1.5\*IQR Low Value =Q1-r1 High Value = Q3+r1**

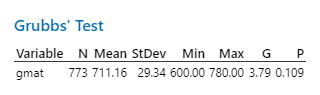
**60 630 790**

**Note : Outliers lying between 630 – 790**

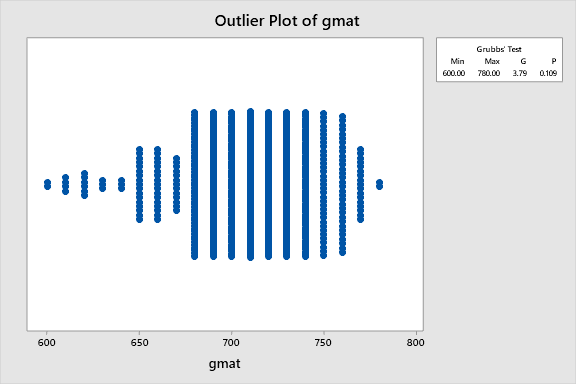
**(OR)**

**Go to Minitab 🡪 select “stat”🡪 select “Basic statistics” 🡪 select “outlier test”**

****

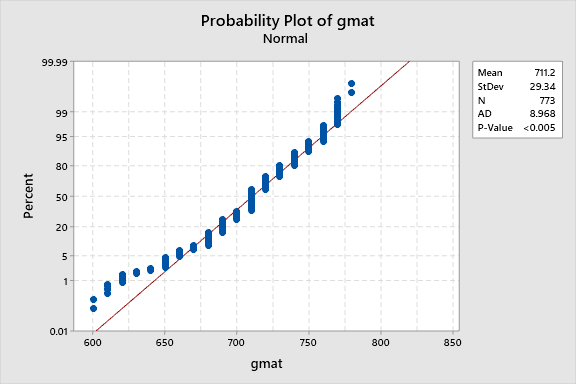
****

****

****

1. **Calculate Normalization:**

**Go to Minitab 🡪 select “stat”🡪 select “Basic statistics” 🡪 select “Normality test”**

****

**LOGISTIC REGRESSION**

1. **Calculate the count in Column using R code:**

There are two ways to calculate the count using R code.

1. **a<- table(claimants$CLMSEX)**

**a**

0 1

586 742

Total Female = 586

Total Male = 742

1. **b<-table(claimants$ATTORNEY)**

**b**

0 1

685 655

Total Hiring a Attorney = 655

Total not hiring a Attorney = 685

1. **c<-table(claimants$CLMINSUR)**

**c**

0 1

120 1179

Total no.of people eligible for Claim insurance = 1179

Total no.of people not eligible for Claim insurance = 120

1. **d<-table(claimants$SEATBELT)**

**d**

0 1

1270 22

Total no.of persons using seatbelt = 22

Toal no.of persons not using seatbelt = 1270

1. **temp1<-as.data.frame(table(claimants$ATTORNEY))**

**temp1**

**Var1 Freq**

**1 0 685**

**2 1 655**

1. **temp2<-as.data.frame((table(claimants$CLMSEX)))**

**temp2**

**Var1 Freq**

**1 0 586**

**2 1 742**

1. **temp3<-as.data.frame(table(claimants$CLMINSUR))**

**temp3**

**Var1 Freq**

**1 0 120**

**2 1 1179**

1. **temp4<-as.data.frame(table(claimants$SEATBELT))**

**temp4**

**Var1 Freq**

**1 0 1270**

**2 1 22**

1. **Calculate Logistic regression:**

In Logistic regression, we can calculate only factor level (Ex: 0,1 ) and we can use the method is glm.

**reg1<-glm(claimants$ATTORNEY~factor(claimants$CLMSEX)+factor(claimants$CLMINSUR)+factor(claimants$SEATBELT)+claimants$CLMAGE+claimants$LOSS , family = binomial , data = claimants)**

**summary(reg1)**

**OutPut:**

Deviance Residuals:

Min 1Q Median 3Q Max

-1.74474 -1.01055 -0.02547 0.95764 2.78320

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -0.199978 0.246769 -0.810 0.41772

factor(claimants$CLMSEX)1 0.432996 0.135706 3.191 0.00142 \*\*

factor(claimants$CLMINSUR)1 0.602173 0.231030 2.606 0.00915 \*\*

factor(claimants$SEATBELT)1 -0.781079 0.566125 -1.380 0.16768

claimants$CLMAGE 0.006487 0.003324 1.952 0.05097 .

claimants$LOSS -0.385044 0.034845 -11.050 < 2e-16 \*\*\*

In this output, we have identified Beta values are negative and we have to convert them into positive values using the exponential method.

**exp(coef(reg1))**

Output:

(Intercept) factor(claimants$CLMSEX)1 factor(claimants$CLMINSUR)1

0.8187490 1.5418701 1.8260829

factor(claimants$SEATBELT)1 claimants$CLMAGE claimants$LOSS

0.4579119 1.0065085 0.6804208

1. **Calculate Confusion matrix:**

We need calculate the accuracy of the data using confusion matrix.

If there are fewer errors in confusion matrix then Accuracy is high.

If there are high errors in confusion matrix then Accuracy is less.

**reg2 <- predict(reg1,type = c("response"),claimants)**

**reg2**

**confusion <- table(reg2>0.5 ,claimants$ATTORNEY)**

**confusion**

**0 1**

**FALSE 380 125**

**TRUE 198 393**

1. **Calculate Accuracy of the data:**

We need to calculate the accuracy of the data and if it is >50% then the accuracy is 1

<50% then accuracy is 0

**Accuracy <- sum(diag(confusion) / sum(confusion))**

**Accuracy**

0.705292 (70%)

Now accuracy is >50%.

1. **Replace Mean with missing values in the original data**

**claimants$CLMAGE <- ifelse(is.na(claimants$CLMAGE), ave(claimants$CLMAGE , FUN = function(x) mean(x,na.rm = TRUE)), claimants$CLMAGE)**

**claimants$CLMAGE**

We need to convert them into table format using the below code.

**c<-table(claimants$CLMAGE)**

**c**

**View(claimants)**

**mean(claimants$CLMAGE)**

I am going to save this file into excel using below R code.Before this,we need to install the packages call “xlsx” and “Rjava”.

**write.xlsx(claimants ,file = "16-jun-19.xlsx")**

we need to use the below command to check where the file saved.

**getwd()**

"C:/Users/sundara.rao.ext/Documents"

**MACHINE LEARNING TECHNIQUES**

There are two types of machine learning techniques

1. **Supervised Learning**

If output variable Y is known then we can use supervised learning

1. Logistic Regression
2. Decision Tree
3. Support Vector Machine(SVM)
4. K-nearest Neighbors
5. Naïve Bayes
6. Random Forest
7. Linear Regression
8. **Un-Supervised Learning**

If output variable Y is unknown then we can use un-supervised learning

1. Hierarchical Clustering
2. K Means Clustering
3. Principal component analysis
4. **Semi-Supervised Learning**
5. **Reinforcement Learning**

**Steps:**

1. **Standardization**

In the dataset, first and second columns are not required for analysis .We need to remove them from data.

**# Remove the first and second columns for analysis**

**mydata <- University\_Clustering[, c(3:8)]**

**mydata**

**View(mydata)**

**# Normalize the data**

**normalize\_data<-scale(mydata)**

**normalize\_data**

**View(normalize\_data)**

1. **Calculate distance between using variables using Euclidean Method**

**distance <- dist(normalize\_data , method = "euclidean")**

**distance**

**summary(distance)**

1. **Grouping the variables using single linkage method or complete linkage method**

**fit <- hclust(distance,method = "complete")**

**fit**

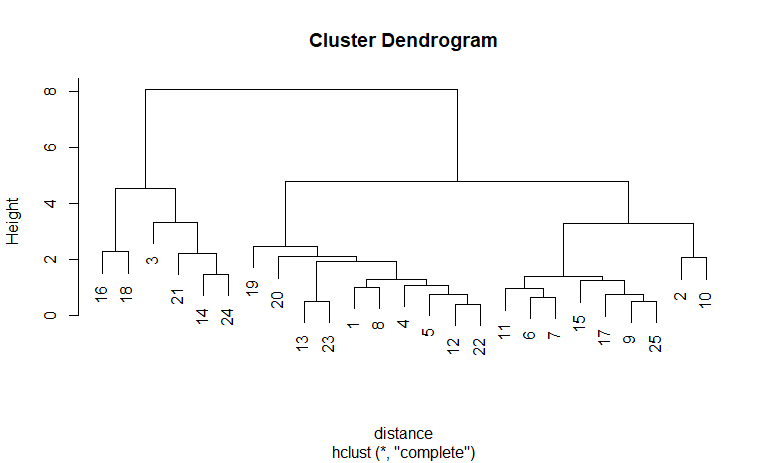
Cluster method : complete

Distance : euclidean

Number of objects: 25

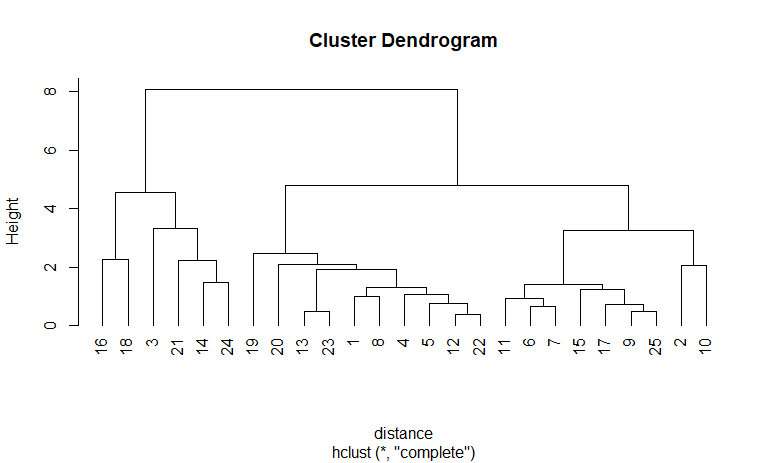
**# check the clustering using Dendogram**

**plot(fit)**



**# make all the variable in one line**

**plot(fit,hang=-1)**



1. **Clustering**

**# looking into dendogram decide how many cluster we need**

**#we are diving the dendogram into 3 clustering by taking k=3**

**groups<-cutree(fit,k=3)**

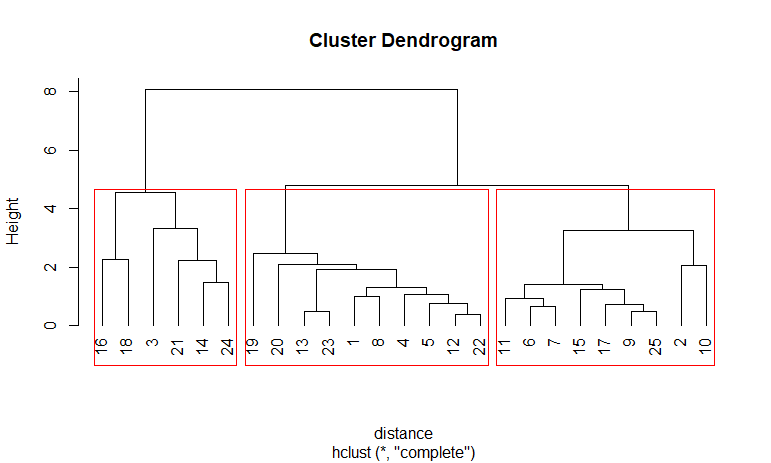
**groups**

groups

[1] 1 2 3 1 1 2 2 1 2 2 2 1 1 3 2 3 2 3 1 1 3 1 1 3 2

**#labelling the groups with red colour**

**rect.hclust(fit,k=3,border="red")**



**#to check the syntax for cutting tree**

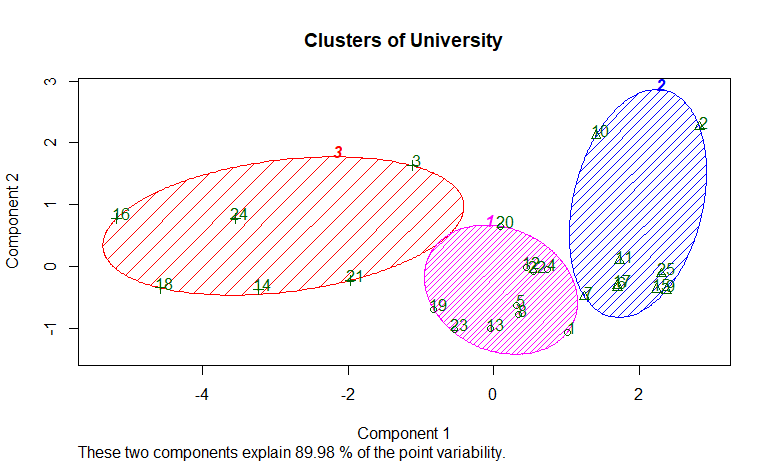
**library(cluster)**

**clusplot(normalize\_data,**

**groups,**

**lines = 0,labels = 2,color = TRUE, shade = TRUE,**

**main = paste('Clusters of University'))**



**# in data set label the clusters or grouping in dataset**

**membership<-as.matrix(groups)**

**View(membership)**

**final1<-data.frame(University\_Clustering,membership)**

**View(final1)**

**# save the file on desktop**

**write.xlsx(final1,file="jan1st.xlsx")**

**getwd()**

getwd()

[1] "C:/Users/sundara.rao.ext/Documents"