ADMISSION OF A STUDENT ANALYSIS

SC614 STATISTICAL METHODS WITH LAB R

Where knowledge meets ANALYSIS;
Every INSIGHT is valuable

By: Nishant Koradia (202118009)
Dhairya Lakhani (202118012)

Table of Contents

1. Loading and Displaying the data

```
> library(readr)

> add<-
read.csv("https://raw.githubusercontent.com/Datamanim/datarepo/main/admission/train.csv")

> View(add)
```

2. Displaying first 10 and last 10 records of the dataset

```
> head(add,10)          #gives first 10 rows of dataset

   Serial.No. GRE.Score TOEFL.Score University.Rating SOP LOR CGPA Research
Chance.of.Admit
1        246       328        110                4 4.0 2.5 9.02        1            0.81
2        100       323        113                3 4.0 4.0 8.88        1            0.79
3         79       296         95               2 3.0 2.0 7.54        1            0.44
4         53       334        116               4 4.0 3.0 8.00        1            0.78
5        444       321        114                5 4.5 4.5 9.16        1            0.87
6        475       308        105                4 3.0 2.5 7.95        1            0.67
7        137       312        103                3 5.0 4.0 8.45        0            0.76
8        249       324        110                3 3.5 4.0 8.87        1            0.80
9         58       298         99               2 4.0 2.0 7.60        0            0.46
10        108       338        117                4 3.5 4.5 9.46        1            0.91


> tail(add,10)   #gives last 10 rows of a dataset

    Serial.No. GRE.Score TOEFL.Score University.Rating SOP LOR CGPA Research
Chance.of.Admit
391        494       300         95               2 3.0 1.5 8.22        1            0.62
392        165       329        111                4 4.5 4.0 9.01        1            0.81
393        351       318        107                3 3.0 3.5 8.27        1            0.74
394         10       323        108               3 3.5 3.0 8.60        0            0.45
395        450       315        101                3 3.5 4.5 9.13        0            0.79
```

```
396       47      329       114                5 4.0 5.0 9.30       1            0.86
397       14      307       109                3 4.0 3.0 8.00       1            0.62
398       28      298        98               2 1.5 2.5 7.50       1            0.44
399      149       339       116                4 4.0 3.5 9.80       1            0.96
400      470       326       114                4 4.0 3.5 9.16       1            0.86
```

3. Displaying mean of every column

```
> install.packages('dplyr')
> library(dplyr)
> add %>% summarise_each(funs(mean))
#gives mean of all the columns at the same time
  Serial.No. GRE.Score TOEFL.Score University.Rating   SOP     LOR    CGPA
Research Chance.of.Admit
1     250.09  316.2225    107.1475           3.0875 3.375 3.48625 8.56025    0.555
0.719225
```

4. Displaying median

```
> median(add$GRE.Score)                    #gives median of the colums of our dataset
[1] 317
> median(add$TOEFL.Score)
[1] 107
```

5. Displaying mode

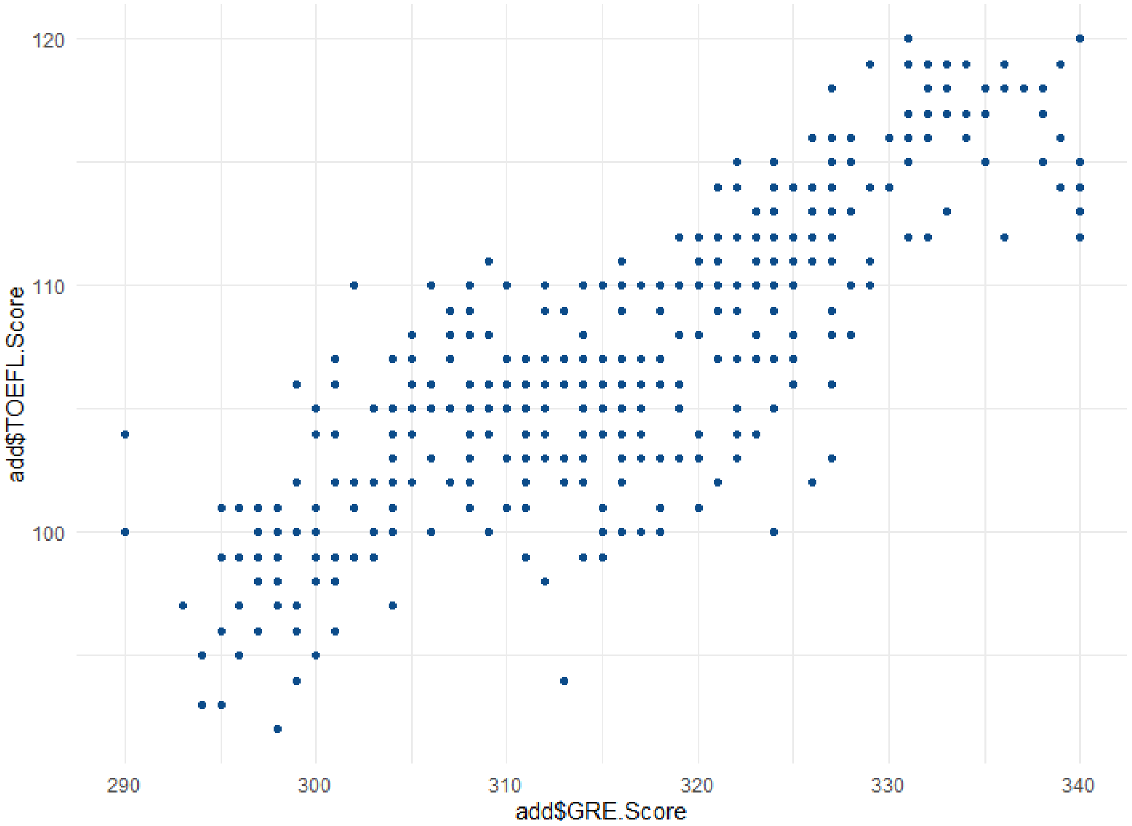
```
> require(modeest)                     #gives mode of the column of the dataset
Loading required package: modeest
> mfv(add$SOP)
[1] 3.5
```

6. Correlation between GRE Score and TOEFL Score

```
> cor(add$GRE.Score,add$TOEFL.Score)
[1] 0.8399105
```

7. Scatter plot between GRE Score and TOEFL Score

```
> library(ggplot2)                    # gives the scatter plot for the above
correlation for better visualization
>   ggplot(add)   +
+         aes(x=add$GRE.Score,y=add$TOEFL.Score)+
+         geom_point(colour="#0c4c8a")+
+        theme_minimal()
```

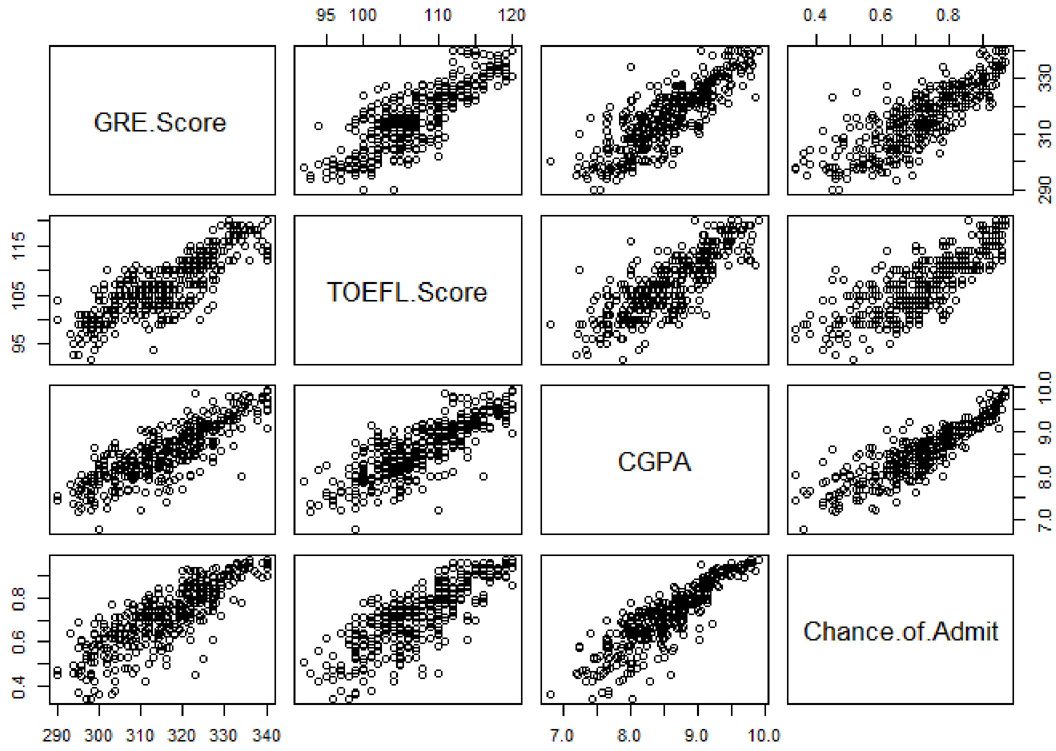


Inference:-

The plot is known as scatter plot which is often used to visualize correlation. From the plot, we can see that the two variables are positively correlated which can also be seen in the above analysis.

8. Plotting a pair plot

```
>   pairs(add[,c(2,3,7,9)])          #gives pairwiseplot for the columns of dataset
```

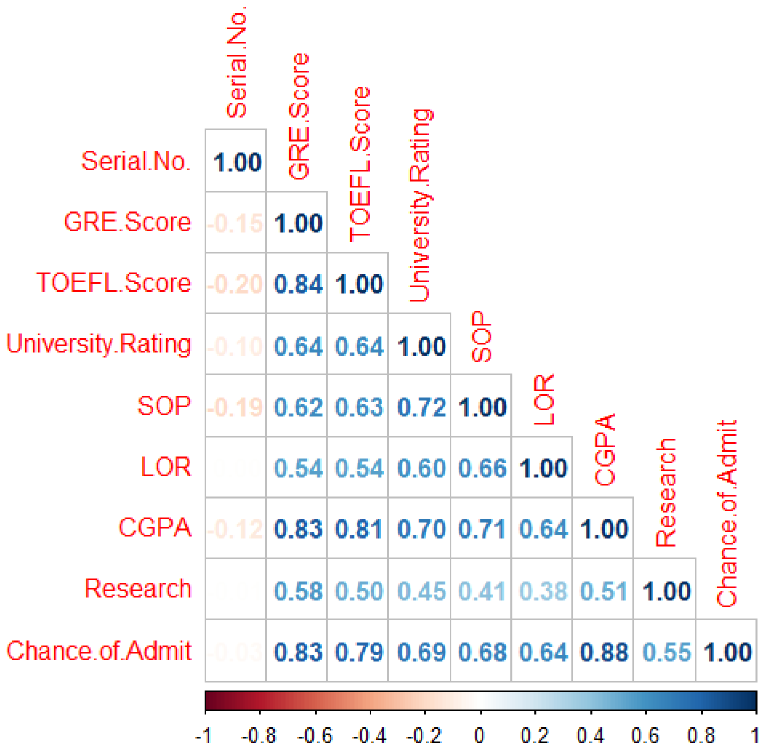


Inference:-

Pairwise plot is the plot in which we can correlate many variables at a time. So from the graph, we can get the information that how the two variables are correlated with each other.

9. Plotting the corrplot of columns in the Dataset

```
>   require(corrplot)         #gives heat map for columns of dataset
>   corrplot(cor(add[,c(1,2,3,4,5,6,7,8,9)]),method= "number",type="lower")
```



	Serial.No.	GRE.Score	TOEFL.Score	University.Rating	SOP	LOR	CGPA	Research	Chance.of.Admit
Serial.No.	1.00								
GRE.Score	-0.15	1.00							
TOEFL.Score	-0.20	0.84	1.00						
University.Rating	-0.10	0.64	0.64	1.00					
SOP	-0.19	0.62	0.63	0.72	1.00				
LOR	0.00	0.54	0.54	0.60	0.66	1.00			
CGPA	-0.12	0.83	0.81	0.70	0.71	0.64	1.00		
Research	-0.01	0.58	0.50	0.45	0.41	0.38	0.51	1.00	
Chance.of.Admit	-0.03	0.83	0.79	0.69	0.68	0.64	0.88	0.55	1.00

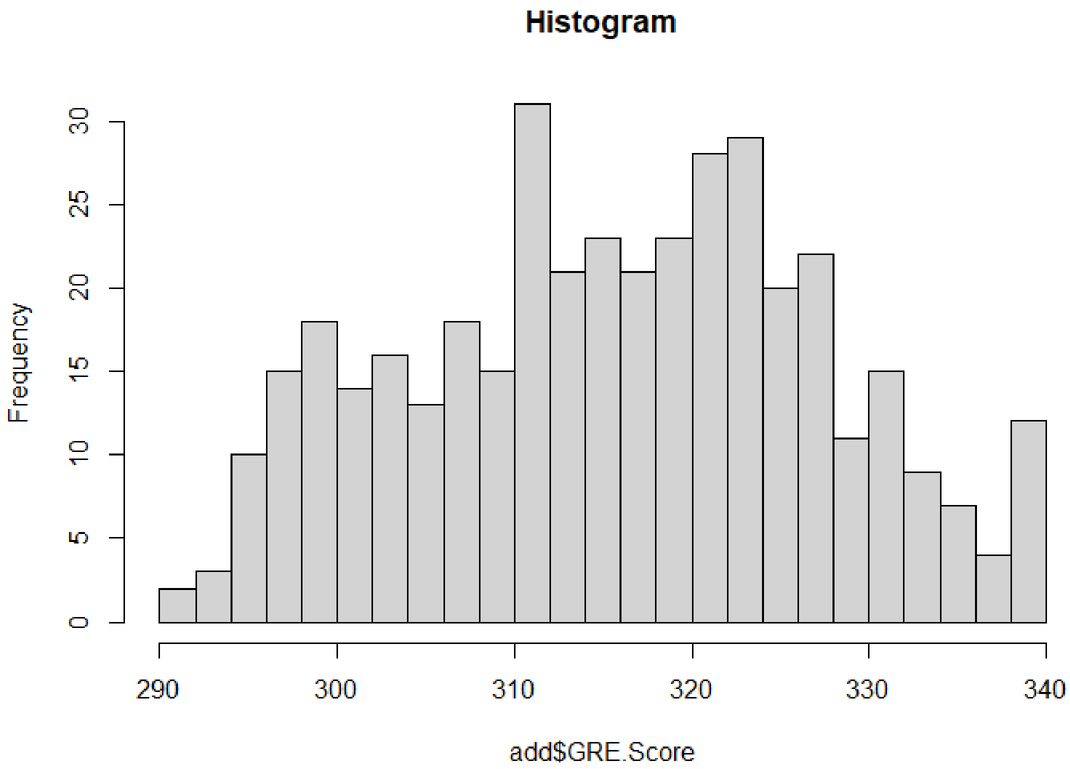
Inference:-

The plot is the lower type correlation matrix which gives the correlation of all the columns. This map is also known as a heat map in which the color shows us the correlation of those two variables. The darker color indicates that it is highly correlated and the light color indicates that it is less correlated.

10. Plotting histogram of GRE Score

```
> hist(add$GRE.Score,breaks = 20,main = "Histogram")
```

#plotting a histogram for the GRE score and its frequency



Inference:-

We are here plotting the histogram of GRE score which by looking at a glance shows that the values are normally distributed.

11. Plotting a histogram and comparing it with the probability density function of the chi-square distribution

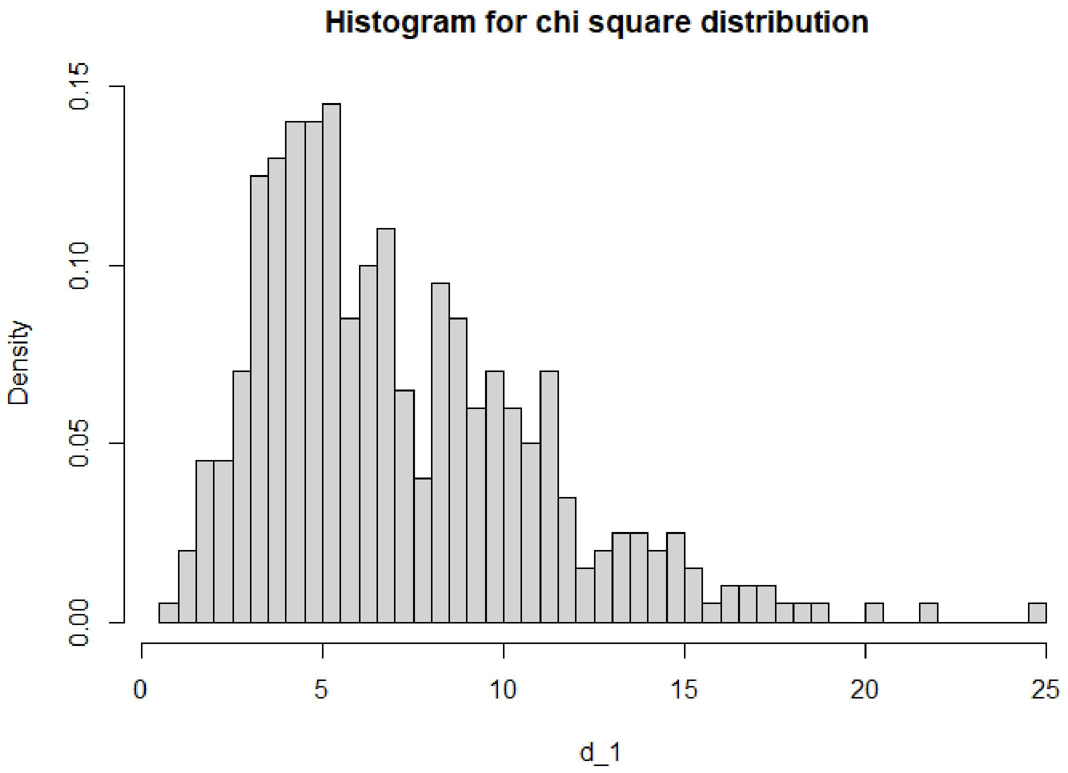
```
>  d_1<-rchisq(add$CGPA,df=7)
```

#gives historam for the random samples generated by chisq from the cgpa

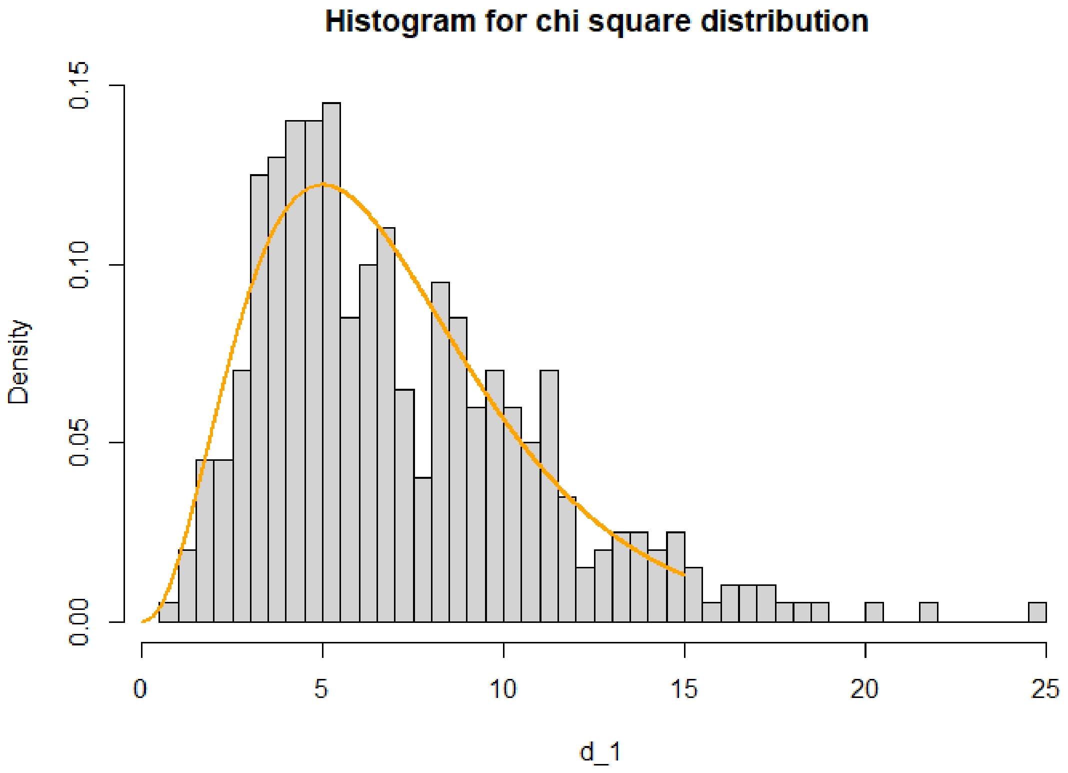
```
>  hist(d_1,freq = FALSE,breaks = 50)
>  x<-d_1
```

#we plot a histogram and compare it to the probability density function of the 2-distribution with df=7

```
>    hist(d_1,
+                          breaks = 50,
+                          freq = FALSE,
+                          main = ('Histogram for chi square distribution '))
```



```
>     curve(dchisq(x, df = 7), from = 0, to = 15, n = 5000, col= 'orange', lwd=2, add =
T)
```

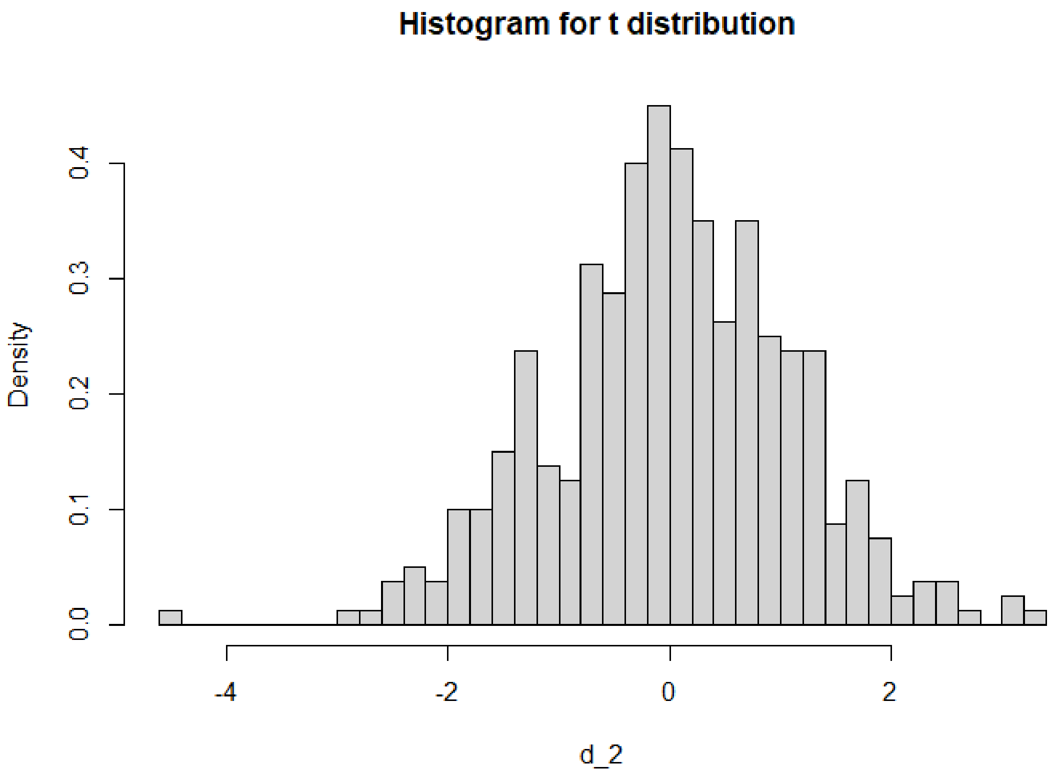


Inference:-

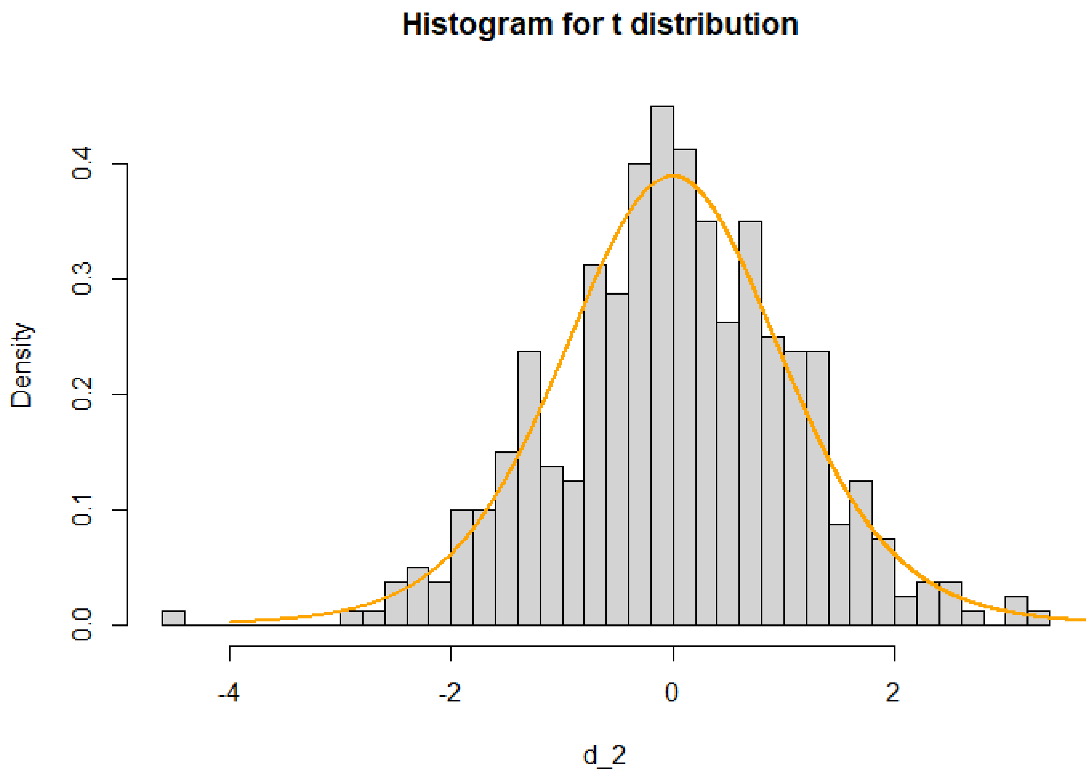
Chi-square plot is plotted to check the expected value of the distribution. Here the histogram which we have plotted is the actual distribution and the chi-square plot tells that how the distribution of the expected values would have been. In our case maximum values are similar to that expected, the others which are going up are the outliers of the plot.

12. Plotting a histogram and comparing it with the probability density function of the t-distribution

```
>    d_2<-rt(add$CGPA,df=10)                                    #we plot
a histogram and compare it to the probability density function of the t-distribution
with df=10
>    hist(d_2,freq = FALSE,breaks=50)
>    x<-d_2
>    hist(d_2,
+                          breaks = 50,
+                           freq = FALSE,
+                         main = ('Histogram for t distribution '))
```



```
>     curve(dt(x, df = 10), from = -4, to = 4, n = 5000, col= 'orange', lwd=2, add = T)
```

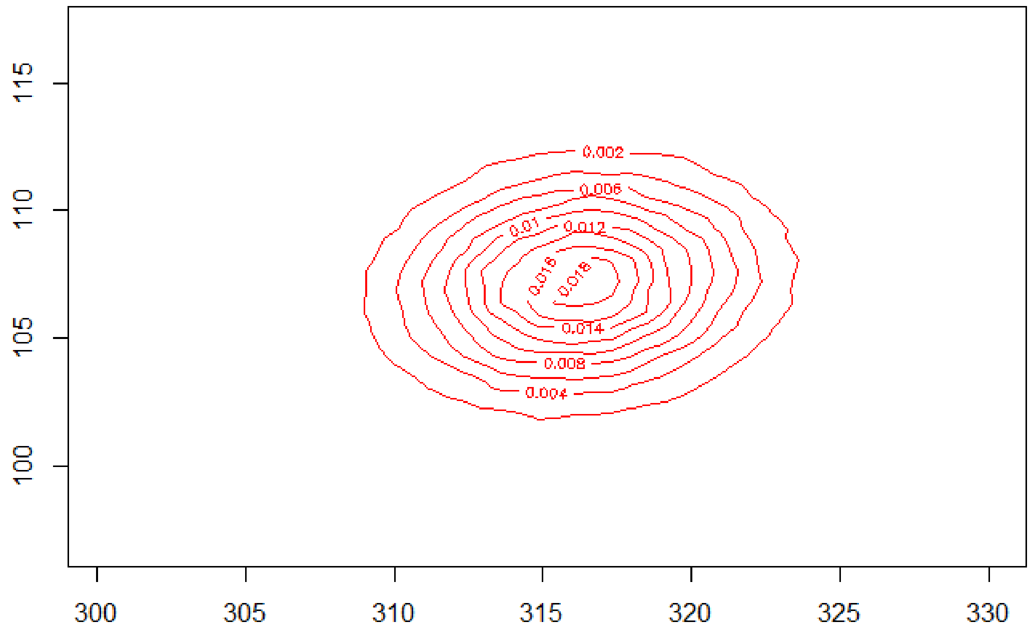


Inference:-

By the t distribution of the plot, we get to know that our dataset is normally distributed.

13. Function for Contour Plot

```
>     contour_plot<-function(x,y)
+     {
+       mu1<-mean(x)
+       mu2<-mean(y)
+       mu3<-c(mu1,mu2)
+       c1<-cor(x,y)
+       s1<-sqrt(var(x))
+       s2<-sqrt(var(y))
+       sigma1<-matrix(c(s1,c1,c1,s2),ncol = 2)
+       library(MASS)
+      bivn<-mvrnorm(100000,mu=mu3,Sigma = sigma1)
+      head(bivn)
+      bivn.kde<-kde2d(bivn[,1],bivn[,2],n=50)
+      contour(bivn.kde,col="red")
+    }
>    contour_plot(add$GRE.Score,add$TOEFL.Score)
```

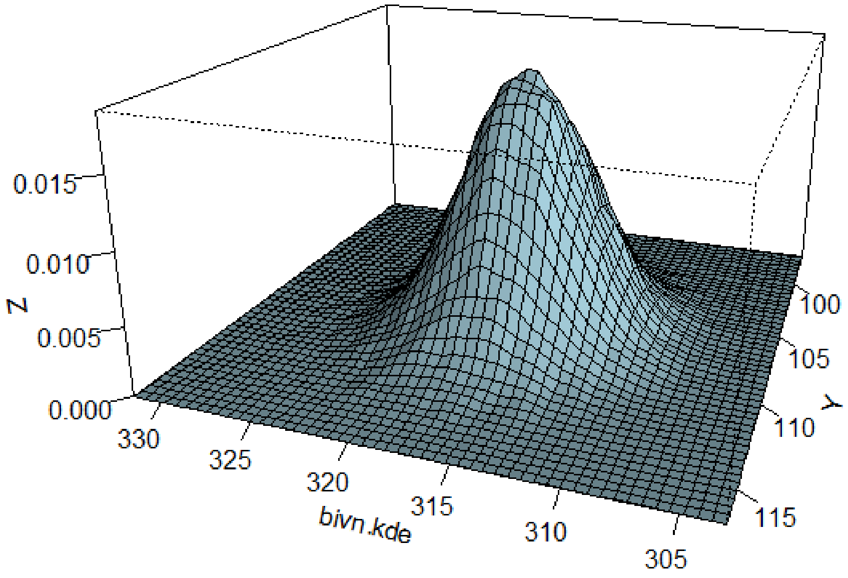


Inference:-

A contour plot is a graphical technique for representing a 3-dimensional surface by plotting constant z slices, called contours, on a 2-dimensional format.

14. Function for Perspective plot

```
>    perspec_plot<-function(x,y)
+    {
+      mu1<-mean(x)
+      mu2<-mean(y)
+      mu3<-c(mu1,mu2)
+       c1<-cor(x,y)
+       s1<-sqrt(var(x))
+       s2<-sqrt(var(y))
+       sigma1<-matrix(c(s1,c1,c1,s2),ncol = 2)
+       library(MASS)
+       bivn<-mvrnorm(100000,mu=mu3,Sigma = sigma1)
+       head(bivn)
+       bivn.kde<-kde2d(bivn[,1],bivn[,2],n=50)
+       persp(bivn.kde, theta = 200, phi = 20,
+             shade = 0.75, col = "light blue", expand = 0.5, r = 2,
+             ltheta = 240, ticktype = "detailed")
+     }
>     perspec_plot(add$GRE.Score,add$TOEFL.Score)
```



Inference:-

When we lift the contour plot on the third axis we get the 3D plot. With this plot, we can infer that how the data is spread between three variables plotted.

15. Two-sided confidence interval

```
>     perspec_plot(add$GRE.Score,add$TOEFL.Score)
>     cd_normalsigma_unknown<-function(n,alpha)
+       {
+          mu<-mean(n)
+           s=sqrt(var(n))
+           len1<-length(n)
+           z1<-qt(1-(alpha/2),df=((len1)-1))
+           f1<-(mu-(z1*(s/sqrt(len1))))
+           f2<-(mu+(z1*(s/sqrt(len1))))
+           f<-c(f1,f2)
+           return(f)
+      }
> cd_normalsigma_unknown(add$GRE.Score,0.05)
[1] 315.0826 317.3624
```

Inference:-

With this function, we can get to know the confidence interval of the variable that we pass in the function. Like for the variable GRE Score, we get the interval (315.0826,317.3624).

16. Hypothesis testing

```
>     null_hypothesis<-function(samp,a,alpha)
+     {
+       xbar<-mean(samp)
+       s<-sqrt(var(samp))
+       len1<-length(samp)
+       z_stat<-qt(1-(alpha/2),df=((len1)-1))
+       z1<-(xbar-a)*sqrt(len1)/s
+       if(abs(z1)<=z_stat){
```

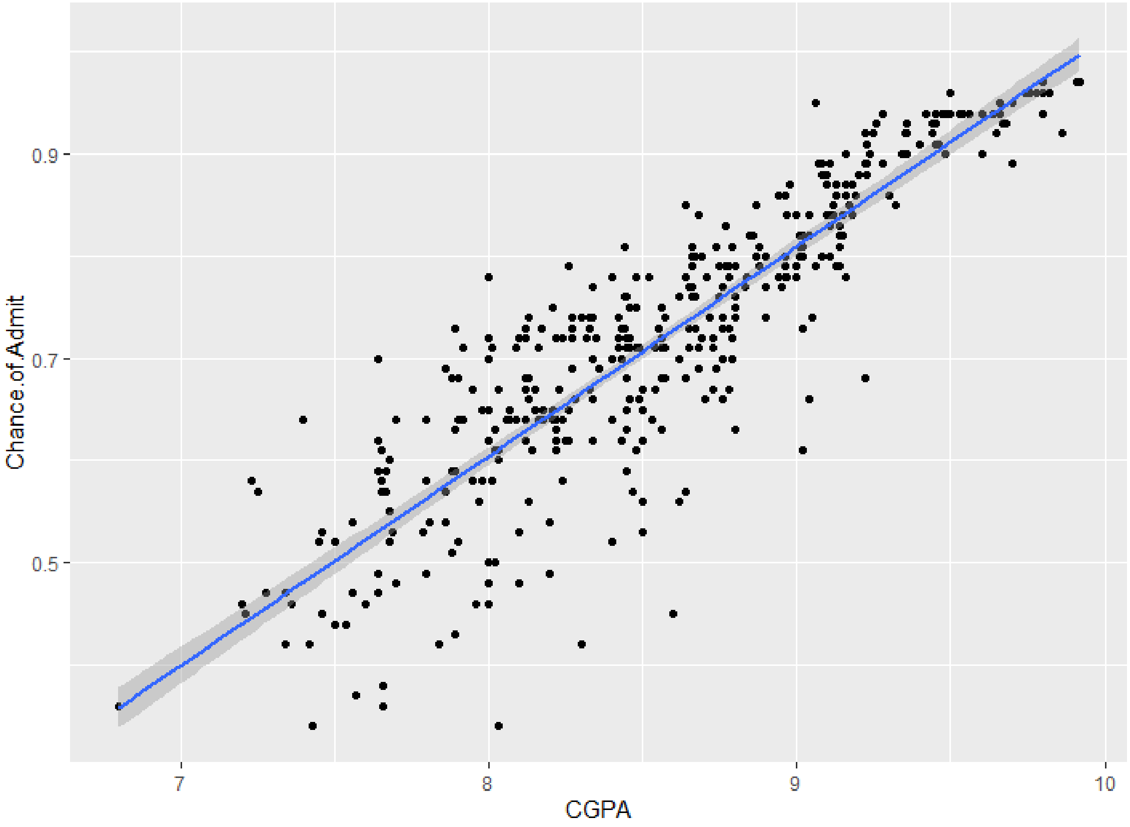
```
+         print("Hyptothesis is in acceptance region")
+       }else{
+         print("Hypothesis is in rejectance region")
+       }
+     }
>     null_hypothesis(add$GRE.Score,545.36,0.05)
[1] "Hypothesis is in rejectance region"
```

Inference:-

Hypothesis testing refers to the concept that we will make an assumption first, and then we perform a test to confirm that whether the assumption is accepted or rejected. For the value shown in the result, we get to know that the value is in the rejectance region.

17. Linear regression Plot

```
>     ggplot(add,aes(x=CGPA,y=Chance.of.Admit))+
+         geom_point()+
+        stat_smooth(method = lm)
```



Inference:-

Linear regression is the concept in which we find a relation between two variables of our dataset and relate them with an intercept and a slope (y=mx+c) concept and we find the value of the response variable for different values of the predictor variable. Here is the plot for our regression analysis.

18. Linear regression model

```
>     model<-lm(Chance.of.Admit~CGPA,data=add)

>     model


Call:

lm(formula = Chance.of.Admit ~ CGPA, data = add)


Coefficients:

(Intercept)        CGPA

    -1.0364       0.2051
```

```
>      new.cgpa<-data.frame(CGPA=c(5.64,9.99,3.65))
>      predict(model,newdata = new.cgpa)
        1          2          3 
 0.1202997  1.0124579 -0.2878370 
```

Inference:-

From this analysis, we get the value of intercept to be -1.0364 and the value of slope to be 0.2051. Then we are predicting various values for our response variable.

19. Conf interval for Linear Regression model

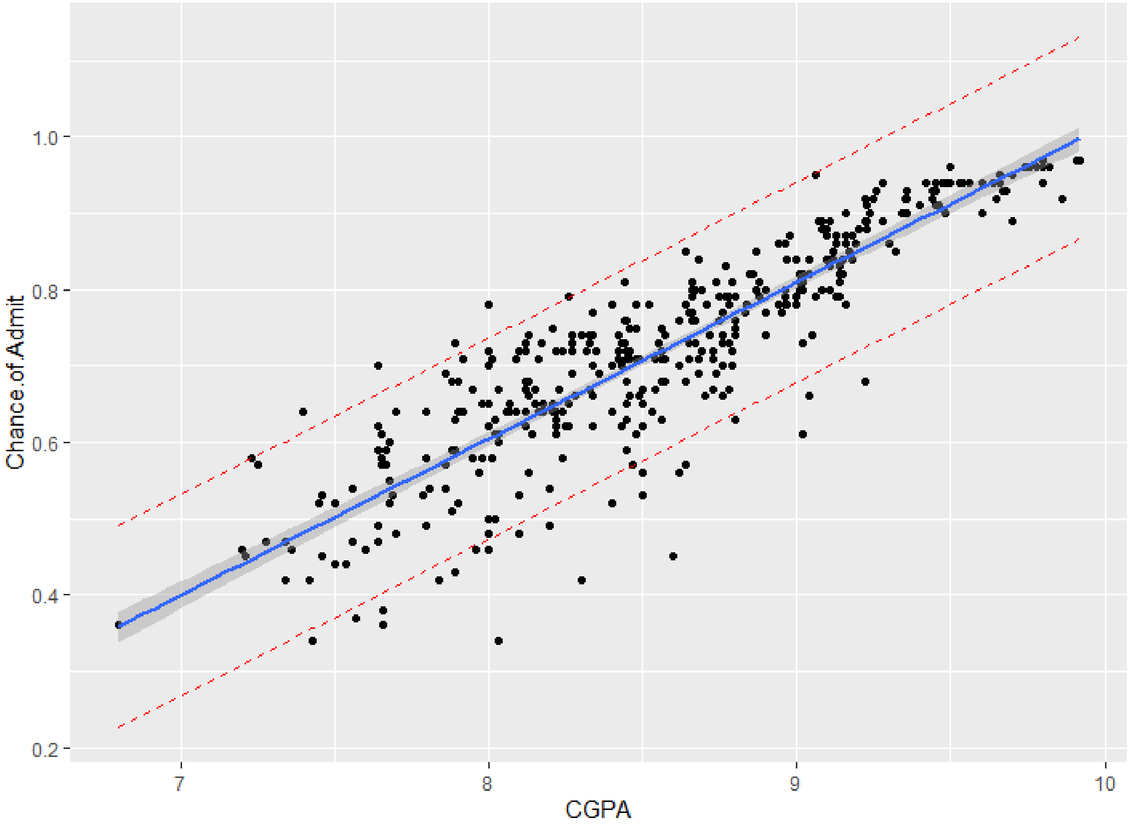
```
>      confint(model)
                 2.5 %     97.5 %
(Intercept) -1.1283469 -0.9445121
CGPA         0.1943835  0.2158042
```

Inference:-

From this analysis, we get the confidence interval of our model.

20. Plot with a prediction interval

```
>      pred.int<-predict(model,interval="prediction")
>      mydata<-cbind(add,pred.int)
>      p<-ggplot(mydata,aes(CGPA,Chance.of.Admit))+
+        geom_point()+
+         stat_smooth(method=lm)
>       p+geom_line(aes(y=lwr),color="red", linetype="dashed")+
+         geom_line(aes(y=upr),color="red", linetype="dashed")
```



Inference:-

From this plot, we get to know the prediction interval for our model. This means that in which range our response variable would show output for our prediction values.

21. Conclusion and Learning

Data visualisation in R and RStudio makes it possible to easily use basic plotting functions, or apply more advanced functions through packages.

As we have noticed throughout this project, the undeniable **added value** of R/RStudio compared to the more classical resources such as **Excel**, is the ability to produce **publication-ready graphics**. For this, we used default functions and options, which already produce a highly controlled output quality, and also pre-define advanced options and use them in variables.

Moreover, we ensured **reproducibility** of our output, simply by writing our commands in a **script**. This would allow us to apply changes in our data or display with the same final output.

Through this analysis, we get to know that how different factors are connected for getting admissions. Using the linear regression, it can be seen that for a good chance of admit a person should have a good cgpa. We have used here packages dplyr for Data Manipulation, readr for reading Dataset, Modeest to find mode of a column, ggplot2 for graphing Plots, tidyverse to access almost all important libraries in one-go. Overall, it was a great learning experience for us and applying different approaches and visualising different data in graphical format helped us to understand small nuances of RStudio. It also helped us to understand that RStudio can be used dynamically and many useful inferences can be concluded easily.