Choose your own project : Graduation Admissions

Pongsasit Thongpramoon

2019/May/13

##1.Introduction

#1.1introduction This dataset is inspired by the UCLA Graduate Dataset. The test scores and GPA are in the older format. The dataset is owned by Mohan S Acharya. #1.2overview The dataset contains several parameters which are considered important during the application for Masters Programs. The parameters included are : 1. GRE Scores (290 to 340) 2. TOEFL Scores (92 to 120) 3. University Rating (1 to 5) 4. Statement of Purpose and Letter of Recommendation Strength (1 to 5) 5. Undergraduate GPA (6.8 to 9.92) 6. Research Experience (0 or 1) 7. Chance of Admit (0.34 to 0.97) #1.3Goal of this project This dataset was built with the purpose of helping students in shortlisting universities with their profiles. The predicted output gives them a fair idea about their chances for a particular university. #1.4Describe dataset This dataset is created for prediction of graduate admissions and the dataset link is below: https://www.kaggle.com/mohansacharya/graduate-admissions

First Look at the dataset

```
#Download useful package.
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cra
n.us.r-project.org")
```

```
## Loading required package: tidyverse
```

```
## Warning: package 'tidyverse' was built under R version 3.5.3
```

```
## -- Attaching packages ---------------------------------------------
------------------------------------- tidyverse 1.2.1 --
```

```
## √ ggplot2 3.0.0       √ purrr   0.3.2
## √ tibble  2.1.1       √ dplyr   0.8.0.1
## √ tidyr   0.8.1       √ stringr 1.3.1
## √ readr   1.1.1       √ forcats 0.3.0
```

```
## Warning: package 'tibble' was built under R version 3.5.3
```

```
## Warning: package 'purrr' was built under R version 3.5.3
```

```
## Warning: package 'dplyr' was built under R version 3.5.3
```

```
## -- Conflicts -------------------------------------------------------
---------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-p
roject.org")
```

```
## Loading required package: caret
```

```
## Warning: package 'caret' was built under R version 3.5.3
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
if(!require(dplyr)) install.packages("caret", repos = "http://cran.us.r-p
roject.org")
if(!require(corrplot)) install.packages("caret", repos = "http://cran.us.
r-project.org")
```

```
## Loading required package: corrplot
```

```
## Warning: package 'corrplot' was built under R version 3.5.3
```

```
## corrplot 0.84 loaded
```

```
if(!require(rpart)) install.packages("caret", repos = "http://cran.us.r-p
roject.org")
```

```
## Loading required package: rpart
```

```
## Warning: package 'rpart' was built under R version 3.5.3
```

```
if(!require(randomForest)) install.packages("caret", repos = "http://cran.
us.r-project.org")
```

```
## Loading required package: randomForest
```

```
## Warning: package 'randomForest' was built under R version 3.5.3
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(tidyverse)
library(dplyr)
#Define the dataset in admission
admission <- read.csv("C:/Users/pongsasit/Desktop/code/R_datascience/caps
tone/GraduateAdmissions/graduate-admissions/Admission_Predict_Ver1.1.csv
")

#find NA in dataset
str(admission)
```

```
## 'data.frame':    500 obs. of  9 variables:
##  $ Serial.No.       : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ GRE.Score        : int  337 324 316 322 314 330 321 308 302 323 ...
##  $ TOEFL.Score      : int  118 107 104 110 103 115 109 101 102 108 ...
##  $ University.Rating: int  4 4 3 3 2 5 3 2 1 3 ...
##  $ SOP              : num  4.5 4 3 3.5 2 4.5 3 3 2 3.5 ...
##  $ LOR              : num  4.5 4.5 3.5 2.5 3 3 4 4 1.5 3 ...
##  $ CGPA             : num  9.65 8.87 8 8.67 8.21 9.34 8.2 7.9 8 8.6 ...
##  $ Research         : int  1 1 1 1 0 1 1 0 0 0 ...
##  $ Chance.of.Admit  : num  0.92 0.76 0.72 0.8 0.65 0.9 0.75 0.68 0.5 0.
45 ...
```

```
sum(is.na(admission))
```

```
## [1] 0
```

```
#make a table(only head)
head(admission)
```

```
##   Serial.No. GRE.Score TOEFL.Score University.Rating SOP LOR CGPA Rese
arch
## 1          1       337         118                 4 4.5 4.5 9.65
   1
## 2          2       324         107                 4 4.0 4.5 8.87
   1
## 3          3       316         104                 3 3.0 3.5 8.00
   1
## 4          4       322         110                 3 3.5 2.5 8.67
```

```
   1
## 5          5       314         103                  2 2.0 3.0 8.21
   0
## 6          6       330         115                  5 4.5 3.0 9.34
   1
##   Chance.of.Admit
## 1            0.92
## 2            0.76
## 3            0.72
## 4            0.80
## 5            0.65
## 6            0.90
```

```
#summary of dataset
summary(admission)
```

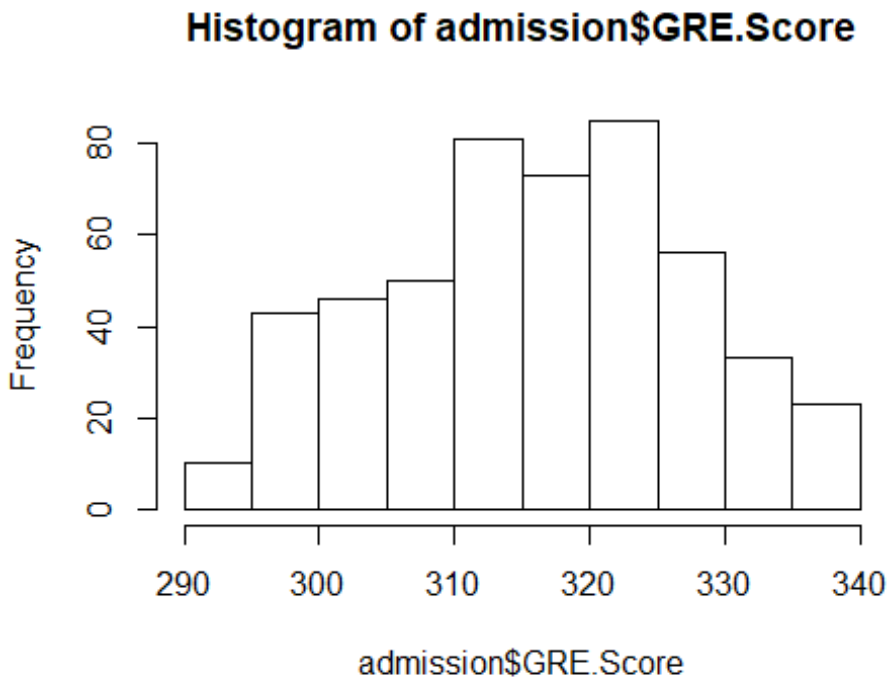
```
##    Serial.No.      GRE.Score      TOEFL.Score    University.Rating
##  Min.   :  1.0   Min.   :290.0   Min.   : 92.0   Min.   :1.000
##  1st Qu.:125.8   1st Qu.:308.0   1st Qu.:103.0   1st Qu.:2.000
##  Median :250.5   Median :317.0   Median :107.0   Median :3.000
##  Mean   :250.5   Mean   :316.5   Mean   :107.2   Mean   :3.114
##  3rd Qu.:375.2   3rd Qu.:325.0   3rd Qu.:112.0   3rd Qu.:4.000
##  Max.   :500.0   Max.   :340.0   Max.   :120.0   Max.   :5.000
##       SOP             LOR             CGPA          Research
##  Min.   :1.000   Min.   :1.000   Min.   :6.800   Min.   :0.00
##  1st Qu.:2.500   1st Qu.:3.000   1st Qu.:8.127   1st Qu.:0.00
##  Median :3.500   Median :3.500   Median :8.560   Median :1.00
##  Mean   :3.374   Mean   :3.484   Mean   :8.576   Mean   :0.56
##  3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:9.040   3rd Qu.:1.00
##  Max.   :5.000   Max.   :5.000   Max.   :9.920   Max.   :1.00
##  Chance.of.Admit
##  Min.   :0.3400
##  1st Qu.:0.6300
##  Median :0.7200
##  Mean   :0.7217
##  3rd Qu.:0.8200
##  Max.   :0.9700
```

Because serial Number is not include as a factor for the prediction.

```
admission <- admission %>% select(GRE.Score,TOEFL.Score,University.Rating,
SOP,LOR,CGPA,Research,Chance.of.Admit)
```

Visualize the data to see how this dataset looklike.

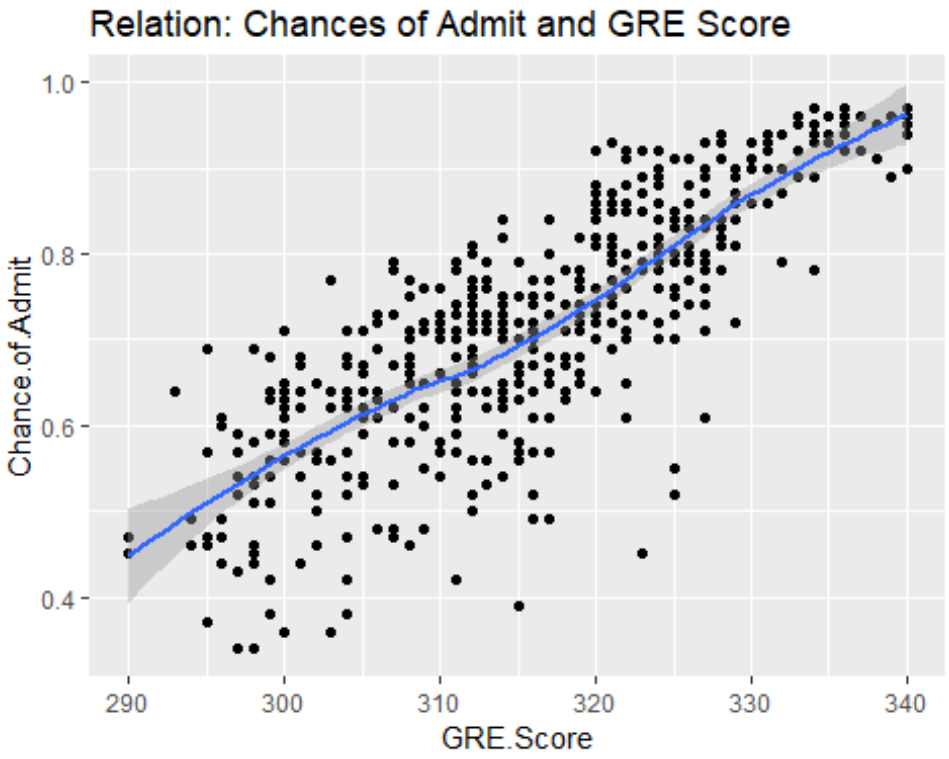
```
#The distribution between GRE score and Amount of people can be shown lik
e below.
hist(admission$GRE.Score)
```



The the relation between chance of admit and GRE score is important to know too.

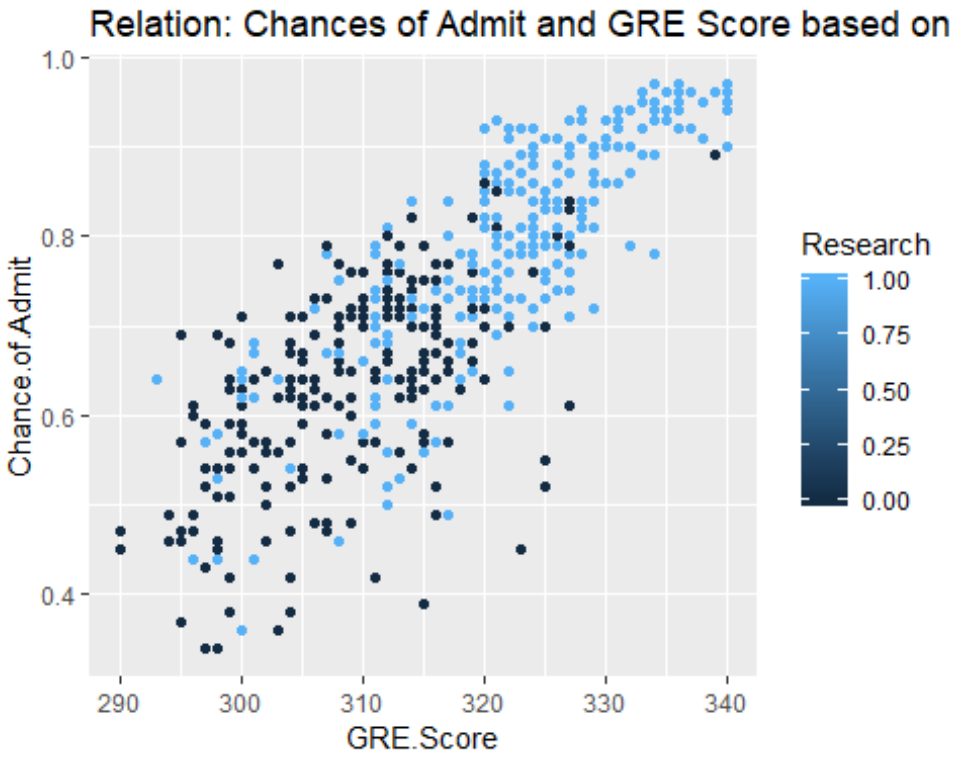
```
#The relation between GRE score and And the chance of admit, shown like b
elow.
ggplot(admission,aes(x=GRE.Score,y=Chance.of.Admit))+geom_point()+geom_sm
ooth()+ggtitle("Relation: Chances of Admit and GRE Score")

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

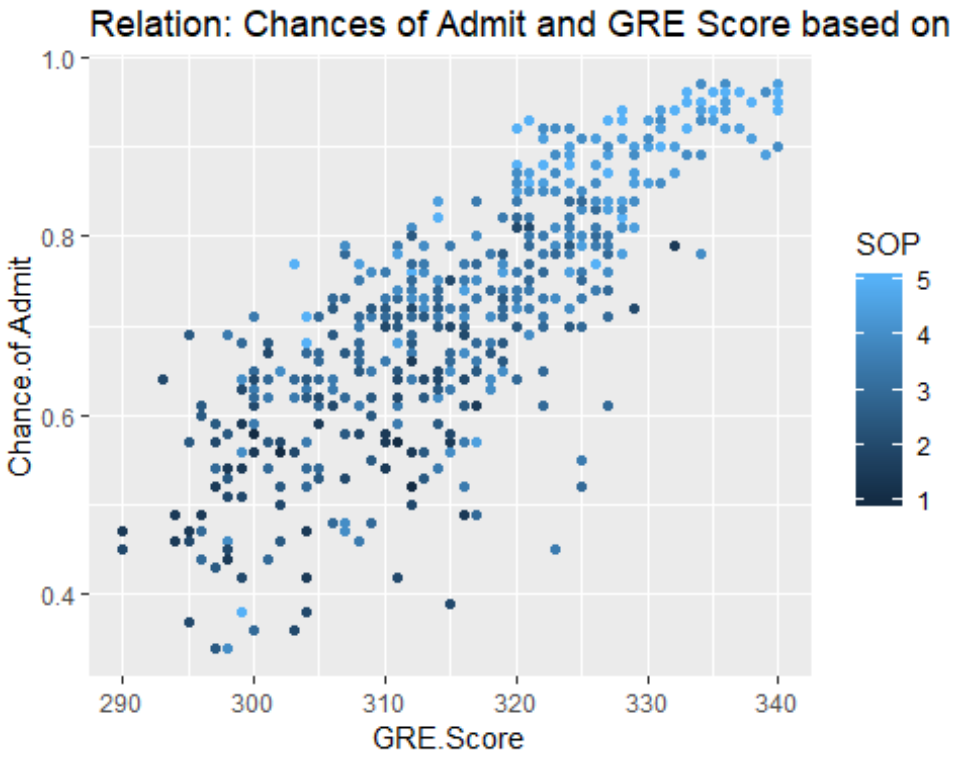


The students have different background so only GRE Score is not enough to judge the result of admission. Now we will plot the relation between GRE Score and Chance of admit base on, Reseach, SOP, LOR, CGPA,TOEFL.Score, University rating as below.

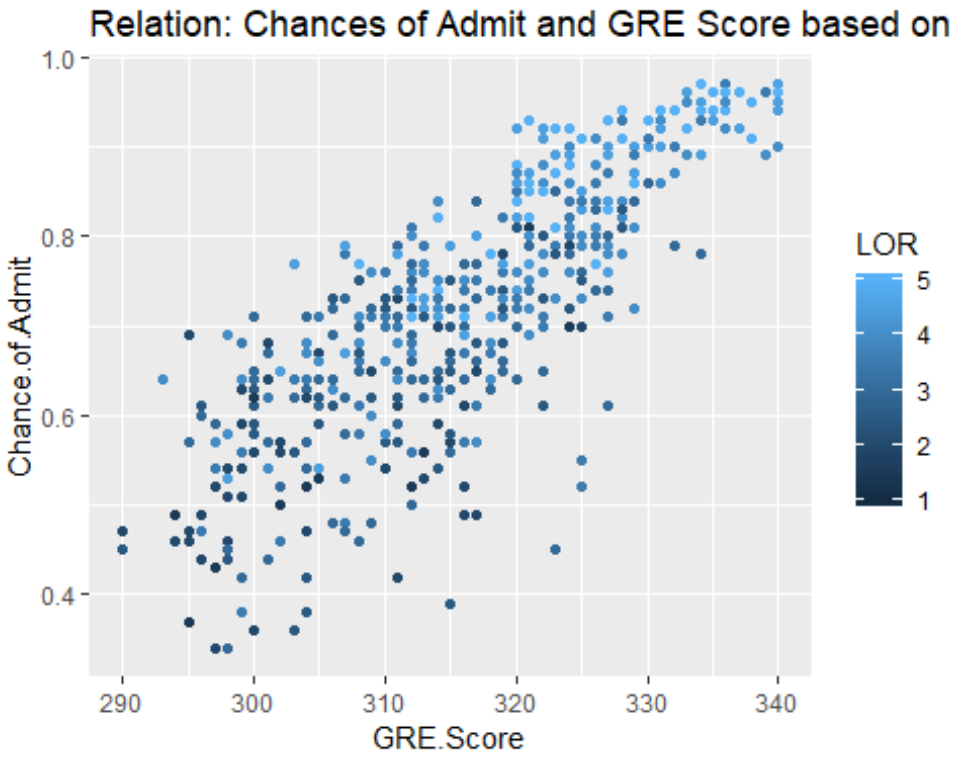
```
ggplot(admission,aes(x=GRE.Score,y=Chance.of.Admit,col=Research))+geom_po
int()+ggtitle("Relation: Chances of Admit and GRE Score based on Research
")
```



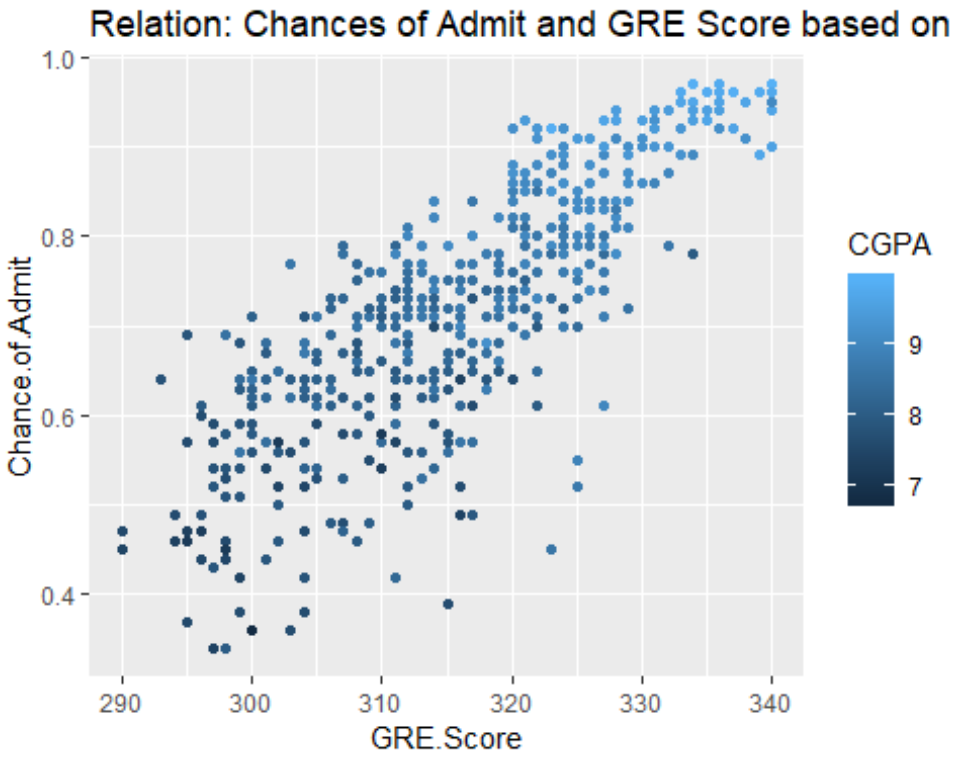

```
ggplot(admission,aes(x=GRE.Score,y=Chance.of.Admit,col=SOP))+geom_point()
+ggtitle("Relation: Chances of Admit and GRE Score based on SOP")
```



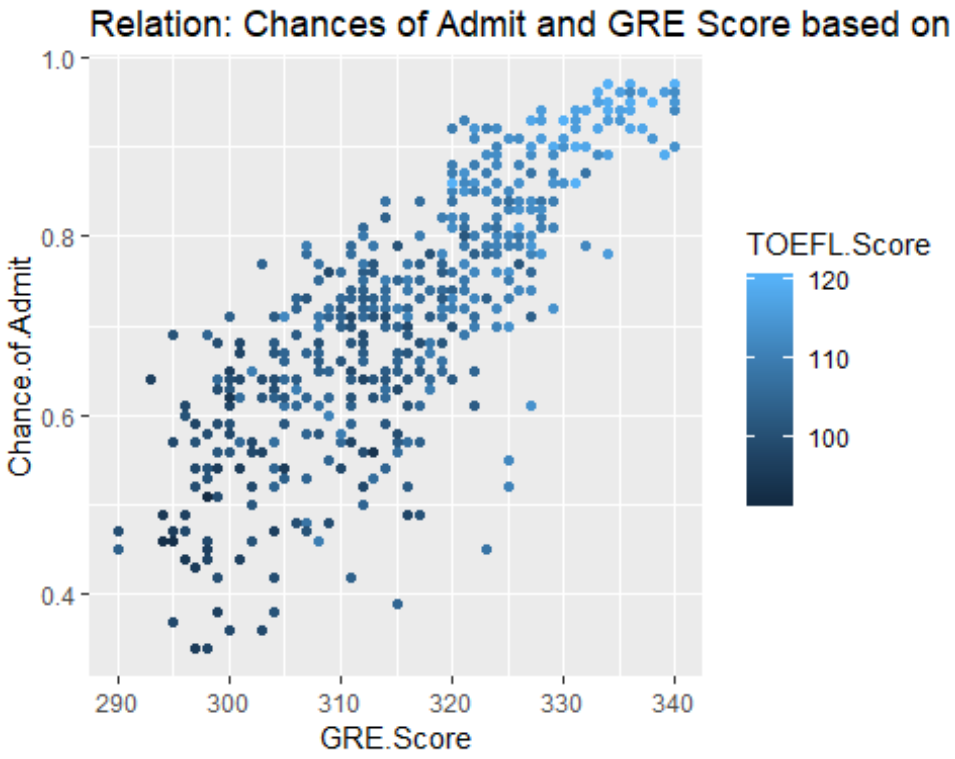

```
ggplot(admission,aes(x=GRE.Score,y=Chance.of.Admit,col=LOR))+geom_point()
+ggtitle("Relation: Chances of Admit and GRE Score based on LOR")
```



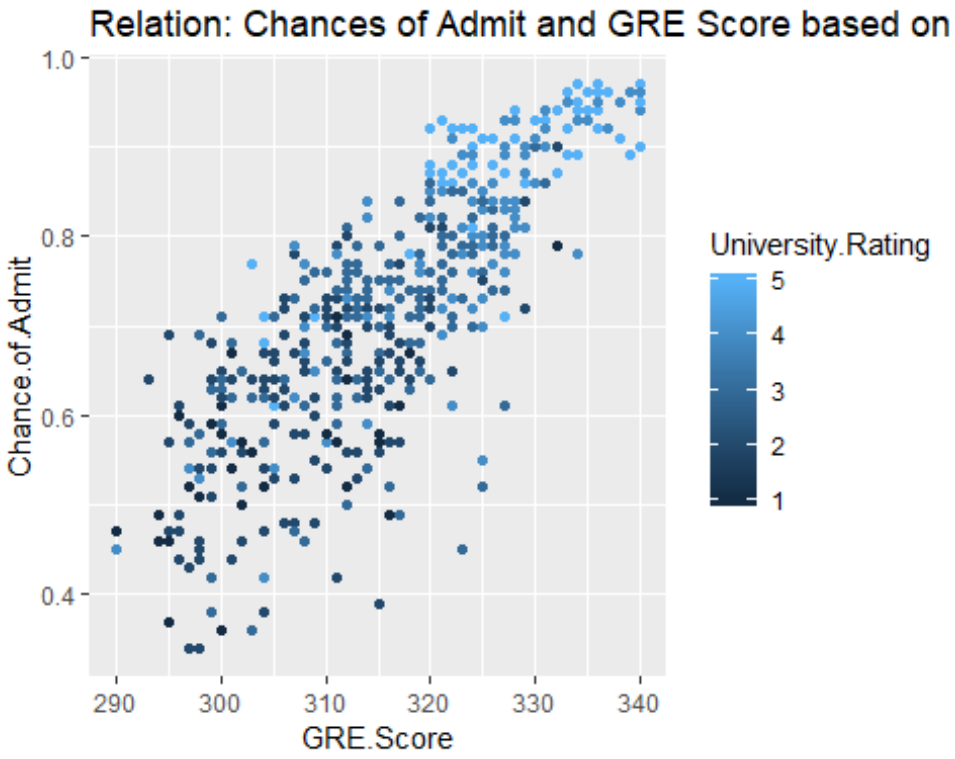

```
ggplot(admission,aes(x=GRE.Score,y=Chance.of.Admit,col=CGPA))+geom_point
()+ggtitle("Relation: Chances of Admit and GRE Score based on CGPA")
```




```
ggplot(admission,aes(x=GRE.Score,y=Chance.of.Admit,col=TOEFL.Score))+geom
_point()+ggtitle("Relation: Chances of Admit and GRE Score based on TOEFL
 Score")
```



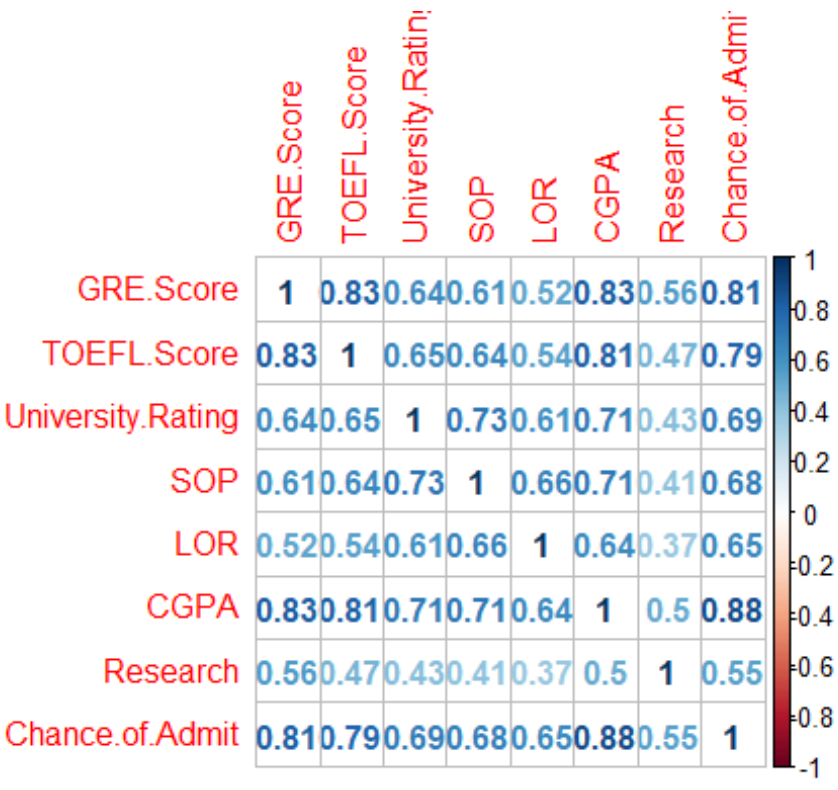

```
ggplot(admission,aes(x=GRE.Score,y=Chance.of.Admit,col=University.Rating))+geom_point()+ggtitle("Relation: Chances of Admit and GRE Score based on University Rating")
```



To make these graph easier to understand I will make a table for corelation.

```
library(corrplot)

C<-cor(admission)
corrplot(C,method='number')
```



As the table above now I know the relation between data and their corelation. ##2.Analysis For make the model to predict the dataset I will split the data in to 2 set. First for training(80%) and second for testing(20%). As below you will see the code.

```
library(caret)
set.seed(1)
test_index <- createDataPartition(y = admission$Chance.of.Admit, times =
1, p = 0.2, list = FALSE)
train <- admission[-test_index,]
test <- admission[test_index,]
```

##2.1 Modeling Method By this data set I will try 3 machine learning method : Linear regression, Decision Tree (and Randomforest) and K-NN. #2.1.1 Linear regression (model1)

```
model1 <- lm(Chance.of.Admit~.,data = train)
summary(model1)

##
## Call:
## lm(formula = Chance.of.Admit ~ ., data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -0.27047 -0.02430  0.00850  0.03565  0.15142
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -1.2630317  0.1175922 -10.741  < 2e-16 ***
## GRE.Score          0.0016698  0.0005672   2.944  0.00343 **
## TOEFL.Score        0.0023964  0.0009862   2.430  0.01555 *
## University.Rating  0.0045944  0.0041455   1.108  0.26843
## SOP                0.0039884  0.0050693   0.787  0.43189
## LOR                0.0137150  0.0046010   2.981  0.00305 **
## CGPA               0.1298225  0.0108385  11.978  < 2e-16 ***
## Research           0.0174249  0.0073658   2.366  0.01849 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05904 on 391 degrees of freedom
## Multiple R-squared:  0.8276, Adjusted R-squared:  0.8245
## F-statistic: 268.1 on 7 and 391 DF,  p-value: < 2.2e-16
```

SOR has only tiny influence in this model so we can exclude it.

```
model1_2 <- lm(Chance.of.Admit~.-SOP,data = train)
summary(model1_2)
```

```
##
## Call:
## lm(formula = Chance.of.Admit ~ . - SOP, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.26937 -0.02310  0.00811  0.03496  0.15244
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -1.2773061  0.1161277 -10.999  < 2e-16 ***
## GRE.Score          0.0016697  0.0005669   2.945 0.003420 **
## TOEFL.Score        0.0024639  0.0009820   2.509 0.012509 *
## University.Rating  0.0057735  0.0038633   1.494 0.135868
## LOR                0.0148667  0.0043598   3.410 0.000717 ***
## CGPA               0.1313190  0.0106651  12.313  < 2e-16 ***
## Research           0.0173917  0.0073621   2.362 0.018648 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05901 on 392 degrees of freedom
## Multiple R-squared:  0.8273, Adjusted R-squared:  0.8247
## F-statistic:   313 on 6 and 392 DF,  p-value: < 2.2e-16
```

Now I use this model to predict using model1_2 on the test dataset.

```
pred<-predict(model1_2,newdata=test)
model1_2_RSME <- sqrt(mean((pred-test$Chance.of.Admit)^2))

rmse_results <- data_frame(method = "Linear regression", RMSE = model1_2_
RSME)

## Warning: `data_frame()` is deprecated, use `tibble()`.
## This warning is displayed once per session.

rmse_results

## # A tibble: 1 x 2
##   method              RMSE
##   <chr>              <dbl>
## 1 Linear regression 0.0642
```

Now we find RMSE of this model is 0.06424821. Which could be better.

#2.1.2 Decision Tree (and Randonforest)

```
library(rpart)
model2_tree <- rpart(Chance.of.Admit~.-SOP, data =train)
```

Now I will check the RMSE.

```
pred<-predict(model2_tree,newdata=test)
Deciciontree_RSME <- sqrt(mean((pred-test$Chance.of.Admit)^2))

rmse_results <- bind_rows(rmse_results,
                          data_frame(method="Decision Tree",
                                     RMSE = Deciciontree_RSME))
rmse_results

## # A tibble: 2 x 2
##   method              RMSE
##   <chr>              <dbl>
## 1 Linear regression 0.0642
## 2 Decision Tree     0.0812
```

This method is worse than Linear regression but, I can improve it using randomforest algorithm

```
library(randomForest)
model2_forest <- randomForest(Chance.of.Admit~.-SOP, data = train)

pred<-predict(model2_forest,newdata=test)
RandomForest_RMSE <- sqrt(mean((pred-test$Chance.of.Admit)^2))
```

```
rmse_results <- bind_rows(rmse_results,
                          data_frame(method="RandomForest",
                                     RMSE = RandomForest_RMSE))
rmse_results
```

```
## # A tibble: 3 x 2
##   method              RMSE
##   <chr>              <dbl>
## 1 Linear regression 0.0642
## 2 Decision Tree     0.0812
## 3 RandomForest      0.0659
```

The RMSE value is smaller.

#2.1.3 KNN method

```
library(caret)
model3_knn <- knn3(Chance.of.Admit~.-SOP, data =train)

pred<-predict(model3_knn,newdata=test)
knn_RMSE <- sqrt(mean((pred-test$Chance.of.Admit)^2))

rmse_results <- bind_rows(rmse_results,
                          data_frame(method="RandomForest",
                                     RMSE = knn_RMSE))
rmse_results
```

```
## # A tibble: 4 x 2
##   method              RMSE
##   <chr>              <dbl>
## 1 Linear regression 0.0642
## 2 Decision Tree     0.0812
## 3 RandomForest      0.0659
## 4 RandomForest      0.726
```

KNN model is the worst.

#2.1.4 Logistic regression

```
model4_LR <- glm(Chance.of.Admit~.-SOP, data =train)

pred<-predict(model4_LR,newdata=test)
logistic_regression_RMSE <- sqrt(mean((pred-test$Chance.of.Admit)^2))

rmse_results <- bind_rows(rmse_results,
                          data_frame(method="Logistic Regression",
                                     RMSE = logistic_regression_RMSE))
rmse_results
```

```
## # A tibble: 5 x 2
##   method                RMSE
##   <chr>                <dbl>
## 1 Linear regression   0.0642
## 2 Decision Tree       0.0812
## 3 RandomForest        0.0659
## 4 RandomForest        0.726
## 5 Logistic Regression 0.0642
```

##Result section

```
rmse_results
```

```
## # A tibble: 5 x 2
##   method                RMSE
##   <chr>                <dbl>
## 1 Linear regression   0.0642
## 2 Decision Tree       0.0812
## 3 RandomForest        0.0659
## 4 RandomForest        0.726
## 5 Logistic Regression 0.0642
```

As you see above the model that can predict the best is Linear regression model. The RMSE value is 0.06424821. Now I will use this linear regression model to predict chance for admissions for the given values (some value are mine).

```
predict(model1_2,data.frame(GRE.Score=330,TOEFL.Score=103,University.Rating=4,SOP=3.5,LOR=3.5,CGPA=7.5,Research=1))
```

```
##        1
## 0.604889
```

By this result I think I will take a chance to admission next year.

##Conclusion This Graduate Admissions project I used the machine learning methods that the first project (MoviesLens) didn't use. So I think I used this project for my revision and practice the machine learning skill on this work. And because of this, if I want to admit to my dream University, I have to retake my GRE test to renew my score. Because of this projects I learned alot.