# Data Science Project 1

*Filiberto Asare-Akuffo*

*February 5, 2019*

```r
#import libraries
library(ggplot2)
library(ggthemes)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(corrgram)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```r
library(caTools)
```

```r
#Importing dataset
df = read.csv('Admission_Predict_Ver1.1.csv')
head(df)
```

```
##   Serial.No. GRE.Score TOEFL.Score University.Rating SOP LOR CGPA Research
## 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
```

```r
str(df)
```

```
## '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 ...
```

The dataset has 500 observations with 9 variables. Most of the variables are in numeric and integer as such will not have to be concern with factor variables. I want to go ahead and explore the data to understand it very well

```
# Drop the  Serial No. columns of the dataframe since we will not need it in our analysis
dataset <- select (df,-c(Serial.No.))
```

```
#I want to know the descriptive statistics of the data
summary(dataset)
```

```
##    GRE.Score      TOEFL.Score    University.Rating      SOP
##  Min.   :290.0   Min.   : 92.0   Min.   :1.000     Min.   :1.000
##  1st Qu.:308.0   1st Qu.:103.0   1st Qu.:2.000     1st Qu.:2.500
##  Median :317.0   Median :107.0   Median :3.000     Median :3.500
##  Mean   :316.5   Mean   :107.2   Mean   :3.114     Mean   :3.374
##  3rd Qu.:325.0   3rd Qu.:112.0   3rd Qu.:4.000     3rd Qu.:4.000
##  Max.   :340.0   Max.   :120.0   Max.   :5.000     Max.   :5.000
##       LOR             CGPA          Research    Chance.of.Admit
##  Min.   :1.000   Min.   :6.800   Min.   :0.00   Min.   :0.3400
##  1st Qu.:3.000   1st Qu.:8.127   1st Qu.:0.00   1st Qu.:0.6300
##  Median :3.500   Median :8.560   Median :1.00   Median :0.7200
##  Mean   :3.484   Mean   :8.576   Mean   :0.56   Mean   :0.7217
##  3rd Qu.:4.000   3rd Qu.:9.040   3rd Qu.:1.00   3rd Qu.:0.8200
##  Max.   :5.000   Max.   :9.920   Max.   :1.00   Max.   :0.9700
```

```
#I want to check if there are any missin values(na) in my data set
any(is.na(dataset))
```

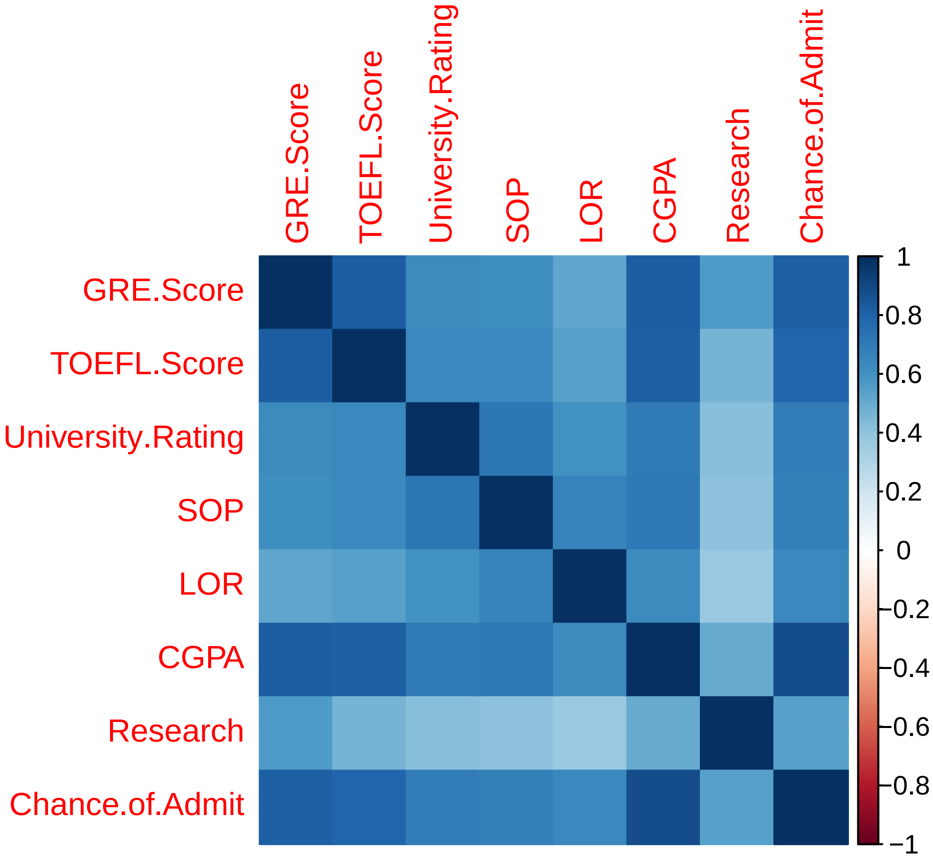
```
## [1] FALSE
```

The result shows there are no missing values(na) values in the dataset

```
#Visualize the data to see the relationship between variables
num.col <- sapply(dataset, is.numeric)
cor.data <- cor(dataset[,num.col])
print(cor.data)
```

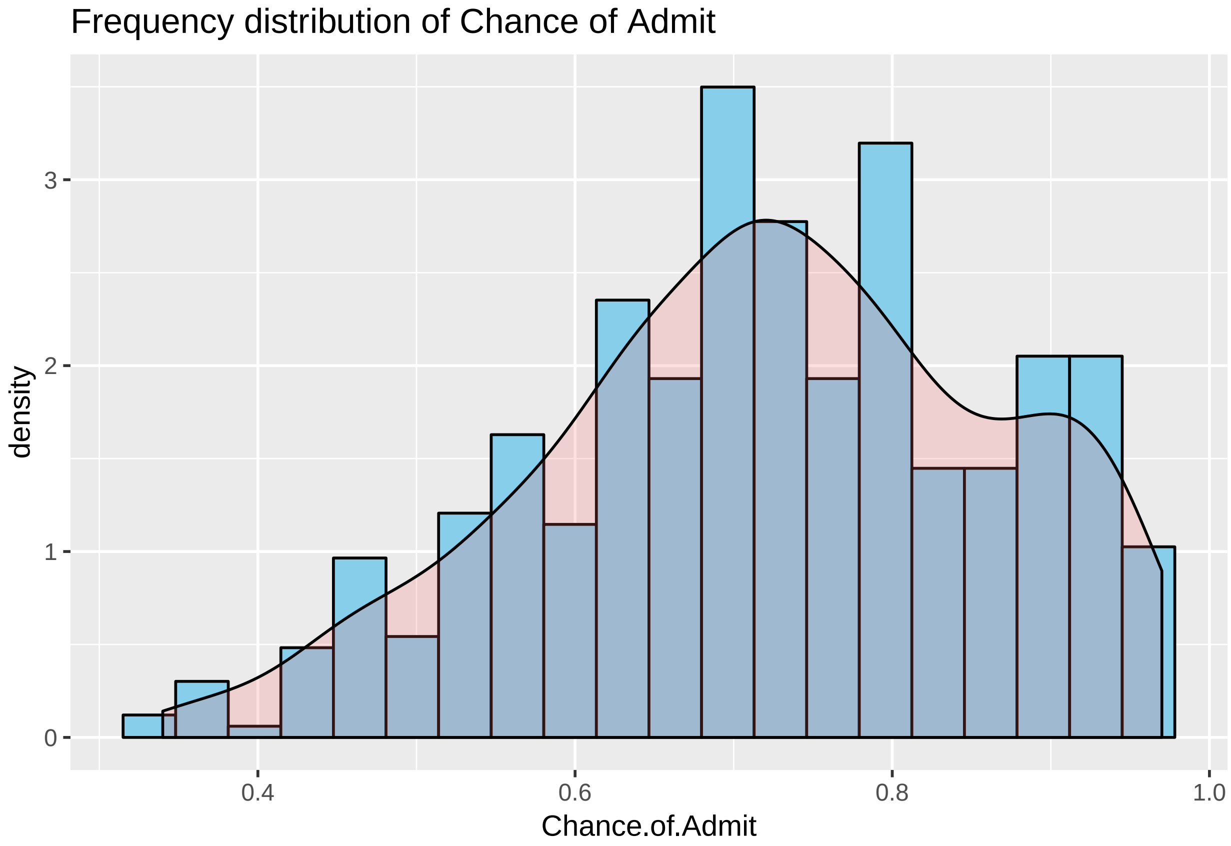
```
##                   GRE.Score TOEFL.Score University.Rating       SOP
## GRE.Score         1.0000000   0.8272004         0.6353762 0.6134977
## TOEFL.Score       0.8272004   1.0000000         0.6497992 0.6444104
## University.Rating 0.6353762   0.6497992         1.0000000 0.7280236
## SOP               0.6134977   0.6444104         0.7280236 1.0000000
## LOR               0.5246794   0.5415633         0.6086507 0.6637069
## CGPA              0.8258780   0.8105735         0.7052543 0.7121543
## Research          0.5633981   0.4670121         0.4270475 0.4081158
## Chance.of.Admit   0.8103506   0.7922276         0.6901324 0.6841365
##                         LOR      CGPA  Research Chance.of.Admit
## GRE.Score         0.5246794 0.8258780 0.5633981       0.8103506
## TOEFL.Score       0.5415633 0.8105735 0.4670121       0.7922276
## University.Rating 0.6086507 0.7052543 0.4270475       0.6901324
## SOP               0.6637069 0.7121543 0.4081158       0.6841365
## LOR               1.0000000 0.6374692 0.3725256       0.6453645
## CGPA              0.6374692 1.0000000 0.5013110       0.8824126
## Research          0.3725256 0.5013110 1.0000000       0.5458710
## Chance.of.Admit   0.6453645 0.8824126 0.5458710       1.0000000
```

```
print(corrplot(cor.data, method = 'color'))
```



```
##                   GRE.Score TOEFL.Score University.Rating       SOP
## GRE.Score         1.0000000   0.8272004         0.6353762 0.6134977
## TOEFL.Score       0.8272004   1.0000000         0.6497992 0.6444104
## University.Rating 0.6353762   0.6497992         1.0000000 0.7280236
## SOP               0.6134977   0.6444104         0.7280236 1.0000000
## LOR               0.5246794   0.5415633         0.6086507 0.6637069
## CGPA              0.8258780   0.8105735         0.7052543 0.7121543
## Research          0.5633981   0.4670121         0.4270475 0.4081158
## Chance.of.Admit   0.8103506   0.7922276         0.6901324 0.6841365
##                         LOR      CGPA  Research Chance.of.Admit
## GRE.Score         0.5246794 0.8258780 0.5633981       0.8103506
## TOEFL.Score       0.5415633 0.8105735 0.4670121       0.7922276
## University.Rating 0.6086507 0.7052543 0.4270475       0.6901324
## SOP               0.6637069 0.7121543 0.4081158       0.6841365
## LOR               1.0000000 0.6374692 0.3725256       0.6453645
## CGPA              0.6374692 1.0000000 0.5013110       0.8824126
## Research          0.3725256 0.5013110 1.0000000       0.5458710
## Chance.of.Admit   0.6453645 0.8824126 0.5458710       1.0000000
```

```
ggplot(dataset, aes(x=Chance.of.Admit)) +
    geom_histogram(aes(y=..density..),
                   bins = 20,
                   colour="black", fill="skyblue") +
    geom_density(alpha=.2, fill="#FF6666") +
  ggtitle('Frequency distribution of Chance of Admit')
```



The results shows that on average there is a good correlation between the various variables.

```
#Splitting the datatset in to training set and testing set
set.seed(123)
split = sample.split(dataset$Chance.of.Admit, SplitRatio = 0.7)

training_set = subset(dataset, split == TRUE)
#70%  of the dataset will be used for training

test_set = subset(dataset, split == FALSE)
#30% of the dataset will be used for testing
```

```
#Building the multiple regression model
model = lm(formula = Chance.of.Admit ~ .,data = training_set)

summary(model)
```

```
##
## Call:
## lm(formula = Chance.of.Admit ~ ., data = training_set)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.23988 -0.02440  0.00781  0.03427  0.15260
##
```

```
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)    
## (Intercept)       -1.2846618  0.1211134 -10.607  < 2e-16 ***
## GRE.Score          0.0020068  0.0005832   3.441 0.000651 ***
## TOEFL.Score        0.0024428  0.0010114   2.415 0.016252 *  
## University.Rating  0.0033208  0.0043780   0.759 0.448662    
## SOP                0.0096181  0.0052401   1.835 0.067303 .  
## LOR                0.0130221  0.0046956   2.773 0.005854 ** 
## CGPA               0.1174281  0.0112846  10.406  < 2e-16 ***
## Research           0.0254138  0.0077168   3.293 0.001094 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.0584 on 342 degrees of freedom
## Multiple R-squared:  0.8336, Adjusted R-squared:  0.8302 
## F-statistic: 244.8 on 7 and 342 DF,  p-value: < 2.2e-16
```

```
#Predicting the Test set results
y_pred = predict(model, newdata = test_set)
y_pred
```

```
##         3         5         6         9        10        12        13 
## 0.6527623 0.6261021 0.8796391 0.5520641 0.7199219 0.8353254 0.8515178 
##        19        25        27        30        36        37        41 
## 0.7430682 0.9571698 0.7666142 0.4836131 0.8617708 0.6578984 0.6496628 
##        47        52        55        57        59        61        62 
## 0.8917276 0.6054733 0.6540459 0.5358769 0.4413514 0.6054408 0.6285518 
##        66        68        70        92        94        96        99 
## 0.7811154 0.7367329 0.8641898 0.5554444 0.5816379 0.5505790 0.9027024 
##       105       109       110       116       127       130       134 
## 0.8139698 0.9196391 0.7083672 0.7985025 0.8469069 0.9250154 0.7818139 
##       138       142       143       145       147       148       149 
## 0.6375207 0.8898783 0.8950907 0.7997757 0.6615097 0.8210920 0.9525410 
##       152       154       156       159       168       169       170 
## 0.9099031 0.7404276 0.7060496 0.6026452 0.6320190 0.5595935 0.5881112 
##       173       178       181       183       187       193       194 
## 0.8446255 0.7749598 0.6123265 0.5660500 0.7402465 0.8271930 0.9473637 
##       199       200       202       205       206       208       210 
## 0.6952579 0.7323221 0.7144063 0.6684005 0.5172515 0.6495047 0.6480721 
##       212       218       223       224       226       231       232 
## 0.8544364 0.8271538 0.7824149 0.6795411 0.5573699 0.7202966 0.6942064 
##       234       239       241       242       250       251       256 
## 0.5935861 0.6491381 0.5243459 0.6168636 0.7886811 0.7147013 0.6933536 
##       258       262       263       265       267       270       274 
## 0.7633507 0.6412100 0.6749206 0.7583824 0.6486458 0.6982721 0.5817735 
##       281       285       288       292       295       301       304 
## 0.7339350 0.9509770 0.8589666 0.5378077 0.6553479 0.5970346 0.7435326 
##       305       306       311       316       318       320       324 
## 0.6490447 0.7737616 0.7476862 0.6048733 0.5453005 0.7794661 0.5955644 
##       326       328       331       334       335       337       338 
## 0.8473316 0.5337748 0.7726224 0.7319649 0.7526508 0.7277716 0.9371000 
##       339       340       344       346       353       357       362 
## 0.7862484 0.7716091 0.6099669 0.5051558 0.6206279 0.7887905 0.9119758 
##       364       369       370       377       383       386       388 
## 0.6324711 0.5120288 0.5887560 0.4750804 0.8427745 0.9817774 0.6105342 
##       390       392       393       398       401       402       403 
## 0.7335490 0.6964467 0.8368307 0.9160204 0.6143074 0.6578782 0.7897108 
##       404       418       425       430       432       437       439 
## 0.8675460 0.5726045 0.9024890 0.8911693 0.7734138 0.5504011 0.7233931 
##       443       444       450       452       453       454       456 
## 0.9120439 0.8575315 0.7685383 0.8683567 0.9165989 0.7472872 0.5273442 
##       460       464       467       475       481       482       492 
## 0.8734765 0.5924535 0.7434093 0.6235781 0.7878331 0.7208022 0.5580152 
##       494       496       498 
## 0.5951379 0.8419983 0.9437315
```

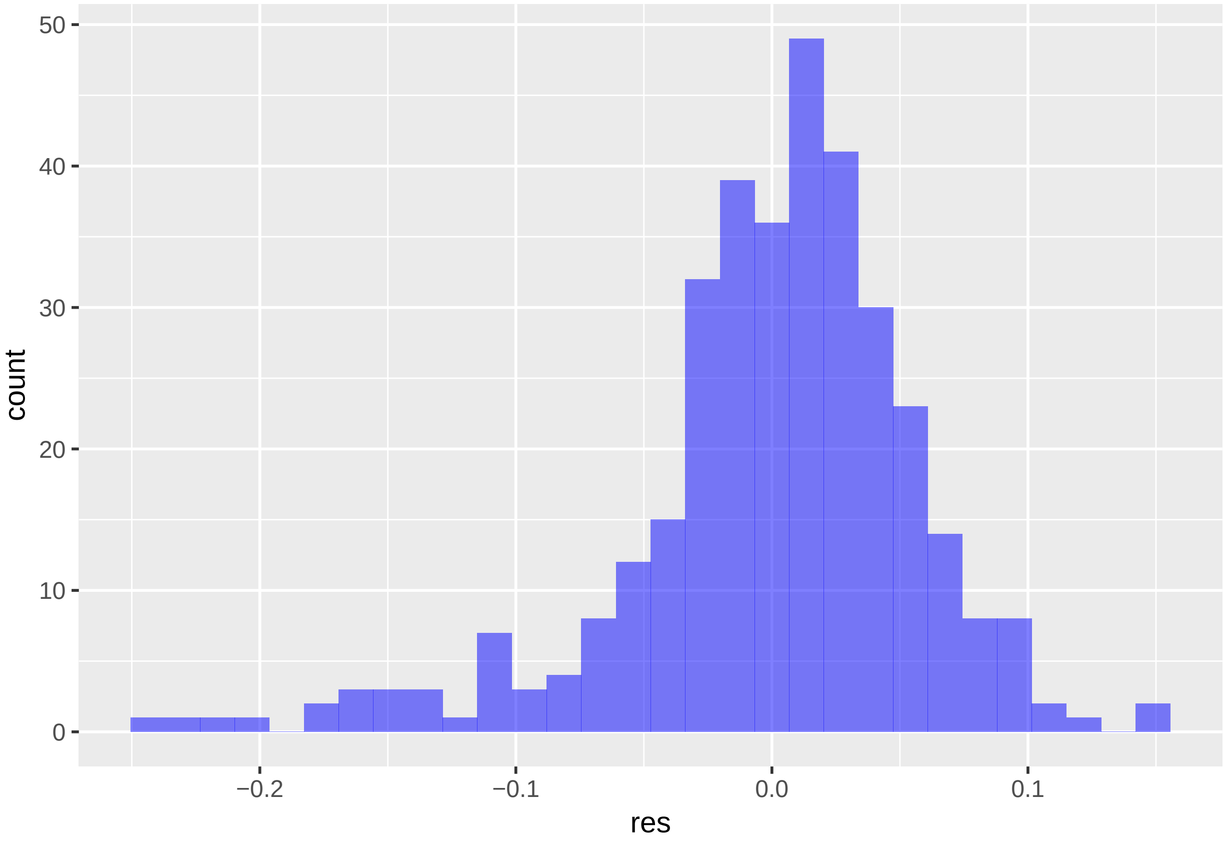
From the model, it can be seen that GRE score, CGPA, LOR, and Research are highly significance to the Chance of Admit. TOEFL and SOP are least significant. University Rating have no significnace to a persons chance of getting admission.

```
#Plotting the residuals
res <- residuals(model)
head(res)
```

```
##           1           2           4           7           8          11 
## -0.03363002 -0.04426833  0.05007661  0.04499320  0.08458662 -0.21400022
```

```
res <- as.data.frame(res)
ggplot(res,aes(res)) + geom_histogram(fill = 'blue', alpha = 0.5)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



This residuals shows the diference between the actual data points and the predicted regression model

```r
#Testing the model with the testing dataset
prediction <- predict(model,test_set)

results <- cbind(prediction, test_set$Chance.of.Admit)
colnames(results) <- c('Predicted', 'Actual')
print(results)
```

```
##     Predicted Actual
## 3   0.6527623   0.72
## 5   0.6261021   0.65
## 6   0.8796391   0.90
## 9   0.5520641   0.50
## 10  0.7199219   0.45
## 12  0.8353254   0.84
## 13  0.8515178   0.78
## 19  0.7430682   0.63
## 25  0.9571698   0.97
## 27  0.7666142   0.76
## 30  0.4836131   0.54
## 36  0.8617708   0.88
## 37  0.6578984   0.64
## 41  0.6496628   0.46
## 47  0.8917276   0.86
## 52  0.6054733   0.56
## 55  0.6540459   0.70
## 57  0.5358769   0.64
## 59  0.4413514   0.36
## 61  0.6054408   0.48
## 62  0.6285518   0.47
## 66  0.7811154   0.55
## 68  0.7367329   0.57
## 70  0.8641898   0.78
## 92  0.5554444   0.38
## 94  0.5816379   0.44
## 96  0.5505790   0.42
## 99  0.9027024   0.90
## 105 0.8139698   0.74
## 109 0.9196391   0.93
## 110 0.7083672   0.68
## 116 0.7985025   0.66
## 127 0.8469069   0.85
## 130 0.9250154   0.92
## 134 0.7818139   0.79
## 138 0.6375207   0.71
## 142 0.8898783   0.90
## 143 0.8950907   0.92
## 145 0.7997757   0.80
## 147 0.6615097   0.75
## 148 0.8210920   0.83
## 149 0.9525410   0.96
## 152 0.9099031   0.94
## 154 0.7404276   0.79
## 156 0.7060496   0.77
## 159 0.6026452   0.61
## 168 0.6320190   0.64
## 169 0.5595935   0.64
## 170 0.5881112   0.65
## 173 0.8446255   0.86
## 178 0.7749598   0.82
## 181 0.6123265   0.71
## 183 0.5660500   0.68
## 187 0.7402465   0.84
## 193 0.8271930   0.86
## 194 0.9473637   0.94
## 199 0.6952579   0.70
## 200 0.7323221   0.72
## 202 0.7144063   0.72
## 205 0.6684005   0.69
## 206 0.5172515   0.57
## 208 0.6495047   0.66
## 210 0.6480721   0.68
## 212 0.8544364   0.82
## 218 0.8271538   0.85
## 223 0.7824149   0.76
## 224 0.6795411   0.71
## 226 0.5573699   0.61
## 231 0.7202966   0.73
## 232 0.6942064   0.74
## 234 0.5935861   0.64
## 239 0.6491381   0.70
## 241 0.5243459   0.60
## 242 0.6168636   0.65
## 250 0.7886811   0.77
## 251 0.7147013   0.74
## 256 0.6933536   0.79
## 258 0.7633507   0.78
## 262 0.6412100   0.71
## 263 0.6749206   0.70
## 265 0.7583824   0.75
## 267 0.6486458   0.72
## 270 0.6982721   0.77
## 274 0.5817735   0.52
## 281 0.7339350   0.68
## 285 0.9509770   0.94
## 288 0.8589666   0.89
## 292 0.5378077   0.56
## 295 0.6553479   0.61
## 301 0.5970346   0.62
## 304 0.7435326   0.73
## 305 0.6490447   0.62
## 306 0.7737616   0.74
## 311 0.7476862   0.76
## 316 0.6048733   0.65
## 318 0.5453005   0.58
## 320 0.7794661   0.80
## 324 0.5955644   0.62
## 326 0.8473316   0.81
## 328 0.5337748   0.69
## 331 0.7726224   0.80
## 334 0.7319649   0.71
## 335 0.7526508   0.73
## 337 0.7277716   0.72
## 338 0.9371000   0.94
## 339 0.7862484   0.81
## 340 0.7716091   0.81
## 344 0.6099669   0.59
## 346 0.5051558   0.49
## 353 0.6206279   0.64
## 357 0.7887905   0.79
## 362 0.9119758   0.93
## 364 0.6324711   0.69
## 369 0.5120288   0.51
## 370 0.5887560   0.67
## 377 0.4750804   0.34
## 383 0.8427745   0.82
## 386 0.9817774   0.96
## 388 0.6105342   0.53
## 390 0.7335490   0.76
## 392 0.6964467   0.71
## 393 0.8368307   0.84
## 398 0.9160204   0.91
## 401 0.6143074   0.63
## 402 0.6578782   0.66
## 403 0.7897108   0.78
## 404 0.8675460   0.91
## 418 0.5726045   0.52
## 425 0.9024890   0.91
## 430 0.8911693   0.95
## 432 0.7734138   0.73
## 437 0.5504011   0.58
## 439 0.7233931   0.67
## 443 0.9120439   0.92
## 444 0.8575315   0.87
## 450 0.7685383   0.79
## 452 0.8683567   0.89
## 453 0.9165989   0.93
## 454 0.7472872   0.73
## 456 0.5273442   0.59
## 460 0.8734765   0.89
## 464 0.5924535   0.57
## 467 0.7434093   0.71
## 475 0.6235781   0.67
## 481 0.7878331   0.80
## 482 0.7208022   0.78
## 492 0.5580152   0.54
## 494 0.5951379   0.62
## 496 0.8419983   0.87
## 498 0.9437315   0.93
```

```
#Use Backward elimination to build an optimal model
model1 = lm(formula = Chance.of.Admit ~ GRE.Score + TOEFL.Score + SOP + LOR + CGPA + Research, data = tr
```

```r
summary(model1)
```

```
##
## Call:
## lm(formula = Chance.of.Admit ~ GRE.Score + TOEFL.Score + SOP +
##     LOR + CGPA + Research, data = training_set)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.242368 -0.025747  0.007207  0.033983  0.152234
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.3010723  0.1190916 -10.925  < 2e-16 ***
## GRE.Score    0.0020025  0.0005828   3.436 0.000663 ***
## TOEFL.Score  0.0025254  0.0010049   2.513 0.012430 *
## SOP          0.0110048  0.0049078   2.242 0.025581 *
## LOR          0.0134231  0.0046628   2.879 0.004243 **
## CGPA         0.1189182  0.0111054  10.708  < 2e-16 ***
## Research     0.0261395  0.0076526   3.416 0.000712 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05836 on 343 degrees of freedom
## Multiple R-squared:  0.8333, Adjusted R-squared:  0.8304
## F-statistic: 285.8 on 6 and 343 DF,  p-value: < 2.2e-16
```