DATA311_Project

Parsa Rajabi, Chelsey Hvingelby, Mackenzie Salloum, Cameron Chong, Jeff Bulmer

2019-03-31

Introduction

```
##   GRE.Score TOEFL.Score University.Rating SOP LOR CGPA Research
## 1       337         118                 4 4.5 4.5 9.65        1
## 2       324         107                 4 4.0 4.5 8.87        1
## 3       316         104                 3 3.0 3.5 8.00        1
## 4       322         110                 3 3.5 2.5 8.67        1
## 5       314         103                 2 2.0 3.0 8.21        0
## 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
```

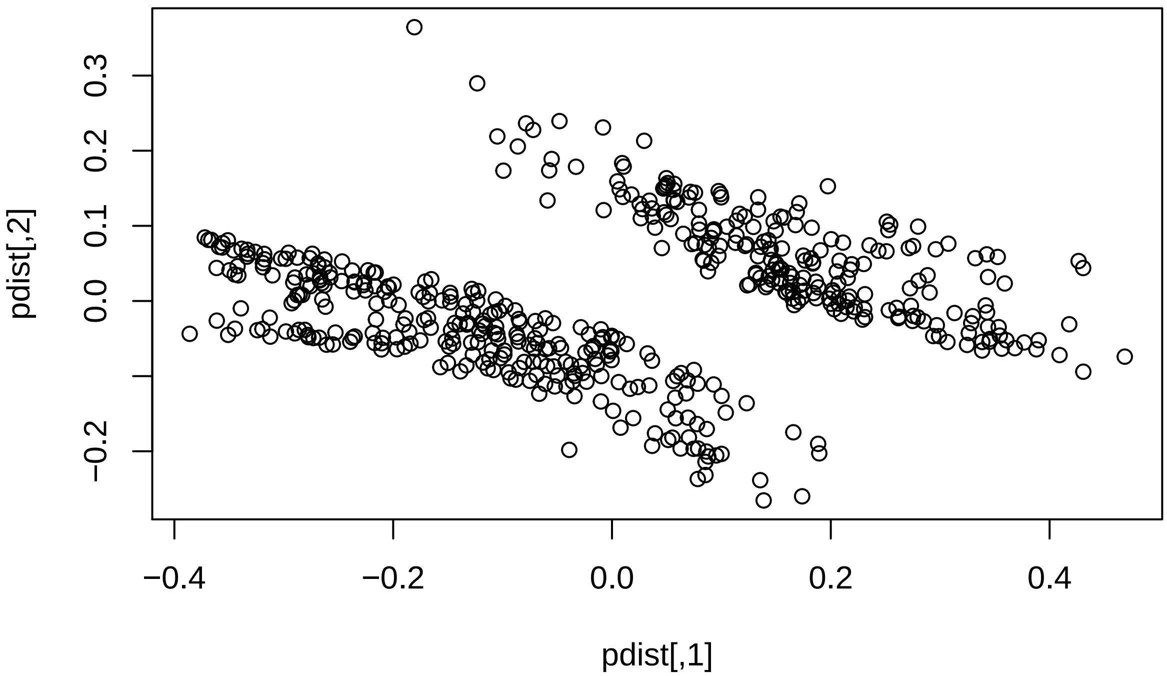
Data was collected from 500 prospective graduate students, including various scores achieved in the Test of English as a First Language (TOEFL) and Graduate Record Examinations (GRE), the strength of each candidates Statement of Purpose (SOP) and Letter of Recommendation (LOR), Undergraduate GPA (CGPA), and Research Experience. Finally, each candidate was polled about their confidence of being accepted into graduate school (Chance of Admit).

Using this data, we will perform analyses to attempt to predict variables within the dataset. The most obvious candidate for prediction is Chance of Admit, however, it may be worthwhile to attempt to predict other variables. In order to determine this, we will first run clustering on the dataset to see if any clear groups appear.

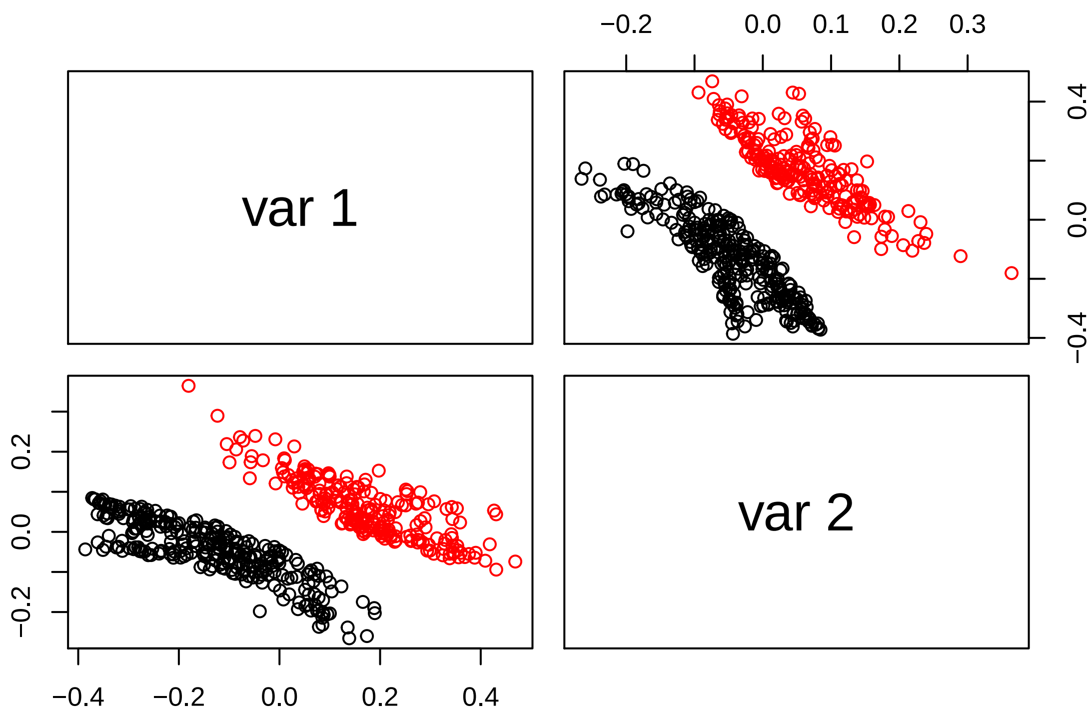
Clustering

We begin by computing the respective pairwise distances in our data, and plotting the output.

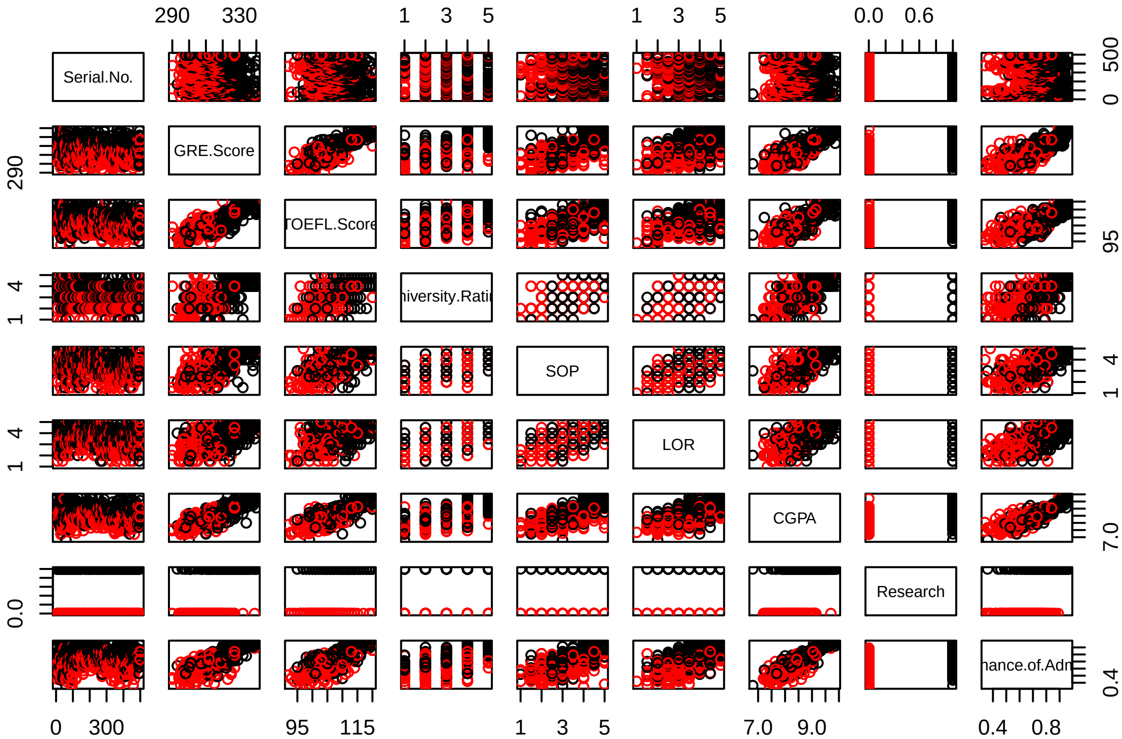
```
## Warning in daisy(admissionsData, metric = "gower"): binary variable(s) 8
## treated as interval scaled
```



We quickly see that two clear groups appear. We can isolate these two groups using hierarchical clustering with single-linkage chaining.



We can then use scatterplots to show the entirety of the data, while still keeping the groups intact, to see if we can determine which predictors most affect these clusters.



We notice that, using the single linkage chaining from above, we can predict whether or not a student performs research almost perfectly. Additionally, we get decent results pertaining to the Chance of Admission and the GRE Score. Based on these clustering results, our two groups (Research and Non-Research) divide up pretty well as follows: In general, Research has a higher chance of admission than Non-Research, as well as a higher GRE and TOEFL score.

So, by applying Gower's Distance on all predictors and using single-linkage chaining, we have two clear clusters directly coinciding with the presence of a research variable. From here, we can try and run some predictive models using research as the response variable.

Principal Component Analysis

With Response Variable Chance.of.Admit

The variable we are interested in predicting, Chance.of.Admit, is the 9th variable.

Run PCA on the data and remove the response variable (chance of admit) and the unique identifier (serial number)

```r
set.seed(43849)
pca.admin <- prcomp(as.matrix(admissionsData[,-c(1,9)]), scale = TRUE)
summary(pca.admin)
```

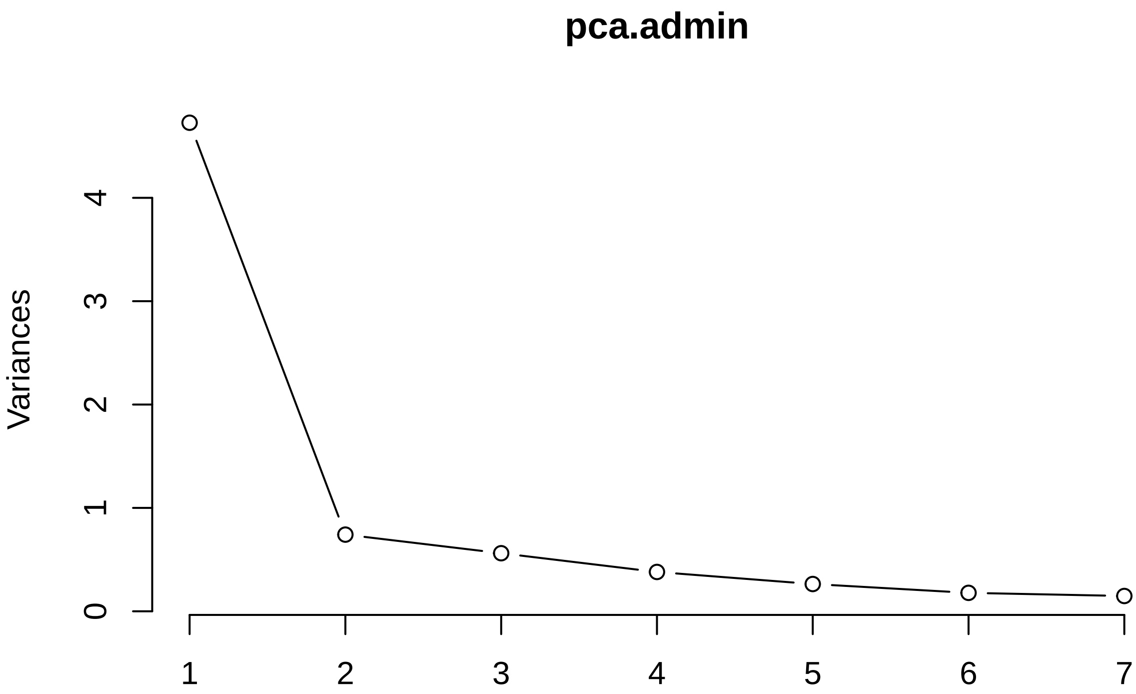
```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6
## Standard deviation     2.1740 0.8612 0.74942 0.61674 0.51349 0.42223
## Proportion of Variance 0.6752 0.1060 0.08023 0.05434 0.03767 0.02547
## Cumulative Proportion  0.6752 0.7812 0.86139 0.91573 0.95340 0.97886
##                            PC7
## Standard deviation     0.38464
## Proportion of Variance 0.02114
## Cumulative Proportion  1.00000
```

To choose the number of principal components to keep, we can either use the Kaiser criterion, cumulative proportion/percent of variance, or a scree plot.

Using the Kaiser criterion, we keep all principal components with a standard deviation greater than 1 (since the data is scaled). Hence the Kaiser criterian is telling us to keep the first principal component.

I will now compare this with a scree plot.

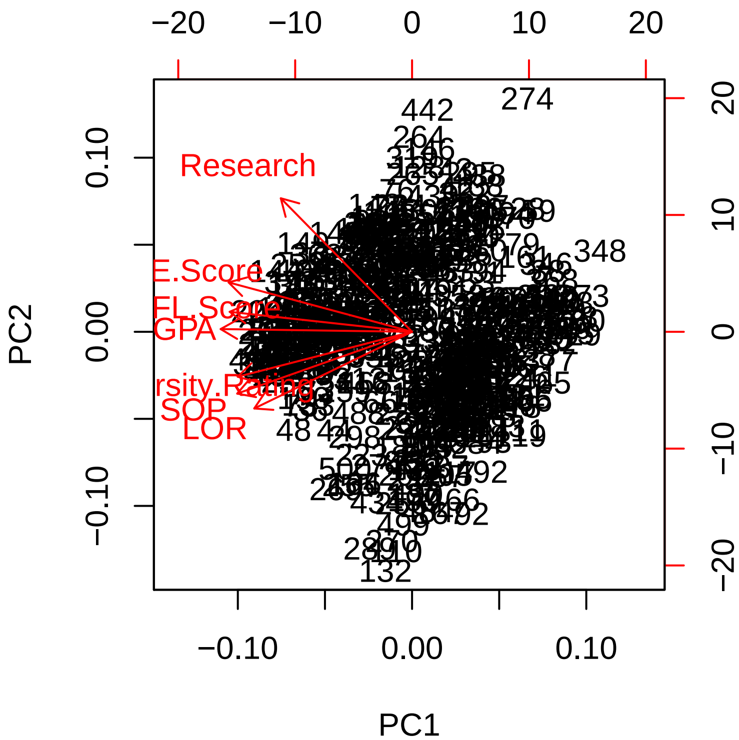
```
plot(pca.admin, type="lines")
```



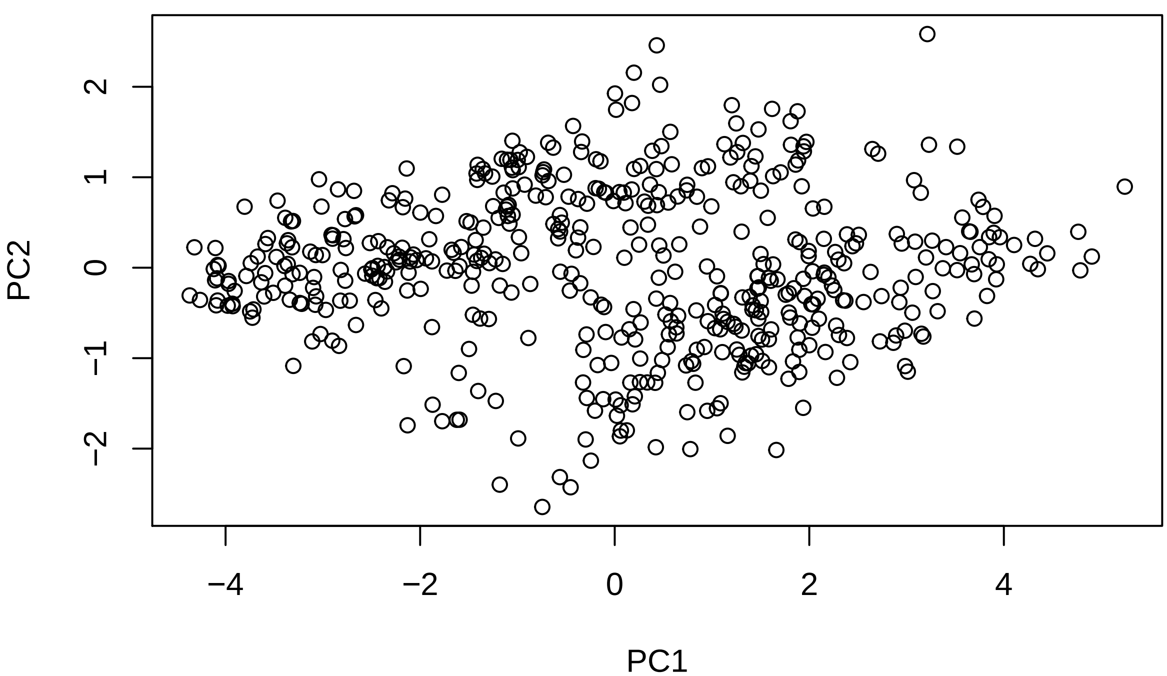
The above scree plot plots the monotonically decreasing eigenvalues and the location of an 'elbow' or plateau indicates the number of principal components. The scree plot suggests probably 2 principal components.

The first two principal components that will be retained explain 78% of the variation in the data. We can now view the data projected onto the components using a biplot.

```
biplot(pca.admin)
```

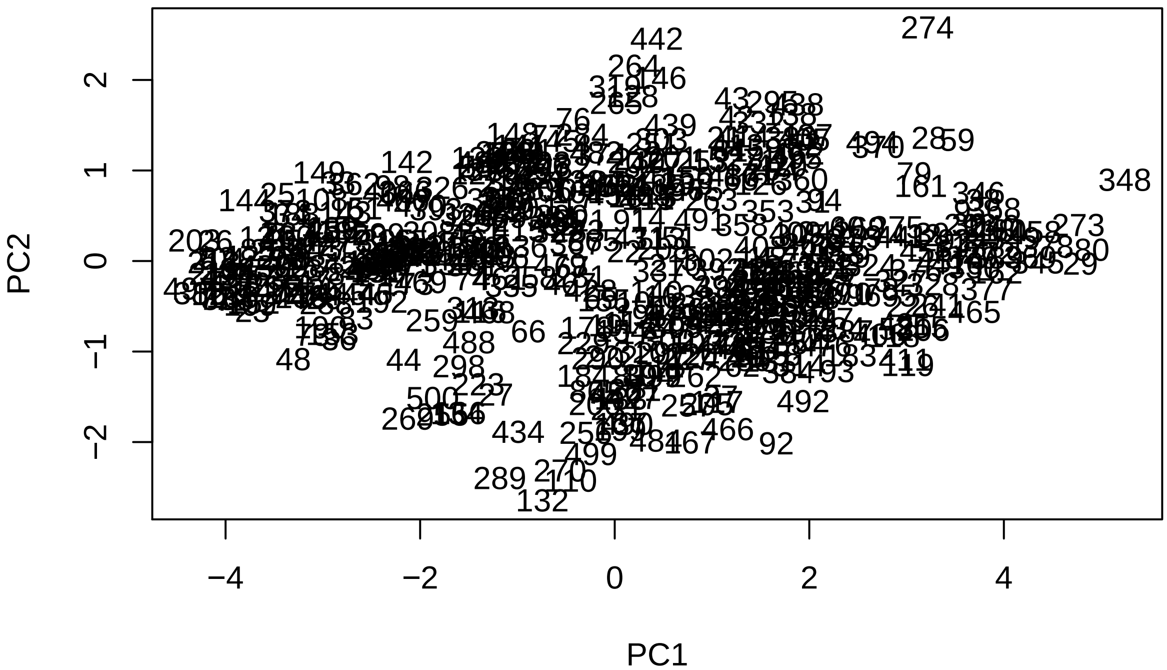



```r
plot(pca.admin$x[,1:2])
```



We can put data labels on the biplot by observation number

```r
plot(pca.admin$x[,1:2], type = "n")
text(pca.admin$x[,1:2], labels = 1:nrow(admissionsData))
```



It looks like there are two groups in the above principal component plots.

Take a look at the component loadings (eigenvectors) which provide the coefficients of the original variables, rounded to 2 decimal places.

```
round(pca.admin$rotation[,1:2], 2)
```

```
##                     PC1   PC2
## GRE.Score         -0.40  0.27
## TOEFL.Score       -0.40  0.11
## University.Rating -0.38 -0.25
## SOP               -0.38 -0.34
## LOR               -0.35 -0.43
## CGPA              -0.42  0.02
## Research          -0.29  0.74
```

These are the coefficients of the original variables. The magnitudes are pretty similar for the first component, perhaps with the exception of research. They are also all containing the same sign. This is a little difficult to interpret, but most likely indicates that the first principal component is equally weighting all predictor variables, with the exception of research.

In the second component, the highest magnitude is the research aspect, along with the letter of recommendation. Perhaps this component indicates previous experience a student has. A reference letter most likely comes from someone you have worked with, conducted research with, volunteered with, or TA'd for. Therefore a good reference letter coupled with research experience could be indicative of research and other activities in both academic and non-academic settings.

We can now look at the four students who scored highest on PC1:

```
admissionsData[order(pca.admin$x[,1], decreasing = TRUE)[1:4],1:9]
```

```
##     Serial.No. GRE.Score TOEFL.Score University.Rating SOP LOR CGPA
## 348        348       299          94                 1 1.0 1.0 7.34
## 80          80       294          93                 1 1.5 2.0 7.36
## 29          29       295          93                 1 2.0 2.0 7.20
## 273        273       294          95                 1 1.5 1.5 7.64
```

```
##     Research Chance.of.Admit
## 348        0            0.42
## 80         0            0.46
## 29         0            0.46
## 273        0            0.49
```

It is noted that the four students who performed highest on PC1 all had a low belief of their chance of admit. None of them had research, and all had a similar cumulative GPA. In addition, the universities where all rated low (1 to be exact) and the students had similar GRE and TOEFL scores (well below the average). These students in general seem to be ones who are not performing scoring very well across all predictors.

And the four students who scored highest on PC2:

```r
admissionsData[order(pca.admin$x[,2], decreasing = TRUE)[1:4], 1:9]
```

```
##     Serial.No. GRE.Score TOEFL.Score University.Rating SOP LOR CGPA
## 274        274       312          99                 1 1.0 1.5 8.01
## 442        442       332         112                 1 1.5 3.0 8.66
## 264        264       324         111                 3 2.5 1.5 8.79
## 146        146       320         113                 2 2.0 2.5 8.64
##     Research Chance.of.Admit
## 274        1            0.52
## 442        1            0.79
## 264        1            0.70
## 146        1            0.81
```

Notice that the four students who performed highest on PC2 all have research experience. In general, these students are scoring better than the students in principal component 1 across the board.

With Response Variable Research

The variable we are interested in predicting, Chance.of.Admit, is the 8th variable.

Run PCA on the data and remove the response variable (research) and the unique identifier (serial number)

```r
set.seed(43849)
pca.admin2 <- prcomp(as.matrix(admissionsData[,-c(1,8)]), scale = TRUE)
summary(pca.admin2)
```

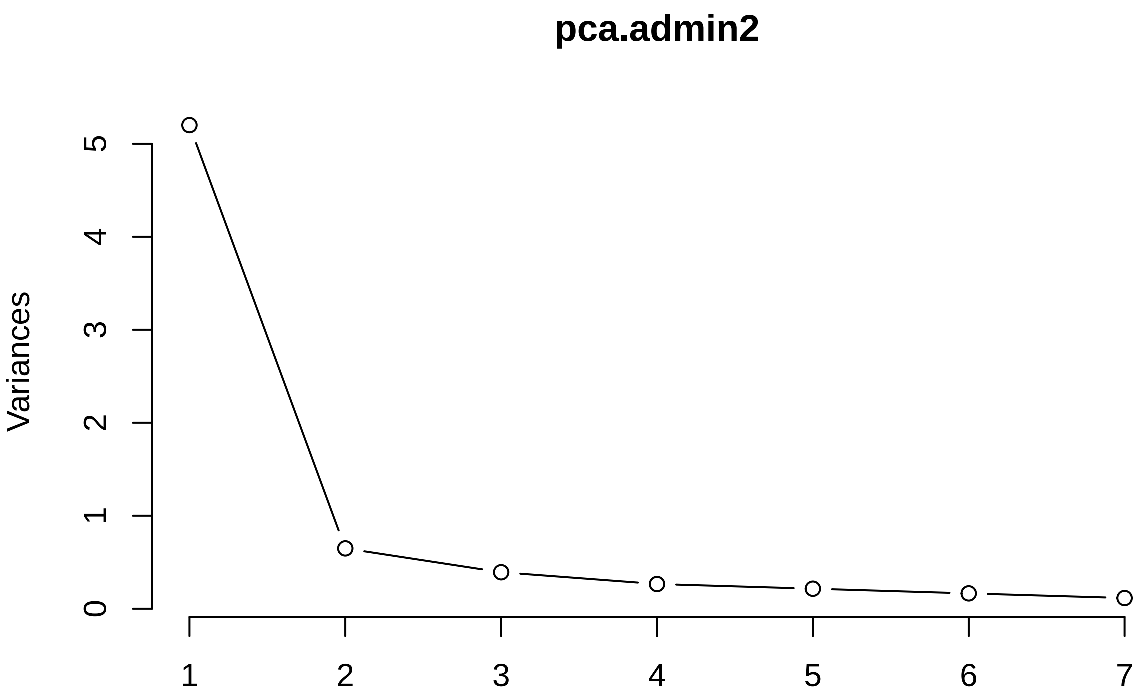
```
## Importance of components:
##                           PC1     PC2     PC3    PC4     PC5     PC6
## Standard deviation     2.2803 0.80529 0.62599 0.5150 0.46369 0.40586
## Proportion of Variance 0.7429 0.09264 0.05598 0.0379 0.03071 0.02353
## Cumulative Proportion  0.7429 0.83549 0.89147 0.9294 0.96008 0.98361
##                            PC7
## Standard deviation     0.33868
## Proportion of Variance 0.01639
## Cumulative Proportion  1.00000
```

To choose the number of principal components to keep, we can either use the Kaiser criterian, cumulative proportion/percent of variance, or a scree plot.

Using the Kaiser criterian, we keep all principal components with a standard deviation greater than 1 (since the data is scaled). Hence the Kaiser criterian is telling us to keep the first principal component.

I will now compare this with a scree plot.

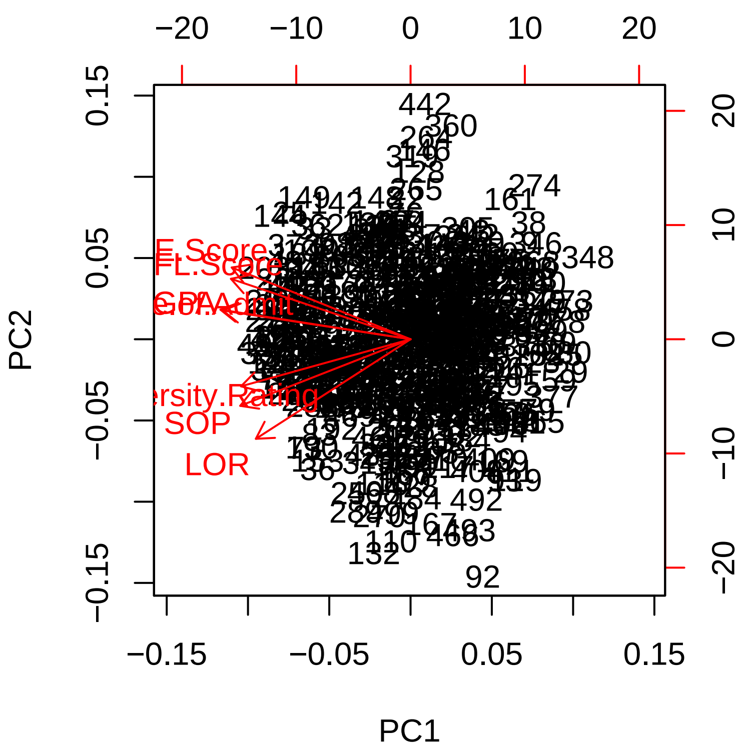
```r
plot(pca.admin2, type="lines")
```



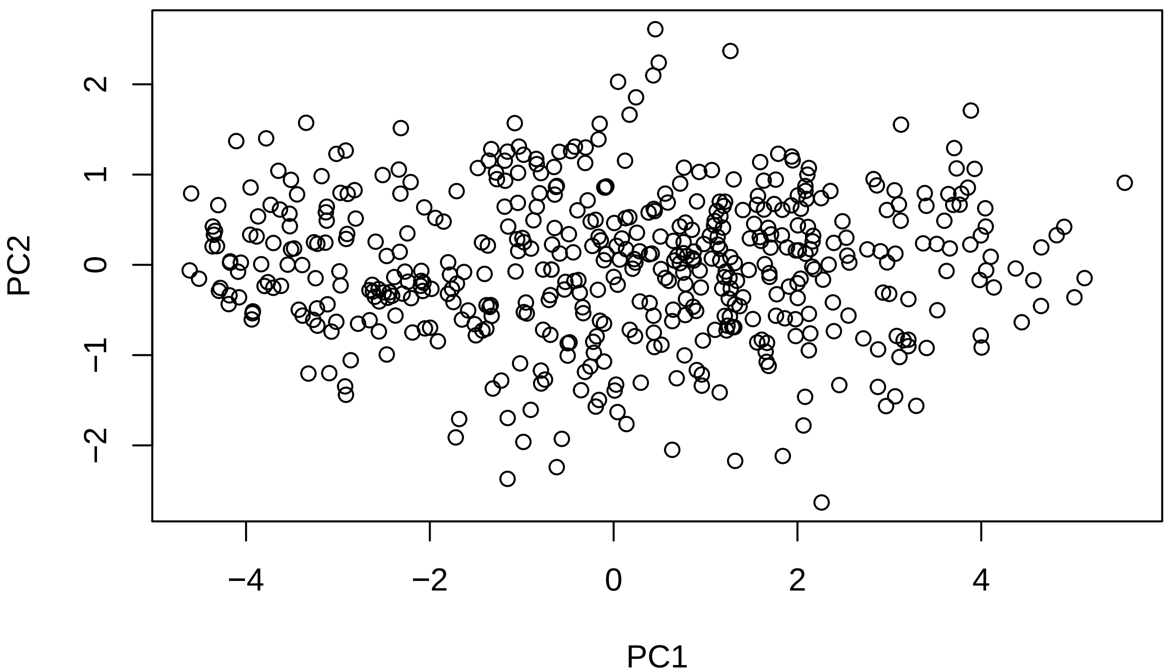
The above scree plot plots the monotonically decreasing eigenvalues and the location of an 'elbow' or plateau indicates the number of principal components. The scree plot suggests probably 2 principal components.

The first two principal components that will be retained explain 84% of the variation in the data. We can now view the data projected onto the components using a biplot.

```r
biplot(pca.admin2)
```

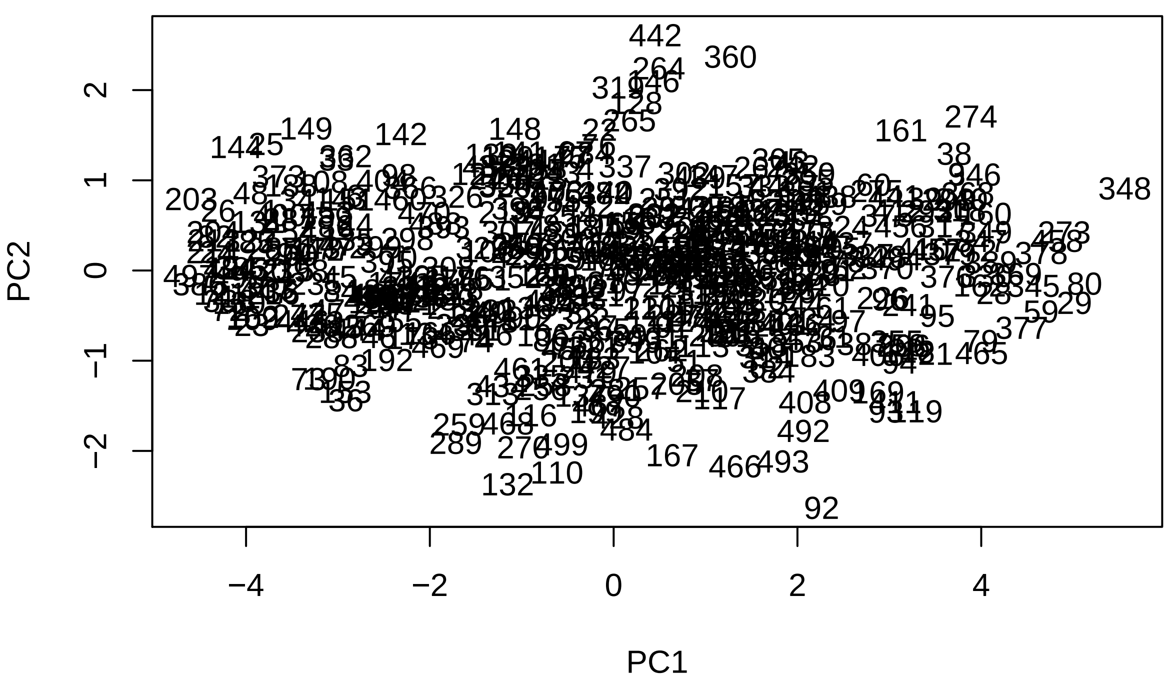


```r
plot(pca.admin2$x[,1:2])
```



We can put data labels on the biplot by observation number

```r
plot(pca.admin2$x[,1:2], type = "n")
text(pca.admin2$x[,1:2], labels = 1:nrow(admissionsData))
```



It looks like there are two groups in the above principal component plots.

Take a look at the component loadings (eigenvectors) which provide the coefficients of the original variables, rounded to 2 decimal places.

```r
round(pca.admin2$rotation[,1:2], 2)
```

```
##                     PC1   PC2
## GRE.Score         -0.38  0.44
## TOEFL.Score       -0.39  0.37
## University.Rating -0.36 -0.29
## SOP               -0.37 -0.40
## LOR               -0.33 -0.61
## CGPA              -0.41  0.18
## Chance.of.Admit   -0.40  0.17
```

These are the coefficients of the original variables. The magnitudes are extremely similar for the first component. They are also all containing the same sign. This is a little difficult to interpret again, but most likely indicates that the first principal component is equally weighting all predictor variables.

In the second component, the highest magnitude is the lettor of recommendation which has a negative sign. Other variables with the same sign include the SOP score and the university rating. Variables of opposite sign with higher magnitude include GRE Score, TOEFL Score, as well as CGPA and Chance of Admit having a lower magnitude. Students who score high on this principal component, likely scored high on their standardized tests.

We can now look at the four students who scored highest on PC1:

```r
admissionsData[order(pca.admin2$x[,1], decreasing = TRUE)[1:4],1:9]
```

```
##     Serial.No. GRE.Score TOEFL.Score University.Rating SOP LOR CGPA
## 348        348       299          94                 1 1.0 1.0 7.34
## 80          80       294          93                 1 1.5 2.0 7.36
## 29          29       295          93                 1 2.0 2.0 7.20
## 273        273       294          95                 1 1.5 1.5 7.64
##     Research Chance.of.Admit
## 348        0            0.42
## 80         0            0.46
## 29         0            0.46
## 273        0            0.49
```

The top four students in this first principal component are the same as the first four students in the previous PC1 (compared using Serial.No.). Even when looking at the loadings, this principal component is very similar to the principal component in the previous section.

And the four students who scored highest on PC2:

```r
admissionsData[order(pca.admin2$x[,2], decreasing = TRUE)[1:4], 1:9]
```

```
##     Serial.No. GRE.Score TOEFL.Score University.Rating SOP LOR CGPA
## 442        442       332         112                 1 1.5 3.0 8.66
## 360        360       321         107                 2 2.0 1.5 8.44
## 264        264       324         111                 3 2.5 1.5 8.79
## 146        146       320         113                 2 2.0 2.5 8.64
##     Research Chance.of.Admit
## 442        1            0.79
## 360        0            0.81
## 264        1            0.70
## 146        1            0.81
```

As hypothesized above, the first four students in PC2 are scoring higher on their standardized tests (GRE.Score and TOEFL.Score). These students are performing the at, or above average on these standardized tests. However, they all have a below average score on SOP, and LOR. The CGPA of the students scoring high on

PC2 hovers fairly close to the mean. This proves the initial hypothesis that standardized testing is most important for PC2.

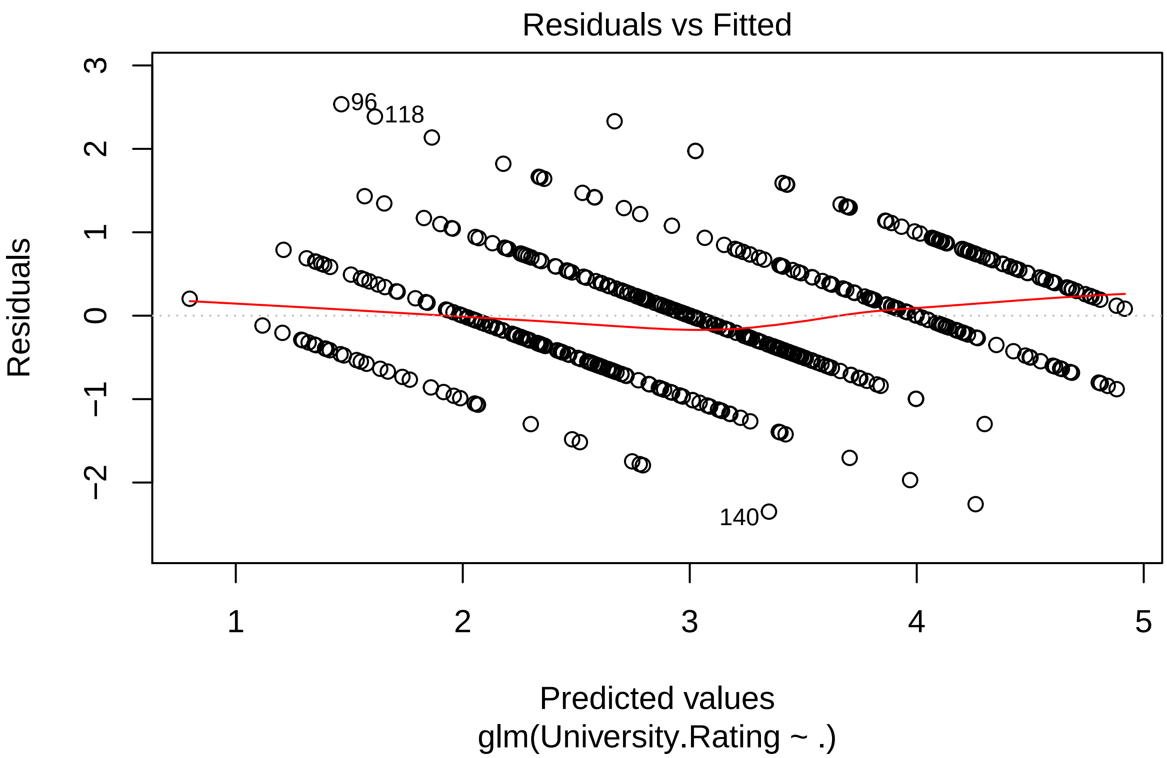
Logmod Analysis and Plots

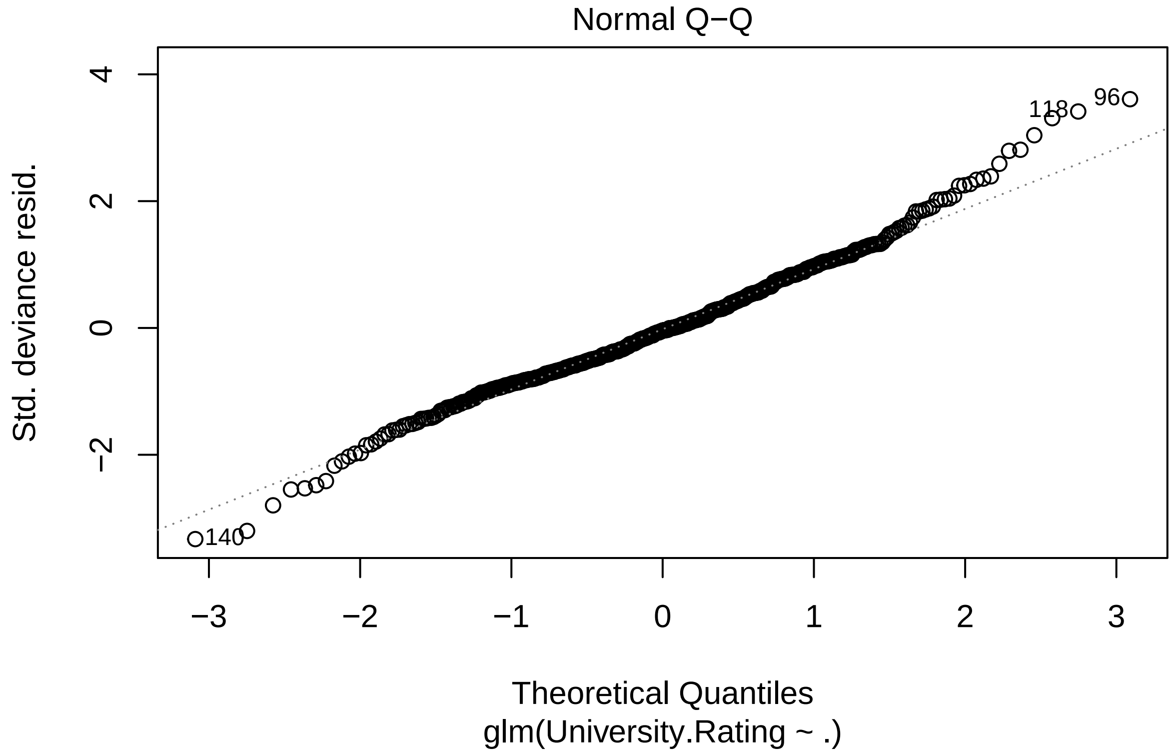
Here's a logmod analysis. No variable selection performed though.

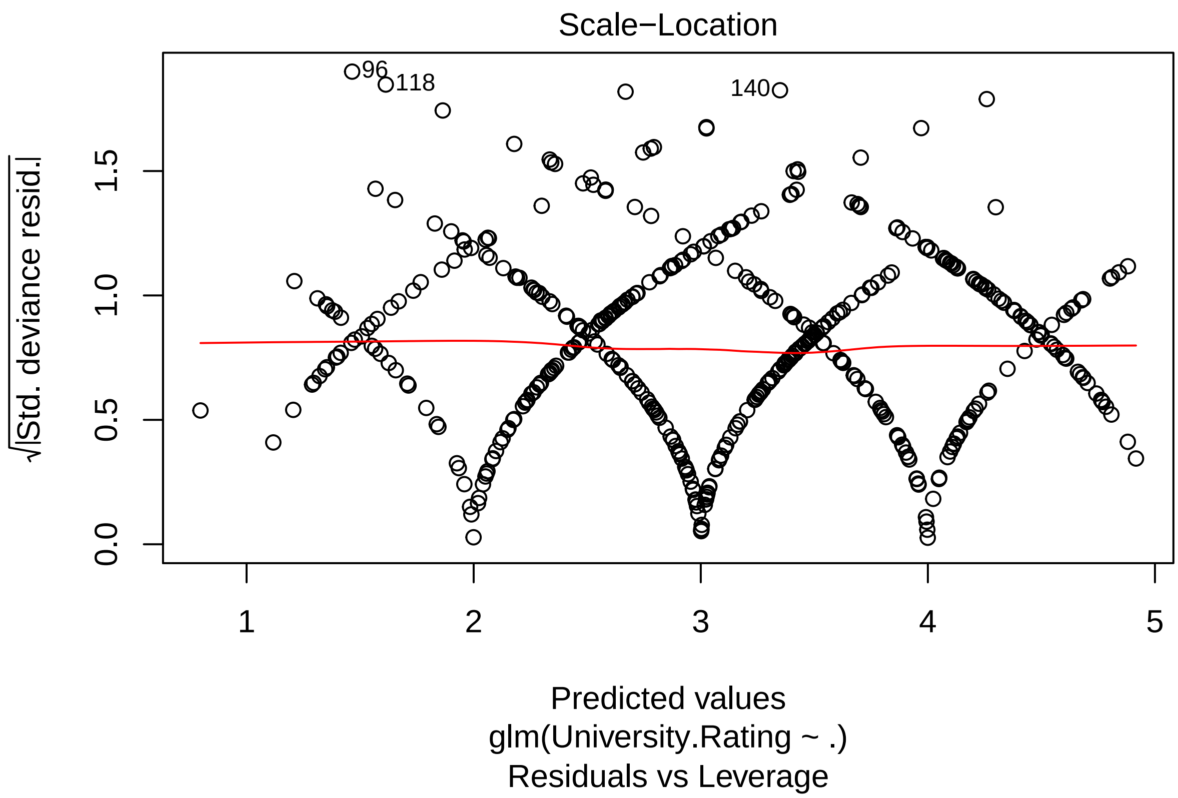
```
University.Rating <- factor(University.Rating)
Research <- factor(Research)
logmod <- glm(University.Rating ~., data=admissionsData)
summary(logmod)
```

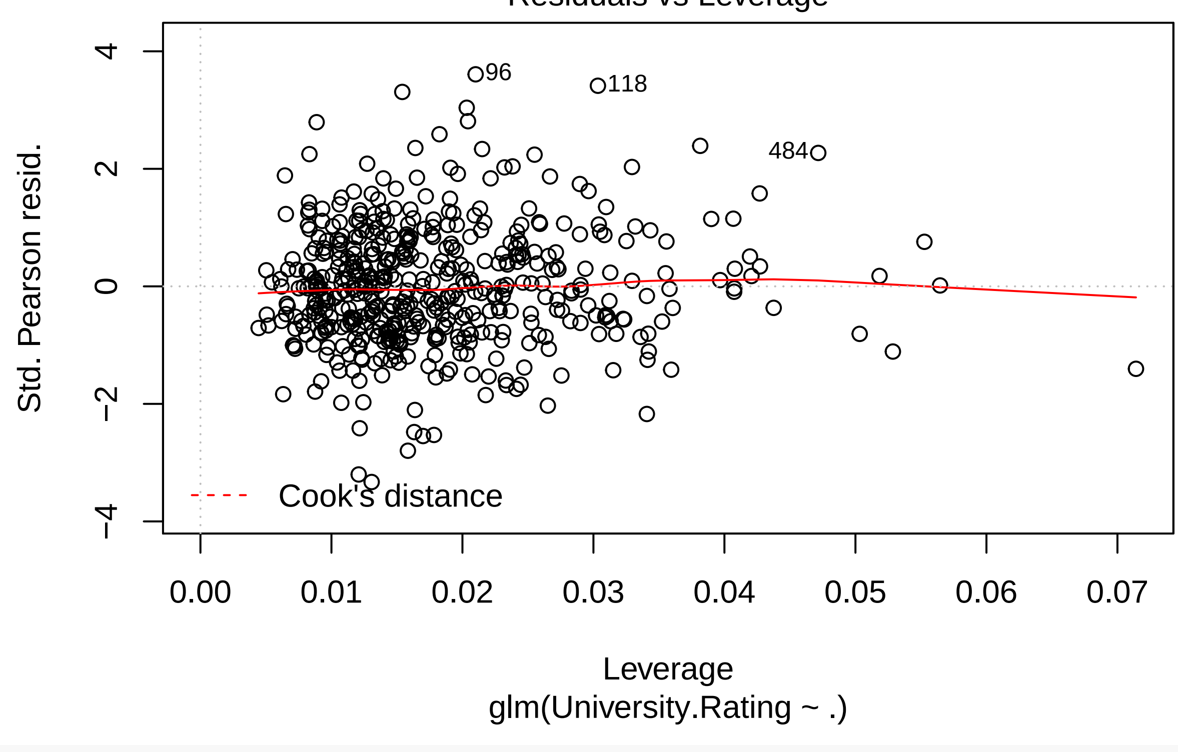
```
##
## Call:
## glm(formula = University.Rating ~ ., data = admissionsData)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.34889  -0.46404  -0.02909   0.43638   2.53513
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -5.3520556  1.4229030  -3.761 0.000189 ***
## Serial.No.       0.0001131  0.0002308   0.490 0.624275
## GRE.Score        0.0050723  0.0060361   0.840 0.401135
## TOEFL.Score      0.0184033  0.0104963   1.753 0.080172 .
## SOP              0.4420126  0.0508516   8.692  < 2e-16 ***
## LOR              0.1376178  0.0495241   2.779 0.005665 **
## CGPA             0.2666732  0.1306889   2.041 0.041833 *
## Research         0.0744728  0.0792227   0.940 0.347657
## Chance.of.Admit  0.7761573  0.5441596   1.426 0.154405
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.5042716)
##
##     Null deviance: 652.5  on 499  degrees of freedom
## Residual deviance: 247.6  on 491  degrees of freedom
## AIC: 1087.5
##
## Number of Fisher Scoring iterations: 2
```

```
plot(logmod)
```


Residuals vs Fitted
Residuals
Predicted values
glm(University.Rating ~ .)
96
118
140
3
2
1
0
−1
−2
1
2
3
4
5


Normal Q−Q
Std. deviance resid.
Theoretical Quantiles
glm(University.Rating ~ .)
118
96
140
4
2
0
−2
−3
−2
−1
0
1
2
3






```
linear.full <- lm(Chance.of.Admit ~., data=admissionsData)
linear.null <- lm(Chance.of.Admit ~ 1, data=admissionsData)

linear.rank.full <- lm(University.Rating ~., data=admissionsData)
linear.null.full <- lm(University.Rating ~ 1, data=admissionsData)
```

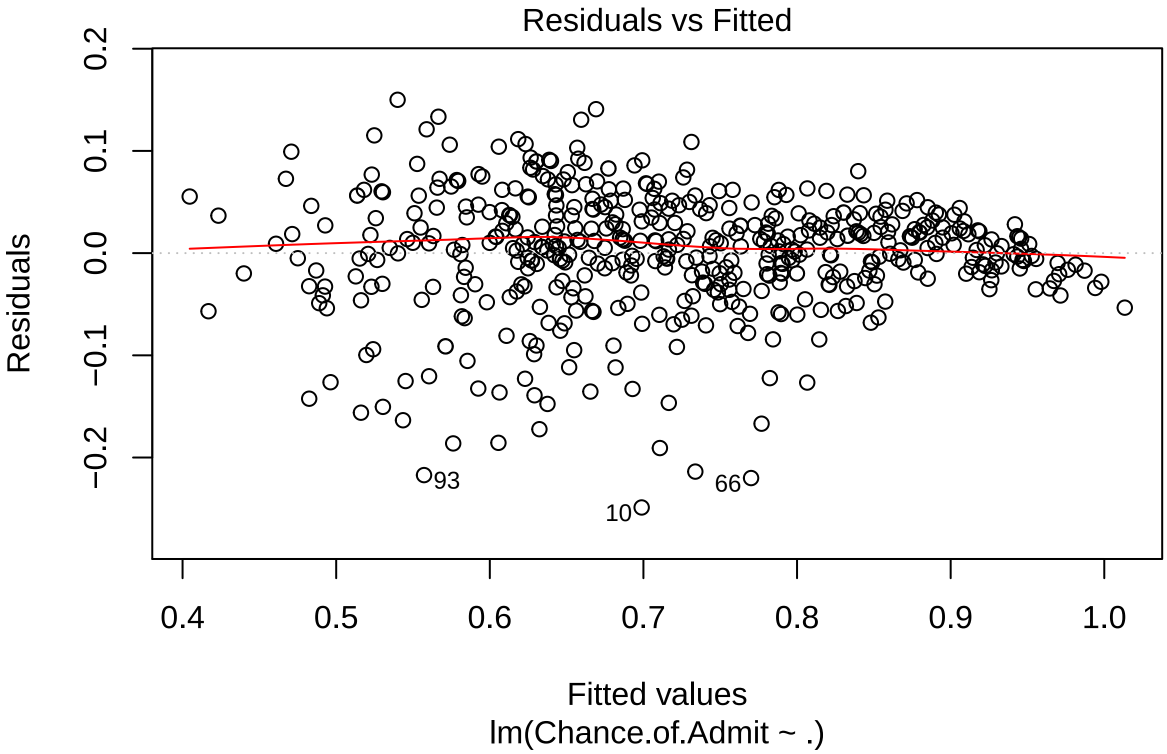
Linear Regression and some plots

Here's a linear model with a few plots.

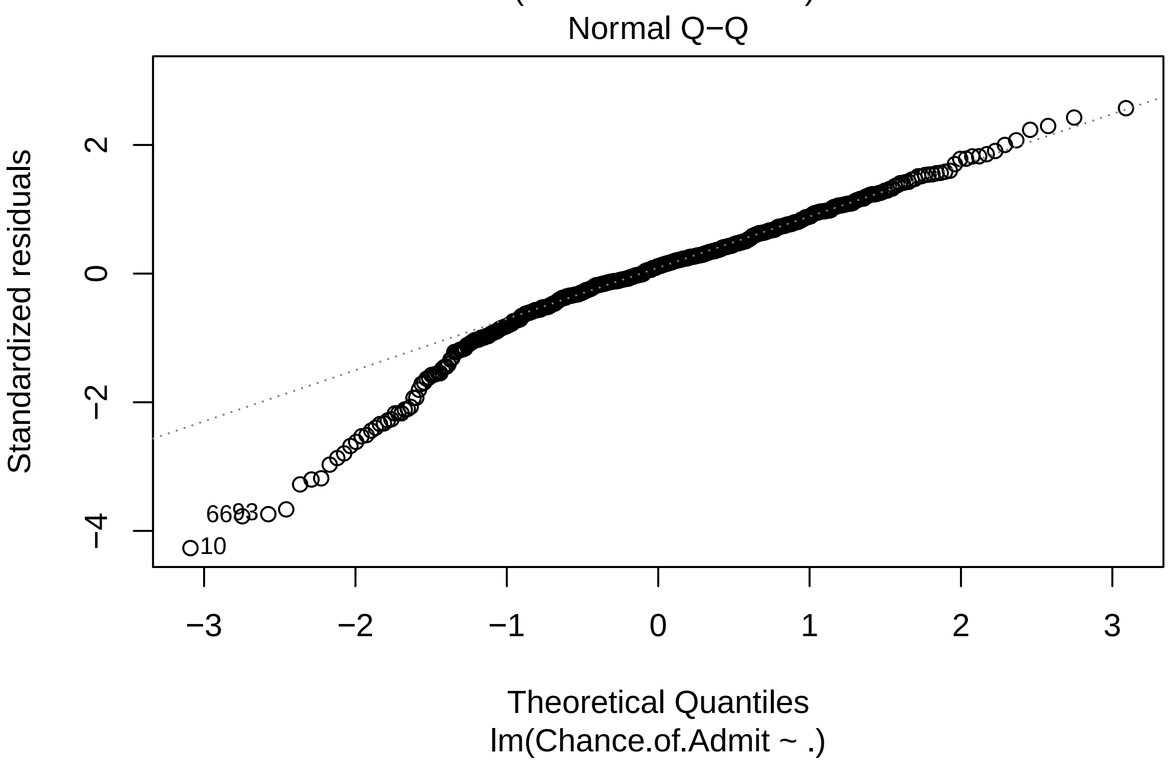
```
linear <- lm(Chance.of.Admit ~., data=admissionsData)
summary(linear)
```

```
##
## Call:
## lm(formula = Chance.of.Admit ~ ., data = admissionsData)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.248847 -0.025984  0.006627  0.036671  0.150015
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -1.3379983  0.1030617 -12.982  < 2e-16 ***
## Serial.No.         0.0000868  0.0000187   4.641 4.44e-06 ***
## GRE.Score          0.0019217  0.0004923   3.903 0.000108 ***
## TOEFL.Score        0.0031928  0.0008594   3.715 0.000227 ***
## University.Rating  0.0053164  0.0037273   1.426 0.154405
## SOP                0.0045661  0.0045161   1.011 0.312489
## LOR                0.0149151  0.0040757   3.660 0.000280 ***
## CGPA               0.1155561  0.0095282  12.128  < 2e-16 ***
## Research           0.0225254  0.0064834   3.474 0.000557 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05877 on 491 degrees of freedom
## Multiple R-squared:  0.8294, Adjusted R-squared:  0.8266
## F-statistic: 298.4 on 8 and 491 DF,  p-value: < 2.2e-16
```

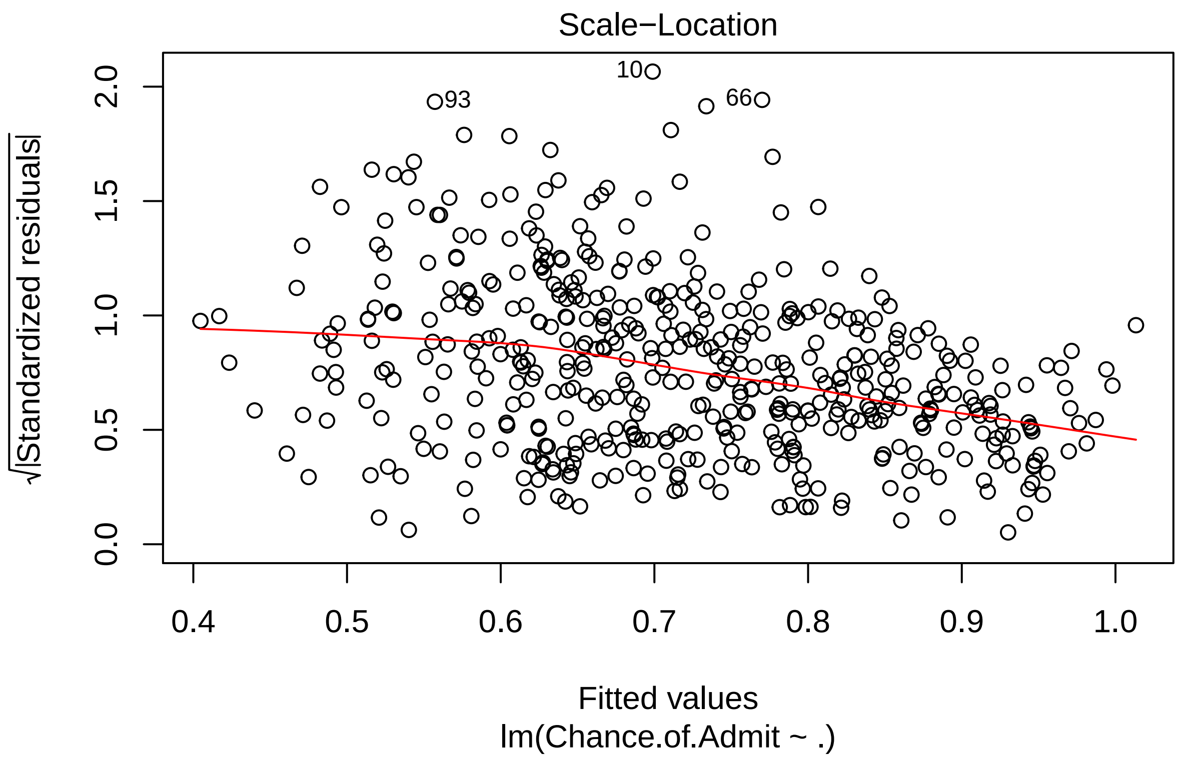
```
plot(linear)
```

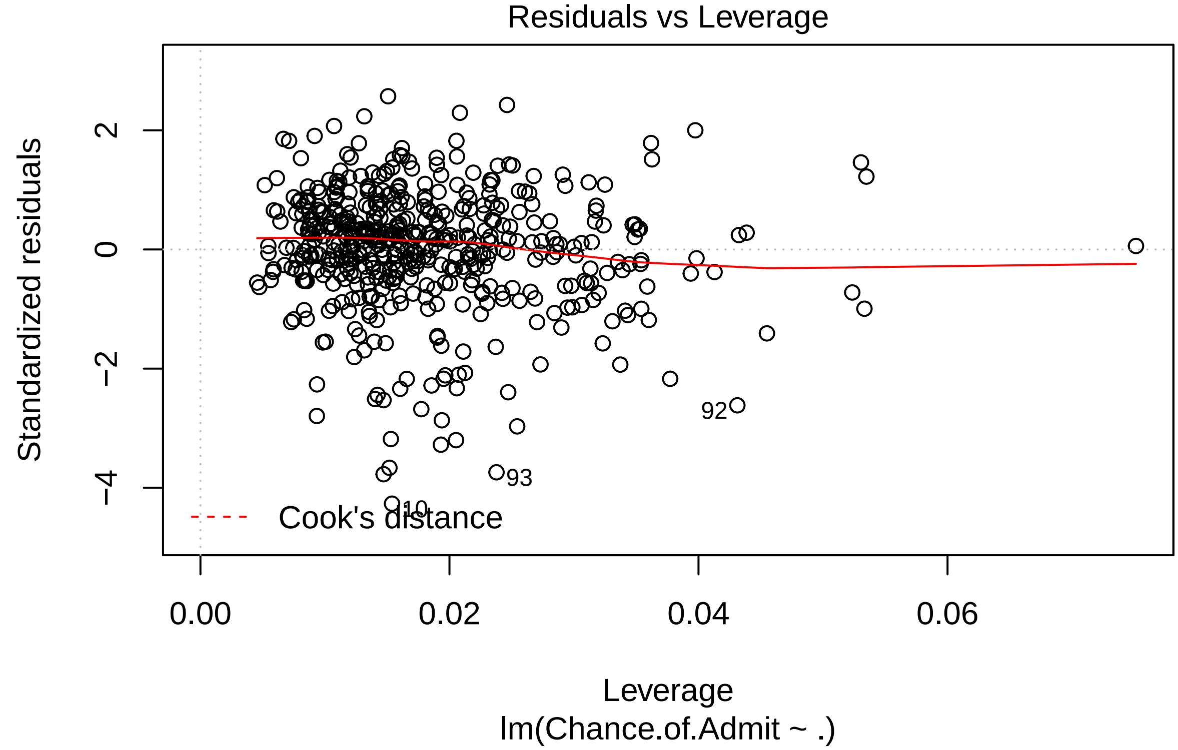


Residuals vs Fitted
Residuals
0.2
0.1
0.0
−0.1
−0.2
93
66
10
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Fitted values
lm(Chance.of.Admit ~ .)



Normal Q−Q
Standardized residuals
2
0
−2
−4
6693
10
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(Chance.of.Admit ~ .)





Variable Selection for Chance of Admittion

By performing backwards selection, we will remove the least significant values until all values are significant.

```r
linear <- lm(Chance.of.Admit~ ., data = admissionsData )
summary(linear)
```

```
##
## Call:
## lm(formula = Chance.of.Admit ~ ., data = admissionsData)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.248847 -0.025984  0.006627  0.036671  0.150015
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -1.3379983  0.1030617 -12.982  < 2e-16 ***
## Serial.No.         0.0000868  0.0000187   4.641 4.44e-06 ***
## GRE.Score          0.0019217  0.0004923   3.903 0.000108 ***
## TOEFL.Score        0.0031928  0.0008594   3.715 0.000227 ***
## University.Rating  0.0053164  0.0037273   1.426 0.154405
## SOP                0.0045661  0.0045161   1.011 0.312489
## LOR                0.0149151  0.0040757   3.660 0.000280 ***
## CGPA               0.1155561  0.0095282  12.128  < 2e-16 ***
## Research           0.0225254  0.0064834   3.474 0.000557 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05877 on 491 degrees of freedom
## Multiple R-squared:  0.8294, Adjusted R-squared:  0.8266
## F-statistic: 298.4 on 8 and 491 DF,  p-value: < 2.2e-16
```

```r
#Remove University Ranking because it has the highest non significant p value
linear <- lm(Chance.of.Admit~ Serial.No. + GRE.Score + TOEFL.Score + SOP +LOR + CGPA + Research , data =
summary(linear)
```

```
##
## Call:
## lm(formula = Chance.of.Admit ~ Serial.No. + GRE.Score + TOEFL.Score +
##     SOP + LOR + CGPA + Research, data = admissionsData)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.249225 -0.026058  0.005588  0.037182  0.150359
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.372e+00  1.004e-01 -13.673  < 2e-16 ***
## Serial.No.   8.776e-05  1.871e-05   4.691 3.53e-06 ***
## GRE.Score    1.957e-03  4.922e-04   3.975 8.09e-05 ***
## TOEFL.Score  3.304e-03  8.568e-04   3.857 0.000130 ***
## SOP          6.945e-03  4.201e-03   1.653 0.098981 .
## LOR          1.571e-02  4.041e-03   3.888 0.000115 ***
## CGPA         1.175e-01  9.444e-03  12.437  < 2e-16 ***
## Research     2.302e-02  6.481e-03   3.551 0.000420 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.05883 on 492 degrees of freedom
## Multiple R-squared:  0.8287, Adjusted R-squared:  0.8262
## F-statistic:   340 on 7 and 492 DF,  p-value: < 2.2e-16
```

```
#Remove SOP has the second highest non significant p value
linear <- lm(Chance.of.Admit~ Serial.No. + GRE.Score + TOEFL.Score +LOR + CGPA + Research , data = admi
#All variables are now significant
summary(linear)
```

```
##
## Call:
## lm(formula = Chance.of.Admit ~ Serial.No. + GRE.Score + TOEFL.Score +
##     LOR + CGPA + Research, data = admissionsData)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.247948 -0.026442  0.005457  0.036306  0.152463
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.406e+00  9.844e-02 -14.280  < 2e-16 ***
## Serial.No.   8.348e-05  1.856e-05   4.498 8.58e-06 ***
## GRE.Score    1.941e-03  4.930e-04   3.937 9.42e-05 ***
## TOEFL.Score  3.478e-03  8.518e-04   4.083 5.18e-05 ***
## LOR          1.831e-02  3.729e-03   4.911 1.23e-06 ***
## CGPA         1.215e-01  9.132e-03  13.310  < 2e-16 ***
## Research     2.357e-02  6.484e-03   3.635 0.000307 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05894 on 493 degrees of freedom
## Multiple R-squared:  0.8277, Adjusted R-squared:  0.8256
## F-statistic: 394.8 on 6 and 493 DF,  p-value: < 2.2e-16
```

CV for linear model - Chance of Admission

```
set.seed(7861)

cvlm <- list()
msecv <- NA
coef <-matrix(nrow = 500, ncol=length(linear$coefficients))
for(i in 1:nrow(admissionsData)){
  #Fit the linear model
cvlm[[i]] <- lm(Chance.of.Admit[-i] ~ Serial.No.[-i] + GRE.Score[-i] + TOEFL.Score[-i] +LOR[-i] + CGPA[
# Calculate MSE for ith model
msecv[i] <- (predict(cvlm[[i]], newdata = data.frame(Serial.No.[-i] + GRE.Score[-i] + TOEFL.Score[-i] +
#coef[[i]] <- cvlm[[i]]$coefficients
  for(j in 1:length(linear$coefficients)){
    coef[i,j] <- cvlm[[i]]$coefficients[j]
  }
#msecv[i]
}
```

```r
#output mean of MSE
mean(msecv)
```

```
## [1] 0.0666215
```

The chance of being admitted to univeristy is +/- 6.66%.

Variable Selection for Research

```r
linear <- lm(Research~ Serial.No. + GRE.Score + TOEFL.Score + University.Rating + SOP +LOR + CGPA, data
#summary(linear)

#Remove LOR
linear <- lm(Research~ Serial.No. + GRE.Score + TOEFL.Score + University.Rating  +LOR + CGPA, data = adm
summary(linear)
```

```
##
## Call:
## lm(formula = Research ~ Serial.No. + GRE.Score + TOEFL.Score +
##     University.Rating + LOR + CGPA, data = admissionsData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0861 -0.3358  0.0128  0.2852  0.9840
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -6.3639911  0.6526388  -9.751  < 2e-16 ***
## Serial.No.         0.0001593  0.0001284   1.240    0.215
## GRE.Score          0.0217245  0.0032763   6.631 8.79e-11 ***
## TOEFL.Score       -0.0051749  0.0059488  -0.870    0.385
## University.Rating  0.0365662  0.0240158   1.523    0.129
## LOR                0.0361827  0.0268979   1.345    0.179
## CGPA               0.0377370  0.0650001   0.581    0.562
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4084 on 493 degrees of freedom
## Multiple R-squared:  0.3325, Adjusted R-squared:  0.3243
## F-statistic: 40.92 on 6 and 493 DF,  p-value: < 2.2e-16
```

```r
#Remove CGPA
linear <- lm(Research~ Serial.No. + GRE.Score + TOEFL.Score + University.Rating  +LOR, data = admissions
summary(linear)
```

```
##
## Call:
## lm(formula = Research ~ Serial.No. + GRE.Score + TOEFL.Score +
##     University.Rating + LOR, data = admissionsData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -1.0821 -0.3360  0.0127  0.2866  0.9834
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -6.4346661  0.6407546 -10.042  < 2e-16 ***
## Serial.No.         0.0001623  0.0001283   1.265   0.2064
## GRE.Score          0.0225289  0.0029669   7.593 1.57e-13 ***
## TOEFL.Score       -0.0041137  0.0056572  -0.727   0.4675
## University.Rating  0.0398047  0.0233433   1.705   0.0888 .
## LOR                0.0405427  0.0258109   1.571   0.1169
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4082 on 494 degrees of freedom
## Multiple R-squared:  0.332,  Adjusted R-squared:  0.3253
## F-statistic: 49.11 on 5 and 494 DF,  p-value: < 2.2e-16
```

```
#Remove LOR
linear <- lm(Research~ Serial.No. + GRE.Score + TOEFL.Score + University.Rating, data = admissionsData )
summary(linear)
```

```
##
## Call:
## lm(formula = Research ~ Serial.No. + GRE.Score + TOEFL.Score +
##     University.Rating, data = admissionsData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1057 -0.3428  0.0090  0.2871  1.0214
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -6.5778405  0.6351775 -10.356  < 2e-16 ***
## Serial.No.         0.0001785  0.0001280   1.394   0.1638
## GRE.Score          0.0228912  0.0029623   7.727 6.16e-14 ***
## TOEFL.Score       -0.0029714  0.0056186  -0.529   0.5971
## University.Rating  0.0536923  0.0216362   2.482   0.0134 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4088 on 495 degrees of freedom
## Multiple R-squared:  0.3287, Adjusted R-squared:  0.3233
## F-statistic: 60.59 on 4 and 495 DF,  p-value: < 2.2e-16
```

```
#Remove TOEFL
linear <- lm(Research~ Serial.No. + GRE.Score + University.Rating, data = admissionsData )
summary(linear)
```

```
##
## Call:
## lm(formula = Research ~ Serial.No. + GRE.Score + University.Rating,
##     data = admissionsData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.10835 -0.34957  0.00049  0.28952  1.02269
```

```
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -6.5389338  0.6304444 -10.372   <2e-16 ***
## Serial.No.         0.0001855  0.0001272   1.458   0.1455
## GRE.Score          0.0217887  0.0021030  10.361   <2e-16 ***
## University.Rating  0.0504027  0.0207077   2.434   0.0153 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4085 on 496 degrees of freedom
## Multiple R-squared:  0.3283, Adjusted R-squared:  0.3242
## F-statistic: 80.81 on 3 and 496 DF,  p-value: < 2.2e-16
```

```r
#Remove Serial Number
linear <- lm(Research~ + GRE.Score + University.Rating, data = admissionsData )
summary(linear)
```

```
##
## Call:
## lm(formula = Research ~ +GRE.Score + University.Rating, data = admissionsData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.14033 -0.35017  0.00906  0.29255  1.00181
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -6.415603   0.625451 -10.258   <2e-16 ***
## GRE.Score          0.021546   0.002099  10.266   <2e-16 ***
## University.Rating  0.050337   0.020731   2.428   0.0155 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4089 on 497 degrees of freedom
## Multiple R-squared:  0.3254, Adjusted R-squared:  0.3227
## F-statistic: 119.9 on 2 and 497 DF,  p-value: < 2.2e-16
```

CV for linear model - Research

```r
set.seed(7861)

cvlm <- list()
msecv <- NA
for(i in 1:nrow(admissionsData)){
  #Fit the linear model
cvlm[[i]] <- lm(Research[-i] ~ GRE.Score[-i] + University.Rating[-i])
# Calculate MSE for ith model
msecv[i] <- (predict(cvlm[[i]], newdata = data.frame(GRE.Score[-i] + University.Rating[-i]))-Research[i
#msecv[i]
}
#output mean of MSE
mean(msecv)
```

```
## [1] NA
```

Variable Selection for University Ranking

```r
linear <- lm(University.Rating~ Serial.No. + GRE.Score + TOEFL.Score + SOP +LOR + CGPA + Research, data
summary(linear)
```

```
##
## Call:
## lm(formula = University.Rating ~ Serial.No. + GRE.Score + TOEFL.Score +
##     SOP + LOR + CGPA + Research, data = admissionsData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.34352 -0.46556 -0.03557  0.44046  2.44809
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.4170319  1.2125399  -5.292 1.82e-07 ***
## Serial.No.   0.0001812  0.0002260   0.802  0.42307
## GRE.Score    0.0065910  0.0059476   1.108  0.26833
## TOEFL.Score  0.0209679  0.0103520   2.025  0.04336 *
## SOP          0.4474027  0.0507642   8.813  < 2e-16 ***
## LOR          0.1498125  0.0488318   3.068  0.00227 **
## CGPA         0.3578395  0.1141124   3.136  0.00182 **
## Research     0.0923371  0.0783086   1.179  0.23891
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7109 on 492 degrees of freedom
## Multiple R-squared:  0.619,  Adjusted R-squared:  0.6135
## F-statistic: 114.2 on 7 and 492 DF,  p-value: < 2.2e-16
```

```r
#Remove Serial Number
linear <- lm(University.Rating~  GRE.Score + TOEFL.Score + SOP +LOR + CGPA + Research, data = admissions
summary(linear)
```

```
##
## Call:
## lm(formula = University.Rating ~ GRE.Score + TOEFL.Score + SOP +
##     LOR + CGPA + Research, data = admissionsData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.36251 -0.47140 -0.04223  0.45376  2.41297
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.295220   1.202548  -5.235 2.45e-07 ***
## GRE.Score    0.006468   0.005943   1.088  0.27705
## TOEFL.Score  0.020128   0.010295   1.955  0.05114 .
## SOP          0.441757   0.050255   8.790  < 2e-16 ***
## LOR          0.154072   0.048524   3.175  0.00159 **
```

```
## CGPA         0.364222   0.113793   3.201  0.00146 **
## Research     0.096184   0.078133   1.231  0.21890
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7106 on 493 degrees of freedom
## Multiple R-squared:  0.6185, Adjusted R-squared:  0.6138
## F-statistic: 133.2 on 6 and 493 DF,  p-value: < 2.2e-16
```

```r
#Remove GRE
linear <- lm(University.Rating~   TOEFL.Score + SOP +LOR + CGPA + Research, data = admissionsData )
summary(linear)
```

```
##
## Call:
## lm(formula = University.Rating ~ TOEFL.Score + SOP + LOR + CGPA +
##     Research, data = admissionsData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.37560 -0.47448 -0.03629  0.45065  2.41676
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.243653   0.715856  -7.325 9.79e-13 ***
## TOEFL.Score  0.025353   0.009109   2.783  0.00559 **
## SOP          0.440906   0.050259   8.773  < 2e-16 ***
## LOR          0.151540   0.048478   3.126  0.00188 **
## CGPA         0.414718   0.103920   3.991 7.59e-05 ***
## Research     0.120784   0.074805   1.615  0.10702
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7107 on 494 degrees of freedom
## Multiple R-squared:  0.6176, Adjusted R-squared:  0.6137
## F-statistic: 159.5 on 5 and 494 DF,  p-value: < 2.2e-16
```

```r
#Remove Research
linear <- lm(University.Rating~   TOEFL.Score + SOP +LOR + CGPA, data = admissionsData )
summary(linear)
```

```
##
## Call:
## lm(formula = University.Rating ~ TOEFL.Score + SOP + LOR + CGPA,
##     data = admissionsData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.46231 -0.46269 -0.04935  0.45262  2.39211
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.62010    0.67792  -8.290 1.07e-15 ***
## TOEFL.Score  0.02695    0.00907   2.971  0.00311 **
## SOP          0.44423    0.05030   8.832  < 2e-16 ***
## LOR          0.15563    0.04849   3.210  0.00142 **
```

```
## CGPA         0.44360    0.10254   4.326 1.83e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7119 on 495 degrees of freedom
## Multiple R-squared:  0.6155, Adjusted R-squared:  0.6124
## F-statistic: 198.1 on 4 and 495 DF,  p-value: < 2.2e-16
```

CV for linear model - University Rating

```
set.seed(7861)

cvlm <- list()
msecv <- NA
for(i in 1:nrow(admissionsData)){
  #Fit the linear model
cvlm[[i]] <- lm(University.Rating[-i] ~ TOEFL.Score[-i] + SOP[-i] + LOR[-1] + CGPA[-i])
# Calculate MSE for ith model
msecv[i] <- (predict(cvlm[[i]], newdata = data.frame(TOEFL.Score[-i] + SOP[-i] + LOR[-1] + CGPA[-i]))-Un
#msecv[i]
}
#output mean of MSE
mean(msecv)
```

```
## [1] NA
```

Trees and Forests

```
admissionsData <- admissionsData[,-1]
#head(admissionsData)
dim(admissionsData)
```

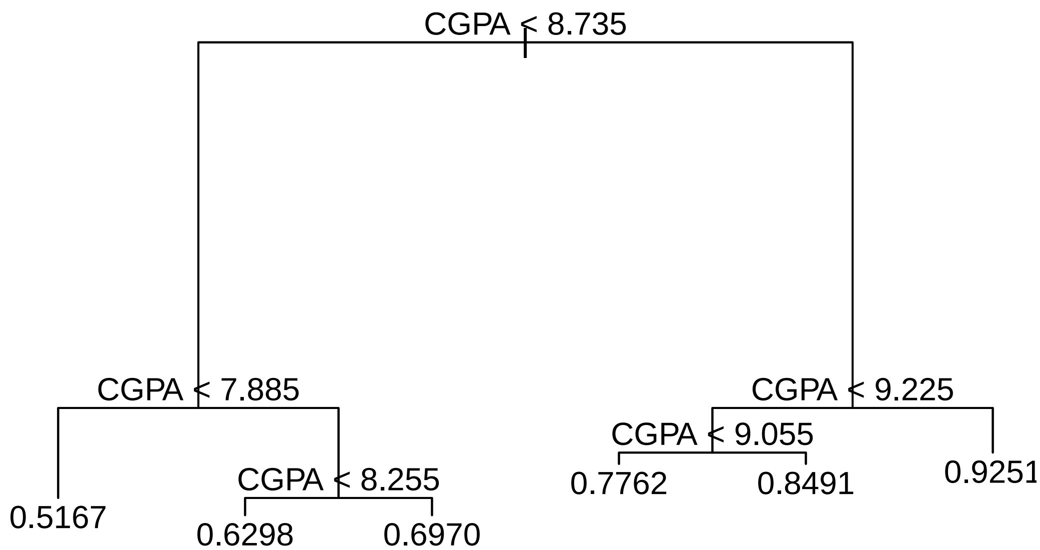
```
## [1] 500   8
```

```
trainindex <- sample(1:nrow(admissionsData), 350)
admissionsTrain <- admissionsData[trainindex, ]
admissionsTest <- admissionsData[-trainindex, ]
```

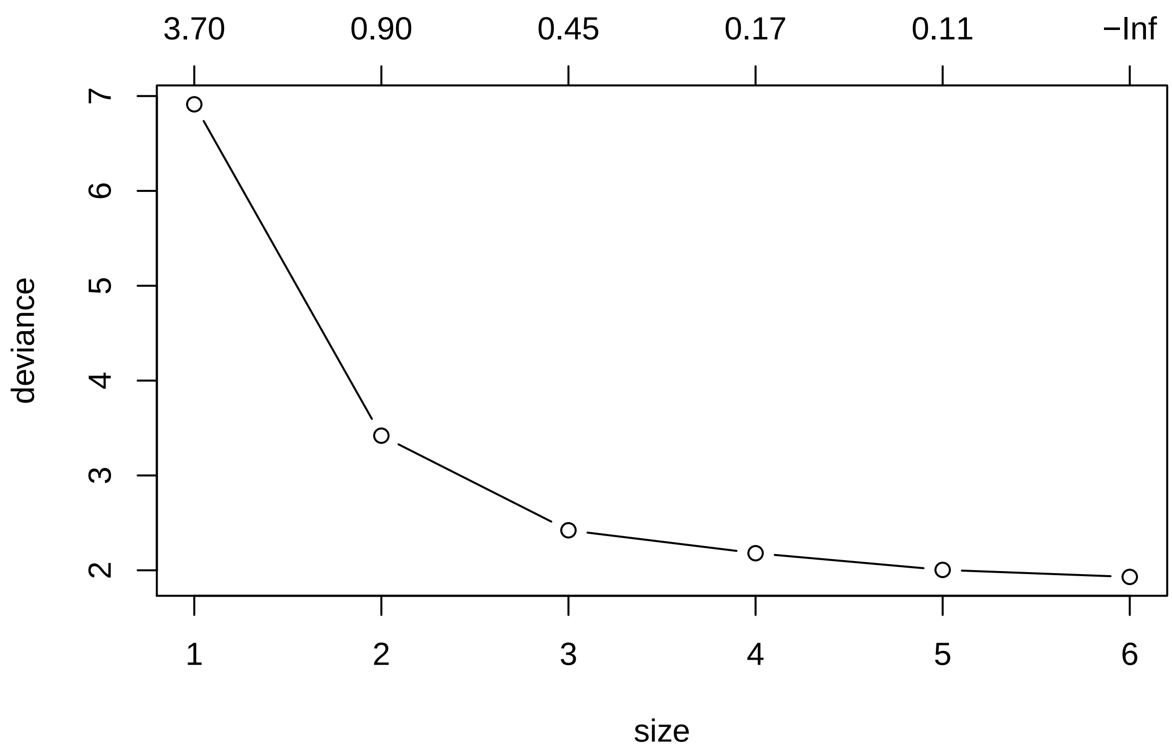
Chance of Admittance

I am going to do a 70/30 split of training and testing data. There are 500 observations, so we will have 350 training observations and 150 testing points.

```
set.seed(110101010)
admissionTree <- tree(Chance.of.Admit~., data = admissionsTrain)
plot(admissionTree)
text(admissionTree, pretty=0)
```




```r
admissionTreeCV <- cv.tree(admissionTree, FUN = prune.tree, K = 10)
plot(admissionTreeCV, type = "b")
```




```r
admissionTreeCV
```

```
## $size
## [1] 6 5 4 3 2 1
##
## $dev
## [1] 1.930357 2.004156 2.180121 2.421745 3.419459 6.912745
##
## $k
## [1]      -Inf 0.1138598 0.1737316 0.4487043 0.9046623 3.6750542
##
```

```
## $method
## [1] "deviance"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

```
admissionTreeCV$dev
```

```
## [1] 1.930357 2.004156 2.180121 2.421745 3.419459 6.912745
```

```
admissionTreeCV$size
```

```
## [1] 6 5 4 3 2 1
```

```
which.min(admissionTreeCV$dev)
```

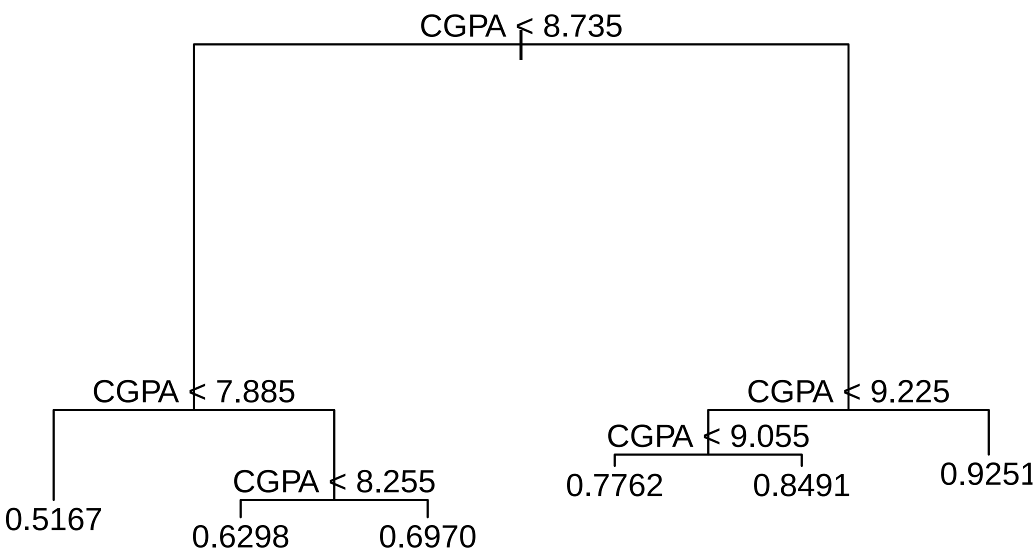
```
## [1] 1
```

Cross validation suggest 7 nodes would be best, so we will prune the tree using 7 terminal nodes.

```
pruneAdmissionTreeCV <- prune.tree(admissionTree, best=7)
```

```
## Warning in prune.tree(admissionTree, best = 7): best is bigger than tree
## size
```

```
plot(pruneAdmissionTreeCV)
text(pruneAdmissionTreeCV, pretty = 0)
```




```
summary(pruneAdmissionTreeCV)
```

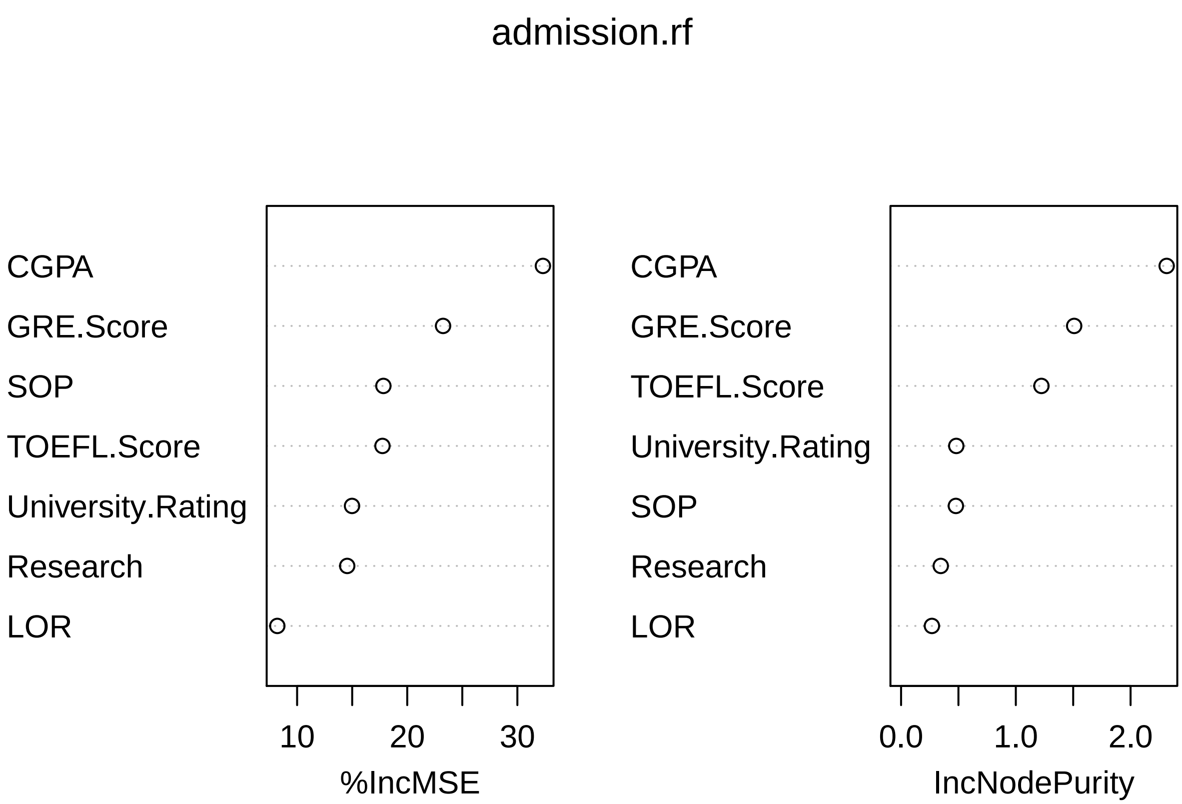
```
##
## Regression tree:
## tree(formula = Chance.of.Admit ~ ., data = admissionsTrain)
## Variables actually used in tree construction:
## [1] "CGPA"
## Number of terminal nodes:  6
## Residual mean deviance:  0.004512 = 1.552 / 344
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.28980 -0.02914  0.01016  0.00000  0.03884  0.18330
```

```r
set.seed(1000101010)
admission.rf <- randomForest(Chance.of.Admit~., data = admissionsTrain, importance = TRUE)
admission.rf
```

```
##
## Call:
##  randomForest(formula = Chance.of.Admit ~ ., data = admissionsTrain,      importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 0.0041354
##                     % Var explained: 78.93
```

Since Random Forest uses out-of-bag which is similar to cross validation so no cross validation was performed. We can look at the importance of the variables.

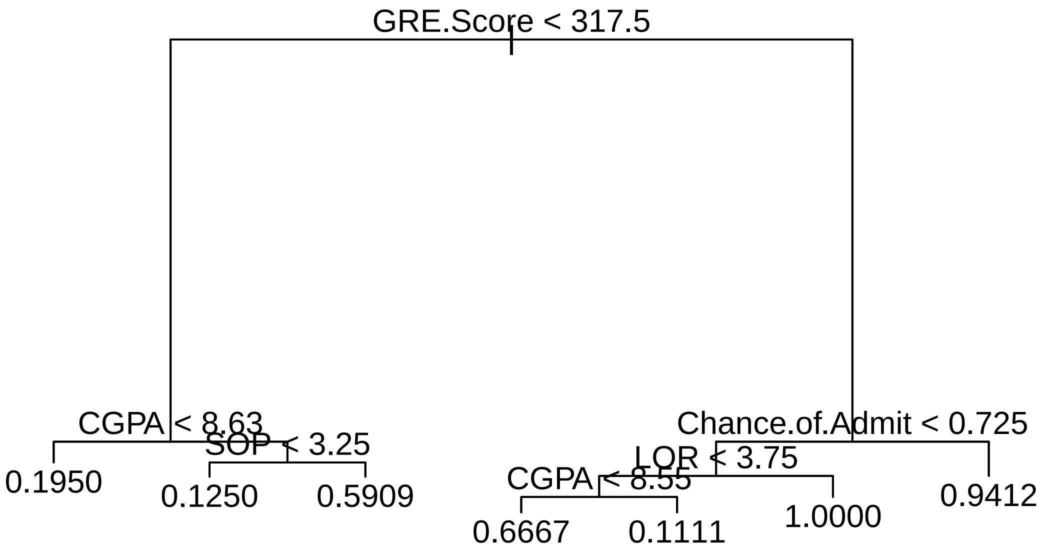
```r
varImpPlot(admission.rf)
```



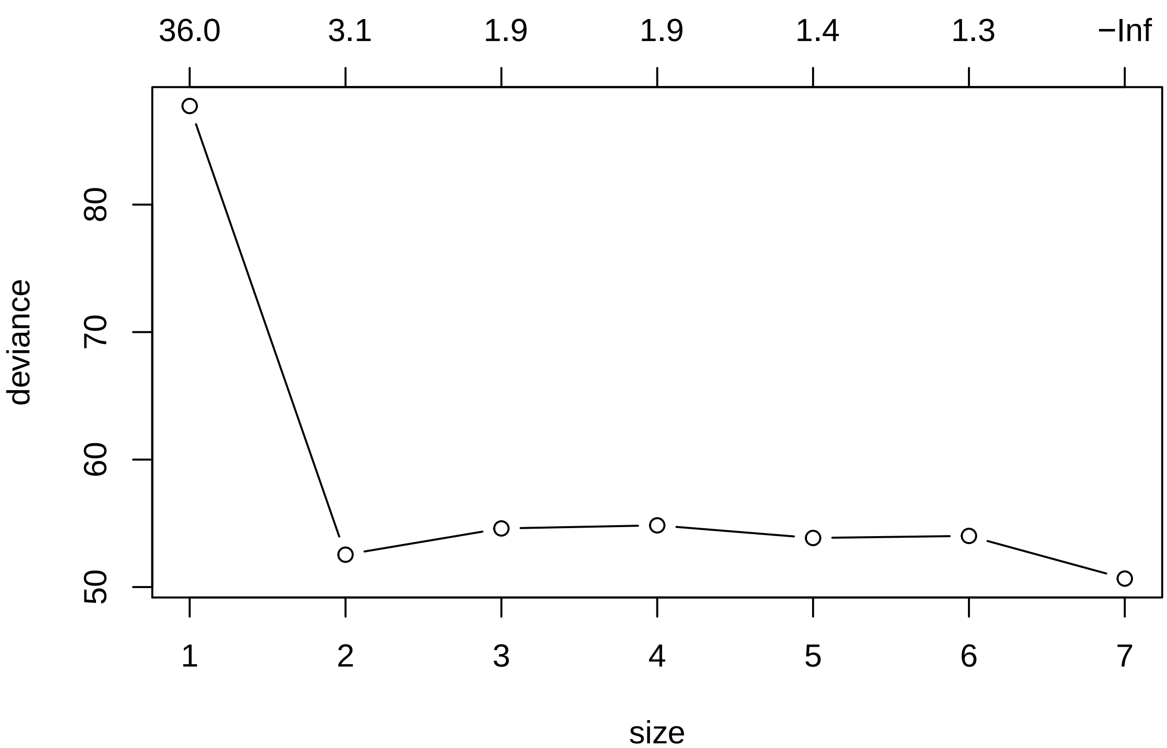
As seen from the Importance Plot the most important variables are CGPA, GRE Score and TOEFL scores when using chance of admission as a response variable.

Research

```r
set.seed(1388582293)
researchTree <- tree(Research~., data = admissionsTrain)
plot(researchTree)
text(researchTree, pretty=0)
```




```r
researchTreeCV <- cv.tree(researchTree, FUN = prune.tree, K = 10)
plot(researchTreeCV, type = "b")
```




```r
which.min(researchTreeCV$dev)
```

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

```r
researchTreeCV$dev
```

```
## [1] 50.66486 54.01171 53.85052 54.83924 54.59803 52.53946 87.73502
```

```r
researchTreeCV$dev
```

```
## [1] 50.66486 54.01171 53.85052 54.83924 54.59803 52.53946 87.73502
```

```r
researchTreeCV$size
```

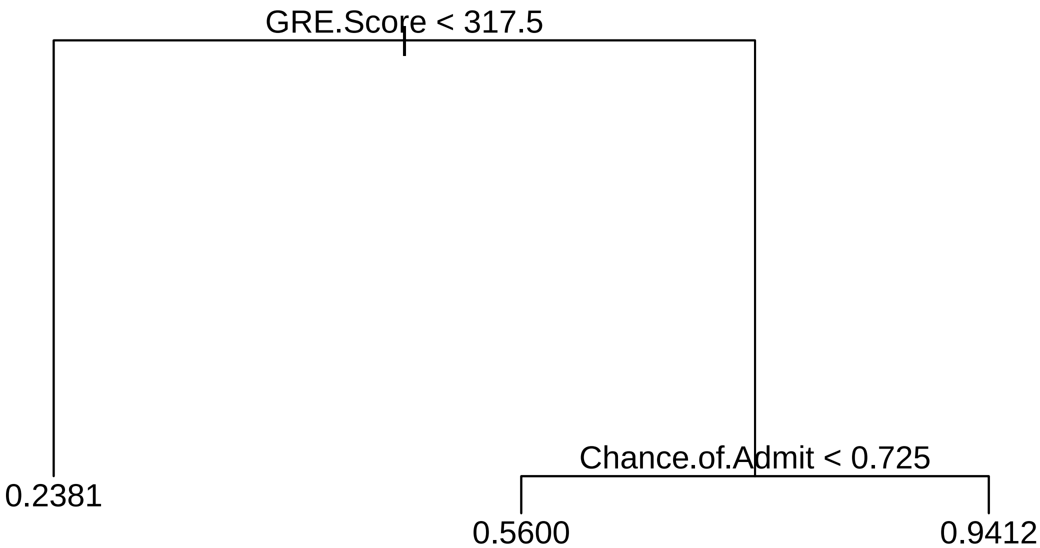
```
## [1] 7 6 5 4 3 2 1
```

```r
which.min(researchTreeCV$dev)
```

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

Cross Validation Suggests 3 terminal nodes would be best.

```r
pruneResearchTreeCV <- prune.tree(researchTree, best=3)
plot(pruneResearchTreeCV)
text(pruneResearchTreeCV, pretty = 0)
```




```r
summary(pruneResearchTreeCV)
```

```
##
## Regression tree:
## snip.tree(tree = researchTree, nodes = c(2L, 6L))
## Variables actually used in tree construction:
## [1] "GRE.Score"       "Chance.of.Admit"
## Number of terminal nodes:  3
## Residual mean deviance:  0.1383 = 47.98 / 347
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.94120 -0.23810  0.05882  0.00000  0.05882  0.76190
```

```r
set.seed(1413755523)
research.rf <- randomForest(Research~., data = admissionsTrain, importance = TRUE)
```

```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values. Are you sure you want to do regression?
```

```r
research.rf
```

```
##
## Call:
##  randomForest(formula = Research ~ ., data = admissionsTrain,      importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
```

```
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 0.144092
##                     % Var explained: 42.09
```

```r
varImpPlot(research.rf)
```