SDM_Assignment2_3

Sri Balaji Muruganandam

04/10/2021

Setting Working Directory

```
rm(list = ls())
setwd("G:\\SDM_Sem01\\Assignment2")
```

Importing necessary libraries

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.1
```

```
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 4.1.1
```

```
library(caret)  #Used to perform cross validations
```

```
## Warning: package 'caret' was built under R version 4.1.1
```

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

```
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 4.1.1
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-2
```

```
data(College)
```

(a) Split the data set into a training set and a test set. Fit a linear model using least squares on the training set and report the test error obtained.

```
co_data <-College
names(co_data)
```

```
##  [1] "Private"     "Apps"        "Accept"      "Enroll"      "Top10perc"
##  [6] "Top25perc"   "F.Undergrad" "P.Undergrad" "Outstate"    "Room.Board"
## [11] "Books"       "Personal"    "PhD"         "Terminal"    "S.F.Ratio"
## [16] "perc.alumni" "Expend"      "Grad.Rate"
```

```
head(co_data)
```

```
##                              Private Apps Accept Enroll Top10perc Top25perc
## Abilene Christian University     Yes 1660   1232    721        23        52
## Adelphi University               Yes 2186   1924    512        16        29
## Adrian College                   Yes 1428   1097    336        22        50
## Agnes Scott College              Yes  417    349    137        60        89
## Alaska Pacific University        Yes  193    146     55        16        44
## Albertson College                Yes  587    479    158        38        62
##                              F.Undergrad P.Undergrad Outstate Room.Board Books
## Abilene Christian University        2885         537     7440       3300   450
## Adelphi University                  2683        1227    12280       6450   750
## Adrian College                      1036          99    11250       3750   400
## Agnes Scott College                  510          63    12960       5450   450
## Alaska Pacific University            249         869     7560       4120   800
## Albertson College                    678          41    13500       3335   500
##                              Personal PhD Terminal S.F.Ratio perc.alumni Expend
## Abilene Christian University     2200  70       78      18.1          12   7041
## Adelphi University               1500  29       30      12.2          16  10527
## Adrian College                   1165  53       66      12.9          30   8735
## Agnes Scott College               875  92       97       7.7          37  19016
## Alaska Pacific University        1500  76       72      11.9           2  10922
## Albertson College                 675  67       73       9.4          11   9727
##                              Grad.Rate
## Abilene Christian University        60
## Adelphi University                  56
## Adrian College                      54
## Agnes Scott College                 59
## Alaska Pacific University           15
## Albertson College                   55
```

```
dim(co_data)
```

```
## [1] 777  18
```

Eliminating Null values

```
head(na.omit(co_data),3)
```

```
##                              Private Apps Accept Enroll Top10perc Top25perc
## Abilene Christian University     Yes 1660   1232    721        23        52
## Adelphi University               Yes 2186   1924    512        16        29
## Adrian College                   Yes 1428   1097    336        22        50
##                              F.Undergrad P.Undergrad Outstate Room.Board Books
## Abilene Christian University        2885         537     7440       3300   450
## Adelphi University                  2683        1227    12280       6450   750
## Adrian College                      1036          99    11250       3750   400
##                              Personal PhD Terminal S.F.Ratio perc.alumni Expend
## Abilene Christian University     2200  70       78      18.1          12   7041
## Adelphi University               1500  29       30      12.2          16  10527
## Adrian College                   1165  53       66      12.9          30   8735
##                              Grad.Rate
## Abilene Christian University        60
## Adelphi University                  56
## Adrian College                      54
```

Converting Categorical variable to numerical value in the Private Column

```
co_data$Private <- as.numeric(as.factor(co_data$Private)) - 1
head(co_data$Private)
```

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

Transforming Rating to the range of 1 to 10

```
co_data$Grad.Rate <- co_data$Grad.Rate/100
head(co_data$Grad.Rate)
```

```
## [1] 0.60 0.56 0.54 0.59 0.15 0.55
```

Splitting Training and the Testing Data

```
set.seed(23)
cross_val_data = createDataPartition(co_data$Apps, p = 0.75, list = FALSE)
training_data  <- co_data[cross_val_data, ]
testing_data <- co_data[-cross_val_data, ]
dim(training_data)
```

```
## [1] 585  18
```

```
dim(testing_data)
```

```
## [1] 192  18
```

Fitting a Linear Model

```
model_fit <- lm(Apps~., data = training_data)
summary(model_fit)
```

```
##
## Call:
## lm(formula = Apps ~ ., data = training_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4879.3  -425.5   -26.6   294.1  7630.8
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -474.76535  477.94282  -0.993  0.32096
## Private     -453.58007  164.31772  -2.760  0.00596 **
## Accept         1.60190    0.04657  34.397  < 2e-16 ***
## Enroll        -0.97255    0.22943  -4.239 2.62e-05 ***
## Top10perc     49.09779    6.54082   7.506 2.37e-13 ***
## Top25perc    -13.61468    5.30854  -2.565  0.01058 *
## F.Undergrad    0.06598    0.04015   1.644  0.10083
## P.Undergrad    0.05505    0.03963   1.389  0.16535
## Outstate      -0.09081    0.02248  -4.040 6.09e-05 ***
## Room.Board     0.15427    0.06036   2.556  0.01085 *
## Books         -0.04054    0.27181  -0.149  0.88149
## Personal       0.07823    0.07708   1.015  0.31057
## PhD           -8.40495    5.38246  -1.562  0.11895
## Terminal      -1.71118    5.94490  -0.288  0.77357
## S.F.Ratio     12.87083   15.14046   0.850  0.39563
## perc.alumni    3.26004    4.97835   0.655  0.51283
## Expend         0.07218    0.01485   4.860 1.52e-06 ***
## Grad.Rate    660.60195  348.74733   1.894  0.05871 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1075 on 567 degrees of freedom
## Multiple R-squared:  0.9272, Adjusted R-squared:  0.9251
## F-statistic:   425 on 17 and 567 DF,  p-value: < 2.2e-16
```

Predicting the Number of Applications for the test data

```
model_predict <- predict.lm(model_fit, testing_data)
summary(model_predict)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -714.0   762.9  1684.1  3043.9  3796.2 19005.6
```

Calculating the Model Performance Metrics

```
train_y = training_data$Apps
test_y = testing_data$Apps
test_MSE <- mean((test_y - model_predict)^2)
model_r2 = R2(model_predict,test_y)
model_rmse = RMSE(model_predict,test_y)

print(model_r2)
```

```
## [1] 0.9346875
```

```
print(test_MSE)
```

```
## [1] 891547.3
```

```
print(model_rmse)
```

```
## [1] 944.2178
```

R2 value of the linear model is 0.9346 which means the model almost fits 93.45% of the data

The Mean Square Error of the model is 891547.3 and the RMSE is 944.21

(b) Fit a ridge regression model on the training set, with λ chosen by crossvalidation. Report the test error obtained.

```
ridge.mod = glmnet(training_data[,-2], train_y, alpha=0)
names(ridge.mod)
```

```
##  [1] "a0"        "beta"      "df"        "dim"       "lambda"    "dev.ratio"
##  [7] "nulldev"   "npasses"   "jerr"      "offset"    "call"      "nobs"
```

```
#coef(ridge.mod)
dim(coef(ridge.mod))
```

```
## [1]  18 100
```

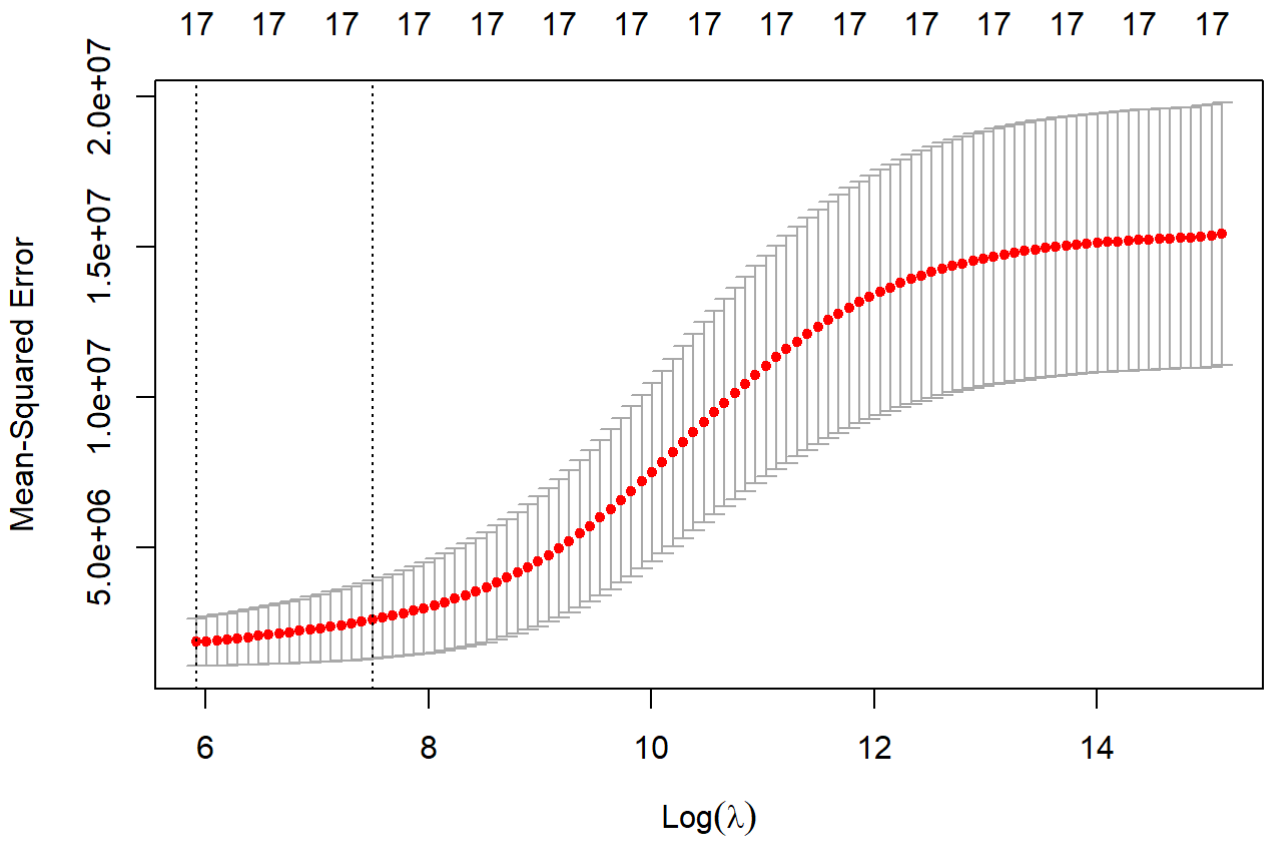
Finding the lambda

```
set.seed(2323)

#cv_train <- sample(1:nrow(co_data), round(nrow(co_data)/2))
#cv.out <- cv.glmnet(co_data[cv_train,-2], data.frame(co_data[cv_train,2]), alpha = 0)
#df_train = data.frame(training_data[,-2])
df_train_y = data.frame(train_y)
print(class(df_train_y))
```

```
## [1] "data.frame"
```

```
cv_out <- cv.glmnet(as.matrix(training_data[,-2]),as.matrix(train_y), alpha=0)
plot(cv_out)
```




```
#head(df_train)
```

Best Lambda for the model

```
best_lambda <- cv_out$lambda.min
best_lambda
```

```
## [1] 370.2356
```

The best lambda for the ridge model is 370.23

Fitting a Ridge Regression Model

```
ridge.pred <- predict(ridge.mod, s= best_lambda, type = "coefficients")

y_test_predict <- predict(ridge.mod, s = best_lambda, newx = as.matrix(testing_data[,-2]), ty
pe = "response")

test_error <- sum((y_test_predict - test_y)^2)
ridge_test_MSE <- mean((y_test_predict - test_y)^2)
```

The test error of the model is 203767671

```
print(test_error)
```

```
## [1] 203767671
```

Similarly, the MSE is 1061290

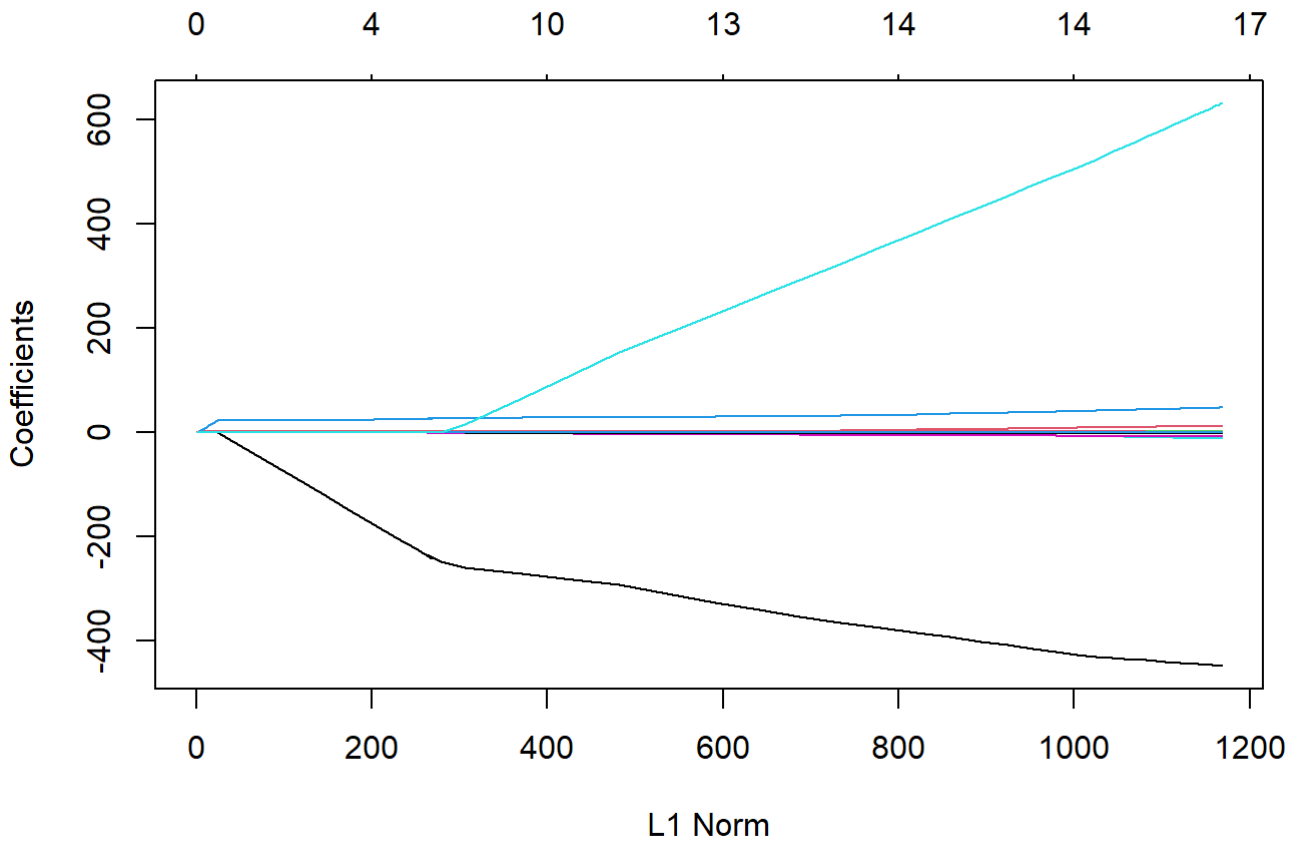
```
print(ridge_test_MSE)
```

```
## [1] 1061290
```

(c) Fit a lasso model on the training set, with λ chosen by crossvalidation.Report the test error obtained, along with the number of non-zero coefficient estimates.

Lasso Model

```
lasso.mod <- glmnet(training_data[,-2], train_y, alpha=1)
plot(lasso.mod)
```




```
cv_out_lasso = cv.glmnet(as.matrix(training_data[,-2]),as.matrix(train_y), alpha=1)
best_lambda_lasso = cv_out_lasso$lambda.min

lasso.pred <- predict(lasso.mod, s = best_lambda_lasso, type = "coefficients")

y_test_predict_lasso <- predict(lasso.mod, s = best_lambda_lasso, newx = as.matrix(testing_da
ta[,-2]), type = "response")

test_error_lasso <- sum((y_test_predict_lasso - test_y)^2)
lasso_test_MSE <- mean((y_test_predict_lasso - test_y)^2)
```

Best Lambda for the model is 15.298

```
print(best_lambda_lasso)
```

```
## [1] 15.29818
```

The test error is 174591500 and the MSE is 909330.7 respectively

```
print(test_error_lasso)
```

```
## [1] 174591500
```

```
print(lasso_test_MSE)
```

```
## [1] 909330.7
```

Finding the number of Non Zero coefficients

```
non_z = lasso.pred[1:length(lasso.mod$beta)]
length(non_z[non_z!=0])
```

```
## [1] 1162
```

The number of non-zero coefficient estimates are 1162

(d) Among those that are not predicted well, do you notice any common trend shared between the colleges?

The Ridge model provides a MSE of 1061290 and the Lasso model provides a MSE of 909330.7. It shows that the Lasso model predicted the number of application better than the Ridge Model.

```
ridge_diff = abs(y_test_predict - test_y)
lasso_diff = abs(y_test_predict_lasso - test_y)
print(ridge_diff[ridge_diff>350])
```

```
##   [1] 1494.7687  453.5889  567.8477 1990.9209 1480.4203  569.6366 2572.9309
##   [8]  775.9175  619.5262  394.2152  703.2545  852.1737  851.5367  981.0892
##  [15]  958.2204  351.0373 1302.5338  373.9607  523.9096  521.5674  350.4825
##  [22]  423.0645  592.8904  691.3723  540.7826  877.8560  385.2432 1318.2201
##  [29]  440.1786  948.5056  747.2405 2236.9435  399.1424  708.1904  751.7303
##  [36]  539.3381  499.4980  624.7035 1155.0824 2022.5300 3307.1250  639.1753
##  [43]  358.8215  367.6745  524.0090  682.8203  597.4585  667.9001  791.7970
##  [50]  470.8909  727.5542  580.6505 1483.8622 1539.1424 1538.1203 1121.3442
##  [57]  948.3353  460.5127  691.0033  360.1408  798.1655 1222.2472 2629.7535
##  [64]  615.6132 7151.6549  530.9652  958.3426  586.4455  911.7925 1771.8378
##  [71]  516.0719  595.4027  856.2067  479.4939  935.6592  465.2664  550.2479
##  [78] 2880.9339 1753.2843  415.8596 2644.3161  633.3825 1582.9271  561.2839
##  [85] 1151.6521 2868.1983  455.0103  967.7865 1995.5872  357.9766 1843.6703
##  [92]  486.5617  402.7951  816.0470  842.0680  703.2340  839.2047 2187.7053
##  [99] 1039.3401  370.8339 2426.8932  386.1905  969.8151  451.7092  589.9862
## [106] 2978.7513
```

```
print(lasso_diff[lasso_diff>350])
```

```
##  [1] 1562.2654  503.0244 1876.9293 1607.8715  964.8883  855.5273  539.2750
##  [8] 1022.5783  910.5198 1167.0387  592.1208  950.2305 1152.7639  437.9169
## [15]  535.7455  427.9640  622.6853  506.4147 1467.4524  612.7747 1062.6919
## [22]  471.2218 1940.0182  362.0244  569.9499  561.0954  547.0122  495.2199
## [29]  545.0770  385.2153 1713.6372 3373.3023  605.8908  430.2846  622.6987
## [36]  427.0580  501.8071  416.0641  400.8577  435.7573  387.1719 1332.0094
## [43] 1346.4274 1749.5386  802.8950  633.2192  779.1762 2316.6022 7545.5423
## [50]  503.1211  764.6311  518.6628 2090.1151  403.7209  495.0767  645.5695
## [57]  483.4113  574.4580  536.9202  644.6303 1743.3620 1470.3429  472.3270
## [64]  532.9285  941.5684  748.5234  547.2771  894.2562 2174.3224 1046.7714
## [71]  951.4883 1362.4355 1488.8460 1509.1296  803.9120  674.8676  770.3542
## [78] 1629.7447 1894.3573  387.3432  836.0416  505.5480 2206.5058  866.9336
## [85]  393.0170  939.5222 3420.6878
```

```
x1= list(which(ridge_diff>350))
x2=list(which(lasso_diff>350))
not_predicted = intersect(x1[[1]], x2[[1]])
print(not_predicted)
```

```
##  [1]   1   5   6  10  14  16  18  23  24  25  27  28  34  35  39  40  44  46  50
## [20]  51  54  55  57  58  59  60  63  64  65  66  68  70  76  86  87  94  95  96
## [39]  97  99 100 104 109 111 116 117 119 120 124 128 129 131 134 135 136 139 140
## [58] 141 143 144 149 150 156 158 159 160 161 162 166 168 169 170 171 173 181 185
## [77] 189 190 191
```

Keeping the threshold of the difference as 350 Applications

Colleges that are not predicted well by the models

```
unpre <- testing_data[not_predicted,]
summary(unpre)
```

```
##     Private            Apps           Accept            Enroll
##  Min.   :0.0000   Min.   :  174   Min.   :  146.0   Min.   :  63.0
##  1st Qu.:0.0000   1st Qu.: 1438   1st Qu.:  933.5   1st Qu.: 348.5
##  Median :1.0000   Median : 2362   Median : 1951.0   Median : 700.0
##  Mean   :0.5949   Mean   : 4564   Mean   : 2972.0   Mean   :1244.3
##  3rd Qu.:1.0000   3rd Qu.: 6176   3rd Qu.: 4035.5   3rd Qu.:1664.0
##  Max.   :1.0000   Max.   :20192   Max.   :13243.0   Max.   :6392.0
##    Top10perc       Top25perc      F.Undergrad     P.Undergrad
##  Min.   : 4.00   Min.   :18.00   Min.   :  494   Min.   :   1.0
##  1st Qu.:18.00   1st Qu.:46.00   1st Qu.: 1266   1st Qu.: 105.5
##  Median :32.00   Median :66.00   Median : 3089   Median : 594.0
##  Mean   :33.24   Mean   :63.05   Mean   : 5986   Mean   :1268.5
##  3rd Qu.:43.50   3rd Qu.:79.50   3rd Qu.: 8345   3rd Qu.:1574.0
##  Max.   :95.00   Max.   :99.00   Max.   :31643   Max.   :9310.0
##     Outstate       Room.Board       Books           Personal         PhD
##  Min.   : 3648   Min.   :2190   Min.   : 300.0   Min.   : 400   Min.   :36.00
##  1st Qu.: 6827   1st Qu.:3522   1st Qu.: 500.0   1st Qu.: 910   1st Qu.:71.50
##  Median : 9766   Median :4200   Median : 525.0   Median :1258   Median :80.00
##  Mean   :10247   Mean   :4292   Mean   : 576.7   Mean   :1527   Mean   :78.23
##  3rd Qu.:13045   3rd Qu.:4822   3rd Qu.: 650.0   3rd Qu.:2000   3rd Qu.:88.50
##  Max.   :19960   Max.   :7000   Max.   :1200.0   Max.   :6800   Max.   :99.00
##     Terminal        S.F.Ratio      perc.alumni        Expend
##  Min.   : 41.00   Min.   : 4.60   Min.   : 4.00   Min.   : 3365
##  1st Qu.: 76.50   1st Qu.:11.35   1st Qu.:13.00   1st Qu.: 7493
##  Median : 87.00   Median :13.70   Median :24.00   Median : 9209
##  Mean   : 83.85   Mean   :14.03   Mean   :25.51   Mean   :11205
##  3rd Qu.: 93.50   3rd Qu.:16.80   3rd Qu.:34.50   3rd Qu.:12842
##  Max.   :100.00   Max.   :28.80   Max.   :60.00   Max.   :42926
##    Grad.Rate
##  Min.   :0.2100
##  1st Qu.:0.5550
##  Median :0.6600
##  Mean   :0.6629
##  3rd Qu.:0.7800
##  Max.   :1.0000
```

```
#head(unpre,50)
print(range(unpre$Grad.Rate))
```

```
## [1] 0.21 1.00
```

1. The number of applications for the colleges that are not predicted well is more than 1000

2. The difference between applications and the number of students accepted is very less

3. Most of the applications are from other states not from the state where the college is located

4. The Grade is very high for the Colleges. Even the 1st Quartile is 0.5550