World University Rankings

Group No.: Group 16

Student Names: Shubham Wadekar & Tiange Yang

Executive Summary:

The goal of the study is to analyze different properties from The Times Higher Education World University Ranking to predict the rankings of the test data. The data contains the first 200 rankings of year 2016 and was processed and cleaned for regression analysis. Virtualizations such as boxplots, scatterplots and co-relation plots were performed to find potential correlations. Regression techniques such as MLR, CART (Regression Tree and Random Forest), Logistic Regression are used. The value of RMSE is mainly factor we consider on accuracy evaluation; residuals Analysis were applied to the models to ensure reliability. We conclude at the end of the study that the MLR model is the best fit for the dataset.

I. Background and Introduction

The significance of university rankings in most cases is to provide a powerful reference for students when make application for universities, and students with different academic backgrounds often pay attention to schools in a specific ranking range. The Times Higher Education World University Ranking is widely regarded as one of the most influential and widely observed university measures. Founded in the United Kingdom in 2010, it has been criticized for its commercialization and for undermining non-English-instructing institutions. Each year, the Times university ranking organizations will conduct a multi-faceted assessment of nearly 1,000 universities in the world. The parameters of the assessment include the learning environment, research influence, income and reputation and so on. Each of these indicators can be regarded as independent variables.

a. The Problem

This project aims to apply different models such as Multiple Linear Regression, Regression Tree and Logistic Ordinal Regression algorithm to extract the training dataset from the entire dataset, find out the category of variables with the highest degree of relevance to the university ranking, built and optimize different models with these variables and then use the test set to evaluate the prediction accuracy of the model, thereby we can offer references when predicting the ranking of each university of future.

b. The Goal of the Study

- To determine which parameter affects the rankings of the university significantly.
- To build a prediction model to determine the university rankings based on the ranking parameters.

c. The Possible Solution

- One of the parameters dominate it’s influence on the rankings of the universities.
- Two or more parameters have combined influence on the quality of the university rankings.
- A predictive model can be built to predict the future rankings of the universities all over the world.

II. Data Exploration and Visualization

- Variables types

The dataset contains two types of variables: categorical and quantitative.
The categorical variables in our data are:

Column	Metadata	Datatype
university_name	name of university	String
country	country of each university	String
international_students	Percentage of students who are international	String
female_male_ratio	Female student to Male student ratio	String

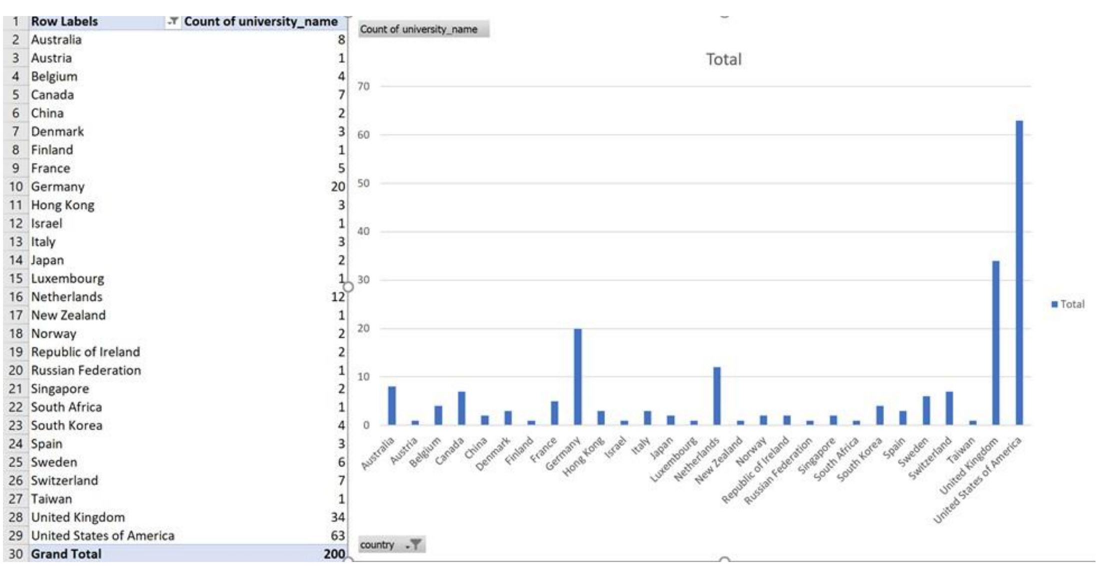
Similarly, the quantitative variables are:

Column	Metadata	Datatype
world_rank	world rank for the university. Contains rank ranges and equal ranks (eg.	Numeric
Teaching	university score for teaching (the learning environment)	Numeric
international	university score international outlook (staff, students, research)	Numeric
research	university score for research (volume, income and reputation)	Numeric
citations	university score for citations (research influence)	Numeric
income	university score for industry income (knowledge transfer)	Numeric
total_score	total score for university, used to determine rank	Numeric
num_students	number of students at the university	Numeric
student_staff_ratio	Number of students divided by number of staff	Numeric

In our analysis with regression, we try to answer the questions regarding the variables influencing the rank of each university in the world. For multiple linear regression, we train the first 150 records for the regression analysis and predict ranks for the remaining 50 records.

- Statistical Analysis

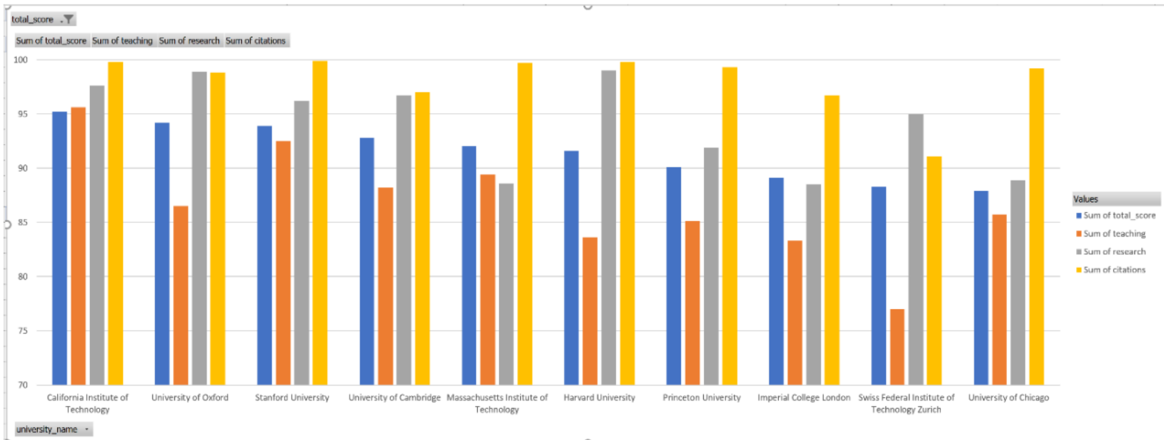
Histograms



	Row Labels	Count of university_name
2	Australia	8
3	Austria	1
4	Belgium	4
5	Canada	7
6	China	2
7	Denmark	3
8	Finland	1
9	France	5
10	Germany	20
11	Hong Kong	3
12	Israel	1
13	Italy	3
14	Japan	2
15	Luxembourg	1
16	Netherlands	12
17	New Zealand	1
18	Norway	2
19	Republic of Ireland	2
20	Russian Federation	1
21	Singapore	2
22	South Africa	1
23	South Korea	4
24	Spain	3
25	Sweden	6
26	Switzerland	7
27	Taiwan	1
28	United Kingdom	34
29	United States of America	63
30	Grand Total	200

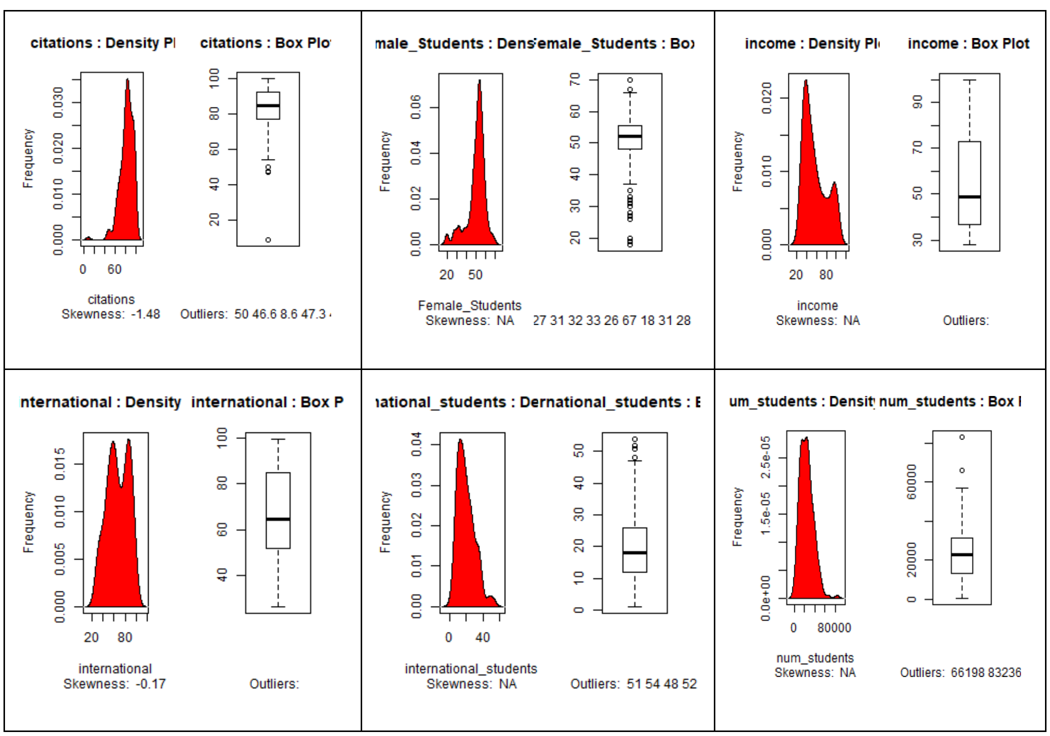
The above graph has been plotted to analyze which of the world countries have the greatest number of universities in the top 200 universities. And we've concluded that United States of America has the highest number of Universities in the top 200

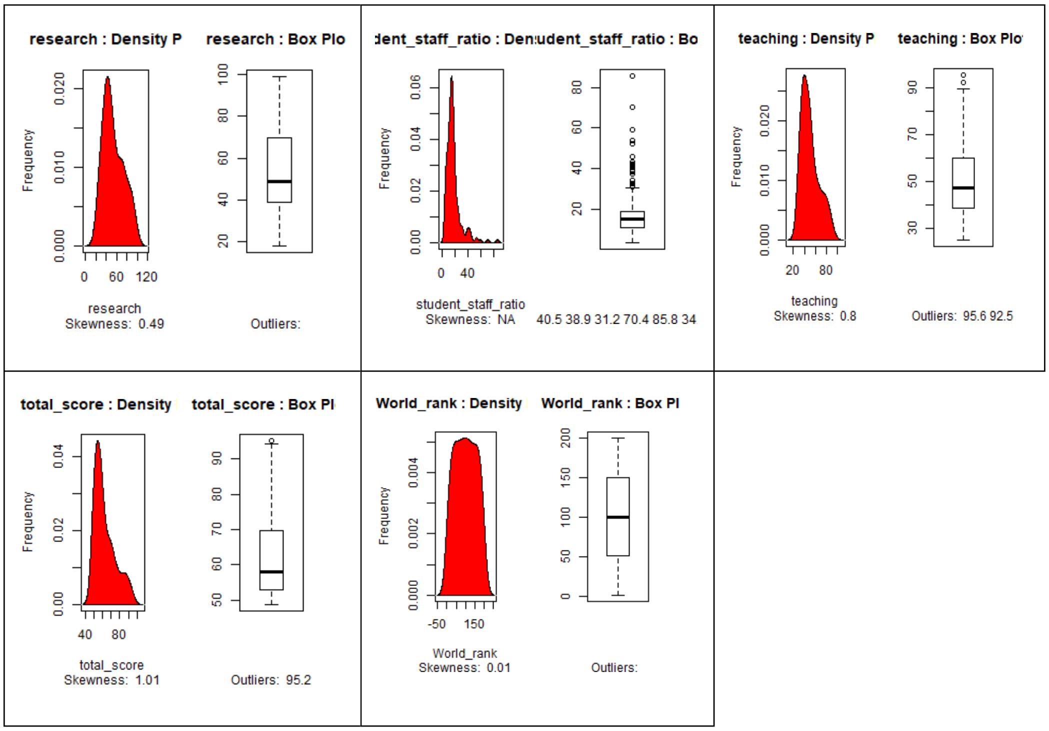
Universities of the world followed by United Kingdom then by Germany and Netherlands.



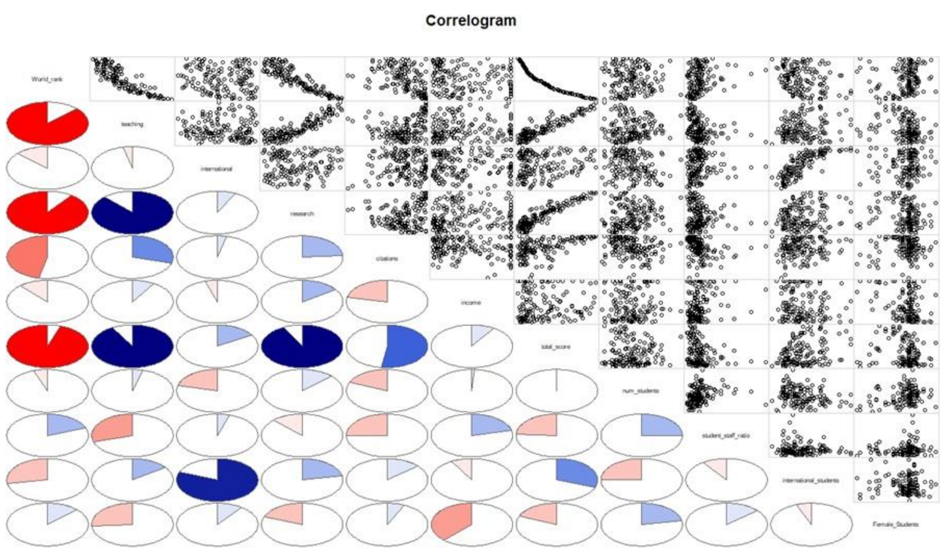
In the above graph, total score, teaching, research, citations have been plotted. If we take top 10 universities and see the top five and below five as two parts, we can see that some values in the below 5 universities have greater values when compared to the top 5 universities. This states that the total score doesn't depend only on a single variable but depends on combination of variables.

Examining Distributions





- Relationships



- **international_students** are highly correlated to **international outlook**.
- The highest correlation is between **teaching** and **research**.
- **World_rank** and **Total_score** is each highly correlated to Teaching, research and citations.

III. Data Preparation and Preprocessing

- Change columns to numeric.

```
## Change Columns to numeric
tm$income = sub('-',' ',tm$income)
tm$income = as.numeric(as.character(tm$income))
tm$num_students = gsub(',','',tm$num_students)
tm$num_students = as.numeric(as.character(tm$num_students))
tm$international_students = as.numeric(as.character(gsub('%','',tm$international_students)))
colnames(tm)[colnames(tm)=="X._Female_Students"] <- "Female_Students"
colnames(tm)[colnames(tm)=="ï..world_rank"] <- "World_rank"
```

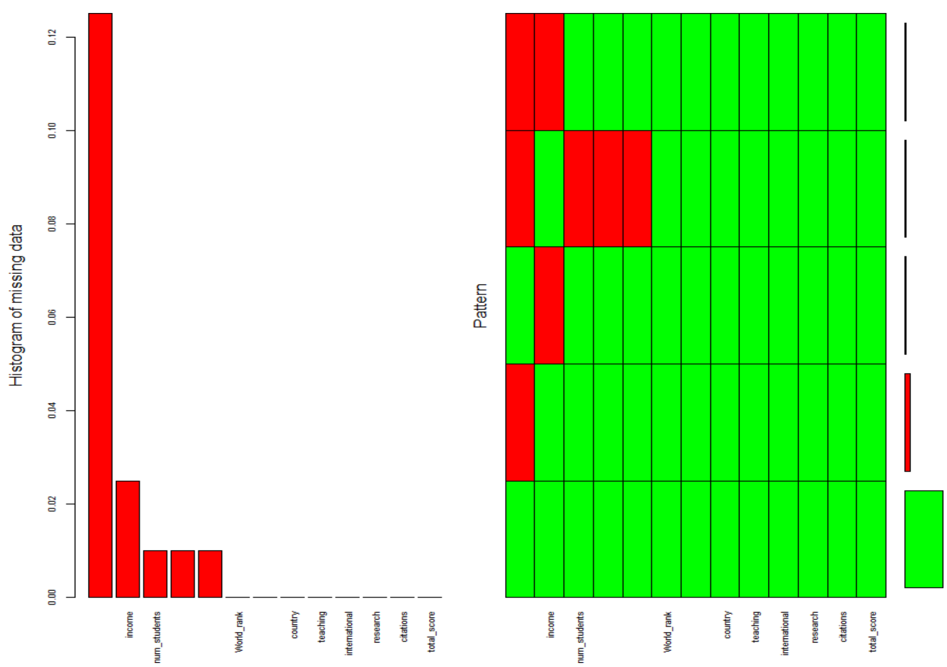
- Missing data Analysis

```
##Missing Data Analysis
pMiss <- function(x){sum(is.na(x))/length(x)*100}
apply(tm,1,pMiss)
apply(tm,2,pMiss)
```

```
> apply(tm,2,pMiss)
         World_rank        university_name                country               teaching
                0.0                    0.0                    0.0                    0.0
      international               research              citations                 income
                0.0                    0.0                    0.0                    2.5
        total_score           num_students  student_staff_ratio international_students
                0.0                    1.0                    1.0                    1.0
    Female_Students
               12.5
```

We can see in the below image that Female_Students missing data is 12.5% which is above 5%. The graph below shows that majority of the values missing are of income which is above considerable level.

```
sapply(tm, function(x) sum(is.na(x)))

aggr_plot <- aggr(tm, col=c('navyblue','red'), numbers=TRUE, sortVars=TRUE, labels=names(tm),
cex.axis=.7, gap=3, ylab=c("Histogram of missing data","Pattern"))
```



- Using mice package to impute missing data,

```
## Data imputation
tempData <- mice(tm,m=5,meth='cart')
sapply(training_set, function(x) sum(is.na(x)))
modelFit2 <- with(tempData, lm(total_score~research+citations+international+income))
summary(modelFit2)
train_complete <- complete(tempData,"long")
training_set = train_complete[(1:150),]
test_set = train_complete[(150:200),]
```

- Outlier Detection

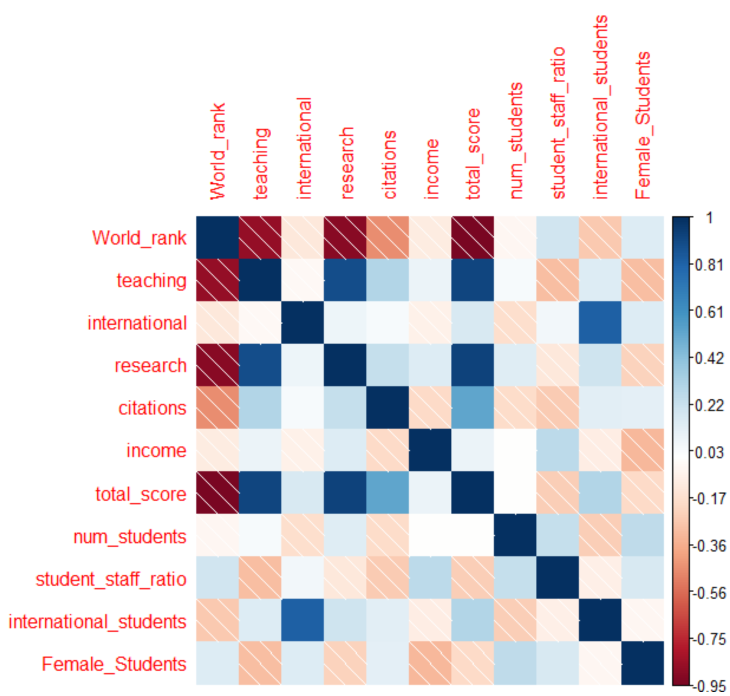
```
## Outlier Detection
outlier(tm[,4:13])
```

```
> outlier(rank_model[,4:13])
           teaching          international               research
               95.6                   26.1                   99.0
          citations                 income            total_score
                8.6                  100.0                   95.2
       num_students    student_staff_ratio international_students
              769.0                   85.8                   54.0
    Female_Students
               18.0
```

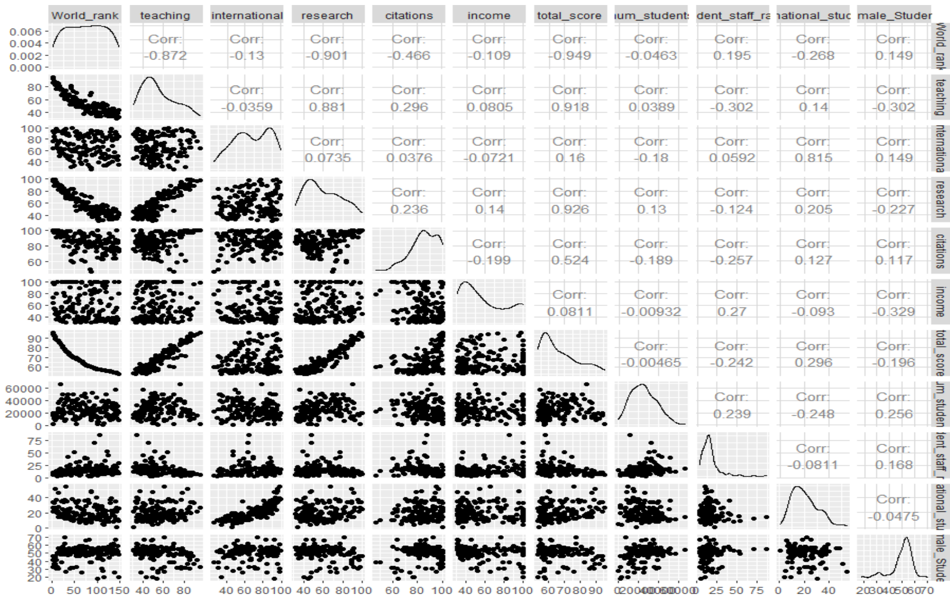
The outliers do not have significant influence over the model. Hence we do not delete the outliers.

- Correlation Plot for the newly imputed dataset

```
q = as.matrix(training_set[,c(3,6:15)])
corrplot(cor(q), method = "shade", number.cex=0.75, is.corr = FALSE)
```




```
par(mfrow=c(2,2))
plot(training_set, col="blue", main="Matrix Scatterplot of all the independent variables")
ggpairs(training_set[,c(3,6:15)])
```



According to the Correlation Plot, World_rank and total_score are highly correlated (0.95), to avoid Multi-collinearity in the process of regression, we consider to do with the total_score which is correlated to the world_rank. Higher the total_score, higher the rank.

IV. Data Mining Techniques and Implementation

1. MLR

- Building different models

```
rank_model = lm(total_score~research+citations+international+teaching+Female_Students
+student_staff_ratio+international_students+num_students+income, data=training_set)
summary(rank_model)
```

```
Call:
lm(formula = total_score ~ research + citations + international +
    teaching + Female_Students + student_staff_ratio + international_students +
    num_students + income, data = training_set)

Residuals:
     Min       1Q   Median       3Q      Max
-1.60794 -0.03450  0.01890  0.06128  0.35747

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)             1.737e-01  1.978e-01   0.878  0.38135
research                3.034e-01  2.175e-03 139.493  < 2e-16 ***
citations               3.003e-01  1.736e-03 173.013  < 2e-16 ***
international           7.706e-02  1.614e-03  47.750  < 2e-16 ***
teaching                2.960e-01  2.611e-03 113.360  < 2e-16 ***
Female_Students        -1.751e-03  2.311e-03  -0.758  0.44981
student_staff_ratio    -5.009e-03  1.764e-03  -2.840  0.00518 **
international_students -2.799e-03  3.098e-03  -0.904  0.36772
num_students           -1.827e-06  1.578e-06  -1.158  0.24891
income                  2.374e-02  8.207e-04  28.925  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2066 on 140 degrees of freedom
Multiple R-squared:  0.9997,    Adjusted R-squared:  0.9997
F-statistic: 5.078e+04 on 9 and 140 DF,  p-value: < 2.2e-16
```

For the first model, the adjusted R-square is 99% which is good but some variables are significant. Hence, we continue to remove the insignificant variables and only include relevant variables and the model is as follows,

```
rank_model = lm(total_score~research+citations+international+Female_Students+student_staff_ra
tio+international_students+num_students+income, data=training_set)
summary(rank_model)
```

```
Call:
lm(formula = total_score ~ research + citations + international +
    Female_Students + student_staff_ratio + international_students +
    num_students + income, data = training_set)

Residuals:
    Min      1Q  Median      3Q     Max
-4.2203 -1.2802 -0.1399  1.2618  6.8384

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)             5.975e+00  1.834e+00   3.258  0.00141 **
research                5.169e-01  1.043e-02  49.542  < 2e-16 ***
citations               3.269e-01  1.651e-02  19.797  < 2e-16 ***
international           5.410e-02  1.537e-02   3.521  0.00058 ***
Female_Students        -4.176e-02  2.192e-02  -1.905  0.05878 .
student_staff_ratio    -6.716e-02  1.609e-02  -4.175  5.2e-05 ***
international_students  6.079e-03  2.973e-02   0.205  0.83825
num_students           -6.691e-06  1.514e-05  -0.442  0.65916
income                  2.104e-02  7.874e-03   2.672  0.00842 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.983 on 141 degrees of freedom
Multiple R-squared:  0.9716,    Adjusted R-squared:   0.97
F-statistic: 602.7 on 8 and 141 DF,  p-value: < 2.2e-16
```

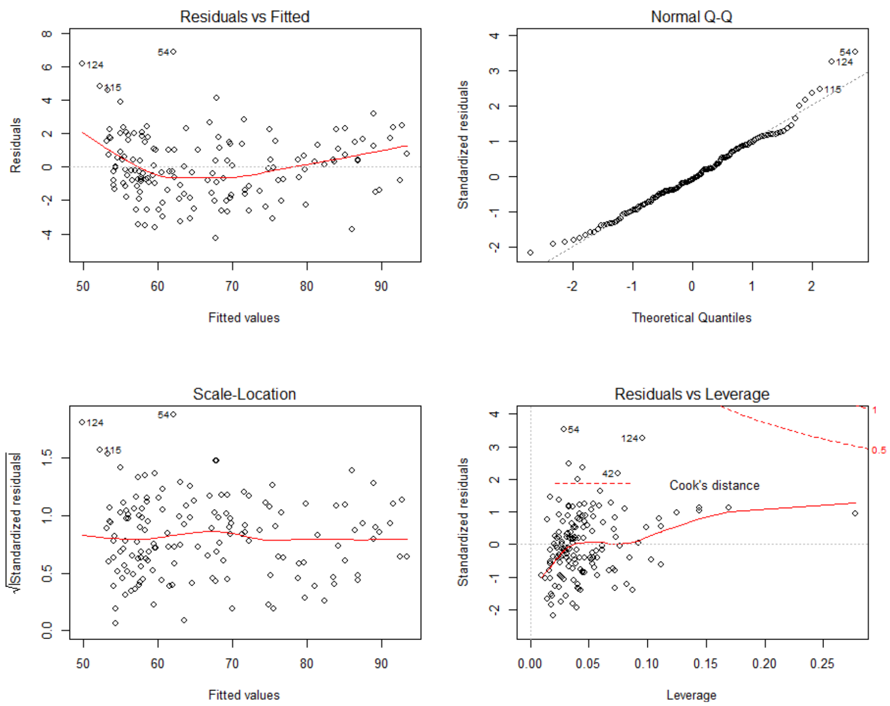
the adjusted R-square is 97% which is good, then we continue to remove the insignificant variables

```
rank_model = lm(total_score~research+citations+international+income+Female_Students+student_s
taff_ratio, data=training_set)
summary(rank_model)
```

```
Call:
lm(formula = total_score ~ research + citations + international +
    income + Female_Students + student_staff_ratio, data = training_set)

Residuals:
    Min      1Q  Median      3Q     Max
-4.2727 -1.2820 -0.1744  1.3390  6.8691

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)          5.794911   1.787891   3.241  0.00148 **
research             0.515745   0.009631  53.549  < 2e-16 ***
citations            0.329194   0.015751  20.900  < 2e-16 ***
international        0.057849   0.008276   6.990 9.68e-11 ***
income               0.021054   0.007743   2.719  0.00735 **
Female_Students     -0.046118   0.019952  -2.311  0.02224 *
student_staff_ratio -0.068810   0.015659  -4.394 2.15e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.971 on 143 degrees of freedom
Multiple R-squared:  0.9715,    Adjusted R-squared:  0.9703
F-statistic: 813.4 on 6 and 143 DF,  p-value: < 2.2e-16
```

The adjusted R-square is 97% which is pretty good and all the variables are significant.

```
par(mfrow=c(2,2))
plot(rank_model)
```



- Predictions for data

```
## Predictions for data
lmpred<-predict (rank_model, test_set)
accuracy(lmpred, test_set$total_score)
```

```
                ME     RMSE      MAE       MPE     MAPE
Test set 0.2227418 1.902467 1.478921 0.4389656 2.902129
```

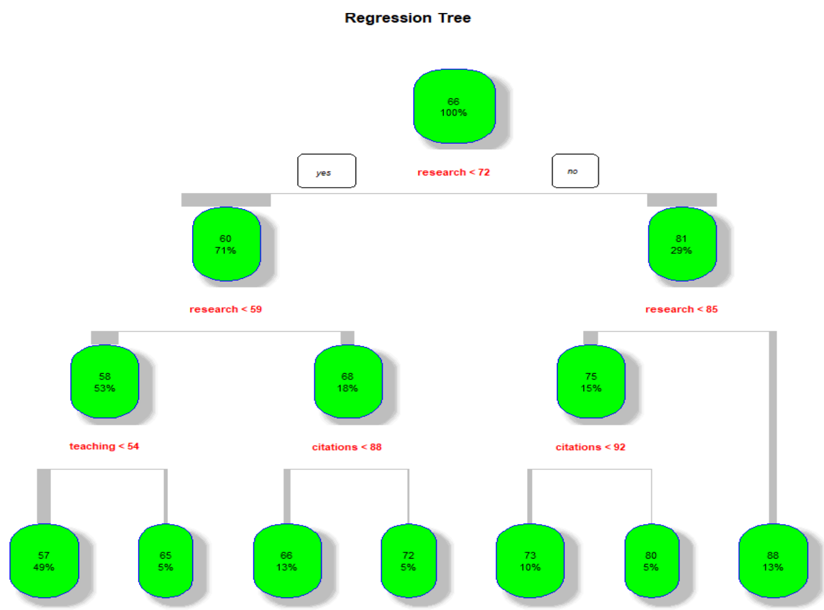
The RMSE is 1.90. Hence, we conclude that our model is significant.

2. CART

a. Regression Tree

- Building model

```
tree1 <- rpart(total_score ~ research+citations+international+teaching+Female_Students
+student_staff_ratio+international_students+num_students+income, data=training_set)

rpart.plot(tree1, branch = 1, branch.type = 1, type = 2,shadow.col='gray', box.col='green'
,border.col='blue', split.col='red',main="Regression Tree")
```



- Predictions for data

```
## Predictions for data
treepred1 <- predict(tree1,test_set)
accuracy(treepred1, test_set$total_score)
                ME     RMSE      MAE       MPE    MAPE
Test set -6.945265 7.177763 6.945265 -13.7303 13.7303
```

The RMSE is 7.17. Hence, we conclude that our model is significant, but not as significant as the MLR model.

b. Random Forest

A random forest is a collection of decision trees, which is a strong modeling technique and much more robust than a single decision tree. They aggregate many decision trees to limit overfitting as well as error due to bias and therefore yield useful results.

- Building model

```
tree2 <- randomForest(total_score ~ research+citations+international+teaching+Female_Students
+student_staff_ratio+international_students+num_students+income, data=training_set)
```

- Predictions for data

```
## Predictions for data
treepred2 <- predict(tree2,test_set)
accuracy(treepred2, test_set$total_score)
                ME     RMSE      MAE       MPE     MAPE
Test set -5.790332 6.212375 5.790332 -11.45747 11.45747
```

Since the randomness of this method, the RMSE is different but basically stable around 6.21. Hence, we conclude that our model is significant and slightly better than regression tree model, but not as significant as the MLR model.

c. Logistic Ordinal Regression

d.

Even though we can predict the rank of a university using the total score, we would like to predict the ranks using the ranks given with the help of ordinal regression.

- We wanted to try with the first 30 ranks and check the regression model, the below code is to order the ranking variable which is originally a numeric variable and needed to be a factor variable in order to perform the ordinal regression.

```
library("ordinal")
library("MASS")
str(tm)
training_set2 = train_complete[(1:30),]
training_set2 = training_set[,c(3,6:15)]
training_set2$World_rank = as.factor (training_set$World_rank)
training_set2$World_rank <- ordered (training_set$World_rank, levels = c(30:1))

model <- polr(World_rank~research+citations+income,training_set2, Hess = TRUE)
summary(model)
```

```
Call:
polr(formula = World_rank ~ research + citations + income, data = training_set2,
    Hess = TRUE)

Coefficients:
            Value Std. Error t value
research  0.57859    0.10773   5.371
citations 0.36725    0.09927   3.699
income    0.02356    0.01596   1.476

Intercepts:
      Value   Std. Error t value
30|29 77.3051 14.9526     5.1700
29|28 78.6617 15.0374     5.2311
28|27 79.4626 15.1067     5.2601
27|26 80.0617 15.1409     5.2878
26|25 80.8535 15.2788     5.2919
25|24 81.6265 15.4294     5.2903
24|23 82.3579 15.5400     5.2997
23|22 83.4164 15.8033     5.2784
22|21 84.2815 15.9950     5.2692
21|20 84.8054 16.0574     5.2814
20|19 85.3605 16.1401     5.2887
19|18 85.9564 16.2519     5.2890
18|17 86.5026 16.3556     5.2889
17|16 86.9927 16.4352     5.2931
16|15 87.3883 16.4783     5.3032
15|14 87.8138 16.5262     5.3136
14|13 88.3189 16.6002     5.3204
13|12 88.7442 16.6496     5.3301
12|11 89.1565 16.6934     5.3408
11|10 89.5706 16.7358     5.3520
10|9  89.9743 16.7699     5.3652
9|8   90.4128 16.8144     5.3771
8|7   90.9604 16.8975     5.3831
7|6   91.7097 17.0417     5.3815
6|5   92.3307 17.1270     5.3909
5|4   92.9384 17.1999     5.4034
4|3   93.7974 17.3386     5.4097
3|2   94.8594 17.5012     5.4202
2|1   96.1400 17.6285     5.4537

Residual Deviance: 141.5232
AIC: 205.5232
(120 observations deleted due to missingness)
```

- The table below gives the coefficients of all independent variables and ranks.

```
(ctable <- cbind(ctable, "p value" = p))
```

```
> (ctable = coef(summary(model)))
                Value  Std. Error   t value
research    0.57858564  0.10773074 5.370664
citations   0.36725051  0.09927195 3.699439
income      0.02355784  0.01596156 1.475911
30|29      77.30513025 14.95260618 5.170010
29|28      78.66169132 15.03738224 5.231076
28|27      79.46259566 15.10671115 5.260086
27|26      80.06165892 15.14093676 5.287761
26|25      80.85346603 15.27877576 5.291881
25|24      81.62648648 15.42940552 5.290320
24|23      82.35787136 15.53999062 5.299738
23|22      83.41635179 15.80328122 5.278420
22|21      84.28145758 15.99499747 5.269239
21|20      84.80537949 16.05738201 5.281395
20|19      85.36047685 16.14006577 5.288732
19|18      85.95644813 16.25194250 5.288995
18|17      86.50255894 16.35557556 5.288873
17|16      86.99271262 16.43520080 5.293073
16|15      87.38833561 16.47832782 5.303228
15|14      87.81379019 16.52622006 5.313604
14|13      88.31886636 16.60015119 5.320365
13|12      88.74421556 16.64964345 5.330097
12|11      89.15653747 16.69344981 5.340810
11|10      89.57061069 16.73580279 5.352036
10|9       89.97426496 16.76987658 5.365231
9|8        90.41276695 16.81438858 5.377107
8|7        90.96043088 16.89747189 5.383079
7|6        91.70967391 17.04165122 5.381502
6|5        92.33072192 17.12700919 5.390943
5|4        92.93844246 17.19988955 5.403433
4|3        93.79742982 17.33859285 5.409749
3|2        94.85941066 17.50116905 5.420176
2|1        96.14000312 17.62850467 5.453668
```

```
p <- pnorm(abs(ctable[, "t value"]), lower.tail = FALSE) * 2
(ctable <- cbind(ctable, "p value" = p))
                Value  Std. Error  t value      p value
research    0.57858564  0.10773074 5.370664 7.844718e-08
citations   0.36725051  0.09927195 3.699439 2.160765e-04
income      0.02355784  0.01596156 1.475911 1.399679e-01
30|29      77.30513025 14.95260618 5.170010 2.340809e-07
29|28      78.66169132 15.03738224 5.231076 1.685261e-07
28|27      79.46259566 15.10671115 5.260086 1.439883e-07
27|26      80.06165892 15.14093676 5.287761 1.238224e-07
26|25      80.85346603 15.27877576 5.291881 1.210645e-07
25|24      81.62648648 15.42940552 5.290320 1.221028e-07
24|23      82.35787136 15.53999062 5.299738 1.159693e-07
23|22      83.41635179 15.80328122 5.278420 1.303027e-07
22|21      84.28145758 15.99499747 5.269239 1.369908e-07
21|20      84.80537949 16.05738201 5.281395 1.282038e-07
20|19      85.36047685 16.14006577 5.288732 1.231674e-07
19|18      85.95644813 16.25194250 5.288995 1.229900e-07
18|17      86.50255894 16.35557556 5.288873 1.230724e-07
17|16      86.99271262 16.43520080 5.293073 1.202781e-07
16|15      87.38833561 16.47832782 5.303228 1.137724e-07
15|14      87.81379019 16.52622006 5.313604 1.074780e-07
14|13      88.31886636 16.60015119 5.320365 1.035592e-07
13|12      88.74421556 16.64964345 5.330097 9.816029e-08
12|11      89.15653747 16.69344981 5.340810 9.253239e-08
11|10      89.57061069 16.73580279 5.352036 8.697033e-08
10|9       89.97426496 16.76987658 5.365231 8.084560e-08
9|8        90.41276695 16.81438858 5.377107 7.569218e-08
8|7        90.96043088 16.89747189 5.383079 7.322227e-08
7|6        91.70967391 17.04165122 5.381502 7.386703e-08
6|5        92.33072192 17.12700919 5.390943 7.008910e-08
5|4        92.93844246 17.19988955 5.403433 6.537756e-08
4|3        93.79742982 17.33859285 5.409749 6.311326e-08
3|2        94.85941066 17.50116905 5.420176 5.954050e-08
2|1        96.14000312 17.62850467 5.453668 4.934142e-08
```

All the p-values seem to be normal and very less indicating that it is a good model.

```
> (ci <- confint(model))
Waiting for profiling to be done...
                 2.5 %    97.5 %
research    0.38674410 0.8121228
citations   0.18565192 0.5810987
income     -0.00767261 0.0556984
>
```

- Predictions for data

```
## Predictions for data
pred <- predict(model, training_set)
print(pred, digits = 3)
```

```
> print(pred, digits = 3)
  [1] 1  2  3  4  7  2  7  14 7  14 7  14 7  14
 22 14 14 18 19 22 19 12 23 26 23 29 29 23 29 30
 30
 [32] 30 30 30 29 26 30 30 30 30 30 30 30 30 30
```

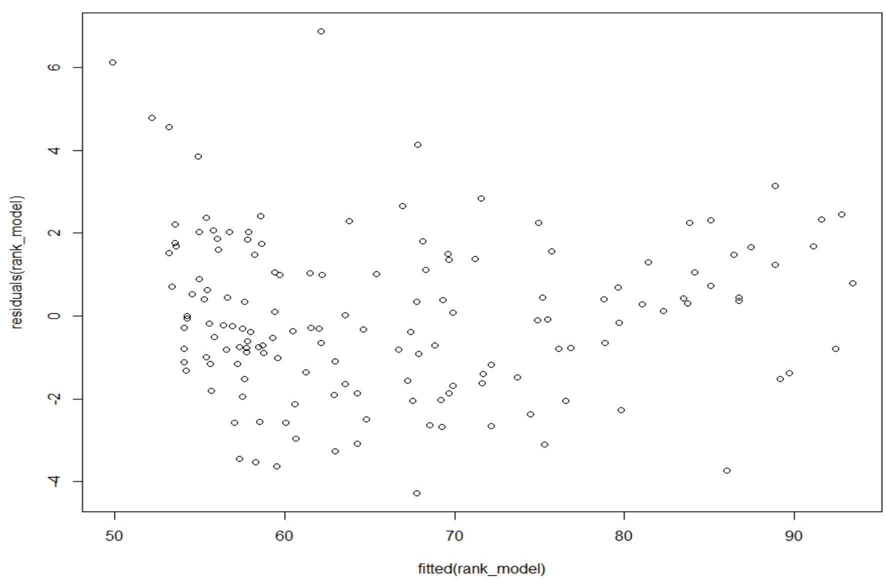
The predictions are almost close to the original first 30 ranks.

V. Performance Evaluation

From the output of 3 regression algorithms we implemented, we can conclude that for the Times University Rankings dataset, the Multiple Linear Regression method has the best performance with the highest accuracy of data prediction. For the further evaluation, we analyze the model from aspects as following,

- Linearity

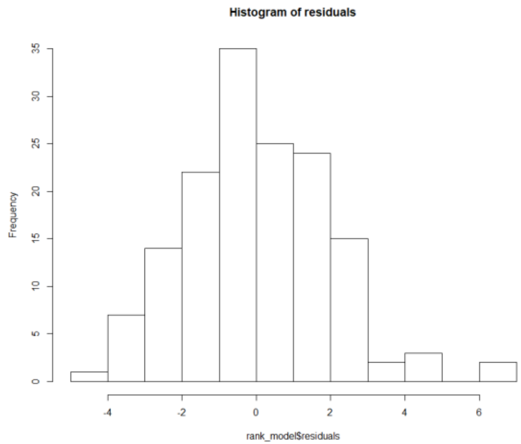
```
## Linearity
plot(fitted(rank_model), residuals(rank_model))
abline(0,1, col="blue", lwd=2)
```

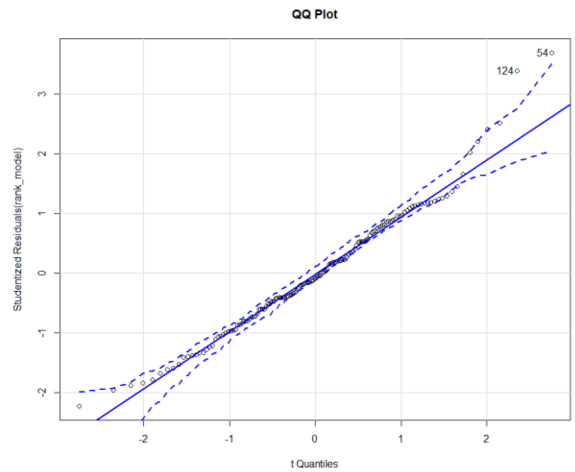


By executing the above commands, we know that our dataset that using the below plot that it is linear, independent and residuals have a constant variance

- Normality

```
##Normality
hist(rank_model$residuals, main ="Histogram of residuals")
qqPlot(rank_model, main="QQ Plot")
```





From the graphs above, we can conclude that the standardized residuals approximately display in a straight line.

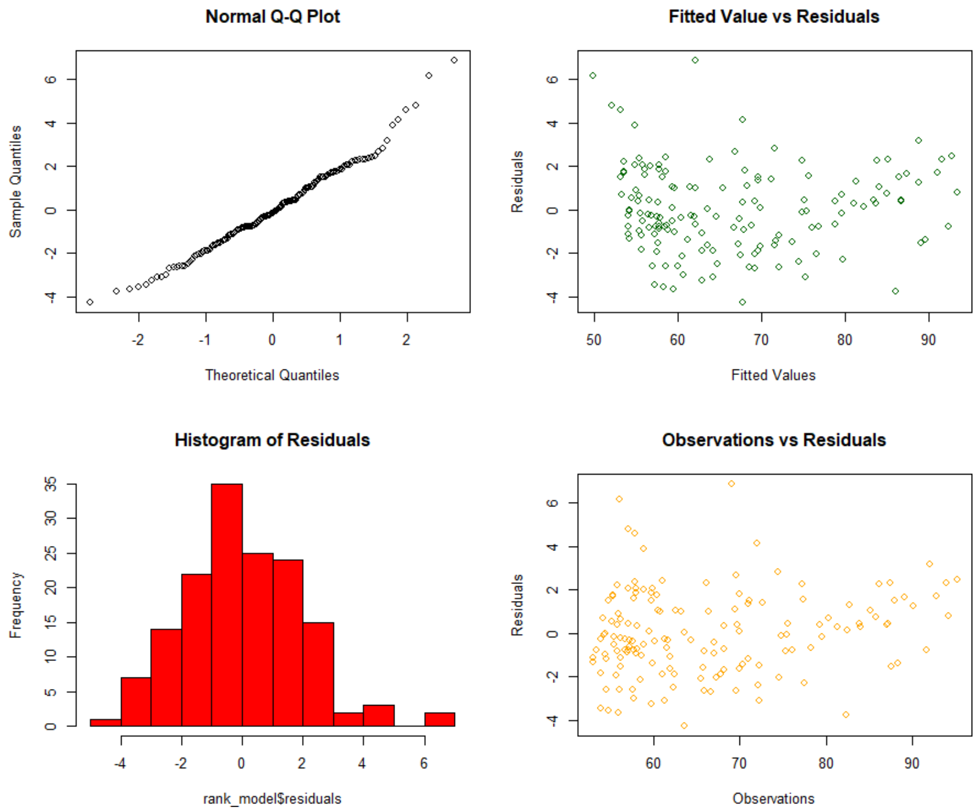
- Multi Collinearity

```
## Multi Collinearity
vif(rank_model)
> vif(rank_model)
           research            citations        international               income
           1.175816             1.195890             1.040620             1.313380
    Female_Students student_staff_ratio
           1.307780             1.252629
```

All the values are below 5. By this, we can say there is no collinearity between predictor variables.

- Residuals

```
## Residuals
layout(matrix(c(1,2,3,4), 2, 2, byrow = TRUE))
qqnorm(rank_model$residuals)
plot(rank_model$fitted.values,rank_model$residuals,main="Fitted Value vs Residuals",xlab
="Fitted Values",ylab="Residuals",col="darkgreen")
hist(rank_model$residuals,col="red",main = "Histogram of Residuals")
plot(training_set$total_score,rank_model$residuals,main="Observations vs Residuals",xlab
="Observations",ylab="Residuals",col="orange")
```



All the residuals plots are normal distributed and there are no deviations in MLR model.

VI. Discussion and Recommendation

- We use 3 kinds of different regression algorithms to build models and test the accuracy of each model, the MLR has the highest accuracy when testing on the test dataset.
- The model of random forest is slightly better than regression tree, the difference should be more obvious when increasing the sample size.
- Some features like 'Teaching' and 'Research' may be strongly correlated with each other in some colleges as the good researchers have great opportunity to become a nice teacher, so multi-collinearity will exist, and if we run a feature selection model (like Lasso), few variables could be dropped. But as the data set's dimension is not that high, dropping them may not result in a huge improvement on model fitting.
- The introduction of cp value and Cross-Validation may optimize the model of CART and increase accuracy.

VII. Summary

1. Data Source –Kaggle
2. Data Exploration – data type transforming, outlier analysis
3. Data Cleaning – Imputing missing values with mice
4. Data Visualization –2 Histograms, 3 Correlation plots

5. Data Splitting – Train 150/Test 50
6. Building Models – Multiple Linear Regression, CART (Regression Tree and Random Forest), Logistic Ordinal Regression
7. Performance evaluations – Linearity, Normality and Multi-Collinearity assumptions, Residuals Analysis
8. Conclusion – Selection of best prediction model – Multiple Linear Regression

Appendix: R Code for use case study

```
## installe packages
install.packages("mice")
install.packages("VIM")
install.packages("corrgram")
install.packages("corrplot")
install.packages("ggplot2")
install.packages("outliers")
install.packages("car")
install.packages("rpart")
install.packages("forecast")
install.packages("randomForest")

## Below are the libraries which are loaded
library("mice")
library("VIM")
library("corrgram")
library("corrplot")
library("outliers")
library("car")
library("ggplot2")
library("rpart")
library("forecast")
library("rpart.plot")
library("randomForest")

## Uploading the data
tm = read.csv("Times_Rankings.csv", header=T)

## Data Preparation and Preprocessing
## Change Columns to numeric
tm$income = sub('-',' ',tm$income)
tm$income = as.numeric(as.character(tm$income))
tm$num_students = gsub(',','',tm$num_students)
tm$num_students = as.numeric(as.character(tm$num_students))
tm$international_students =
as.numeric(as.character(gsub('%','',tm$international_students)))
colnames(tm)[colnames(tm)=="X._Female_Students"] <- "Female_Students"
```

```
colnames(tm)[colnames(tm)=="ï..world_rank"] < - "World_rank"

##Missing Data Analysis
pMiss <- function(x){sum(is.na(x))/length(x)*100}
apply(tm,1,pMiss)
apply(tm,2,pMiss)

## Below gives a clear image of missing data
md.pattern(tm)

## Checking missing values again
sapply(tm, function(x) sum(is.na(x)))
aggr_plot <- aggr(tm, col=c('blue','red'), numbers=TRUE, sortVars=TRUE,
labels=names(tm), cex.axis=.7, gap=3, ylab=c("Histogram of missing data","Pattern"))

## Data imputation
tempData <- mice(tm,m=5,meth='cart')
sapply(training_set, function(x) sum(is.na(x)))
modelFit2 <- with(tempData, lm(total_score~research+citations+international+income))
summary(modelFit2)
train_complete <- complete(tempData,"long")
training_set = train_complete[(1:150),]
test_set = train_complete[(150:200),]

## Outlier Detection
outlier(tm[,4:13])

## Correlation plot of newly imputed data
q = as.matrix(training_set[,c(3,6:15)])
corrplot(cor(q), method = "shade", number.cex=0.75, is.corr = FALSE)

par(mfrow=c(2,2))
plot(training_set, col="blue", main="Matrix Scatterplot of all the independent variables")
ggpairs(training_set[,c(3,6:15)])


## Data Mining Techniques and Implementation
## MLR
## Building different models
rank_model =
lm(total_score~research+citations+international+teaching+Female_Students+student_sta
ff_ratio+international_students+num_students+income, data=training_set)
summary(rank_model)
rank_model =
lm(total_score~research+citations+international+Female_Students+student_staff_ratio+i
nternational_students+num_students+income, data=training_set)
```

```
summary(rank_model)
rank_model =
lm(total_score~research+citations+international+income+Female_Students+student_staff
_ratio, data=training_set)
summary(rank_model)

par(mfrow=c(2,2))
plot(rank_model)

## Predictions for data
lmpred<-predict (rank_model, test_set)
accuracy(lmpred, test_set$total_score)

## CART
## Regression Tree
## Building model
tree1 <- rpart(total_score ~
research+citations+international+teaching+Female_Students+student_staff_ratio+internat
ional_students+num_students+income, data=training_set)
rpart.plot(tree1, branch = 1, branch.type = 1, type = 2,shadow.col='gray',
box.col='green',border.col='blue', split.col='red',main="Regression Tree")

## Predictions for data
treepred1 <- predict(tree1,test_set)
accuracy(treepred1, test_set$total_score)

## Random Forest
## Building model
tree2 <- randomForest(total_score ~
research+citations+international+teaching+Female_Students+student_staff_ratio+internat
ional_students+num_students+income, data=training_set)

## Predictions for data
treepred2 <- predict(tree2,test_set)
accuracy(treepred2, test_set$total_score)

## Logistic Ordinal Regression
library("ordinal")
library("MASS")
str(tm)
training_set2 = train_complete[(1:30),]
training_set2 = training_set[,c(3,6:15)]
training_set2$World_rank = as.factor (training_set$World_rank)
training_set2$World_rank <- ordered (training_set$World_rank, levels = c(30:1))
model <- polr(World_rank~research+citations+income,training_set2, Hess = TRUE)
summary(model)
```

```
## Getting Co-efficients
(ctable = coef(summary(model)))
p <- pnorm(abs(ctable[, "t value"]), lower.tail = FALSE) * 2
(ctable <- cbind(ctable, "p value" = p))

## Confidence Intervals
(ci <- confint(model))
(exp(coef(model)))
exp(cbind(OR = coef(model), ci))

## Predictions for data
pred <- predict(model, training_set)
print(pred, digits = 3)

## Performance Evaluation
## Linearity
plot(fitted(rank_model), residuals(rank_model))
abline(0,1, col="blue", lwd=2)

## Normality
hist(rank_model$residuals, main ="Histogram of residuals")
qqPlot(rank_model, main="QQ Plot")

## Multi Collinearity
vif(rank_model)

## Residuals
layout(matrix(c(1,2,3,4), 2, 2, byrow = TRUE))
qqnorm(rank_model$residuals)
plot(rank_model$fitted.values,rank_model$residuals,main="Fitted Value vs
Residuals",xlab="Fitted Values",ylab="Residuals",col="darkgreen")
hist(rank_model$residuals,col="red",main = "Histogram of Residuals")
plot(training_set$total_score,rank_model$residuals,main="Observations vs
Residuals",xlab="Observations",ylab="Residuals",col="orange")
```