US Research University Prediction Model

Philip Gabriel Andrada

November 14, 2016

Preparation

```
# loading necessary libraries
library(rpart)
library(randomForest)
```

```
## randomForest 4.6-12
```

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

```
library(tree)
library(party)
```

```
## Loading required package: grid
```

```
## Loading required package: mvtnorm
```

```
## Loading required package: modeltools
```

```
## Loading required package: stats4
```

```
## Loading required package: strucchange
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: sandwich
```

```
library(caret)
```

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

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'ggplot2'

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

```r
library(Boruta)
```

```
## Loading required package: ranger

##
## Attaching package: 'ranger'

## The following object is masked from 'package:randomForest':
##
##     importance
```

```r
library(e1071)
library(ROCR)
```

```
## Loading required package: gplots

##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##     lowess
```

```r
library(corrplot)
library(ggplot2)
```

```r
#Reading Data Files
usuniv2010 <- read.csv("C:\\Users\\pandrada\\Desktop\\Capstone\\MERGED2010_11_PP.csv")
usuniv2011 <- read.csv("C:\\Users\\pandrada\\Desktop\\Capstone\\MERGED2011_12_PP.csv")
usuniv2012 <- read.csv("C:\\Users\\pandrada\\Desktop\\Capstone\\MERGED2012_13_PP.csv")
usuniv2013 <- read.csv("C:\\Users\\pandrada\\Desktop\\Capstone\\MERGED2013_14_PP.csv")
usuniv2014 <- read.csv("C:\\Users\\pandrada\\Desktop\\Capstone\\MERGED2014_15_PP.csv")

#Binding All Data Files into One Data Frame
usuniv <- rbind(usuniv2010,usuniv2011,usuniv2012,usuniv2013,usuniv2014)
```

```
## Warning in `[<-.factor`(`*tmp*`, ri, value = c(100200L, 105200L,
## 2503400L, : invalid factor level, NA generated

## Warning in `[<-.factor`(`*tmp*`, ri, value = c(100200L, 105200L,
## 2503400L, : invalid factor level, NA generated

## Warning in `[<-.factor`(`*tmp*`, ri, value = c(100200L, 105200L,
```

```
## 2503400L, : invalid factor level, NA generated

## Warning in `[<-.factor`(`*tmp*`, ri, value = c(100200L, 105200L,
## 2503400L, : invalid factor level, NA generated

## Warning in `[<-.factor`(`*tmp*`, ri, value = c(100200L, 105200L,
## 2503400L, : invalid factor level, NA generated

## Warning in `[<-.factor`(`*tmp*`, ri, value = c(100200L, 105200L,
## 2503400L, : invalid factor level, NA generated
```

```
#Since there are some incomplete Carnegie Classifications, we use usuniv2014 as basis for the classific
usuniv$CCBASIC2 <- usuniv2014$CCBASIC[match(usuniv$OPEID6,usuniv2014$OPEID6)]

#added the ACCEPTED column for those that are research universities (CCBASIC2 is equal to 15 or 16), as
usuniv$ACCEPTED <- ifelse(usuniv$CCBASIC2 %in% c(15,16), 1, 0)

#number of rows in the usuniv data frame
rows_usuniv <- nrow(usuniv)
rows_usuniv
```

```
## [1] 38389
```

```
#number of columns that are in the usuniv data frame
ncol(usuniv)
```

```
## [1] 1745
```

```
#number of rows that are research universities in the data frame before cleansing
rows_usunivaccepted <- nrow(usuniv[usuniv$ACCEPTED == 1,])
rows_usunivaccepted
```

```
## [1] 1154
```

```
#grab a head of research universities to see if we got the correct ones
head(usuniv[usuniv$ACCEPTED == 1,c(4,1744:1745)], 30)
```

```
##                                                 INSTNM CCBASIC2
## 2                  University of Alabama at Birmingham       15
## 4                  University of Alabama in Huntsville       16
## 6                            The University of Alabama       16
## 10                                   Auburn University       16
## 50                         University of South Alabama       16
## 61                      University of Alaska Fairbanks       16
## 82                      Arizona State University-Tempe       15
## 84                              University of Arizona       15
## 113                         Northern Arizona University       16
## 144                             University of Arkansas       15
## 237                  California Institute of Technology      15
## 254                  University of California-Berkeley       15
## 255                     University of California-Davis       15
```

```
## 256                      University of California-Irvine       15
## 257                 University of California-Los Angeles       15
## 258                   University of California-Riverside       15
## 259                   University of California-San Diego       15
## 261               University of California-Santa Barbara       15
## 262                  University of California-Santa Cruz       15
## 294                        Claremont Graduate University       16
## 518                           San Diego State University       16
## 567                  University of Southern California       15
## 604 University of Colorado Denver/Anschutz Medical Campus       16
## 607                       University of Colorado Boulder       15
## 614                             Colorado School of Mines       16
## 616               Colorado State University-Fort Collins       15
## 627                                 University of Denver       16
## 644                      University of Northern Colorado       16
## 675                           University of Connecticut       15
## 720                                      Yale University       15
##     ACCEPTED
## 2          1
## 4          1
## 6          1
## 10         1
## 50         1
## 61         1
## 82         1
## 84         1
## 113        1
## 144        1
## 237        1
## 254        1
## 255        1
## 256        1
## 257        1
## 258        1
## 259        1
## 261        1
## 262        1
## 294        1
## 518        1
## 567        1
## 604        1
## 607        1
## 614        1
## 616        1
## 627        1
## 644        1
## 675        1
## 720        1
```

```
#Create a vector with the columns that is needed from the study
# 19 - institution region (1-New England, 2-Mid East, 3-Great Lakes, 4-Plains, 5-Southeast, 6-Southwest
# 37-38 - admission rate
# 39-61 - SAT and ACT Scores
# 62-99 - percentage of degrees awarded for each field of study
```

```
# 293-299 - total share of enrollment for different ethnicities
# 300 - total share of enrollment that are non-resident aliens (i.e. international students)
# 301 - total share of enrollment that have unknown race
# 314 - share of undergraduate, degree-/certificate-seeking students who are part-time
# 377 - average cost of attendance in an academic year institution
# 379 - in-state tuition and fees
# 380 - out-of-state tuition and fees
# 387 - completion rate of first-time, full-time students at four-year institutions with 150% of expect
# 397-403 - completion rate for first-time, full-time students for different ethnicities
# 404 - completion rate for first-time, full-time students for non-resident aliens
# 405 - completion rate for first-time, full-time students that have unknown race
# 429 - retention rate for first-time, full time students at four-year institutions
# 438 - percent of all federal undergraduate students receiving a federal student loan
# 1412 - percentage of first-generation students
# 1740-1741 - total share of enrollment per gender
# 1745 - acceptance flag
col_select <- c(19,37:38,61:99,293:301,314,377,379:380,387,397:405,429,438,1412,1740:1741, 1744, 1745)

# Create a new data frame with the columns that will be filtered out
usunivfilter <- usuniv[,col_select]

# Change the factor columns to numeric for faster processing
for (i in 1:ncol(usunivfilter)){
  usunivfilter[,i] <- as.numeric(as.character(usunivfilter[,i]))
}
```

```
## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion
```

```
## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion
```

```
## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion
```

```
## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion
```

```
# Clean the results to have all complete
usunivfilter <- usunivfilter[!is.na(usunivfilter$C150_4),]
usunivfilter <- usunivfilter[!is.na(usunivfilter$C150_4_ASIAN),]
usunivfilter <- usunivfilter[!is.na(usunivfilter$C150_4_WHITE),]
usunivfilter <- usunivfilter[!is.na(usunivfilter$C150_4_BLACK),]
usunivfilter <- usunivfilter[!is.na(usunivfilter$C150_4_NRA),]
usunivfilter <- usunivfilter[!is.na(usunivfilter$ADM_RATE_ALL),]
usunivfilter <- usunivfilter[!is.na(usunivfilter$SAT_AVG_ALL),]
usunivfilter <- usunivfilter[!is.na(usunivfilter$UGDS_ASIAN),]
usunivfilter <- usunivfilter[!is.na(usunivfilter$UGDS_WHITE),]
usunivfilter <- usunivfilter[!is.na(usunivfilter$UGDS_BLACK),]
usunivfilter <- usunivfilter[!is.na(usunivfilter$UGDS_NRA),]
usunivfilter <- usunivfilter[!is.na(usunivfilter$UGDS_WOMEN),]
usunivfilter <- usunivfilter[!is.na(usunivfilter$UGDS_MEN),]
usunivfilter <- usunivfilter[!is.na(usunivfilter$COSTT4_A),]
usunivfilter <- usunivfilter[!is.na(usunivfilter$PCIP11),]
usunivfilter <- usunivfilter[!is.na(usunivfilter$PCIP12),]
usunivfilter <- usunivfilter[!is.na(usunivfilter$PCIP14),]
usunivfilter <- usunivfilter[!is.na(usunivfilter$PCIP15),]
usunivfilter <- usunivfilter[!is.na(usunivfilter$PCIP24),]
usunivfilter <- usunivfilter[!is.na(usunivfilter$PCIP26),]
usunivfilter <- usunivfilter[!is.na(usunivfilter$PCIP27),]
usunivfilter <- usunivfilter[!is.na(usunivfilter$PCIP40),]
usunivfilter <- usunivfilter[!is.na(usunivfilter$PCIP45),]
usunivfilter <- usunivfilter[!is.na(usunivfilter$PCIP51),]
usunivfilter <- usunivfilter[!is.na(usunivfilter$PCIP52),]
usunivfilter <- usunivfilter[!is.na(usunivfilter$PCTFLOAN),]
usunivfilter <- usunivfilter[!is.na(usunivfilter$PPTUG_EF),]
usunivfilter <- usunivfilter[!is.na(usunivfilter$RET_FT4),]
usunivfilter <- usunivfilter[!is.na(usunivfilter$PAR_ED_PCT_1STGEN),]

#We will create another data frame for the research universities only
usresearchuniv <- usunivfilter[usunivfilter$CCBASIC2 %in% c(15,16),]

#show number of rows in the filtered usuniv
rows_usunivfilter <- nrow(usunivfilter)
rows_usunivfilter
```

```
## [1] 4247
```

```
#percentage of data from filtered to unfiltered
rows_usunivfilter / rows_usuniv
```

```
## [1] 0.1106306
```

```r
#show number of rows of filtered research universities
rows_usresearchuniv <- nrow(usresearchuniv)
rows_usresearchuniv
```

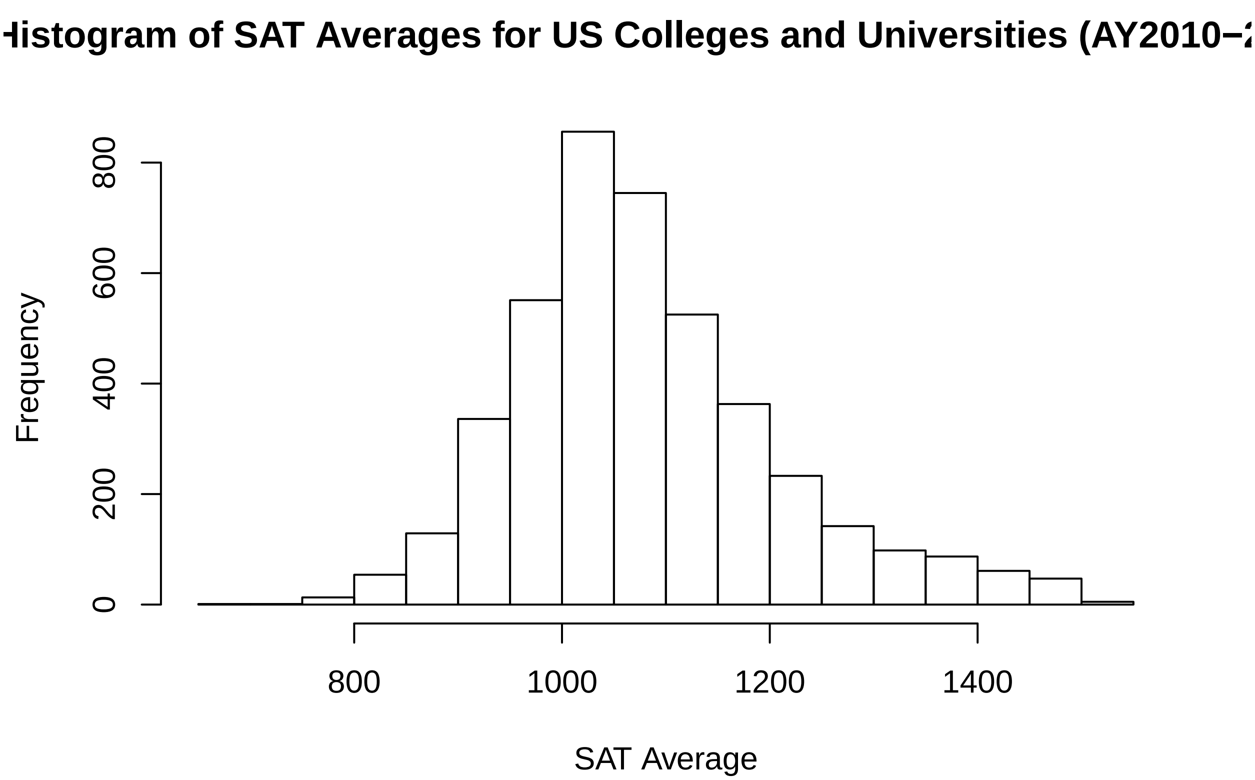
```
## [1] 815
```

```r
#percentage of data from filtered research universities to unfiltered
rows_usresearchuniv / rows_usunivaccepted
```

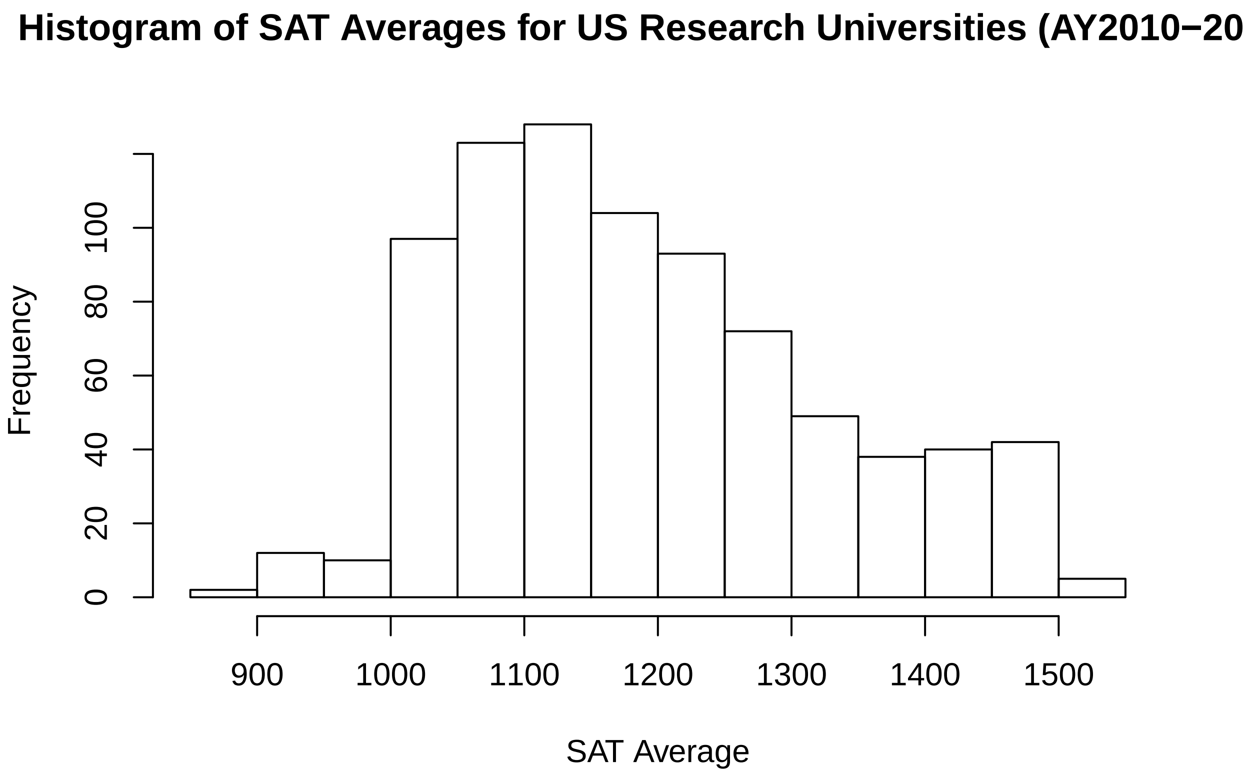
```
## [1] 0.7062392
```

Distributions and Box and Whisker Plots

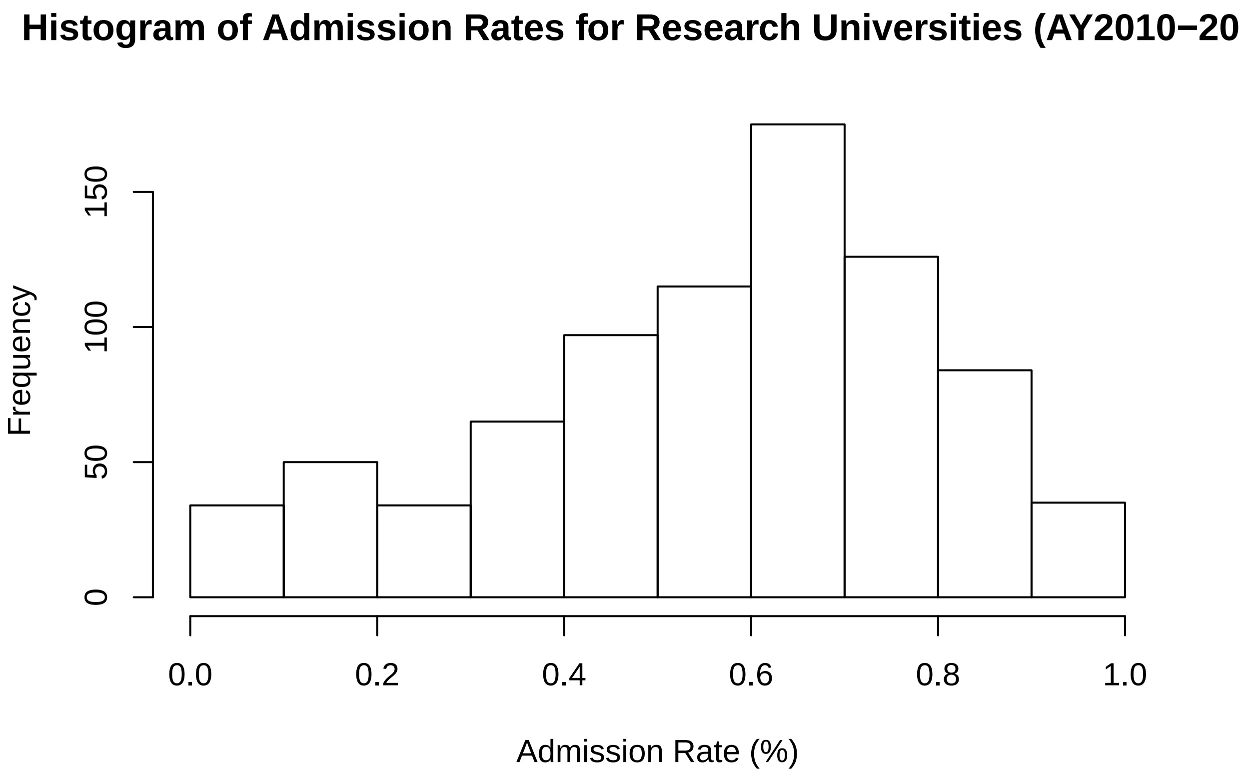
```r
# Histogram of SAT Averages for US Colleges and Universities
hist(usunivfilter$SAT_AVG_ALL, main = "Histogram of SAT Averages for US Colleges and Universities (AY20
```



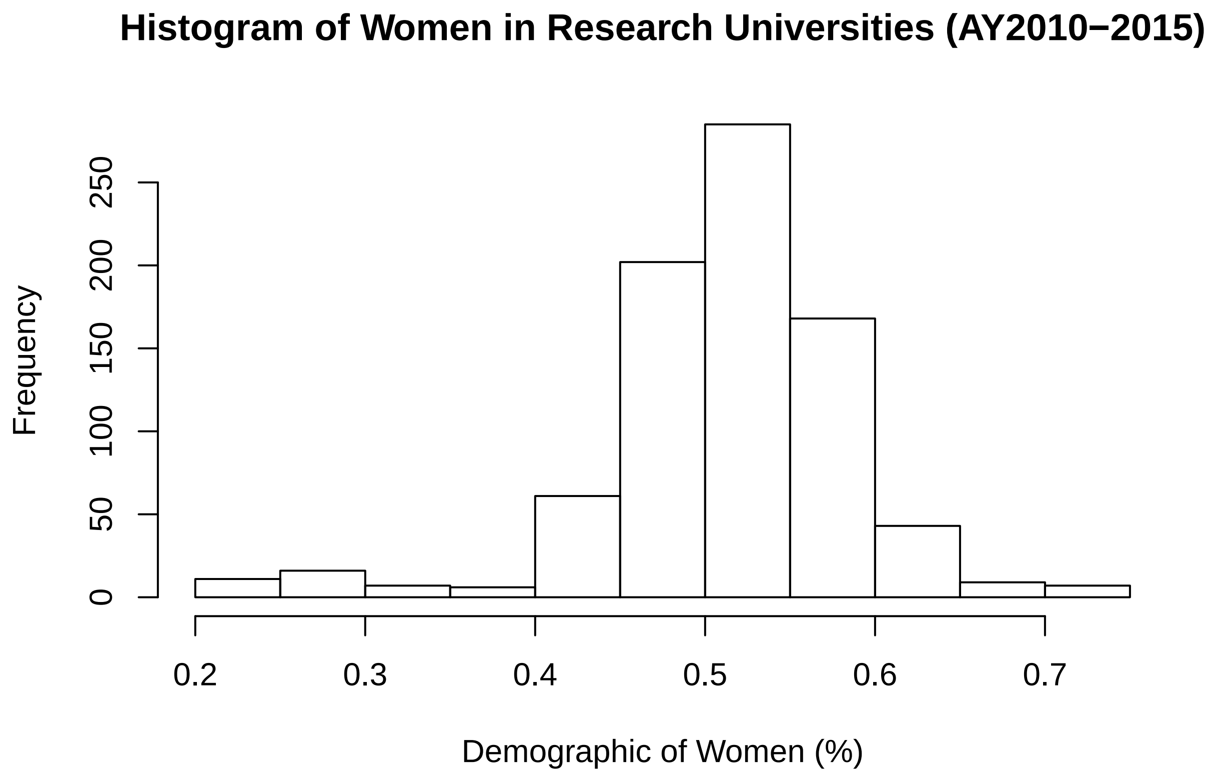

```r
# Histogram of SAT Averages for US Research Universities
hist(usresearchuniv$SAT_AVG_ALL, main = "Histogram of SAT Averages for US Research Universities (AY2010
```



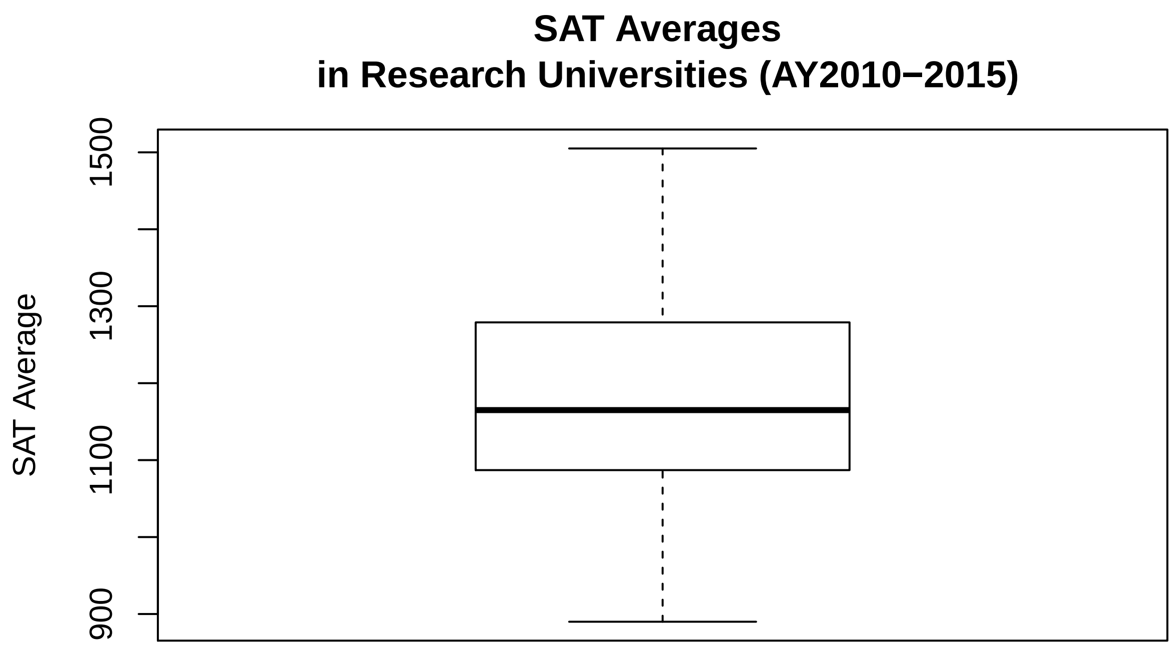

```
# Histogram of Admission Rates for US Research Universities
hist(usresearchuniv$ADM_RATE_ALL, main = "Histogram of Admission Rates for Research Universities (AY201
```



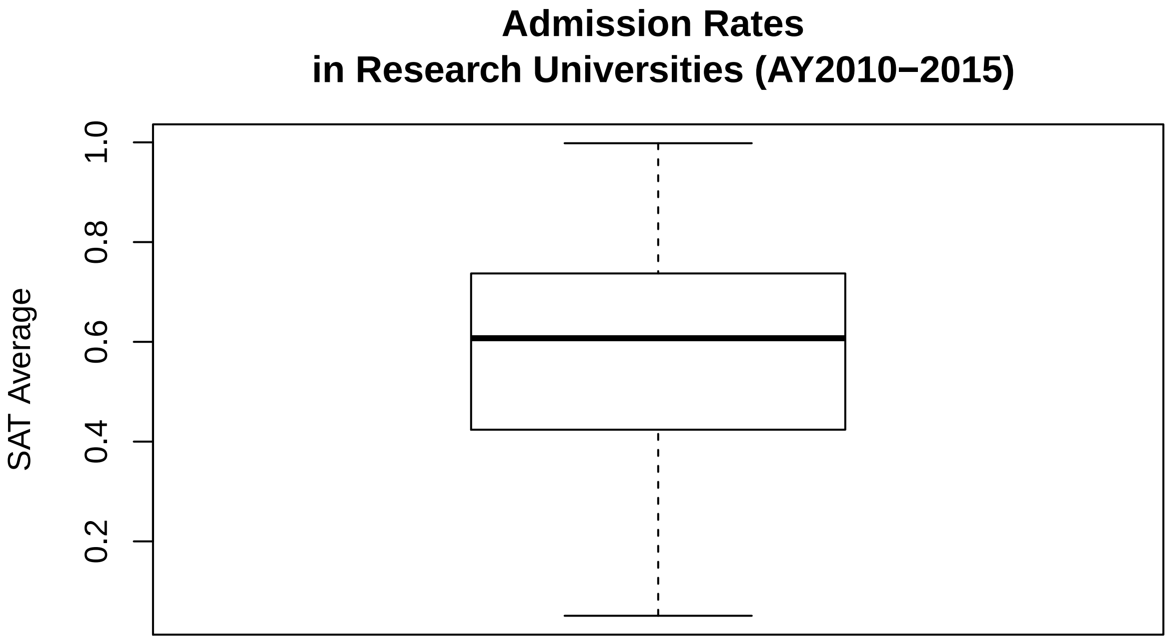

```
# Histogram of Women in US Research Universities
hist(usresearchuniv$UGDS_WOMEN, main = "Histogram of Women in Research Universities (AY2010-2015)", xlal
```



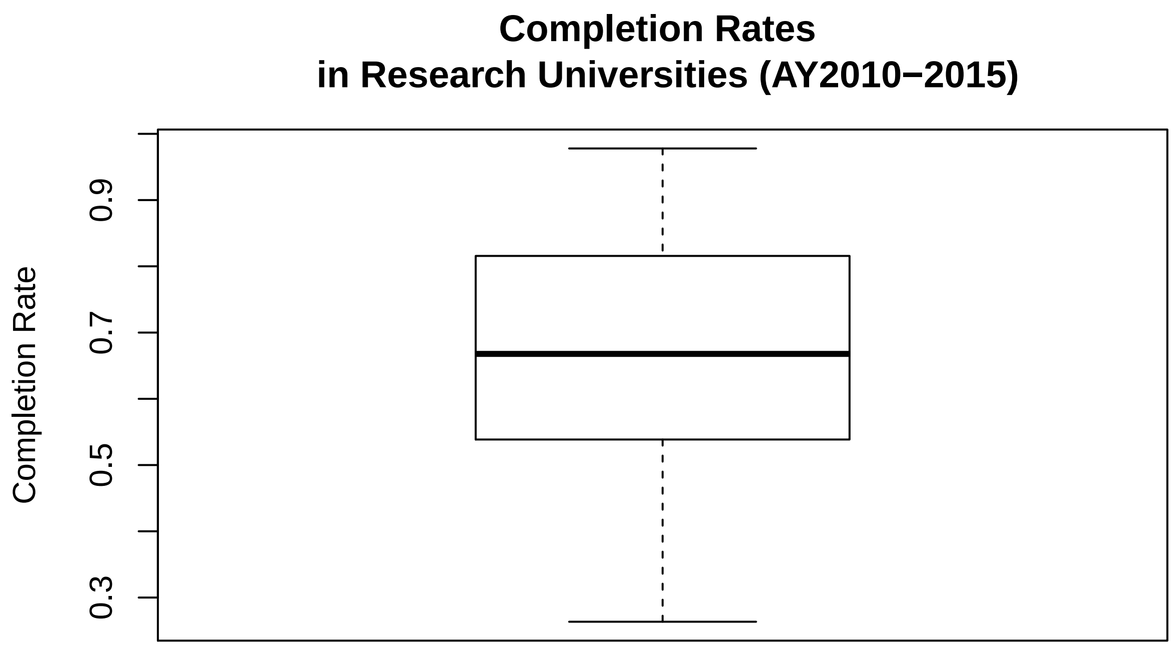

```r
#Boxplot of SAT Average in all US Research Universities
boxplot(usresearchuniv$SAT_AVG_ALL, main = "SAT Averages \n in Research Universities (AY2010-2015)", yla
```



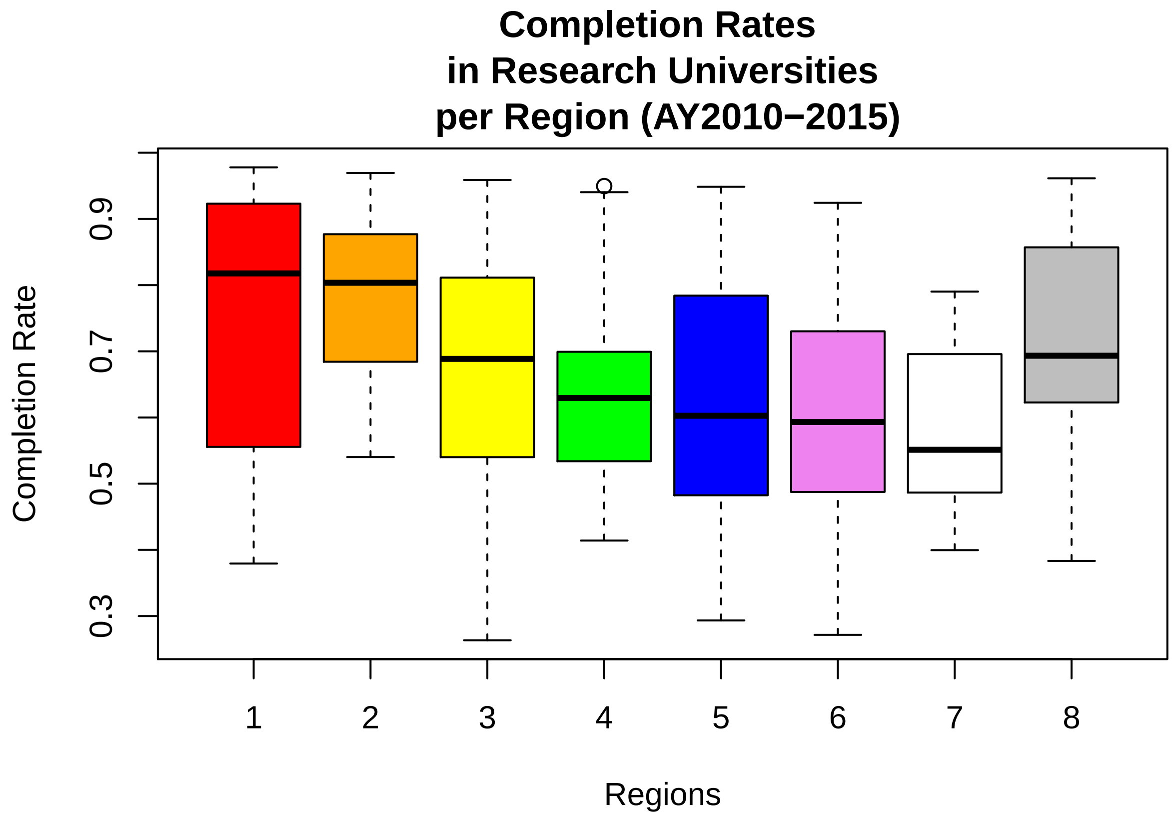

```r
#Boxplot of admission rates in all US Research Universities
boxplot(usresearchuniv$ADM_RATE_ALL, main = "Admission Rates \n in Research Universities (AY2010-2015)"
```



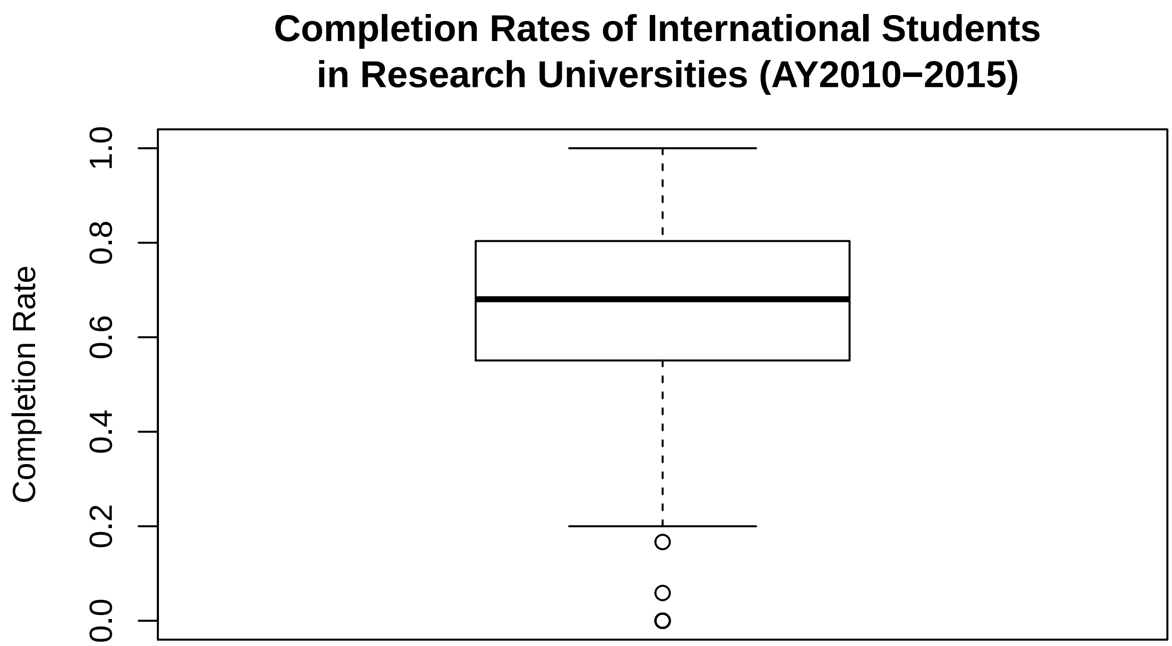

```
#Boxplot of Completion Rates in all US Research Universities
boxplot(usresearchuniv$C150_4, main = "Completion Rates \n in Research Universities (AY2010-2015)", ylal
```



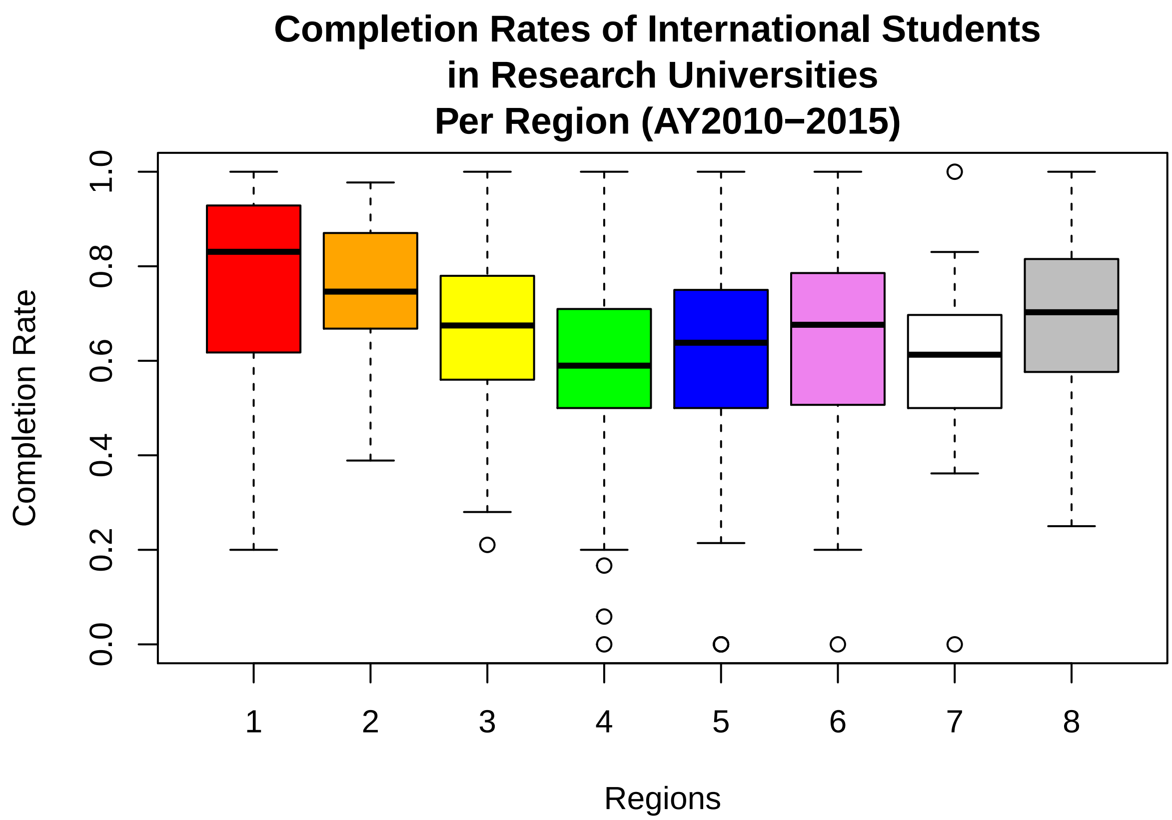

```r
# Boxplot of Completion Rates per Region in US Research Universities
boxplot(C150_4 ~ REGION, usresearchuniv, main = "Completion Rates \n in Research Universities \n per Reg
```




```r
#Boxplot of Completion Rates of International Students in all US Research Universities
boxplot(usresearchuniv$C150_4_NRA, main = "Completion Rates of International Students \n in Research Uni
```




```
# Boxplot of Completion Rates of International Students per Region in US Research Universities
boxplot(C150_4_NRA ~ REGION, usresearchuniv, main = "Completion Rates of International Students \n in Re
```

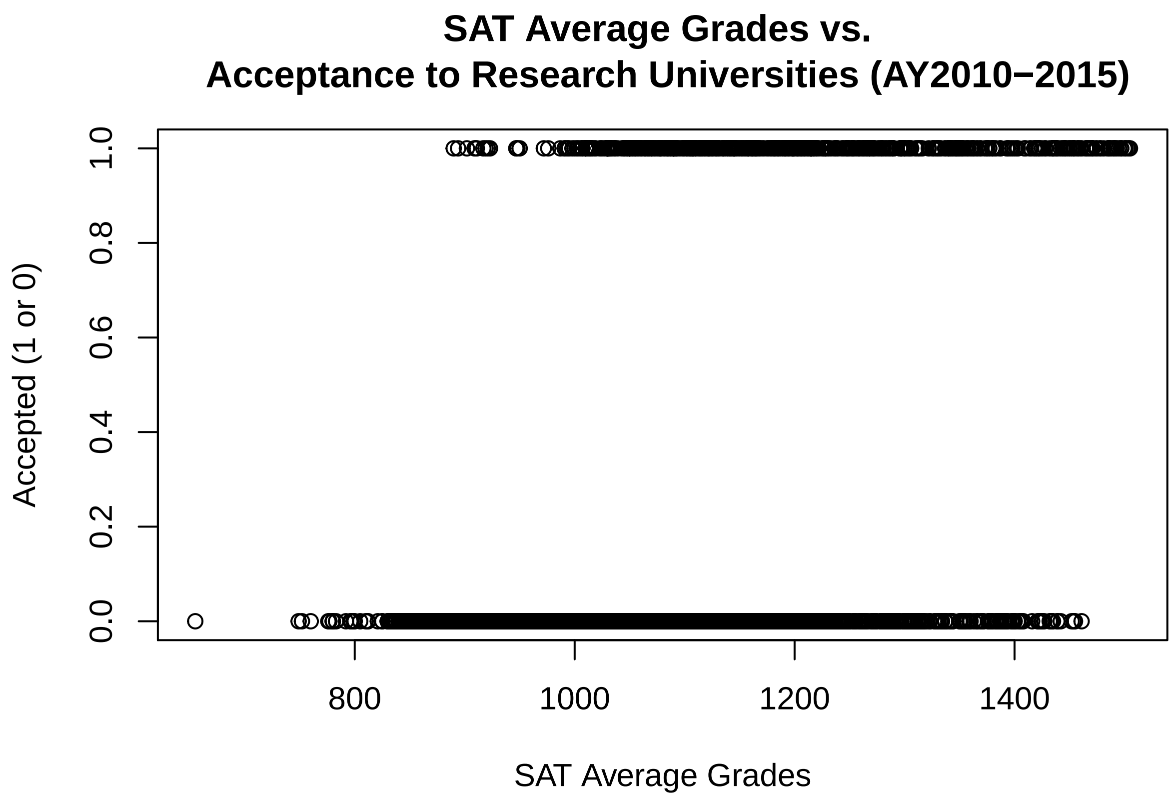



```
nrow(usresearchuniv[usresearchuniv$C150_4_NRA < 0.2,])
```

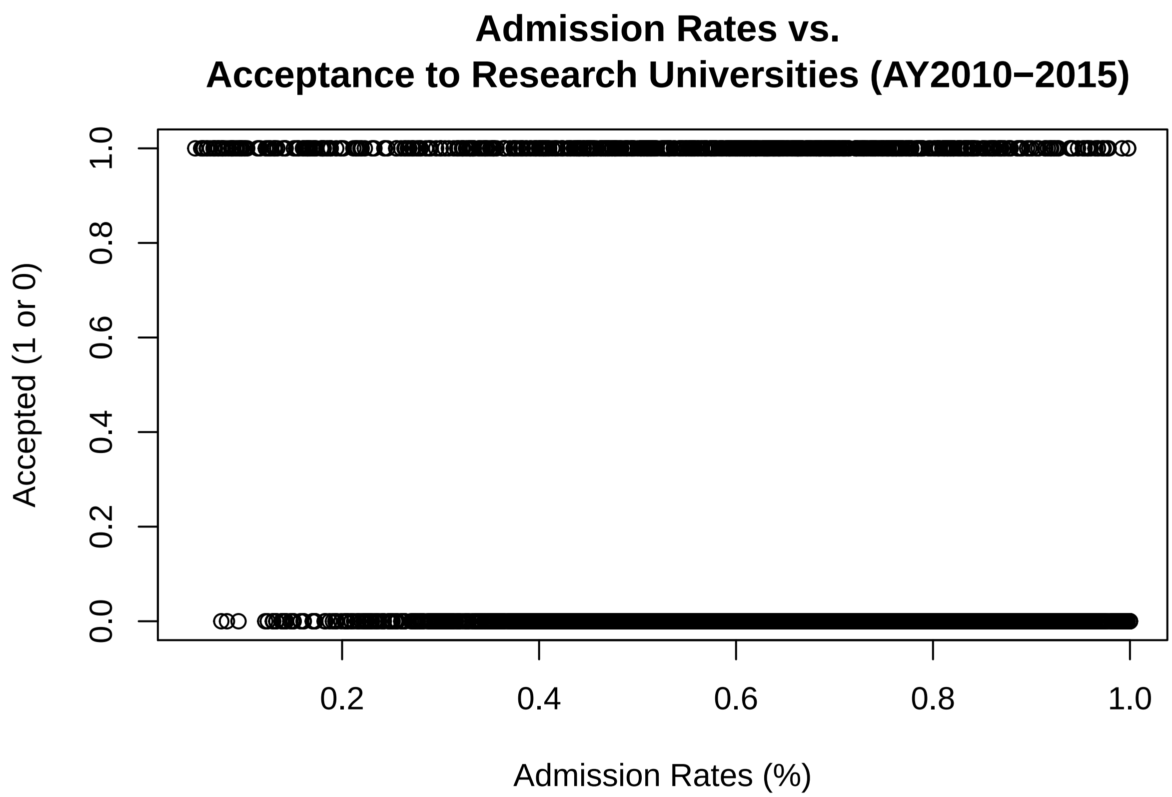
```
## [1] 9
```

Correlations

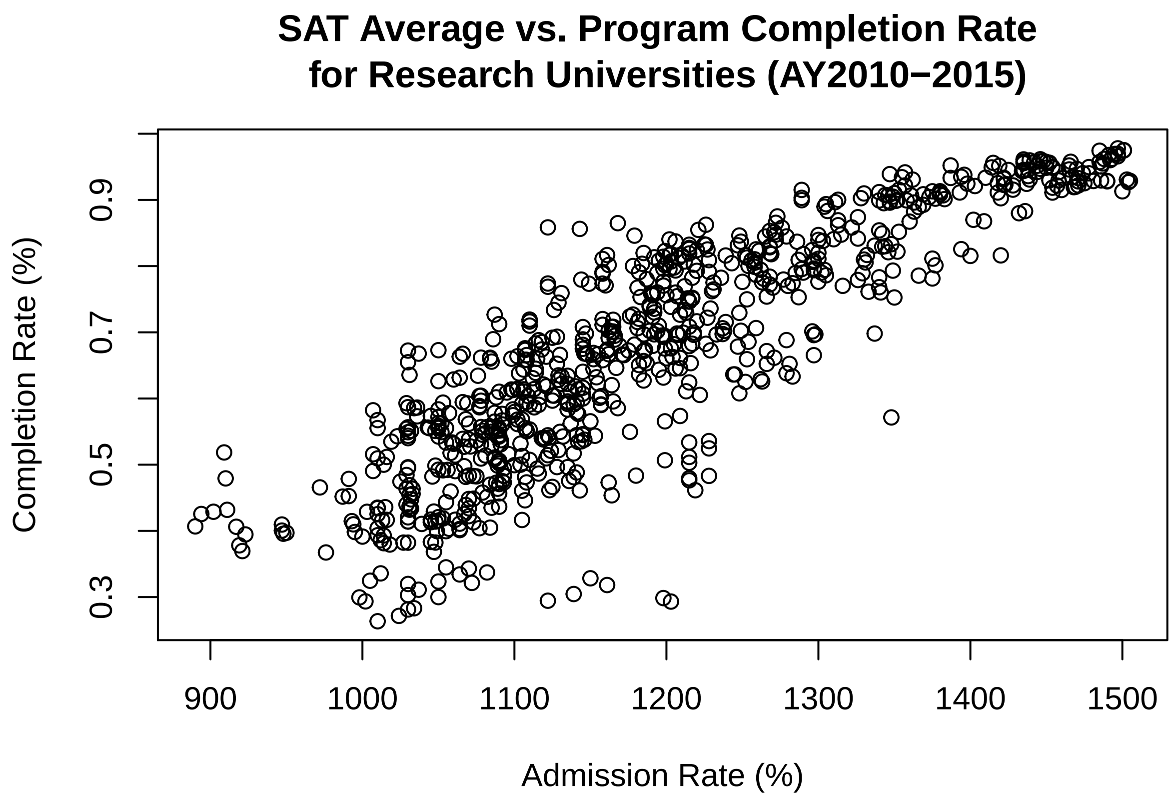
```
#Correlation between the SAT grades and the acceptance for the research universities
plot(usunivfilter$SAT_AVG_ALL, usunivfilter$ACCEPTED, main="SAT Average Grades vs. \n Acceptance to Rese
```




```
#Correlation between the admission rates and the acceptance for the research universities
plot(usunivfilter$ADM_RATE_ALL, usunivfilter$ACCEPTED, main="Admission Rates vs. \n Acceptance to Resea
```




```
#Correlation between admission rate for research universities and program completion rate
plot(usresearchuniv$SAT_AVG_ALL, usresearchuniv$C150_4, main="SAT Average vs. Program Completion Rate \r
```

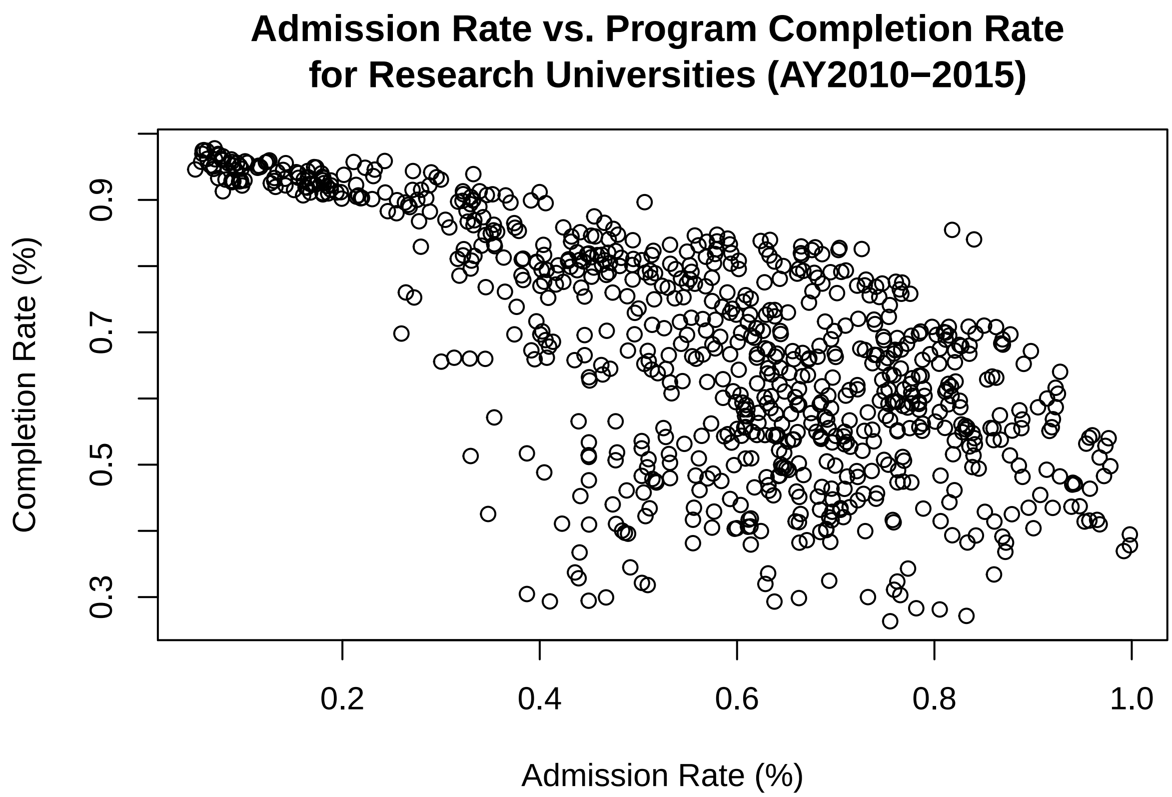



```r
#Correlation coefficient between admission rate and completion rate
cor(usresearchuniv$SAT_AVG_ALL, usresearchuniv$C150_4, method = "pearson")
```

```
## [1] 0.8702261
```

This means that there is a strong positive correlation between the SAT average scores and the completion rate for all students.

```r
#Correlation between admission rate for research universities and program completion rate
plot(usresearchuniv$ADM_RATE_ALL, usresearchuniv$C150_4, main="Admission Rate vs. Program Completion Ra
```

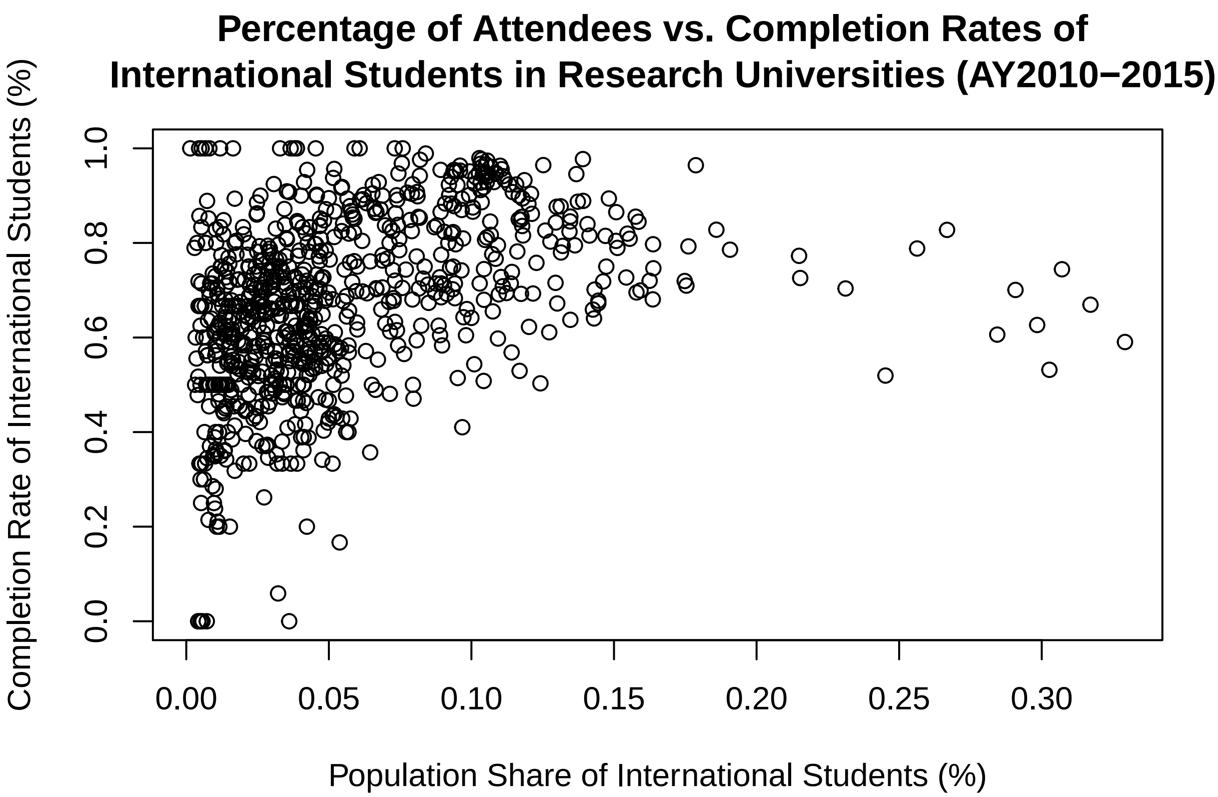



```r
#Correlation coefficient between admission rate and completion rate
cor(usresearchuniv$ADM_RATE_ALL, usresearchuniv$C150_4, method = "pearson")
```

```
## [1] -0.6825525
```

This means that there is a strong negative correlation between the admission rates and the completion rates for the research universities.

```r
#Correlation between attendees and completion rate of non-resident aliens (International Students)
plot(usresearchuniv$UGDS_NRA, usresearchuniv$C150_4_NRA, main="Percentage of Attendees vs. Completion Ra
```

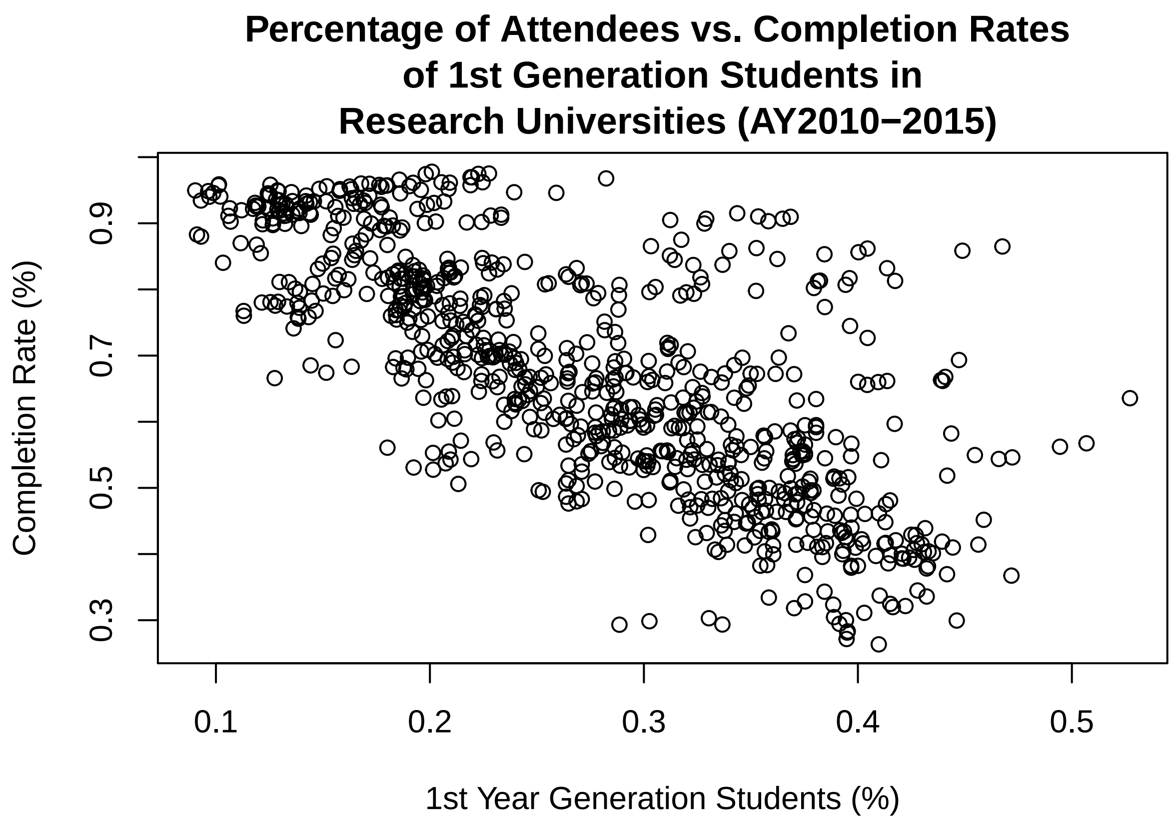



```r
#Correlation coefficient between admission rate and completion rate of international students
cor(usresearchuniv$UGDS_NRA, usresearchuniv$C150_4_NRA, method = "pearson")
```

```
## [1] 0.370641
```

This means that there is a weak positive correlation between international student population and their completion rate.

```r
#Correlation between attendees and completion rate of 1st Generation students in Research Universities
plot(usresearchuniv$PAR_ED_PCT_1STGEN, usresearchuniv$C150_4, main="Percentage of Attendees vs. Complet
```




```
#Correlation coefficient between admission rate and completion rate of 1st Generation students
cor(usresearchuniv$PAR_ED_PCT_1STGEN, usresearchuniv$C150_4, method = "pearson")
```

```
## [1] -0.7419477
```

This means that there is a strong negative correlation between 1st generation students and completion rates in research universities.

U.S. Research University Acceptance Model

In this report section, we are going to create a formula on getting an acceptance to a US Research University based on the College Scorecard statistics. We will try different methods of regression, and find the best regression technique from the following sources.

We will also consider another formula based on an international student taking up science degree/major.

```
# create a training and test model using a 75%/25% from the data set
rm_train <- sample(nrow(usunivfilter), floor(nrow(usunivfilter)*0.75))
univ_train <- usunivfilter[rm_train,]
univ_test <- usunivfilter[-rm_train,]

# create a generic formula for the US research university acceptance model for International Students b
formula_ISAcceptance <- formula(ACCEPTED ~ REGION + ADM_RATE_ALL + SAT_AVG_ALL + UGDS_NRA + COSTT4_A + I
```

We will do a generalized logistic regression formula.

```
# create a logistic regression
fit1 <- glm(formula_ISAcceptance, data = usunivfilter, family  = binomial())
summary(fit1)
```

```
##
## Call:
## glm(formula = formula_ISAcceptance, family = binomial(), data = usunivfilter)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2091  -0.5400  -0.2922  -0.1192   2.7993
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.478e+01  1.029e+00 -14.362  < 2e-16 ***
## REGION        1.246e-01  2.550e-02   4.886 1.03e-06 ***
## ADM_RATE_ALL  7.036e-01  3.297e-01   2.134   0.0328 *
## SAT_AVG_ALL   1.462e-02  7.312e-04  19.999  < 2e-16 ***
## UGDS_NRA      6.637e+00  1.147e+00   5.784 7.28e-09 ***
## COSTT4_A     -9.181e-05  5.441e-06 -16.872  < 2e-16 ***
## PCTFLOAN     -7.486e-01  4.247e-01  -1.763   0.0779 .
## UGDS_WOMEN   -1.995e+00  4.619e-01  -4.318 1.57e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4153.3  on 4246  degrees of freedom
## Residual deviance: 2838.4  on 4239  degrees of freedom
## AIC: 2854.4
##
## Number of Fisher Scoring iterations: 6
```

Based on the logistic regression, the formula will be

$$\frac{1}{1+e^{-x}}$$

where

$x = -14.8 + 0.125 REGION + 0.704 ADM\_RATE\_ALL + 0.0146 SAT\_AVG\_ALL + 6.64 UGDS\_NRA - 0.0000918 COSTT4$

.

We will test this regression with some data types.

```
# this will not accept the person because of the SAT average
df_accept <- data.frame(REGION = 5, SAT_AVG_ALL = 900, ADM_RATE_ALL = .55, UGDS_NRA=.010, COSTT4_A = 20
predict(fit1, type = "response", newdata = df_accept)
```

```
##          1
## 0.03356807
```

```
# this will accept because of the SAT average and the cost
df_accept2 <- data.frame(REGION = 3, SAT_AVG_ALL = 1350, ADM_RATE_ALL = .35, UGDS_NRA=.25, COSTT4_A = 25
predict(fit1, type = "response", newdata = df_accept2)
```

```
##         1
## 0.9667774
```

Now, we will do some testing of performance with the logistic regression. Since we have split the dataset into training and testing set, we will see how the performance will be done.

```
# do a logistic regression model based on this
glm_ISAcceptance <- glm(formula_ISAcceptance, data = univ_train, family = binomial())
summary(glm_ISAcceptance)
```

```
##
## Call:
## glm(formula = formula_ISAcceptance, family = binomial(), data = univ_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2074  -0.5436  -0.3004  -0.1194   2.8135
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.456e+01  1.182e+00 -12.323  < 2e-16 ***
## REGION        1.332e-01  2.925e-02   4.556 5.22e-06 ***
## ADM_RATE_ALL  3.950e-01  3.771e-01   1.047  0.29492
## SAT_AVG_ALL   1.446e-02  8.335e-04  17.350  < 2e-16 ***
## UGDS_NRA      6.496e+00  1.279e+00   5.077 3.83e-07 ***
## COSTT4_A     -9.450e-05  6.212e-06 -15.212  < 2e-16 ***
## PCTFLOAN     -5.552e-01  4.865e-01  -1.141  0.25381
## UGDS_WOMEN   -1.743e+00  5.334e-01  -3.268  0.00108 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3139.9  on 3184  degrees of freedom
## Residual deviance: 2153.7  on 3177  degrees of freedom
## AIC: 2169.7
##
## Number of Fisher Scoring iterations: 6
```

```
# do the first testing with the prediction model
accepted_ind <- predict(glm_ISAcceptance, type="response", newdata = univ_test)
pred1 <- prediction(accepted_ind, univ_test$ACCEPTED)

# create the confusion matrix and accuracy for this prediction model
c1 <- confusionMatrix(as.integer(accepted_ind > 0.5), univ_test$ACCEPTED)
c1$table
```

```
##           Reference
```

```
## Prediction   0   1
##          0 832 109
##          1  35  86
```

```r
#Accuracy of the logistic regression model
c1$overall['Accuracy']
```

```
##  Accuracy
## 0.8644068
```

```r
#Precision of the logistic regression model
c1$byClass['Neg Pred Value']
```

```
## Neg Pred Value
##      0.7107438
```

```r
#Recall of the logistic regression model
c1$byClass['Specificity']
```

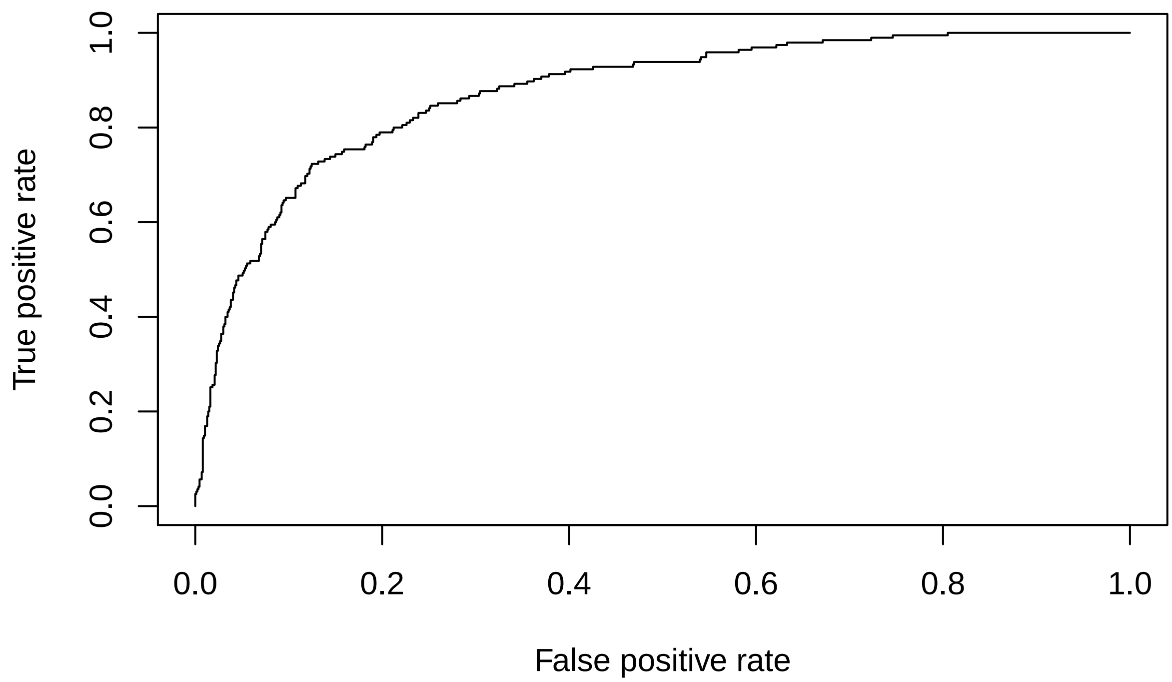
```
## Specificity
##   0.4410256
```

Accuracy shows the correct value. But in precision and recall, it is using "Neg Pred Value" and "Specificity" respectively. It should have been "Pos Pred Value" and "Sensitivity", as defined before. However, I manually calculated for the precision and recall for these values, and they are displayed correctly as it should be.

Precision: TP / (FP + TP) Recall: TP / (FN + TP)

As I show the precision and recall, it would be done the same thing, and verified manually that these are the correct percentages.

```r
# show the curve on the performance
perf1 <- performance(pred1, "tpr", "fpr")
plot(perf1, lty = 1)
```




```r
# Now we check on what acceptable ways we could do for regression
# doing single decision tree
model_dtree1 <- rpart(formula_ISAcceptance, method="anova",data = univ_train)
pred_dtree1 <- predict(model_dtree1, newdata = univ_test)
accu1 = abs(pred_dtree1 - univ_test$ACCEPTED) < 0.5
frac1 = sum(accu1)/length(accu1)
print(frac1)
```

```
## [1] 0.8794727
```

```r
# doing random forest
model_forest1 <- randomForest(formula_ISAcceptance, data = univ_train)
```

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

```r
pred_forest1 <- predict(model_forest1, newdata = univ_test)
accu2 <- abs(pred_forest1 - univ_test$ACCEPTED) < 0.5
frac2 <- sum(accu2)/length(accu2)
print(frac2)
```

```
## [1] 0.9435028
```

```r
# doing support vector machine
model_svm1 <- svm(formula_ISAcceptance, data = univ_train)
pred_svm1 <- predict(model_svm1, newdata = univ_test)
accu3 <- abs(pred_svm1 - univ_test$ACCEPTED) < 0.5
frac3 <- sum(accu3)/length(accu3)
print(frac3)
```

```
## [1] 0.893597
```

```r
# doing simple tree
model_tree1 <- tree(formula_ISAcceptance, data = univ_train)
pred_tree1 <- predict(model_tree1, newdata = univ_test)
accu4 <- abs(pred_tree1 - univ_test$ACCEPTED) < 0.5
frac4 <- sum(accu4)/length(accu4)
print(frac4)
```

```
## [1] 0.8794727
```

```r
# doing conditional inference tree
model_party1 <- ctree(formula_ISAcceptance, data = univ_train)
pred_party1 <- predict(model_party1, newdata = univ_test)
accu5 <- abs(pred_party1 - univ_test$ACCEPTED) < 0.5
frac5 <- sum(accu5)/length(accu5)
print(frac5)
```

```
## [1] 0.8954802
```

Based on the run, random forest is the best regression method to use in this model.

Next, another formula is created. This is an acceptance model for an international student that wants to take up Science degree/major

```r
# create a formula for the US research university acceptance model for International Students taking up
formula_ISSciAcceptance <- formula(ACCEPTED ~ REGION + ADM_RATE_ALL + SAT_AVG_ALL + PCIP11 + PCIP12 + P

# do a logistic regression model based on the formula created
glm_ISSciAcceptance <- glm(formula_ISSciAcceptance, data=univ_train,family=binomial())
summary(glm_ISSciAcceptance)
```

```
##
## Call:
## glm(formula = formula_ISSciAcceptance, family = binomial(), data = univ_train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.54017  -0.48246  -0.25342  -0.08276   3.11831
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.737e+01  1.426e+00 -12.175  < 2e-16 ***
## REGION        1.409e-01  3.144e-02   4.482 7.41e-06 ***
```

```
## ADM_RATE_ALL  7.964e-01  4.187e-01   1.902 0.057152 .
## SAT_AVG_ALL   1.509e-02  1.003e-03  15.046  < 2e-16 ***
## PCIP11        6.826e-01  1.995e+00   0.342 0.732183
## PCIP12       -8.308e+00  2.280e+01  -0.364 0.715535
## PCIP14        5.624e+00  7.713e-01   7.291 3.07e-13 ***
## PCIP15        1.393e+00  2.407e+00   0.579 0.562919
## PCIP24       -5.601e+00  1.250e+00  -4.480 7.45e-06 ***
## PCIP26        7.075e+00  1.730e+00   4.089 4.33e-05 ***
## PCIP27       -2.435e+01  6.616e+00  -3.680 0.000233 ***
## PCIP40       -3.127e+01  4.750e+00  -6.584 4.58e-11 ***
## PCIP45        8.659e+00  1.211e+00   7.149 8.75e-13 ***
## PCIP51        2.233e+00  5.982e-01   3.733 0.000189 ***
## PCIP52        1.237e+00  6.635e-01   1.865 0.062219 .
## UGDS_NRA      8.945e+00  1.439e+00   6.214 5.15e-10 ***
## UGDS_UNKN    -1.291e+00  1.583e+00  -0.816 0.414748
## COSTT4_A     -1.103e-04  7.193e-06 -15.338  < 2e-16 ***
## PCTFLOAN     -4.254e-01  5.594e-01  -0.760 0.446999
## UGDS_WOMEN    6.850e-01  7.917e-01   0.865 0.386893
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3139.9  on 3184  degrees of freedom
## Residual deviance: 1923.3  on 3165  degrees of freedom
## AIC: 1963.3
##
## Number of Fisher Scoring iterations: 6
```

```r
# do the testing with the prediction model
accepted_ind2 <- predict(glm_ISSciAcceptance, type="response", newdata = univ_test)
pred2 <- prediction(accepted_ind2, univ_test$ACCEPTED)

# prepare confusion matrix and accuracy to see the scores
c2 <- confusionMatrix(as.integer(accepted_ind2 > 0.5), univ_test$ACCEPTED)
c2$table
```

```
##           Reference
## Prediction   0   1
##          0 828  85
##          1  39 110
```

```r
c2$overall['Accuracy']
```

```
##  Accuracy
## 0.8832392
```

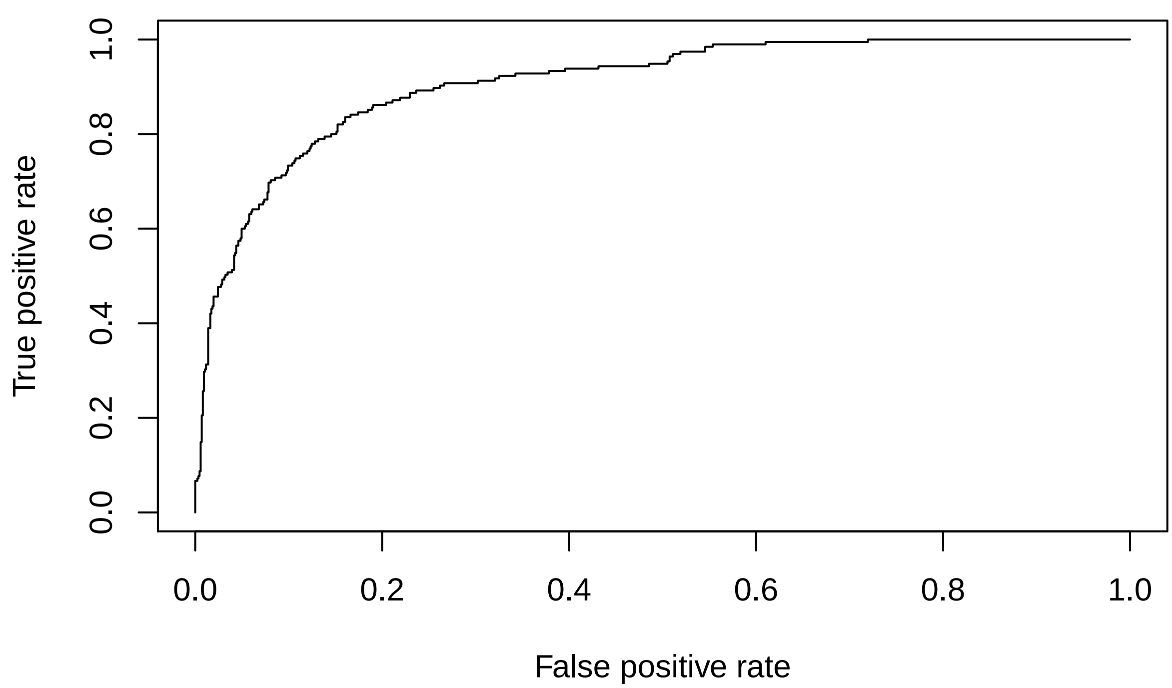
```r
#Precision of the logistic regression model
c2$byClass['Neg Pred Value']
```

```
## Neg Pred Value
##       0.738255
```

```r
#Recall of the logistic regression model
c2$byClass['Specificity']
```

```
## Specificity
##   0.5641026
```

```r
# show the curve on the performance
perf2 <- performance(pred2,"tpr","fpr")
plot(perf2, lty = 1)
```




```r
# Now we check on what acceptable ways we could do for regression
# doing single decision tree
model_dtree2 <- rpart(formula_ISSciAcceptance, method="anova",data = univ_train)
pred_dtree2 <- predict(model_dtree2, newdata = univ_test)
accu6 <- abs(pred_dtree2 - univ_test$ACCEPTED) < 0.5
frac6 <- sum(accu6)/length(accu6)
print(frac6)
```

```
## [1] 0.9048964
```

```r
# doing random forest
model_forest2 <- randomForest(formula_ISSciAcceptance, data = univ_train)
```

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

```r
pred_forest2 <- predict(model_forest2, newdata = univ_test)
accu7 <- abs(pred_forest2 - univ_test$ACCEPTED) < 0.5
frac7 <- sum(accu7)/length(accu7)
print(frac7)
```

```
## [1] 0.9576271
```

```r
# doing support vector machine
model_svm2 <- svm(formula_ISSciAcceptance, data = univ_train)
pred_svm2 <- predict(model_svm2, newdata = univ_test)
accu8 <- abs(pred_svm2 - univ_test$ACCEPTED) < 0.5
frac8 <- sum(accu8)/length(accu8)
print(frac8)
```

```
## [1] 0.9256121
```

```r
# doing simple tree
model_tree2 <- tree(formula_ISSciAcceptance, data = univ_train)
pred_tree2 <- predict(model_tree2, newdata = univ_test)
accu9 <- abs(pred_tree2 - univ_test$ACCEPTED) < 0.5
frac9 <- sum(accu9)/length(accu9)
print(frac9)
```

```
## [1] 0.9048964
```

```r
# doing conditional inference tree
model_party2 <- ctree(formula_ISSciAcceptance, data = univ_train)
pred_party2 <- predict(model_party2, newdata = univ_test)
accu10 <- abs(pred_party2 - univ_test$ACCEPTED) < 0.5
frac10 <- sum(accu10)/length(accu10)
print(frac10)
```

```
## [1] 0.9039548
```

Based on this, random forest is the best regression method to use.

In this project, I have selected a couple of variables that we could use in this model. However, we could use more than a few variables to get the optimal result.

With this in mind, feature selection is very essential, especially with datasets that have many variables for model selection. Although in this report, we have 1745 variables, and deduced it to 72 variables, we have to check which variables will be very useful in doing our research model.

In this portion, we will consider all variables, and use Boruta and RFE to use what variables we could use for doing a better outcome of the model.

Boruta is a package created was written by Miron B. Kursa and Witold R. Rudnicki to use an all relevant feature selection wrapper algorithm. According to their description, it "finds relevant features by comparing original attributes' importance with importance achievable at random, estimated using their permuted copies". (Source: https://cran.r-project.org/web/packages/Boruta/Boruta.pdf)

The Recursive Feature Elimination, or RFE, is a function in R's Caret package that uses the random forest algorithm to evaluate the attributes needed to be able to get an optimal result in the data that we have. (Source: http://machinelearningmastery.com/feature-selection-with-the-caret-r-package/)

Now, we will be doing some feature eliminations using Boruta and RFE.

```r
# First, we will create another copy of the dataset
usunivnoccbasic <- usunivfilter

# Next, we will change those that have "NA" to 0, since there is no data in it
usunivnoccbasic[usunivnoccbasic == "NA"] <- 0

# Next, we will choose rows that have complete cases
usunivnoccbasic <- usunivnoccbasic[complete.cases(usunivnoccbasic),]

# Now that we have the cleansed dataset, we will implement Boruta
boruta.train <- Boruta(ACCEPTED ~ .-CCBASIC2, data=usunivnoccbasic)
print(boruta.train)
```

```
## Boruta performed 99 iterations in 26.637 secs.
##  60 attributes confirmed important: ADM_RATE, ADM_RATE_ALL,
## C150_4, C150_4_2MOR, C150_4_AIAN and 55 more.
##  7 attributes confirmed unimportant: C150_4_NHPI, PCIP12, PCIP25,
## PCIP29, PCIP46 and 2 more.
##  3 tentative attributes left: PCIP10, PCIP22, PCIP41.
```

```r
getSelectedAttributes(boruta.train)
```

```
##  [1] "REGION"            "ADM_RATE"          "ADM_RATE_ALL"
##  [4] "SAT_AVG_ALL"       "PCIP01"            "PCIP03"
##  [7] "PCIP04"            "PCIP05"            "PCIP09"
## [10] "PCIP11"            "PCIP13"            "PCIP14"
## [13] "PCIP15"            "PCIP16"            "PCIP19"
## [16] "PCIP23"            "PCIP24"            "PCIP26"
## [19] "PCIP27"            "PCIP30"            "PCIP31"
## [22] "PCIP38"            "PCIP39"            "PCIP40"
## [25] "PCIP42"            "PCIP43"            "PCIP44"
## [28] "PCIP45"            "PCIP49"            "PCIP50"
## [31] "PCIP51"            "PCIP52"            "PCIP54"
## [34] "UGDS_WHITE"        "UGDS_BLACK"        "UGDS_HISP"
## [37] "UGDS_ASIAN"        "UGDS_AIAN"         "UGDS_NHPI"
## [40] "UGDS_2MOR"         "UGDS_NRA"          "UGDS_UNKN"
## [43] "PPTUG_EF"          "COSTT4_A"          "TUITIONFEE_IN"
## [46] "TUITIONFEE_OUT"    "C150_4"            "C150_4_WHITE"
## [49] "C150_4_BLACK"      "C150_4_HISP"       "C150_4_ASIAN"
## [52] "C150_4_AIAN"       "C150_4_2MOR"       "C150_4_NRA"
## [55] "C150_4_UNKN"       "RET_FT4"           "PCTFLOAN"
## [58] "PAR_ED_PCT_1STGEN" "UGDS_MEN"          "UGDS_WOMEN"
```

```r
# We will print the stats of the variables that would be accepted or not
stats <- attStats(boruta.train)
print(stats)
```

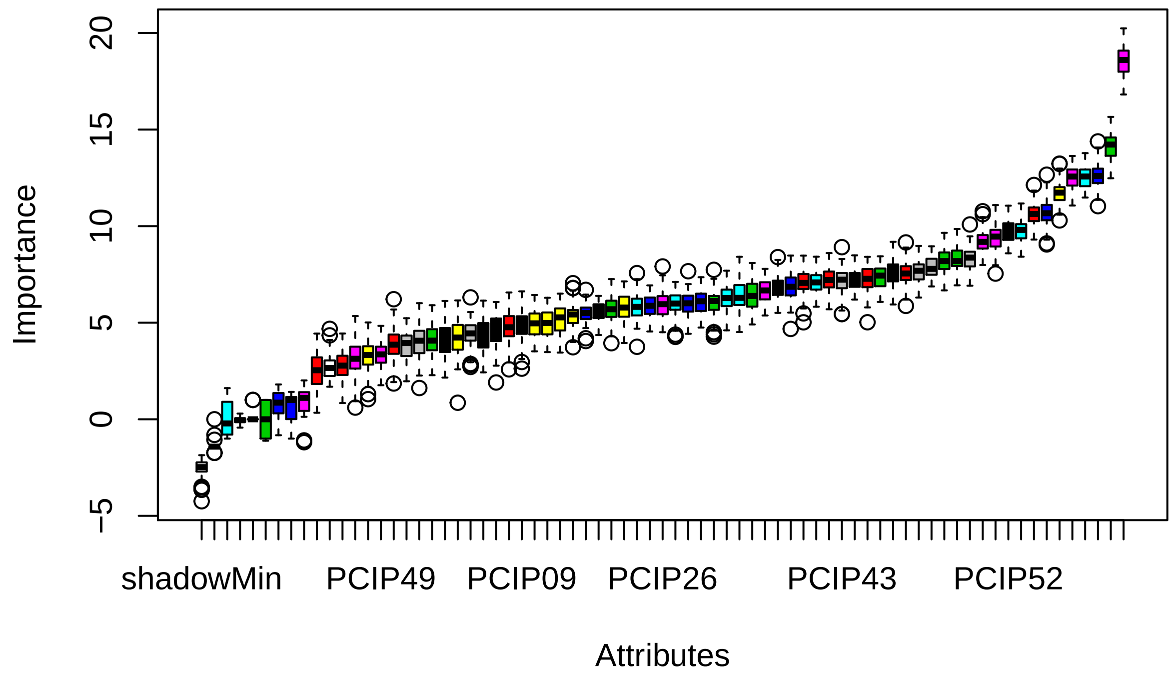
```
##                        meanImp  medianImp     minImp    maxImp  normHits
## REGION             5.580290593  5.6412714  4.3621495  6.388653 1.0000000
## ADM_RATE           7.283779624  7.2719062  5.0233636  8.409523 1.0000000
## ADM_RATE_ALL       7.349433811  7.4380978  6.0784446  8.438585 1.0000000
## SAT_AVG_ALL       12.611747570 12.5937686 11.0348273 14.388176 1.0000000
```

```
## PCIP01             6.260765023  6.2797128  4.6042371  7.692672 1.0000000
## PCIP03             6.672630700  6.6704563  5.3734977  7.786983 1.0000000
## PCIP04            11.721833066 11.7350225 10.2960630 13.236676 1.0000000
## PCIP05             8.316278726  8.3705447  6.9151158 10.087284 1.0000000
## PCIP09             4.828559520  4.9067098  2.6209154  6.623441 0.9797980
## PCIP10             2.766914059  2.7779827  0.8391203  4.440021 0.5454545
## PCIP11             6.413811947  6.3867856  4.9101282  8.091952 1.0000000
## PCIP12             0.853933102  0.8601440 -0.8261891  1.803542 0.0000000
## PCIP13             5.994607507  5.9905457  4.2642988  7.111917 1.0000000
## PCIP14            18.587539241 18.6057063 16.8171862 20.241704 1.0000000
## PCIP15             4.976447599  4.9847167  3.4779331  6.290287 0.9797980
## PCIP16             7.642973079  7.6884167  6.3013722  8.976329 1.0000000
## PCIP19             7.576041330  7.5275058  5.9439577  9.187353 1.0000000
## PCIP22             2.469843506  2.5389380  0.3360699  4.436063 0.3737374
## PCIP23             8.210588000  8.1900357  6.6742308  9.652165 1.0000000
## PCIP24             5.845144140  5.8742855  4.5453803  6.932079 1.0000000
## PCIP25            -1.307734762 -1.4169985 -1.7372501  0.000000 0.0000000
## PCIP26             5.925499129  5.9602279  4.5080171  7.913417 1.0000000
## PCIP27             5.165067650  5.2757789  3.4532610  6.504006 0.9898990
## PCIP29             0.154000231  0.0000000  0.0000000  1.001002 0.0000000
## PCIP30             4.160665729  4.1586651  2.1584809  6.127716 0.8989899
## PCIP31             4.797160057  4.7667441  2.5848583  6.564904 0.9595960
## PCIP38             4.134846289  4.0722433  2.2839887  5.906849 0.9090909
## PCIP39             5.489026292  5.5061392  4.0504527  6.700783 1.0000000
## PCIP40             5.797418167  5.8191872  3.7521265  7.568641 1.0000000
## PCIP41             3.105232998  3.1315796  0.6038768  5.344076 0.6161616
## PCIP42             4.955901609  4.9674919  3.5180368  6.435046 1.0000000
## PCIP43             7.177823940  7.2273608  5.4488360  8.912202 1.0000000
## PCIP44             4.571082948  4.5346569  1.9051171  6.073749 0.9494949
## PCIP45             7.545300416  7.5477453  5.8710765  9.157623 1.0000000
## PCIP46            -0.010852581  0.0000000 -1.1119656  1.001002 0.0000000
## PCIP47             0.720386652  1.0010015 -1.0010015  1.417051 0.0000000
## PCIP48            -0.004272703 -0.2158142 -1.0010015  1.613016 0.0000000
## PCIP49             3.360756712  3.3611401  1.7636617  4.835916 0.7878788
## PCIP50             5.734949236  5.7842447  3.9503136  7.140312 0.9898990
## PCIP51             3.995621875  4.0659655  1.6180637  6.013355 0.9090909
## PCIP52             9.744118264  9.7266522  8.5939382 11.060330 1.0000000
## PCIP54             3.867001927  3.8615735  1.8524449  6.217096 0.8383838
## UGDS_WHITE         8.290457397  8.1911957  6.9366350  9.851062 1.0000000
## UGDS_BLACK        10.721327409 10.6690750  9.0517804 12.664372 1.0000000
## UGDS_HISP          6.433673727  6.2862672  4.5163873  8.412740 1.0000000
## UGDS_ASIAN         9.228954747  9.1852910  7.9866895 10.771389 1.0000000
## UGDS_AIAN          4.245586008  4.2316385  0.8552907  6.153439 0.9191919
## UGDS_NHPI          3.809763594  3.9460035  1.9691965  5.232433 0.8484848
## UGDS_2MOR          4.356671306  4.4977196  2.4299560  6.142771 0.8888889
## UGDS_NRA           7.226087134  7.2096676  5.6963223  8.604983 1.0000000
## UGDS_UNKN          6.031270312  6.1244052  4.2777035  7.739699 1.0000000
## PPTUG_EF           6.837143901  6.8589840  4.6783847  8.475818 1.0000000
## COSTT4_A           9.745600351  9.8046399  8.4170412 11.174214 1.0000000
## TUITIONFEE_IN      9.409257151  9.4590922  7.5422875 11.089863 1.0000000
## TUITIONFEE_OUT     5.417446827  5.4264449  3.7289982  7.047509 1.0000000
## C150_4             7.858751736  7.7875895  6.8771352  8.956235 1.0000000
## C150_4_WHITE       6.821981196  6.7995202  5.5090458  8.395923 1.0000000
## C150_4_BLACK       7.083497423  7.0606669  5.0204832  8.473994 1.0000000
```

```
## C150_4_HISP        5.714078469  5.6980160  3.9395217  7.258872 1.0000000
## C150_4_ASIAN       5.983549995  6.0065988  4.4246711  7.666465 1.0000000
## C150_4_AIAN        7.081581937  7.0874377  5.8264177  8.416640 1.0000000
## C150_4_NHPI        0.786788660  1.1076062 -1.1882131  2.012278 0.0000000
## C150_4_2MOR        3.277041311  3.3290509  1.0415176  5.013939 0.7070707
## C150_4_NRA         4.437597337  4.4473455  2.7083456  6.311558 0.9696970
## C150_4_UNKN        7.208714541  7.2319203  6.1964527  8.488720 1.0000000
## RET_FT4           10.616119968 10.6301183  9.3049553 12.131738 1.0000000
## PCTFLOAN          14.128964597 14.2251125 12.4812453 15.656010 1.0000000
## PAR_ED_PCT_1STGEN  6.055710220  6.1126321  4.7485628  7.356421 1.0000000
## UGDS_MEN          12.535087815 12.5722380 11.4805721 13.779525 1.0000000
## UGDS_WOMEN        12.517172109 12.5688775 11.0683271 13.632081 1.0000000
##                    decision
## REGION            Confirmed
## ADM_RATE          Confirmed
## ADM_RATE_ALL      Confirmed
## SAT_AVG_ALL       Confirmed
## PCIP01            Confirmed
## PCIP03            Confirmed
## PCIP04            Confirmed
## PCIP05            Confirmed
## PCIP09            Confirmed
## PCIP10            Tentative
## PCIP11            Confirmed
## PCIP12             Rejected
## PCIP13            Confirmed
## PCIP14            Confirmed
## PCIP15            Confirmed
## PCIP16            Confirmed
## PCIP19            Confirmed
## PCIP22            Tentative
## PCIP23            Confirmed
## PCIP24            Confirmed
## PCIP25             Rejected
## PCIP26            Confirmed
## PCIP27            Confirmed
## PCIP29             Rejected
## PCIP30            Confirmed
## PCIP31            Confirmed
## PCIP38            Confirmed
## PCIP39            Confirmed
## PCIP40            Confirmed
## PCIP41            Tentative
## PCIP42            Confirmed
## PCIP43            Confirmed
## PCIP44            Confirmed
## PCIP45            Confirmed
## PCIP46             Rejected
## PCIP47             Rejected
## PCIP48             Rejected
## PCIP49            Confirmed
## PCIP50            Confirmed
## PCIP51            Confirmed
## PCIP52            Confirmed
```

```
## PCIP54             Confirmed
## UGDS_WHITE         Confirmed
## UGDS_BLACK         Confirmed
## UGDS_HISP          Confirmed
## UGDS_ASIAN         Confirmed
## UGDS_AIAN          Confirmed
## UGDS_NHPI          Confirmed
## UGDS_2MOR          Confirmed
## UGDS_NRA           Confirmed
## UGDS_UNKN          Confirmed
## PPTUG_EF           Confirmed
## COSTT4_A           Confirmed
## TUITIONFEE_IN      Confirmed
## TUITIONFEE_OUT     Confirmed
## C150_4             Confirmed
## C150_4_WHITE       Confirmed
## C150_4_BLACK       Confirmed
## C150_4_HISP        Confirmed
## C150_4_ASIAN       Confirmed
## C150_4_AIAN        Confirmed
## C150_4_NHPI         Rejected
## C150_4_2MOR        Confirmed
## C150_4_NRA         Confirmed
## C150_4_UNKN        Confirmed
## RET_FT4            Confirmed
## PCTFLOAN           Confirmed
## PAR_ED_PCT_1STGEN Confirmed
## UGDS_MEN           Confirmed
## UGDS_WOMEN         Confirmed
```

```r
# We will plot on the number of variables and its importance for Boruta
plot(boruta.train, type = c("g","o"), cex = 1.0, col = 1:70)
```




```r
#Now, let us try RFE
rfe_control <- rfeControl(functions=rfFuncs, method="cv", number = 10)
rfe.train <- rfe(usunivnoccbasic[,1:70], usunivnoccbasic[,72], sizes = 1:70, rfeControl = rfe_control)
```

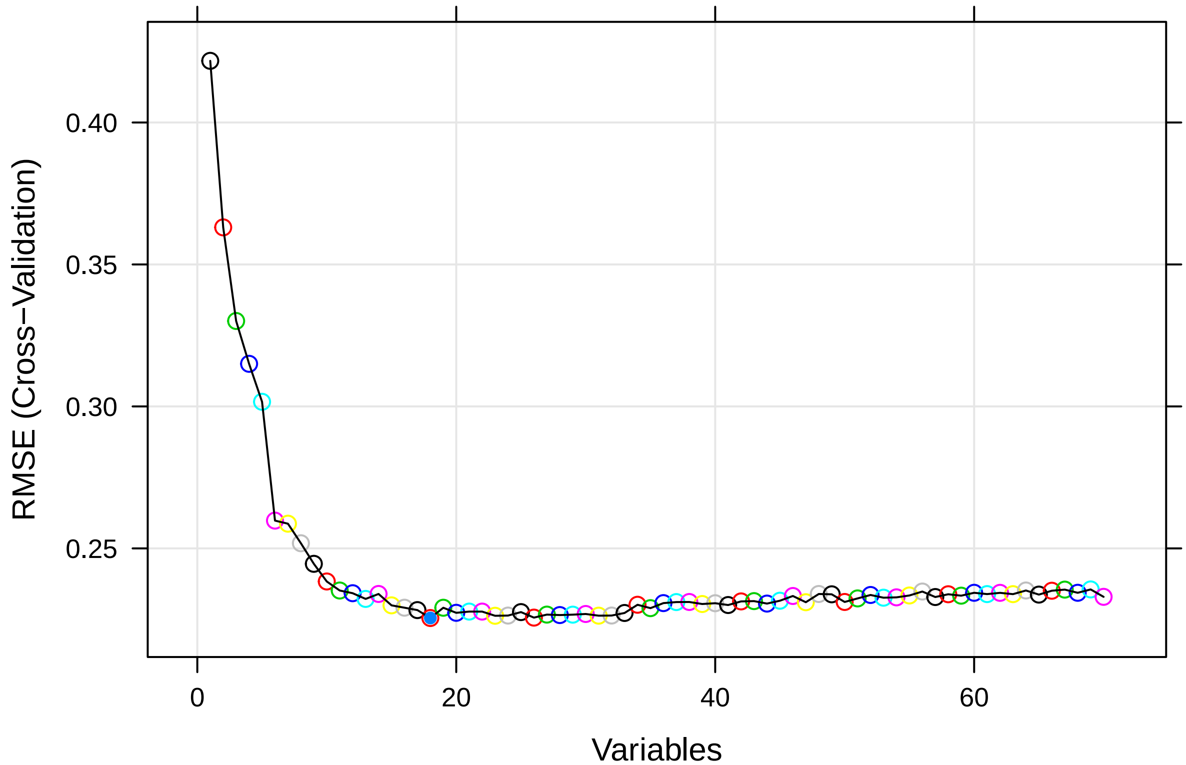
```
##
## Attaching package: 'plyr'

## The following object is masked from 'package:modeltools':
##
##     empty
```

```r
predictors(rfe.train)
```

```
##  [1] "PCIP14"        "PCTFLOAN"      "PCIP04"        "SAT_AVG_ALL"
##  [5] "PCIP52"        "UGDS_BLACK"    "UGDS_WOMEN"    "UGDS_MEN"
##  [9] "PCIP45"        "PCIP43"        "COSTT4_A"      "PCIP23"
## [13] "RET_FT4"       "TUITIONFEE_IN" "UGDS_HISP"     "PCIP39"
## [17] "PCIP16"        "C150_4_AIAN"
```

```r
# We will plot on the number of variables and its importance for RFE
plot(rfe.train, type = c("g","o"), cex = 1.0, col = 1:70)
```



Based on these runs, RFE determines fewer variables needed for the prediction model than Boruta. There would be some cases that the Boruta package could be used, depending on the number of variables.

US Research University Completion Rate Prediction Model

```
rm_train2 <- sample(nrow(usresearchuniv), floor(nrow(usresearchuniv)*0.75))
univ_train2 <- usresearchuniv[rm_train2,]
univ_test2 <- usresearchuniv[-rm_train2,]

formula_completionrate <- formula(C150_4_NRA ~ REGION + ADM_RATE_ALL + UGDS_NRA + PPTUG_EF + COSTT4_A +

# using multivariate linear regression to calculate the completion rate for international students
lm_NRAcompletion <- lm(formula_completionrate, data = univ_train2)
summary(lm_NRAcompletion)
```

```
##
## Call:
## lm(formula = formula_completionrate, data = univ_train2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.62379 -0.05936  0.01046  0.06935  0.50732
##
```

```
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          9.554e-01  4.476e-02  21.342  < 2e-16 ***
## REGION              -2.385e-03  3.321e-03  -0.718 0.473024
## ADM_RATE_ALL        -1.893e-01  3.893e-02  -4.863 1.48e-06 ***
## UGDS_NRA             2.597e-01  1.484e-01   1.750 0.080620 .
## PPTUG_EF            -2.890e-01  8.684e-02  -3.328 0.000928 ***
## COSTT4_A             1.231e-06  6.219e-07   1.980 0.048164 *
## PCTFLOAN            -3.398e-01  5.829e-02  -5.829 9.07e-09 ***
## PAR_ED_PCT_1STGEN -1.249e-01  9.895e-02  -1.263 0.207162
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1403 on 603 degrees of freedom
## Multiple R-squared:  0.4293, Adjusted R-squared:  0.4227
## F-statistic: 64.81 on 7 and 603 DF,  p-value: < 2.2e-16
```

```
# do the testing with the prediction model
accepted_ind3 <- predict(lm_NRAcompletion, interval="prediction", newdata = univ_test2)

# Checking on PRED(25)
errors <- accepted_ind3[,"fit"] - univ_test2$C150_4_NRA
rel_change <- abs(errors) / univ_test2$C150_4_NRA
table(rel_change<0.25)["TRUE"] / nrow(univ_test2)
```

```
##      TRUE
## 0.8088235
```

```
# Now we check on what acceptable ways we could do for regression
# Doing single decision tree
model_dtree3 <- rpart(formula_completionrate, method="anova",data = univ_train2)
pred_dtree3 <- predict(model_dtree3, newdata = univ_test2)
accu11 <- abs(pred_dtree3 - univ_test2$C150_4_NRA) < 0.25
frac11 <- sum(accu11)/length(accu11)
print(frac11)
```

```
## [1] 0.8921569
```

```
# Doing random forest
model_forest3 <- randomForest(formula_completionrate, data = univ_train2)
pred_forest3 <- predict(model_forest3, newdata = univ_test2)
accu12 <- abs(pred_forest3 - univ_test2$C150_4_NRA) < 0.25
frac12 <- sum(accu12)/length(accu12)
print(frac12)
```

```
## [1] 0.9264706
```

```
# Doing support vector machine
model_svm3 <- svm(formula_completionrate, data = univ_train2)
pred_svm3 <- predict(model_svm3, newdata = univ_test2)
accu13 <- abs(pred_svm3 - univ_test2$C150_4_NRA) < 0.25
frac13 <- sum(accu13)/length(accu13)
print(frac13)
```

```
## [1] 0.9215686
```

```r
# doing simple tree
model_tree3 <- tree(formula_completionrate, data = univ_train2)
pred_tree3 <- predict(model_tree3, newdata = univ_test2)
accu14 <- abs(pred_tree3 - univ_test2$C150_4_NRA) < 0.25
frac14 <- sum(accu14)/length(accu14)
print(frac14)
```

```
## [1] 0.9019608
```

```r
# doing conditional inference tree
model_party3 <- ctree(formula_completionrate, data = univ_train2)
pred_party3 <- predict(model_party3, newdata = univ_test2)
accu15 <- abs(pred_party3 - univ_test2$C150_4_NRA) < 0.25
frac15 <- sum(accu15)/length(accu15)
print(frac15)
```

```
## [1] 0.9019608
```

From the regressions that we have run, the random forest is the best regression model to use for determining completion rates for international students.