US Research University Prediction Model

Philip Gabriel Andrada

November 02, 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\\Philip\\Desktop\\Capstone\\MERGED2010_11_PP.csv")
usuniv2011 <- read.csv("C:\\Users\\Philip\\Desktop\\Capstone\\MERGED2011_12_PP.csv")
usuniv2012 <- read.csv("C:\\Users\\Philip\\Desktop\\Capstone\\MERGED2012_13_PP.csv")
usuniv2013 <- read.csv("C:\\Users\\Philip\\Desktop\\Capstone\\MERGED2013_14_PP.csv")
usuniv2014 <- read.csv("C:\\Users\\Philip\\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)

#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 studnets 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
```

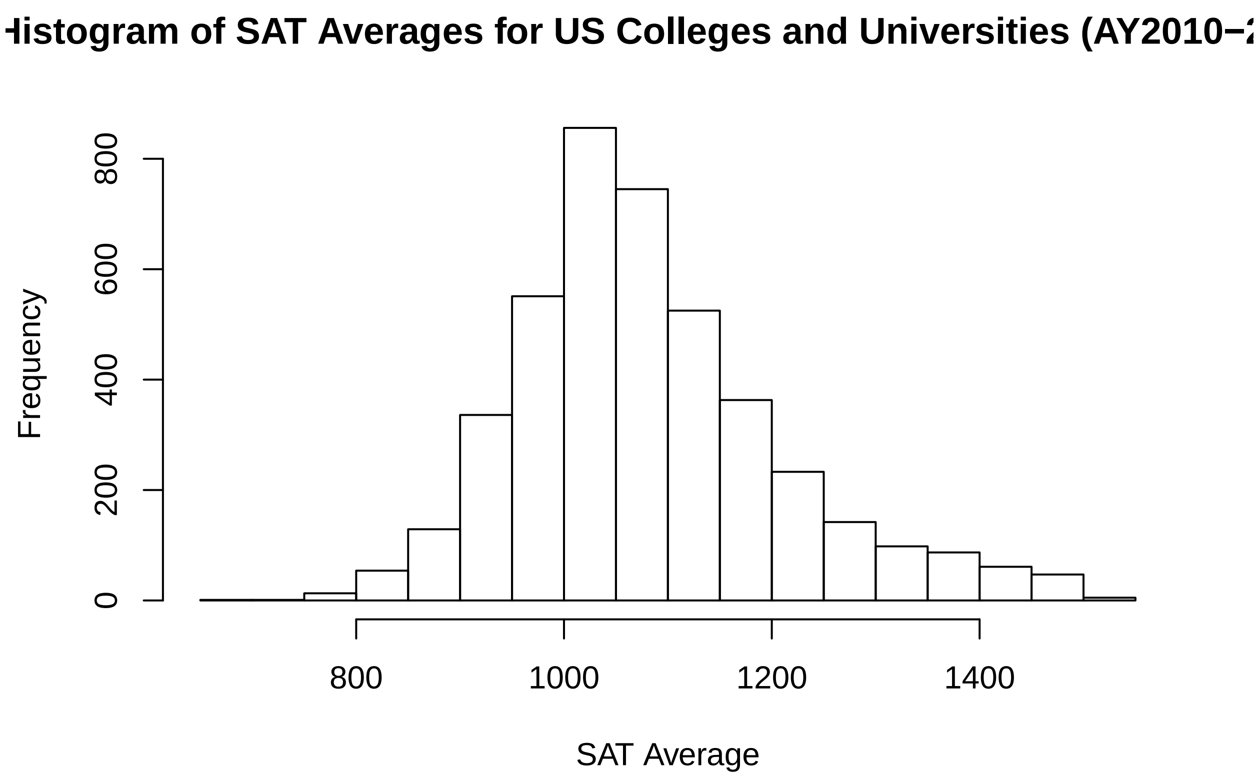
```
## Warning: NAs introduced by coercion

## 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
```

```r
# 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),]
```

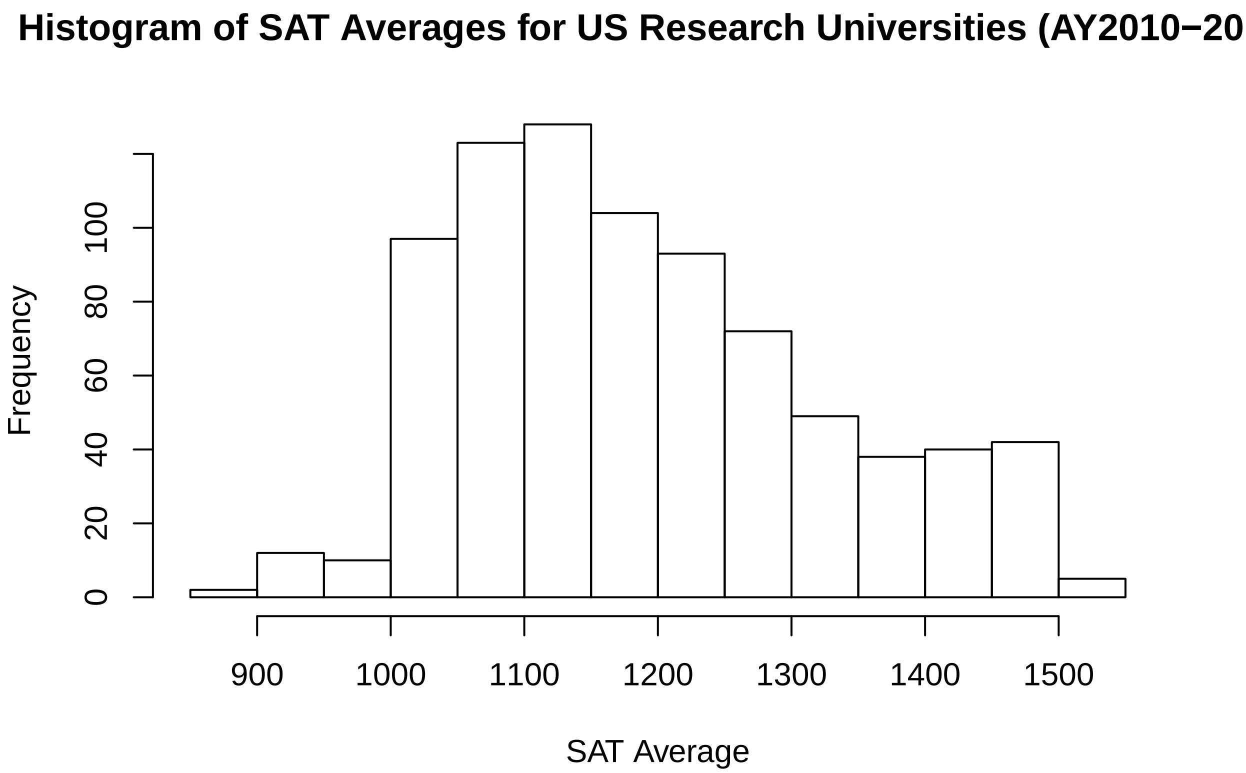
```r
usunivfilter <- usunivfilter[!is.na(usunivfilter$PCIP52),]
usunivfilter <- usunivfilter[!is.na(usunivfilter$PCTFLOAN),]
usunivfilter <- usunivfilter[!is.na(usunivfilter$PPTUG_EF),]
usunivfilter <- usunivfilter[!is.na(usunivfilter$RET_FT4),]
usunivfilter <- usunivfilter[!is.na(usunivfilter$PAR_ED_PCT_1STGEN),]

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

Distributions and Box and Whisker Plots

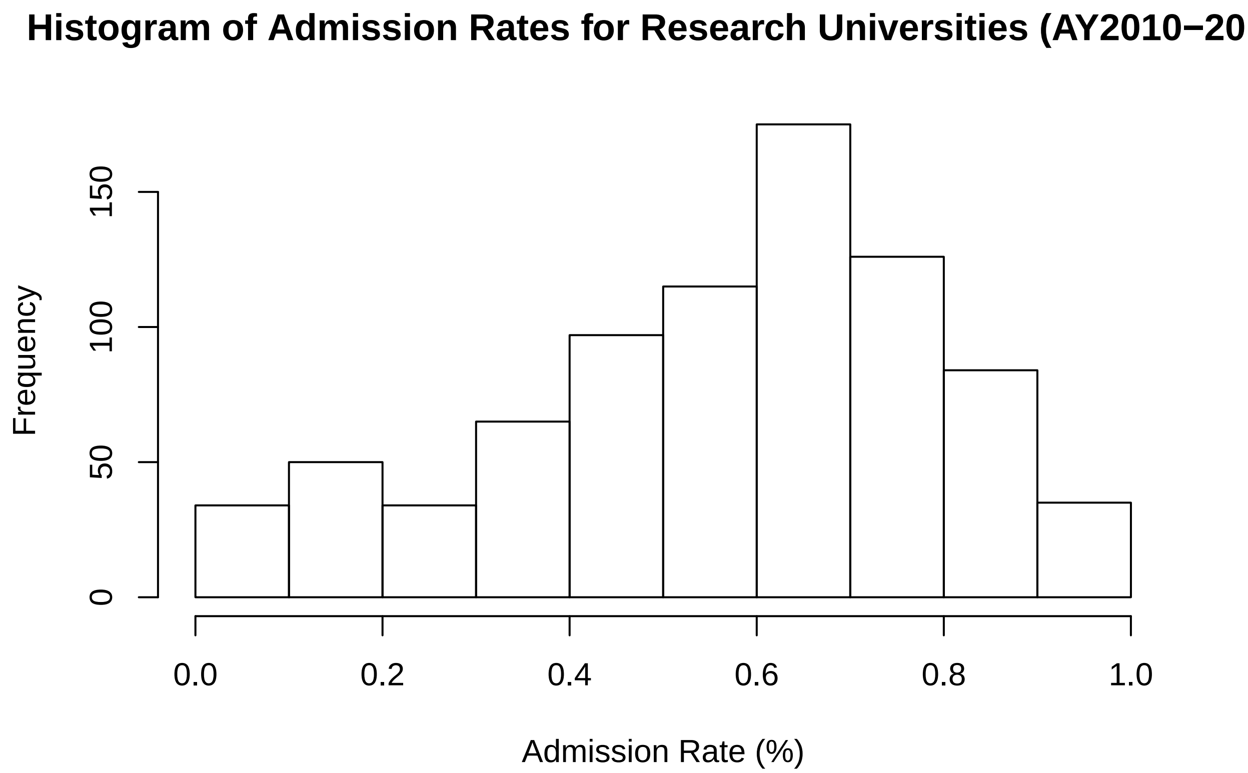
```r
# Histogram of SAT Averages for US Colleges and Universities
hist(usunivfilter$SAT_AVG_ALL, main = "Histogram of SAT Averages for US Colleges and Universities (AY20
```



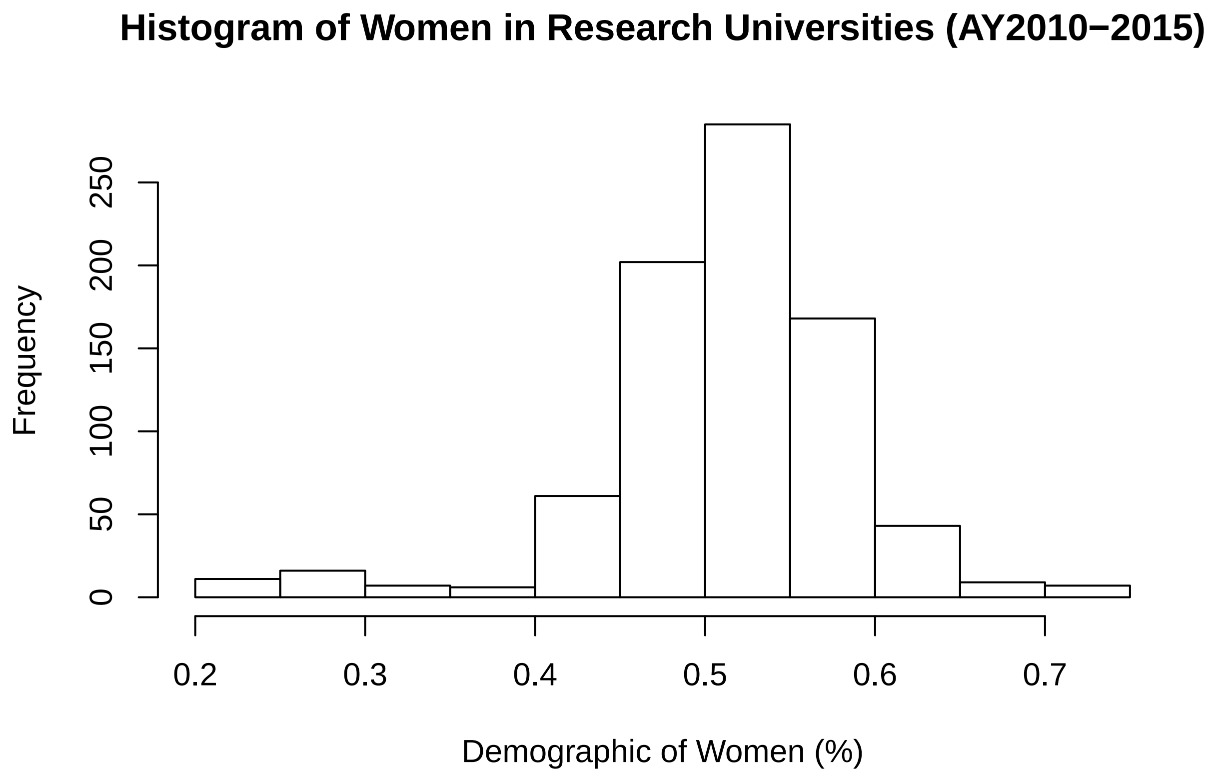

```r
# Histogram of SAT Averages for US Research Universities
hist(usresearchuniv$SAT_AVG_ALL, main = "Histogram of SAT Averages for US Research Universities (AY2010-
```



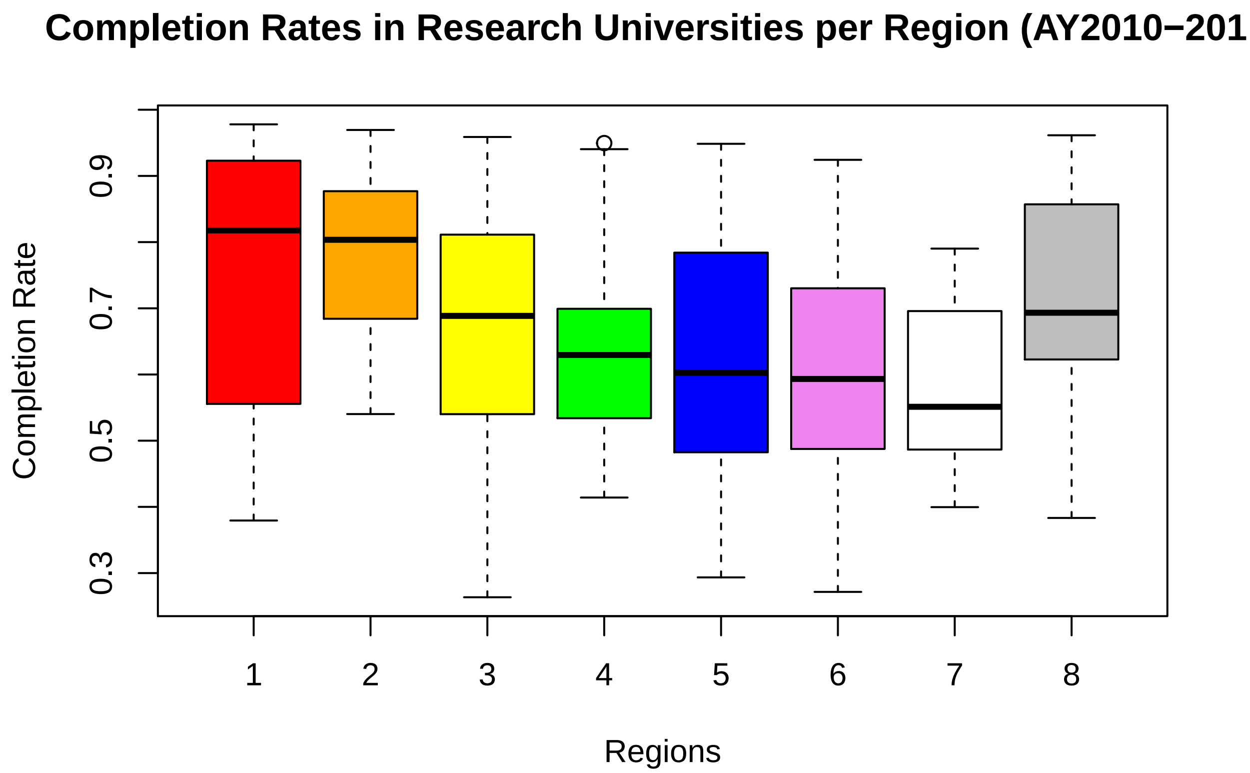

```r
# Histogram of Admission Rates for US Research Universities
hist(usresearchuniv$ADM_RATE_ALL, main = "Histogram of Admission Rates for Research Universities (AY201
```



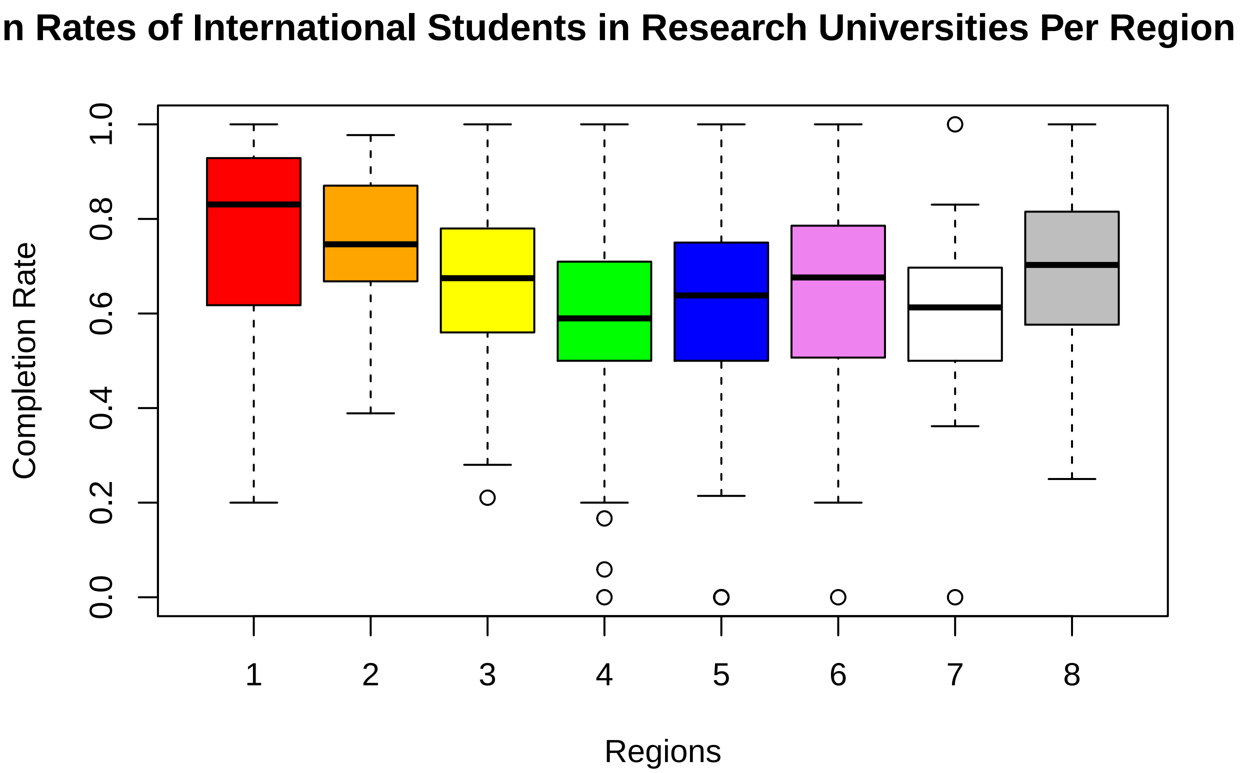

```
# Histogram of Women in US Research Universitie
hist(usresearchuniv$UGDS_WOMEN, main = "Histogram of Women in Research Universities (AY2010-2015)", xlal
```




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

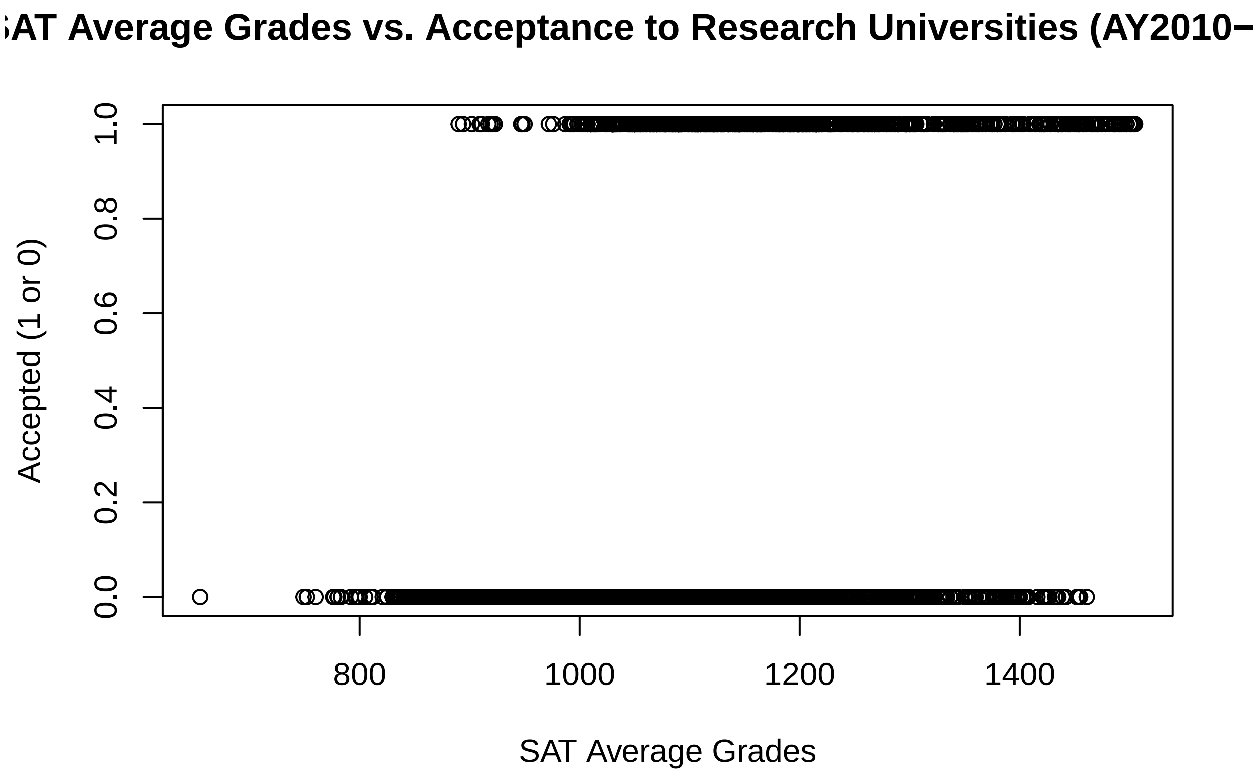



```
# Boxplot of COmpletion Rates of International Students per Region in US Research Universities
boxplot(C150_4_NRA ~ REGION, usresearchuniv, main = "Completion Rates of International Students in Resea
```

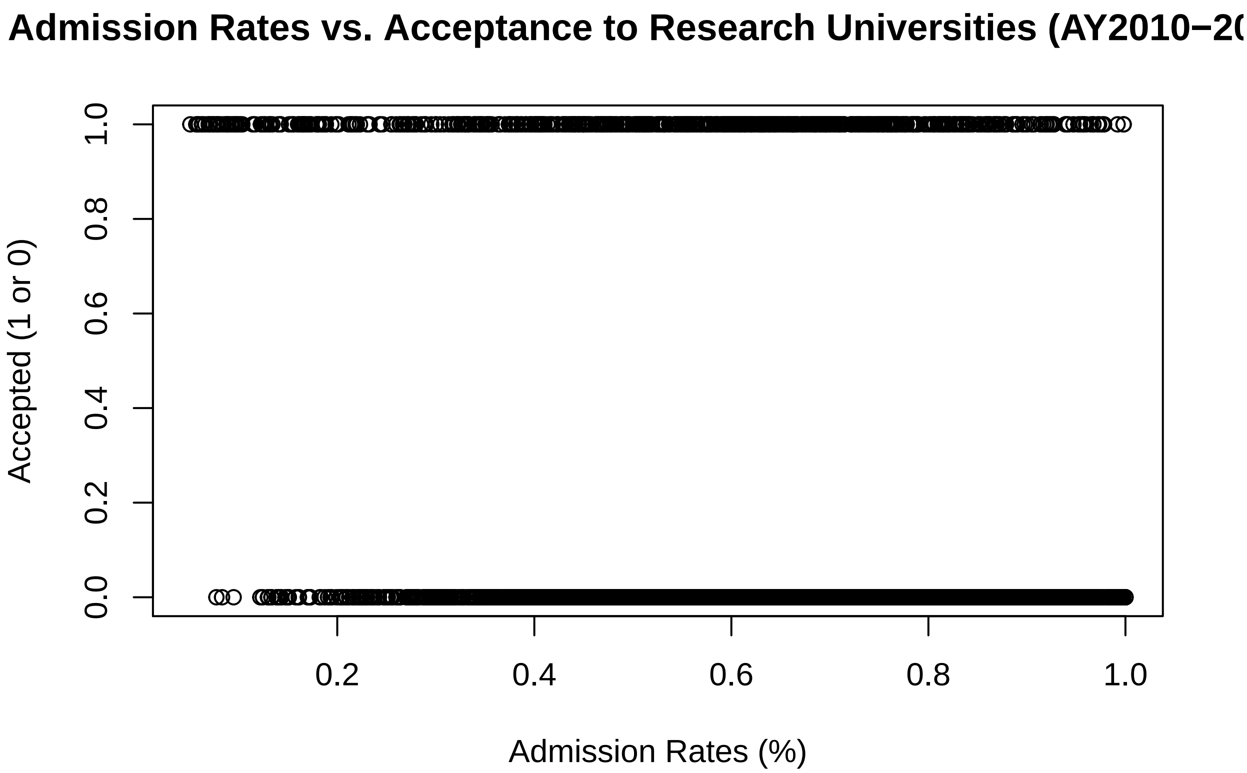


Correlations

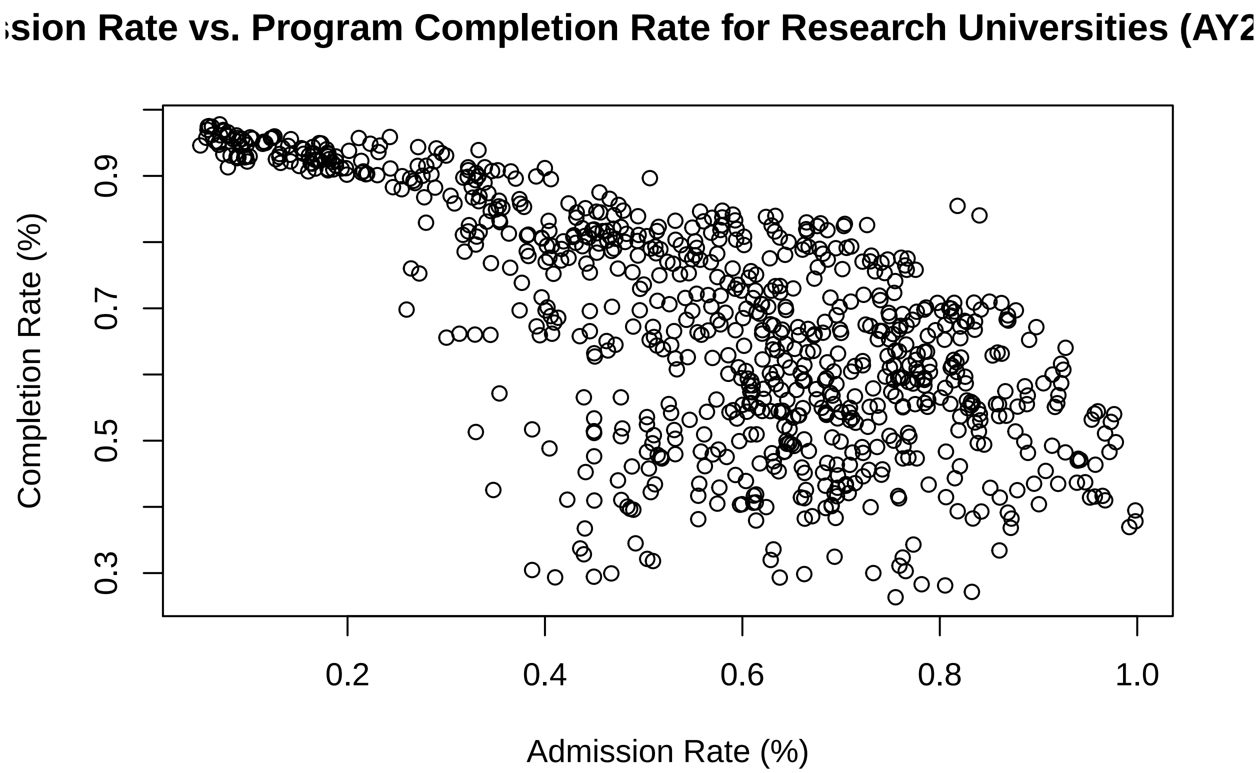
```
#Correlation between the SAT grades and the acceptance for the research universities
plot(usunivfilter$SAT_AVG_ALL, usunivfilter$ACCEPTED, main="SAT Average Grades vs. Acceptance to Researc
```




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

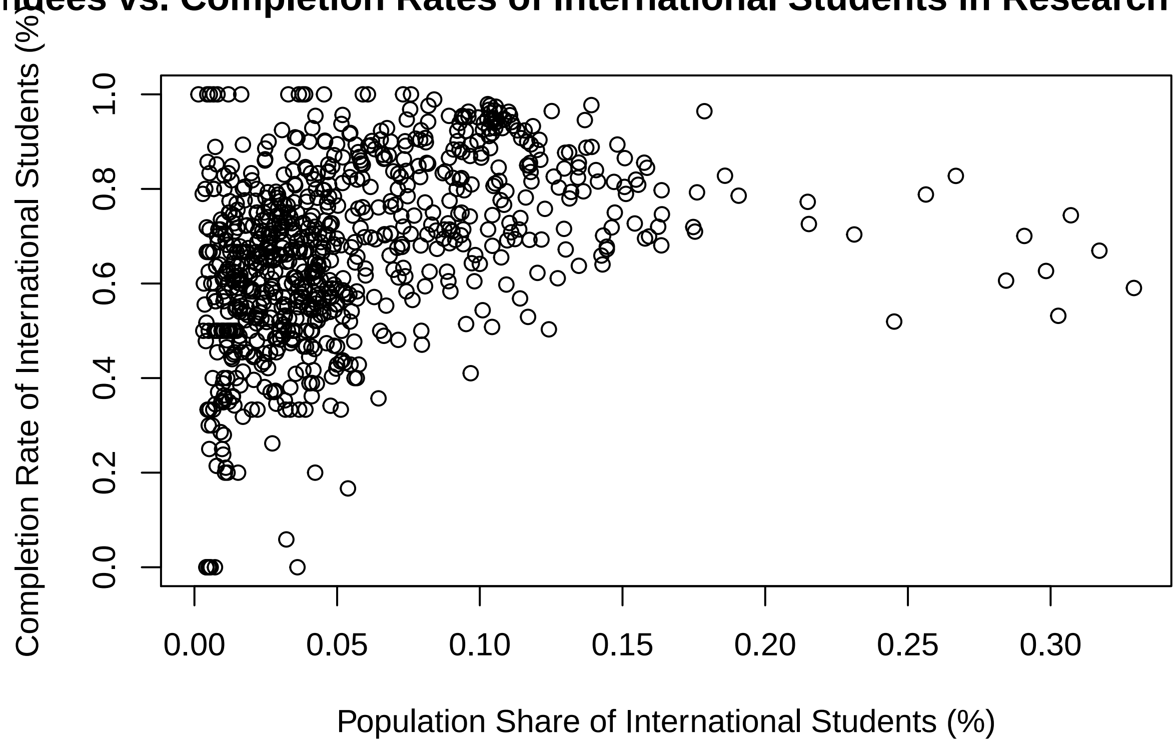



```
#Correlation between admission rate for research universities and program completion rate
plot(usresearchuniv$ADM_RATE_ALL, usresearchuniv$C150_4, main="Admission Rate vs. Program Completion Ra
```

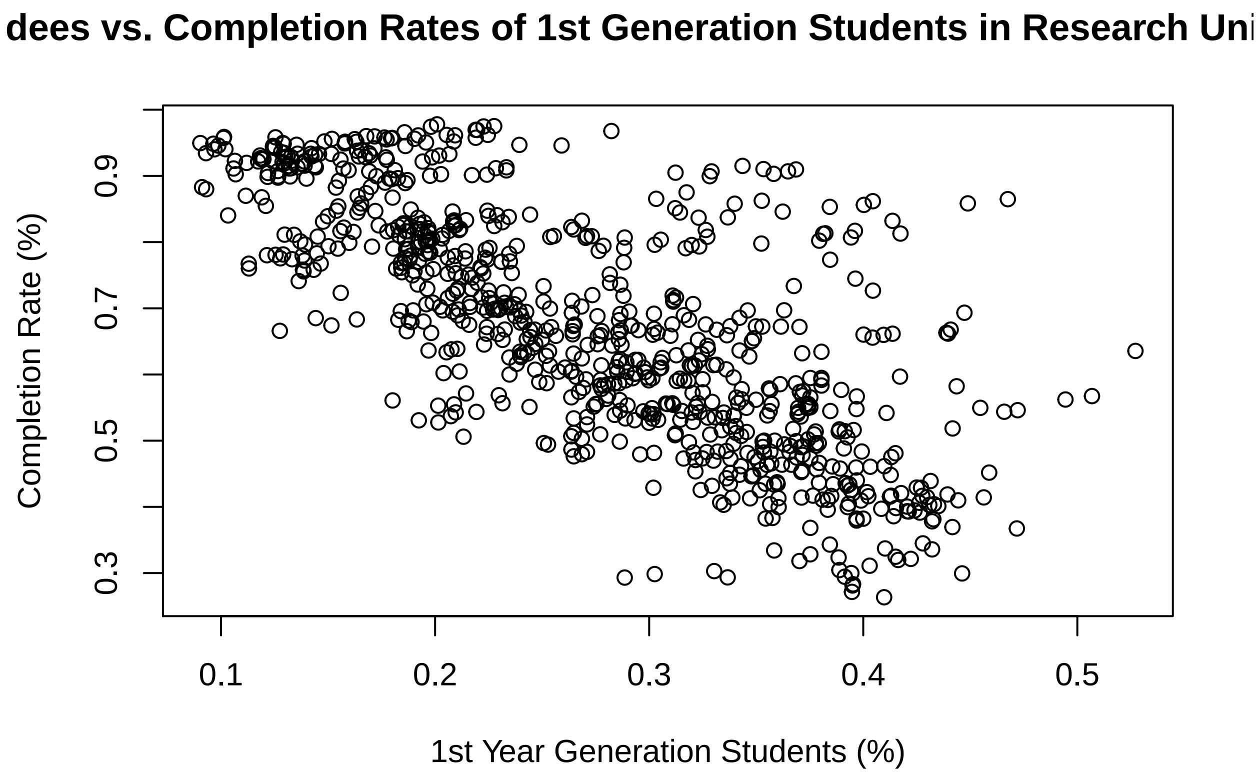



```
#Correlation between attendees and completion rate of non-resident aliens (International Students)
plot(usresearchuniv$UGDS_NRA, usresearchuniv$C150_4_NRA, main="Percentage of Attendees vs. Completion Ra
```

ndees vs. Completion Rates of International Students in Research Univ




```
#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
```



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 + P

# 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
## -2.1512  -0.5435  -0.3005  -0.1230   2.7694
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.454e+01  1.170e+00 -12.425  < 2e-16 ***
## REGION        1.072e-01  2.912e-02   3.679 0.000234 ***
## ADM_RATE_ALL  8.071e-01  3.794e-01   2.127 0.033393 *
## SAT_AVG_ALL   1.433e-02  8.229e-04  17.416  < 2e-16 ***
## UGDS_NRA      6.440e+00  1.276e+00   5.048 4.46e-07 ***
## COSTT4_A     -8.765e-05  6.093e-06 -14.386  < 2e-16 ***
## PCTFLOAN     -8.624e-01  4.805e-01  -1.795 0.072686 .
## UGDS_WOMEN   -1.863e+00  5.234e-01  -3.558 0.000373 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3134.2  on 3184  degrees of freedom
## Residual deviance: 2157.6  on 3177  degrees of freedom
## AIC: 2173.6
##
## Number of Fisher Scoring iterations: 6
```

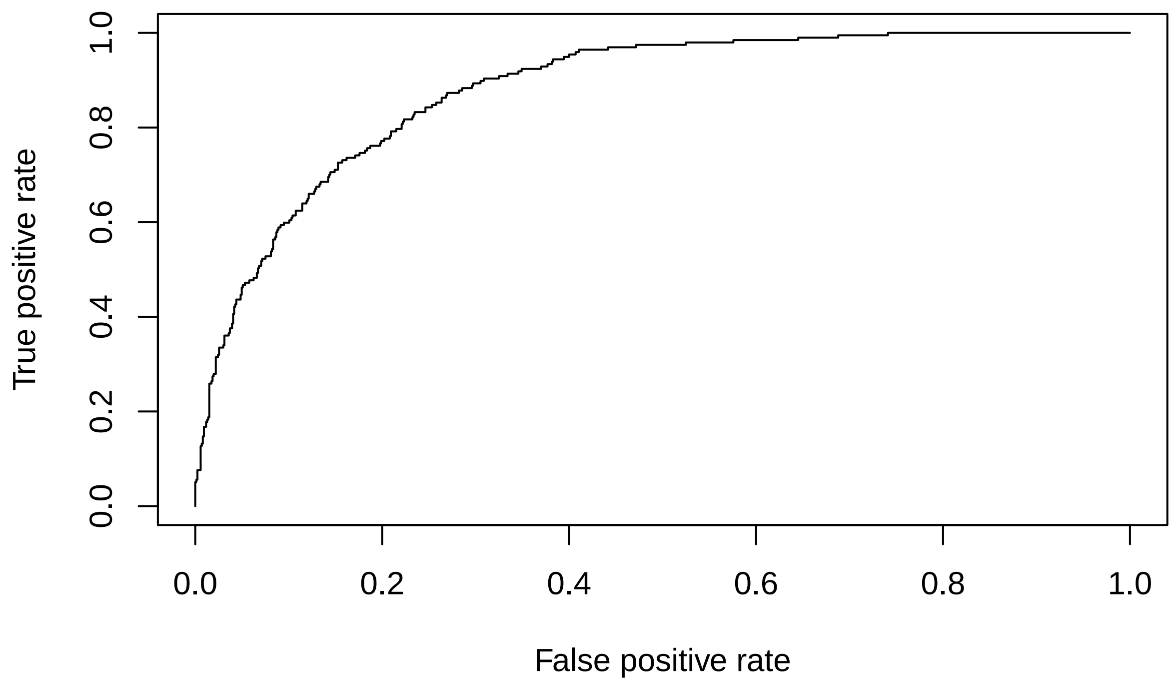
```r
# do the first testing with the prediction model
accepted_ind <- predict(glm_ISAcceptance, type="response", newdata = univ_test)
pred1 <- prediction(accepted_ind, univ_test$ACCEPTED)

# 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
##          0 824 111
##          1  41  86
```

```r
c1$overall['Accuracy']
```

```
##  Accuracy
## 0.8568738
```

```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] 0.8719397
```

```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] 0.9453861
```

```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] 0.8926554
```

```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] 0.8719397
```

```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] 0.8785311
```

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
## -2.49594  -0.48536  -0.25387  -0.08431   3.01218
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.726e+01  1.407e+00 -12.272  < 2e-16 ***
## REGION        1.088e-01  3.120e-02   3.489 0.000486 ***
```

```
## ADM_RATE_ALL  1.167e+00  4.214e-01   2.769 0.005615 **
## SAT_AVG_ALL   1.510e-02  9.920e-04  15.217  < 2e-16 ***
## PCIP11        2.017e+00  1.977e+00   1.020 0.307736
## PCIP12       -2.933e+00  1.855e+01  -0.158 0.874387
## PCIP14        5.631e+00  7.891e-01   7.136 9.63e-13 ***
## PCIP15       -6.102e-01  2.207e+00  -0.277 0.782135
## PCIP24       -5.799e+00  1.251e+00  -4.634 3.58e-06 ***
## PCIP26        6.603e+00  1.681e+00   3.927 8.60e-05 ***
## PCIP27       -3.275e+01  6.868e+00  -4.769 1.85e-06 ***
## PCIP40       -2.944e+01  4.700e+00  -6.264 3.76e-10 ***
## PCIP45        8.149e+00  1.193e+00   6.831 8.41e-12 ***
## PCIP51        1.716e+00  5.896e-01   2.910 0.003612 **
## PCIP52        8.490e-01  6.422e-01   1.322 0.186177
## UGDS_NRA      8.592e+00  1.446e+00   5.940 2.85e-09 ***
## UGDS_UNKN    -1.458e+00  1.585e+00  -0.920 0.357593
## COSTT4_A     -1.027e-04  7.042e-06 -14.588  < 2e-16 ***
## PCTFLOAN     -8.664e-01  5.539e-01  -1.564 0.117799
## UGDS_WOMEN    7.829e-01  7.956e-01   0.984 0.325125
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3134.2  on 3184  degrees of freedom
## Residual deviance: 1922.2  on 3165  degrees of freedom
## AIC: 1962.2
##
## Number of Fisher Scoring iterations: 6
```

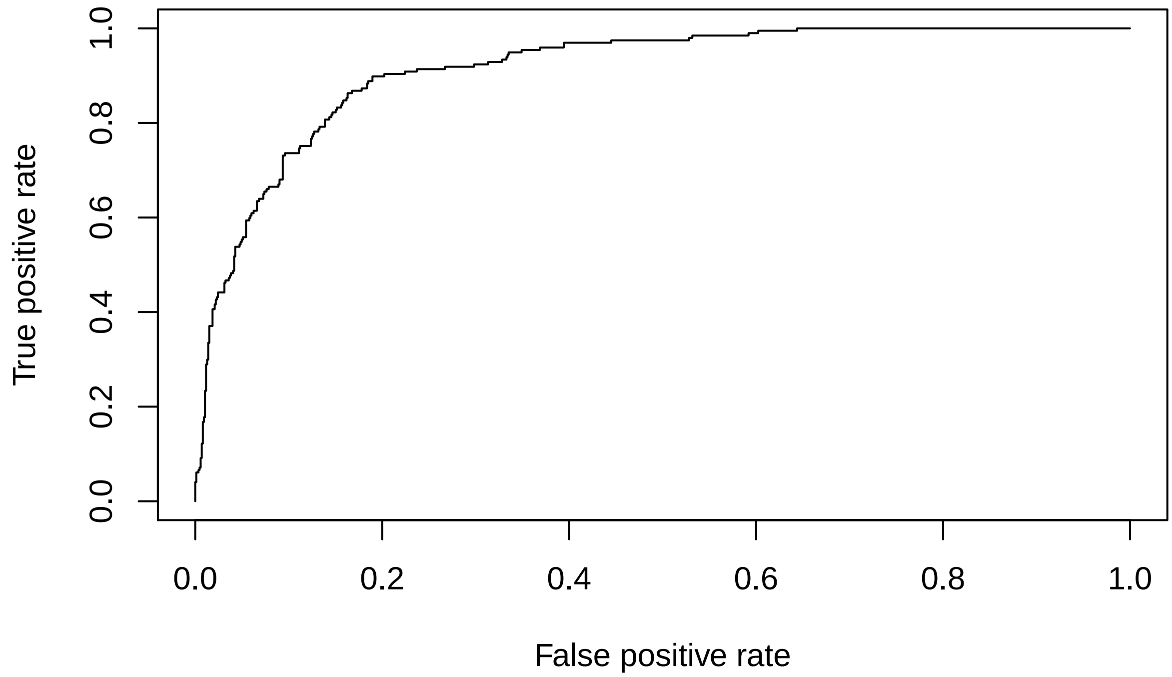
```r
# do the testing with the prediction model
accepted_ind2 <- predict(glm_ISSciAcceptance, type="response", newdata = univ_test)
pred2 <- prediction(accepted_ind2, univ_test$ACCEPTED)

# 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
##          0 824  91
##          1  41 106
```

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

```
##  Accuracy
## 0.8757062
```

```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] 0.9124294
```

```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] 0.9632768
```

```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] 0.9218456
```

```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] 0.9124294
```

```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] 0.9067797
```

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

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 moded

```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 2.766861 mins.
##  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              5.52974696  5.5522392  4.1274905  6.636412 1.0000000
## ADM_RATE            7.24332389  7.2389421  5.5520508  9.076481 1.0000000
## ADM_RATE_ALL        7.23473112  7.2324083  5.3687783  8.779141 1.0000000
## SAT_AVG_ALL        12.65928443 12.6445603 11.2748099 14.612827 1.0000000
## PCIP01              6.28218323  6.1678918  5.1562833  7.559202 1.0000000
## PCIP03              6.67312667  6.6999253  5.0771624  8.660605 1.0000000
## PCIP04             11.61789838 11.5829457 10.3907713 13.056586 1.0000000
## PCIP05              8.40562345  8.4616467  7.2958389  9.761936 1.0000000
## PCIP09              4.94315957  4.9867677  3.1170737  6.586527 0.9898990
## PCIP10              2.58583064  2.5551054  0.6064175  4.344666 0.4848485
## PCIP11              6.54533901  6.5489331  4.8035285  8.137199 1.0000000
## PCIP12              0.98625463  1.0691671 -0.2861730  2.063135 0.0000000
## PCIP13              6.16730011  6.1693622  4.3256899  7.584682 1.0000000
## PCIP14             18.75802228 18.6910930 16.9640469 20.857231 1.0000000
## PCIP15              4.84239074  4.9022054  2.8640709  6.613914 0.9898990
## PCIP16              7.60574948  7.6590083  6.3338622  8.955518 1.0000000
## PCIP19              7.56877672  7.5945312  6.2920698  9.286560 1.0000000
## PCIP22              2.47973072  2.5931591 -0.1263870  4.403460 0.4545455
## PCIP23              8.39697887  8.4017731  6.8941396  9.631402 1.0000000
## PCIP24              5.85378061  5.8151945  4.4987107  7.840930 1.0000000
## PCIP25             -0.89019549 -1.0010015 -1.7369988  1.001002 0.0000000
## PCIP26              5.96839980  5.9135097  4.1765844  7.785011 1.0000000
## PCIP27              5.23813865  5.2532260  3.4266141  7.512010 0.9898990
## PCIP29              0.00000000  0.0000000  0.0000000  0.000000 0.0000000
## PCIP30              4.07367548  4.1886640  1.7034647  6.171970 0.8989899
```

```
## PCIP31              4.71679431  4.6897352  2.8650822  6.116799 0.9696970
## PCIP38              4.25898534  4.3826799  2.3718967  5.707659 0.9393939
## PCIP39              5.48159035  5.5198817  3.9193382  6.874921 1.0000000
## PCIP40              5.65752059  5.7684181  3.6482280  7.128499 0.9898990
## PCIP41              3.18076970  3.2155031  1.1234088  5.683796 0.6565657
## PCIP42              4.88403164  4.9143865  3.0664552  6.572075 1.0000000
## PCIP43              7.25793138  7.2363085  5.7287242  8.509953 1.0000000
## PCIP44              4.46619330  4.3657188  2.2528415  6.168083 0.9696970
## PCIP45              7.60447722  7.6235156  5.9251666  8.935069 1.0000000
## PCIP46             -0.05243582  0.0000000 -1.6049490  1.001002 0.0000000
## PCIP47             -0.32285281 -0.1963086 -1.7365407  1.416994 0.0000000
## PCIP48              0.51271983  0.8700177 -1.2821563  1.825102 0.0000000
## PCIP49              3.48319323  3.4414286  1.6286197  4.717037 0.8585859
## PCIP50              5.76906381  5.6838807  3.9535597  7.825806 1.0000000
## PCIP51              4.01421428  4.1252745  2.1798531  5.711989 0.9494949
## PCIP52              9.64763392  9.6232674  8.4350159 11.032032 1.0000000
## PCIP54              3.88560485  3.9703998  1.2872190  5.829585 0.8484848
## UGDS_WHITE          8.20904517  8.2004673  6.9396906  9.719565 1.0000000
## UGDS_BLACK         10.82595353 10.8193106  9.3249700 12.364024 1.0000000
## UGDS_HISP           6.17341737  6.1823885  3.9280635  7.702337 1.0000000
## UGDS_ASIAN          9.18791801  9.2111193  7.8090650 10.404944 1.0000000
## UGDS_AIAN           4.20097600  4.1589893  2.2761313  7.018973 0.9494949
## UGDS_NHPI           3.75872524  3.7861878  1.4685172  5.471620 0.8787879
## UGDS_2MOR           4.46458627  4.4859280  2.7915489  6.738625 0.9797980
## UGDS_NRA            7.14146478  7.1255599  5.9058436  8.439296 1.0000000
## UGDS_UNKN           6.16815957  6.2305060  3.7643695  7.334462 1.0000000
## PPTUG_EF            6.87361722  6.8003823  5.0360336  8.185720 1.0000000
## COSTT4_A            9.78985169  9.8029715  7.7067549 10.926670 1.0000000
## TUITIONFEE_IN       9.52304862  9.5600661  8.1553253 11.173515 1.0000000
## TUITIONFEE_OUT      5.53635569  5.5690374  4.0255528  7.209874 1.0000000
## C150_4              7.91595324  7.9197952  6.4355296  9.242069 1.0000000
## C150_4_WHITE        6.77501833  6.7656602  5.3877216  8.143375 1.0000000
## C150_4_BLACK        7.09555521  7.0633652  5.6300622  8.295815 1.0000000
## C150_4_HISP         5.69702010  5.6571310  4.4852483  6.783074 1.0000000
## C150_4_ASIAN        6.08139881  6.0873884  4.9150456  7.484236 1.0000000
## C150_4_AIAN         7.12832484  7.1326454  5.4873031  8.659522 1.0000000
## C150_4_NHPI         0.52351273  0.7320779 -1.1000583  2.162487 0.0000000
## C150_4_2MOR         3.14657728  3.2104470  1.3602940  4.806438 0.7070707
## C150_4_NRA          4.45578328  4.5282696  2.4846928  6.178089 0.9595960
## C150_4_UNKN         7.22921967  7.1917223  6.1546852  8.306619 1.0000000
## RET_FT4            10.62500853 10.5983845  9.2692734 11.942381 1.0000000
## PCTFLOAN           13.95504341 13.9784176 12.6211386 15.516570 1.0000000
## PAR_ED_PCT_1STGEN   6.01961719  6.0457136  4.4410298  7.504918 1.0000000
## UGDS_MEN           12.53180009 12.5454873 11.3805948 14.124584 1.0000000
## UGDS_WOMEN         12.40307641 12.3916093 10.8485593 13.668999 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
#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_MEN"     "PCIP45"
##  [9] "UGDS_WOMEN"   "PCIP43"       "COSTT4_A"     "RET_FT4"
## [13] "PCIP23"       "UGDS_HISP"    "TUITIONFEE_IN" "C150_4_AIAN"
## [17] "PCIP39"       "PCIP16"       "UGDS_ASIAN"   "UGDS_WHITE"
## [21] "UGDS_NRA"     "C150_4"       "PCIP19"       "PPTUG_EF"
## [25] "PCIP24"       "PCIP05"       "PCIP50"       "PCIP26"
## [29] "PCIP03"       "PCIP09"       "UGDS_UNKN"
```

Based on these runs, Boruta has 61 attributes that are confirmed important, and 2 that are tentative. On the other hand, RFE confirms less than 30 variables that are very important.

US Research University Completion Rate Prediction Model

```r
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
## -0.63729 -0.05908  0.00529  0.07455  0.49736
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        9.495e-01  4.310e-02  22.029  < 2e-16 ***
## REGION            -3.616e-03  3.181e-03  -1.137  0.25609
## ADM_RATE_ALL      -1.206e-01  3.746e-02  -3.219  0.00136 **
## UGDS_NRA           7.623e-02  1.402e-01   0.544  0.58680
## PPTUG_EF          -3.431e-01  8.318e-02  -4.124 4.24e-05 ***
## COSTT4_A           1.727e-06  6.105e-07   2.829  0.00483 **
## PCTFLOAN          -4.074e-01  5.560e-02  -7.328 7.54e-13 ***
## PAR_ED_PCT_1STGEN -9.655e-02  9.476e-02  -1.019  0.30867
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1354 on 603 degrees of freedom
## Multiple R-squared:  0.4345, Adjusted R-squared:  0.428
## F-statistic:  66.2 on 7 and 603 DF,  p-value: < 2.2e-16
```

```r
# 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
## 0.754902
```

```r
# 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] 0.8431373
```

```r
# 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] 0.8970588
```

```r
# 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] 0.8921569
```

```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] 0.8529412
```

```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] 0.877451
```

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