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

library(Boruta)

## Loading required package: ranger

##   
## Attaching package: 'ranger'

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

library(e1071)  
library(ROCR)

## Loading required package: gplots

##   
## Attaching package: 'gplots'

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

library(corrplot)  
library(ggplot2)

#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 classification for the rest  
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 our focus will be on these  
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, 7-Rocky Mountains, 8-Far West, 9-Outlying Areas)  
# 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 expected time to completion)  
# 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

#show number of rows of filtered research universities  
rows\_usresearchuniv <- nrow(usresearchuniv)  
rows\_usresearchuniv

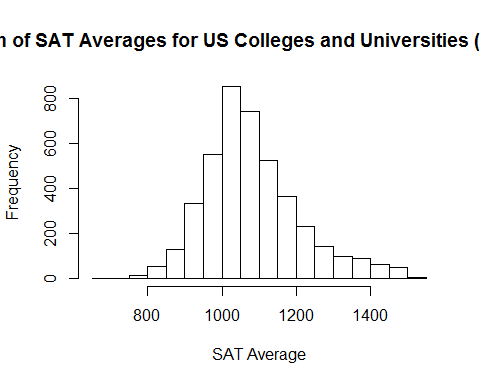
## [1] 815

#percentage of data from filtered research universities to unfiltered  
rows\_usresearchuniv / rows\_usunivaccepted

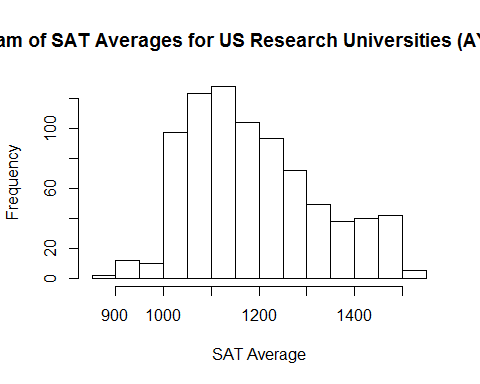
## [1] 0.7062392

# Distributions and Box and Whisker Plots

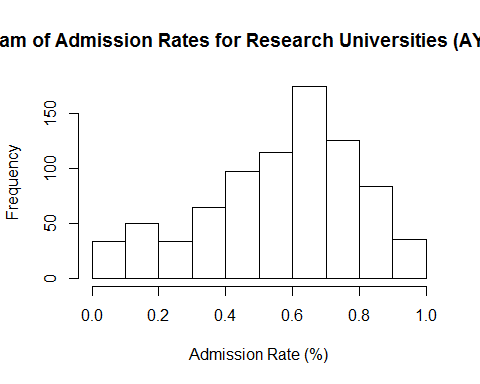
# Histogram of SAT Averages for US Colleges and Universities  
hist(usunivfilter$SAT\_AVG\_ALL, main = "Histogram of SAT Averages for US Colleges and Universities (AY2010-2015)", xlab="SAT Average")



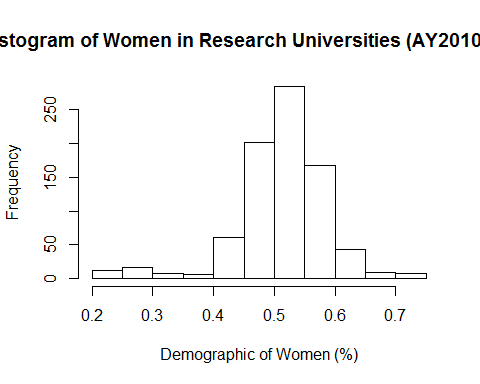
# Histogram of SAT Averages for US Research Universities  
hist(usresearchuniv$SAT\_AVG\_ALL, main = "Histogram of SAT Averages for US Research Universities (AY2010-2015)", xlab="SAT Average")



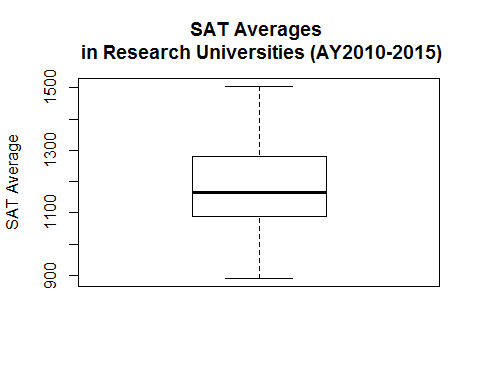
# Histogram of Admission Rates for US Research Universities  
hist(usresearchuniv$ADM\_RATE\_ALL, main = "Histogram of Admission Rates for Research Universities (AY2010-2015)", xlab = "Admission Rate (%)")



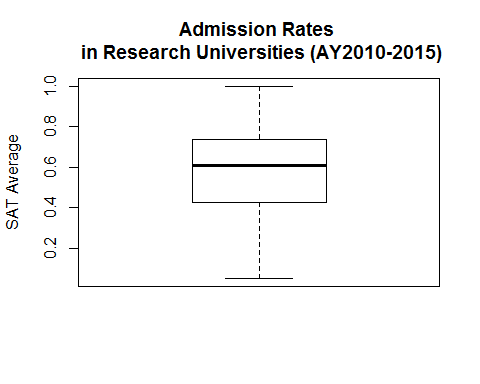
# Histogram of Women in US Research Universities  
hist(usresearchuniv$UGDS\_WOMEN, main = "Histogram of Women in Research Universities (AY2010-2015)", xlab = "Demographic of Women (%)")



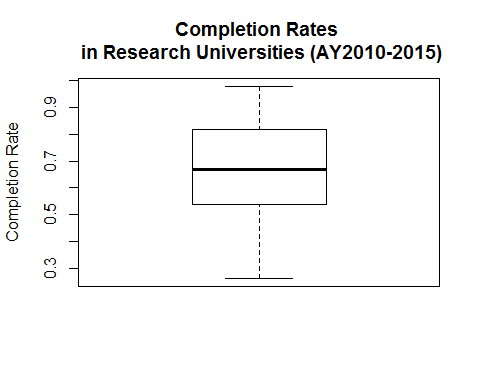
#Boxplot of SAT Average in all US Research Universities  
boxplot(usresearchuniv$SAT\_AVG\_ALL, main = "SAT Averages \n in Research Universities (AY2010-2015)", ylab = "SAT Average")



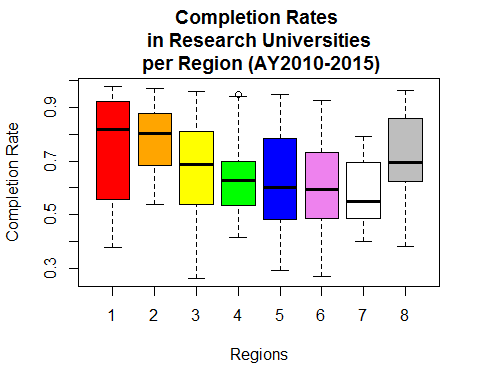
#Boxplot of admission rates in all US Research Universities  
boxplot(usresearchuniv$ADM\_RATE\_ALL, main = "Admission Rates \n in Research Universities (AY2010-2015)", ylab = "SAT Average")



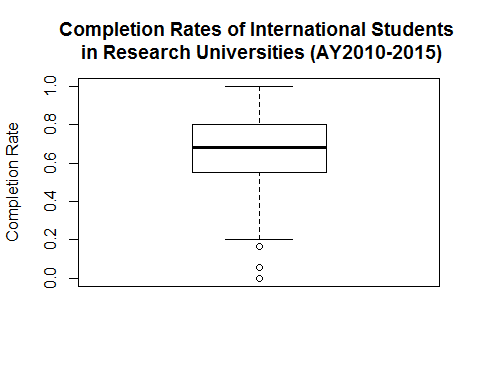
#Boxplot of Completion Rates in all US Research Universities  
boxplot(usresearchuniv$C150\_4, main = "Completion Rates \n in Research Universities (AY2010-2015)", ylab = "Completion Rate")



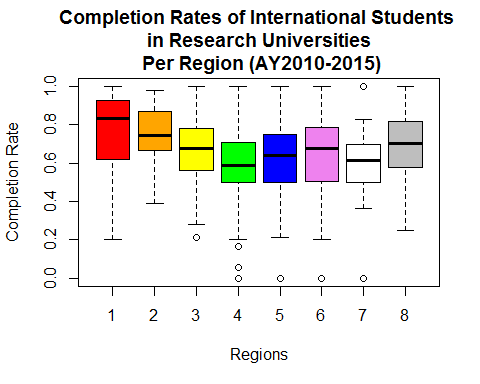
# Boxplot of Completion Rates per Region in US Research Universities  
boxplot(C150\_4 ~ REGION, usresearchuniv, main = "Completion Rates \n in Research Universities \n per Region (AY2010-2015)", col=c("red", "orange", "yellow", "green", "blue", "violet", "white", "gray", "magenta"), ylab = "Completion Rate", xlab = "Regions")



#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 Universities (AY2010-2015)", ylab = "Completion Rate")



# 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 Research Universities \n Per Region (AY2010-2015)", col=c("red", "orange", "yellow", "green", "blue", "violet", "white", "gray", "magenta"), ylab = "Completion Rate", xlab = "Regions")

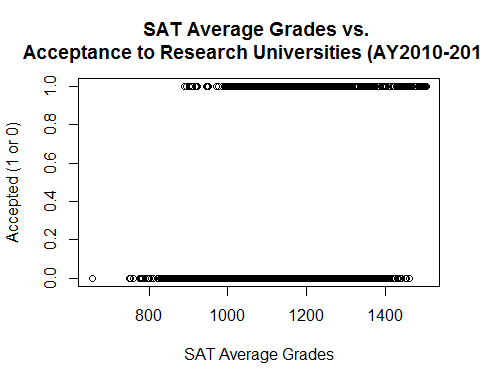


nrow(usresearchuniv[usresearchuniv$C150\_4\_NRA < 0.2,])

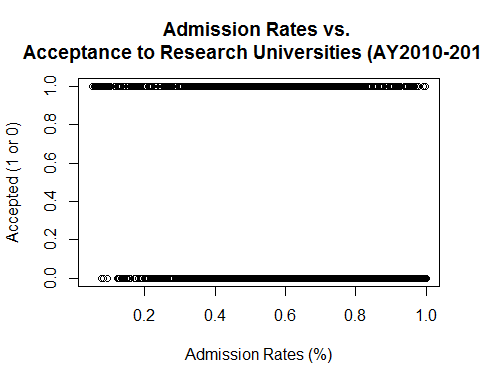
## [1] 9

# Correlations

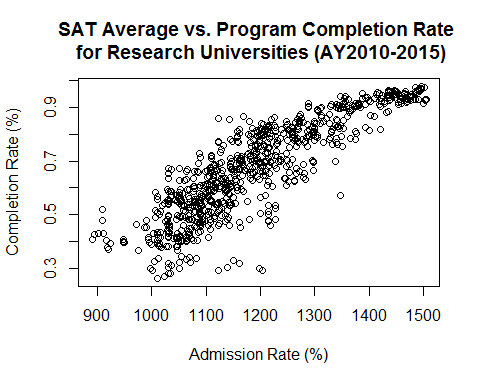
#Correlation between the SAT grades and the acceptance for the research universities  
plot(usunivfilter$SAT\_AVG\_ALL, usunivfilter$ACCEPTED, main="SAT Average Grades vs. \n Acceptance to Research Universities (AY2010-2015)", xlab="SAT Average Grades", ylab="Accepted (1 or 0)")



#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 Research Universities (AY2010-2015)", xlab="Admission Rates (%)", ylab="Accepted (1 or 0)")



#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 \n for Research Universities (AY2010-2015)", xlab="Admission Rate (%)", ylab="Completion Rate (%)")

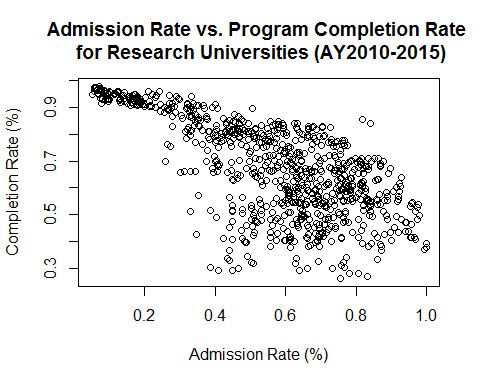


#Correlation coefficient between admission rate and completion rate  
cor(usresearchuniv$SAT\_AVG\_ALL, usresearchuniv$C150\_4, method = "pearson")

## [1] 0.8702261

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

#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 Rate \n for Research Universities (AY2010-2015)", xlab="Admission Rate (%)", ylab="Completion Rate (%)")

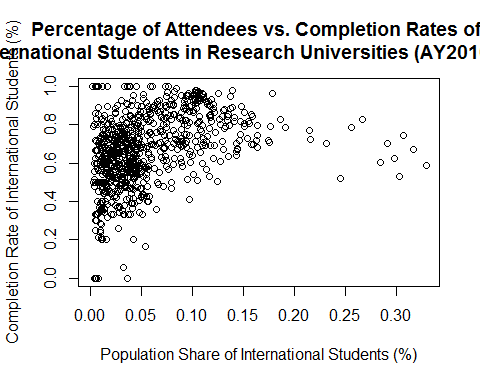


#Correlation coefficient between admission rate and completion rate  
cor(usresearchuniv$ADM\_RATE\_ALL, usresearchuniv$C150\_4, method = "pearson")

## [1] -0.6825525

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

#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 Rates of \n International Students in Research Universities (AY2010-2015)", xlab="Population Share of International Students (%)", ylab="Completion Rate of International Students (%)")

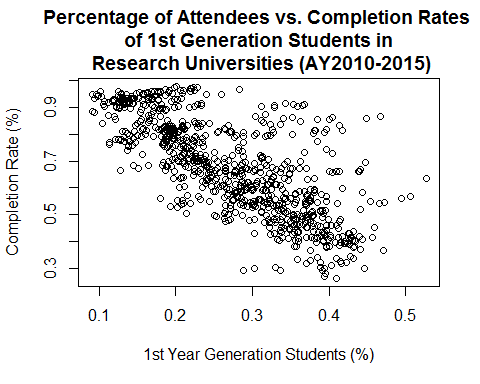


#Correlation coefficient between admission rate and completion rate of international students  
cor(usresearchuniv$UGDS\_NRA, usresearchuniv$C150\_4\_NRA, method = "pearson")

## [1] 0.370641

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

#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. Completion Rates \n of 1st Generation Students in \n Research Universities (AY2010-2015)", xlab="1st Year Generation Students (%)", ylab="Completion Rate (%)")



#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 based on SAT, average cost, loans, and gender  
formula\_ISAcceptance <- formula(ACCEPTED ~ REGION + ADM\_RATE\_ALL + SAT\_AVG\_ALL + UGDS\_NRA + COSTT4\_A + PCTFLOAN + UGDS\_WOMEN)

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

where

.

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 = 20000, PCTFLOAN = 0.33, UGDS\_WOMEN = .37)  
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 = 25600, PCTFLOAN = 0.57, UGDS\_WOMEN = .55)  
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   
## -2.1632 -0.5333 -0.2921 -0.1250 2.8053   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.384e+01 1.205e+00 -11.490 < 2e-16 \*\*\*  
## REGION 1.398e-01 2.976e-02 4.698 2.63e-06 \*\*\*  
## ADM\_RATE\_ALL 1.922e-01 3.812e-01 0.504 0.614236   
## SAT\_AVG\_ALL 1.424e-02 8.565e-04 16.622 < 2e-16 \*\*\*  
## UGDS\_NRA 5.080e+00 1.307e+00 3.887 0.000101 \*\*\*  
## COSTT4\_A -9.092e-05 6.371e-06 -14.271 < 2e-16 \*\*\*  
## PCTFLOAN -4.972e-01 4.986e-01 -0.997 0.318726   
## UGDS\_WOMEN -2.655e+00 5.462e-01 -4.862 1.16e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 3061.7 on 3184 degrees of freedom  
## Residual deviance: 2095.2 on 3177 degrees of freedom  
## AIC: 2111.2  
##   
## 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  
## 0 808 135  
## 1 32 87

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

## Accuracy   
## 0.8427495

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

## Neg Pred Value   
## 0.7310924

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

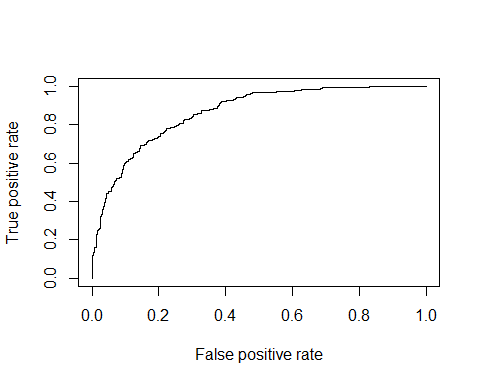
## Specificity   
## 0.3918919

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.

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



# 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] 0.8691149

# 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?

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] 0.9303202

# 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] 0.8700565

# 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] 0.8691149

# 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] 0.8559322

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

# create a formula for the US research university acceptance model for International Students taking up Science degrees/majors  
formula\_ISSciAcceptance <- formula(ACCEPTED ~ REGION + ADM\_RATE\_ALL + SAT\_AVG\_ALL + PCIP11 + PCIP12 + PCIP14 + PCIP15 + PCIP24 + PCIP26 + PCIP27 + PCIP40 + PCIP45 + PCIP51 + PCIP52 + UGDS\_NRA + UGDS\_UNKN + COSTT4\_A + PCTFLOAN + UGDS\_WOMEN)  
  
# 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   
## -2.5901 -0.4620 -0.2422 -0.0856 3.1980   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.715e+01 1.460e+00 -11.745 < 2e-16 \*\*\*  
## REGION 1.520e-01 3.209e-02 4.738 2.16e-06 \*\*\*  
## ADM\_RATE\_ALL 5.590e-01 4.221e-01 1.324 0.185354   
## SAT\_AVG\_ALL 1.518e-02 1.034e-03 14.678 < 2e-16 \*\*\*  
## PCIP11 6.023e-01 1.970e+00 0.306 0.759754   
## PCIP12 1.745e+00 2.110e+01 0.083 0.934092   
## PCIP14 6.498e+00 8.365e-01 7.769 7.92e-15 \*\*\*  
## PCIP15 3.059e-01 2.333e+00 0.131 0.895681   
## PCIP24 -6.033e+00 1.291e+00 -4.674 2.95e-06 \*\*\*  
## PCIP26 6.199e+00 1.743e+00 3.557 0.000374 \*\*\*  
## PCIP27 -2.756e+01 7.037e+00 -3.917 8.96e-05 \*\*\*  
## PCIP40 -3.175e+01 4.773e+00 -6.653 2.87e-11 \*\*\*  
## PCIP45 9.068e+00 1.222e+00 7.418 1.19e-13 \*\*\*  
## PCIP51 1.975e+00 6.319e-01 3.125 0.001775 \*\*   
## PCIP52 9.425e-01 6.670e-01 1.413 0.157686   
## UGDS\_NRA 7.707e+00 1.526e+00 5.050 4.43e-07 \*\*\*  
## UGDS\_UNKN -1.469e+00 1.559e+00 -0.942 0.345973   
## COSTT4\_A -1.107e-04 7.551e-06 -14.657 < 2e-16 \*\*\*  
## PCTFLOAN -1.218e-01 5.777e-01 -0.211 0.832989   
## UGDS\_WOMEN 2.224e-01 8.013e-01 0.278 0.781371   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 3061.7 on 3184 degrees of freedom  
## Residual deviance: 1848.8 on 3165 degrees of freedom  
## AIC: 1888.8  
##   
## Number of Fisher Scoring iterations: 6

# 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  
## 0 801 110  
## 1 39 112

c2$overall['Accuracy']

## Accuracy   
## 0.8596987

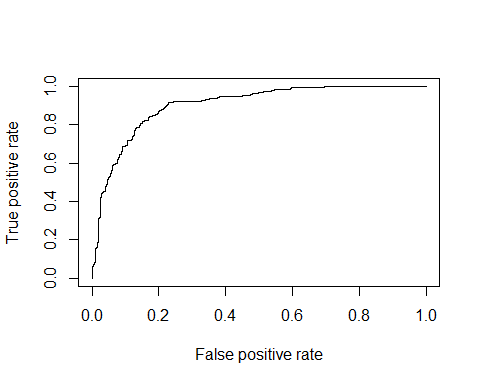
#Precision of the logistic regression model  
c2$byClass['Neg Pred Value']

## Neg Pred Value   
## 0.7417219

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

## Specificity   
## 0.5045045

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



# 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] 0.8832392

# 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?

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] 0.952919

# 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] 0.9011299

# 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] 0.8719397

# 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] 0.8672316

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.

# 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 29.996 secs.  
## 61 attributes confirmed important: ADM\_RATE, ADM\_RATE\_ALL,  
## C150\_4, C150\_4\_2MOR, C150\_4\_AIAN and 56 more.  
## 7 attributes confirmed unimportant: C150\_4\_NHPI, PCIP12, PCIP25,  
## PCIP29, PCIP46 and 2 more.  
## 2 tentative attributes left: PCIP10, PCIP22.

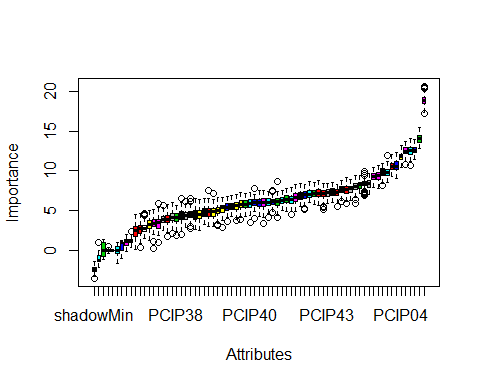
getSelectedAttributes(boruta.train)

## [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] "PCIP41" "PCIP42" "PCIP43"   
## [28] "PCIP44" "PCIP45" "PCIP49"   
## [31] "PCIP50" "PCIP51" "PCIP52"   
## [34] "PCIP54" "UGDS\_WHITE" "UGDS\_BLACK"   
## [37] "UGDS\_HISP" "UGDS\_ASIAN" "UGDS\_AIAN"   
## [40] "UGDS\_NHPI" "UGDS\_2MOR" "UGDS\_NRA"   
## [43] "UGDS\_UNKN" "PPTUG\_EF" "COSTT4\_A"   
## [46] "TUITIONFEE\_IN" "TUITIONFEE\_OUT" "C150\_4"   
## [49] "C150\_4\_WHITE" "C150\_4\_BLACK" "C150\_4\_HISP"   
## [52] "C150\_4\_ASIAN" "C150\_4\_AIAN" "C150\_4\_2MOR"   
## [55] "C150\_4\_NRA" "C150\_4\_UNKN" "RET\_FT4"   
## [58] "PCTFLOAN" "PAR\_ED\_PCT\_1STGEN" "UGDS\_MEN"   
## [61] "UGDS\_WOMEN"

# 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 5.54665053 5.51079042 4.22659609 6.873450 1.0000000  
## ADM\_RATE 7.25154421 7.26650155 5.49475356 8.301225 1.0000000  
## ADM\_RATE\_ALL 7.13406152 7.11592293 5.76577582 8.494882 1.0000000  
## SAT\_AVG\_ALL 12.60688842 12.58060243 11.36721307 13.922811 1.0000000  
## PCIP01 6.25227027 6.28759662 5.02195128 7.543716 1.0000000  
## PCIP03 6.62167113 6.66194353 4.87891266 8.223139 1.0000000  
## PCIP04 11.69709785 11.75830705 10.35549103 13.145013 1.0000000  
## PCIP05 8.38325546 8.39848184 6.99340371 9.715353 1.0000000  
## PCIP09 4.76500202 4.79612176 3.23728203 6.569915 0.9797980  
## PCIP10 2.65037294 2.68587442 0.34868341 4.687841 0.4747475  
## PCIP11 6.45343722 6.38887331 4.82791038 7.756515 1.0000000  
## PCIP12 1.21506603 1.11113563 0.31404198 2.326753 0.0000000  
## PCIP13 6.02167627 6.00143271 4.16230486 7.383240 1.0000000  
## PCIP14 18.84002889 18.81620038 17.16369480 20.610420 1.0000000  
## PCIP15 4.97960833 4.97033489 3.08746688 6.650644 0.9898990  
## PCIP16 7.64786145 7.71943042 6.21379740 8.908762 1.0000000  
## PCIP19 7.59668165 7.56116597 6.13648077 8.839726 1.0000000  
## PCIP22 2.36760850 2.42920970 0.07296174 4.447525 0.3636364  
## PCIP23 8.36652762 8.34375303 6.88663672 9.961165 1.0000000  
## PCIP24 5.95132515 5.97541204 4.26927164 7.553055 1.0000000  
## PCIP25 -1.03069545 -1.34095461 -2.11688238 1.001002 0.0000000  
## PCIP26 5.93909667 5.98325920 3.32494834 7.545225 1.0000000  
## PCIP27 5.24187830 5.34334992 2.87423136 6.634680 1.0000000  
## PCIP29 0.00000000 0.00000000 0.00000000 0.000000 0.0000000  
## PCIP30 4.25902394 4.27236662 1.97724439 6.509423 0.9494949  
## PCIP31 4.85333532 4.84209583 3.00389222 7.550790 1.0000000  
## PCIP38 4.14741248 4.18296838 1.88852846 5.679210 0.9494949  
## PCIP39 5.42627389 5.43281002 3.58600313 6.787014 1.0000000  
## PCIP40 5.86098352 5.92190417 3.53583957 7.321965 1.0000000  
## PCIP41 3.37469196 3.48177061 0.99560362 5.896727 0.7373737  
## PCIP42 4.77337695 4.85083294 3.03437499 7.149081 0.9595960  
## PCIP43 7.21754091 7.22807493 5.88148798 8.431551 1.0000000  
## PCIP44 4.51015453 4.48170696 2.75728500 6.450802 0.9595960  
## PCIP45 7.58166648 7.59186220 5.87381886 9.127617 1.0000000  
## PCIP46 0.02397202 -0.04085623 -1.21446542 1.415606 0.0000000  
## PCIP47 0.33987774 1.00100150 -1.00100150 1.234924 0.0000000  
## PCIP48 -0.03940030 0.00000000 -1.65863664 1.001002 0.0000000  
## PCIP49 3.35333763 3.38475145 0.20131411 5.145311 0.7575758  
## PCIP50 5.77138376 5.73641257 3.85748345 7.321007 1.0000000  
## PCIP51 4.10258206 4.12722175 2.08323566 5.961471 0.9090909  
## PCIP52 9.80156576 9.79661648 8.17964675 10.855110 1.0000000  
## PCIP54 3.90107356 3.89262449 1.70081923 5.764168 0.8484848  
## UGDS\_WHITE 8.20278433 8.22461414 7.08317574 9.393159 1.0000000  
## UGDS\_BLACK 10.67877570 10.58975063 9.08026016 12.287102 1.0000000  
## UGDS\_HISP 6.34029784 6.41245253 4.44358078 7.873665 1.0000000  
## UGDS\_ASIAN 9.26759267 9.28255516 7.89116317 10.471739 1.0000000  
## UGDS\_AIAN 4.51344805 4.54522954 2.49513861 6.416741 0.9898990  
## UGDS\_NHPI 3.91904858 3.87302176 2.46587358 5.656684 0.8888889  
## UGDS\_2MOR 4.49557730 4.49730564 2.55847087 6.546393 0.9595960  
## UGDS\_NRA 7.17845110 7.16990953 5.65811607 8.776867 1.0000000  
## UGDS\_UNKN 6.05861044 6.15328700 4.12901843 8.649083 1.0000000  
## PPTUG\_EF 6.95381329 6.95743584 5.14645961 8.299188 1.0000000  
## COSTT4\_A 9.80808817 9.82628748 8.48121023 11.908417 1.0000000  
## TUITIONFEE\_IN 9.35427226 9.34519162 7.78536064 10.635912 1.0000000  
## TUITIONFEE\_OUT 5.56762522 5.57069085 3.71891085 7.008538 1.0000000  
## C150\_4 8.02543701 8.04686002 5.94249936 9.421203 1.0000000  
## C150\_4\_WHITE 6.82505566 6.78552387 5.52672532 8.369872 1.0000000  
## C150\_4\_BLACK 7.11855863 7.20035765 5.19260901 8.347293 1.0000000  
## C150\_4\_HISP 5.82569436 5.77660973 4.27791596 7.123273 1.0000000  
## C150\_4\_ASIAN 6.07902248 6.13782264 4.81499009 7.467059 1.0000000  
## C150\_4\_AIAN 7.15852460 7.07229072 6.02020546 8.615574 1.0000000  
## C150\_4\_NHPI 1.06272230 1.09766415 -0.09175769 2.023362 0.0000000  
## C150\_4\_2MOR 3.23152665 3.26641490 1.23928720 4.949187 0.7171717  
## C150\_4\_NRA 4.36639617 4.39497071 2.82128610 6.088930 0.9595960  
## C150\_4\_UNKN 7.27819042 7.23031308 6.03582834 8.466388 1.0000000  
## RET\_FT4 10.55212829 10.54225063 9.26114736 12.022287 1.0000000  
## PCTFLOAN 14.07898144 14.08560522 12.82117834 15.457247 1.0000000  
## PAR\_ED\_PCT\_1STGEN 6.00036152 5.94910763 4.04605960 7.752022 1.0000000  
## UGDS\_MEN 12.41701208 12.48780691 10.61445069 14.012695 1.0000000  
## UGDS\_WOMEN 12.48660792 12.48279885 10.84052339 13.620633 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 Confirmed  
## 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

# 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)



#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)

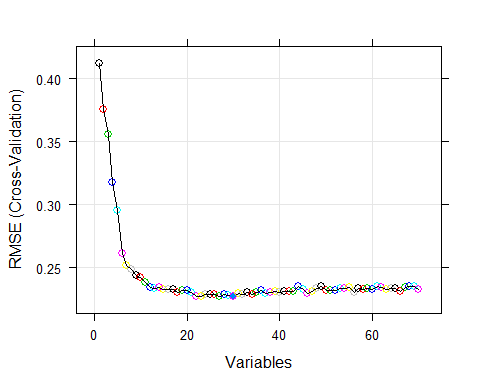
##   
## Attaching package: 'plyr'

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

predictors(rfe.train)

## [1] "PCIP14" "PCTFLOAN" "PCIP04" "SAT\_AVG\_ALL"   
## [5] "PCIP52" "UGDS\_BLACK" "UGDS\_MEN" "PCIP45"   
## [9] "UGDS\_WOMEN" "PCIP43" "COSTT4\_A" "PCIP23"   
## [13] "UGDS\_HISP" "RET\_FT4" "PCIP16" "TUITIONFEE\_IN"  
## [17] "C150\_4\_AIAN" "UGDS\_ASIAN" "PCIP39" "UGDS\_WHITE"   
## [21] "UGDS\_NRA" "C150\_4" "PCIP19" "PCIP05"   
## [25] "PPTUG\_EF" "PCIP24" "PCIP03" "UGDS\_UNKN"   
## [29] "PCIP50" "PCIP26"

# 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 + PCTFLOAN + PAR\_ED\_PCT\_1STGEN)  
  
# 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   
## -0.61726 -0.05817 0.00777 0.07586 0.50915   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 9.341e-01 4.588e-02 20.360 < 2e-16 \*\*\*  
## REGION -6.807e-03 3.341e-03 -2.038 0.042032 \*   
## ADM\_RATE\_ALL -1.418e-01 4.059e-02 -3.493 0.000512 \*\*\*  
## UGDS\_NRA 2.323e-01 1.540e-01 1.509 0.131917   
## PPTUG\_EF -3.300e-01 9.005e-02 -3.665 0.000269 \*\*\*  
## COSTT4\_A 1.615e-06 6.335e-07 2.550 0.011033 \*   
## PCTFLOAN -3.752e-01 6.071e-02 -6.180 1.18e-09 \*\*\*  
## PAR\_ED\_PCT\_1STGEN -5.034e-02 1.020e-01 -0.493 0.621901   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1453 on 603 degrees of freedom  
## Multiple R-squared: 0.4099, Adjusted R-squared: 0.4031   
## F-statistic: 59.84 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] 0.9215686

# 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] 0.9411765

# 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] 0.9362745

# 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] 0.9166667

# 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] 0.9313725

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