Personalized Capstone Project

Emma Jean Boley

2022-11-07

**Introduction**

The main objective is to determine factors/variables that may influence the admission of a student into college within the United States. This analysis will examine seven attributes/variables (Graduate Record Exam Scores (GRE), Grade Point Average (GPA), Rank of University with 1 being very prestigious, socioeconomic status (ranging from 1 to 3 or low to high), gender (female and male were recoded as 0 and 1 respectively) and race (1, 2, 3 indicates that the student identifies as Hispanics, Asian, and African-American respectively). The response or dependent variable in admit, a binary variable where 1 indicates that the student was admitted and 0 not admitted. The dataset was transformed and then baseline prediction was used to first determine the outcome and whether our machine learning algorithm performed better than chance. Additionally, other models were trained and tested to evaluate their accuracy.

**Data Wrangling/Transformation**

The dataset was transformed, categorical variables (rank, socioeconomic status, gender, race, admit) were changed to factors, and then histogram and boxplots were used to examine distribution of numerical variables (GRE & GPA) and variations in categorical variables (socioeconomic status, gender, race, etc). Both GRE and GPA were observed to have outliers. Excluding outliers, females and males had similar variations in gre scores. However, females had higher GPA scores than males. Variations in GRE score and GPA were observed among different socioeconomic status and race groups. 2.26, 220, and 300 were identified as outliers of GPA and GRE scores and these outliers were removed from dataset. The clean data was then split 80:20 into train and test sets using to caret package.

**Data Analysis/Result**

0.45 was the mean probability of being admitted into college. The accuracy of the sex-based prediction was 0.5. 0.625 was the accuracy of the socio-economic status on the probability of college admission. The specificity and sensitivity of the sex model is 0.423 & 0.537 while the specificity and sensitivity of the class based model is 0.346 & 0.759. The balance accuracy is approximately 0.5 for both models. However, the F-means score is 0.592 and 0.732 for the sex and class based models respectively. 0.675 was the accuracy of GRE or GPA as a predictor of admission using the QDA method. The accuracy of model using the glm method with GRE scores, GPA, socio-economic status and race as predictors was 0.725. The KNN value with the highest accuracy 0.675 was identified as 51. The accuracy of the decision tree model was observed at 0.675. The most important variables identified by the random forest model were GRE and GPA.

**Conclusion**

This analysis provided insights on drivers on college admissions examining multiple variables. It was observed that academic performance (GRE scores and GPA) were better predictors of admission than racial and socioeconomic status. To confirm observations, a larger dataset would need to be examined on college admission in the United States.

#Loading packages and librabies

if(!require(lubridate))install.packages("lubridate")

## Loading required package: lubridate

##   
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':  
##   
## date, intersect, setdiff, union

if(!require(readr))install.packages("readr")

## Loading required package: readr

if(!require(ggplot2))install.packages("ggplot2")

## Loading required package: ggplot2

if(!require(tidyverse))install.packages("tidyverse")

## Loading required package: tidyverse

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.2.1 ✔ stringr 1.4.1   
## ✔ purrr 0.3.5 ✔ forcats 0.5.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ lubridate::as.difftime() masks base::as.difftime()  
## ✖ lubridate::date() masks base::date()  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ lubridate::intersect() masks base::intersect()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ lubridate::setdiff() masks base::setdiff()  
## ✖ lubridate::union() masks base::union()

if(!require(caret)) install.packages("caret")

## Loading required package: caret  
## Loading required package: lattice  
##   
## Attaching package: 'caret'  
##   
## The following object is masked from 'package:purrr':  
##   
## lift

if(!require(rpart)) install.packages("rpart")

## Loading required package: rpart

if(!require(dplyr)) install.packages("dplyr")  
if(!require(rpart.plot)) installed.packages("rpart.plot")

## Loading required package: rpart.plot

## Warning: package 'rpart.plot' was built under R version 4.2.2

library(rpart.plot)  
library(caret)  
library(tidyverse)  
library(rpart)  
library(readr)  
library(dplyr)  
library(ggplot2)  
library(lubridate)

#Set working directory

getwd()

## [1] "C:/Users/eboley/OneDrive - Partners In Health/Desktop/Data/project"

setwd("C:/Users/eboley/OneDrive - Partners In Health/Desktop/Data/project")

#Set Relative folder path

College <- read.csv("R/College\_admission.csv")

# 3 significant digits

options(digits = 3)

#Recoding categorical & numerical variables

College\_clean <- College %>%  
 mutate(GreLevels=ifelse(gre<440,"Low",ifelse(gre<580,"Medium","High"))) %>%  
 mutate(Gender=ifelse(Gender\_Male == 0, "female","male")) %>%  
 mutate(Demo = recode(Race,  
 "1" = "Hispanics",  
 "2" = "Asian",  
 "3" = "African-American"))%>%  
 mutate(Socioeco = recode(ses,  
 "1" = "Low",  
 "2" = "Medium",  
 "3" = "High"))

#Summary of data

summary(College\_clean)

## admit gre gpa ses Gender\_Male   
## Min. :0.000 Min. :220 Min. :2.26 Min. :1.00 Min. :0.000   
## 1st Qu.:0.000 1st Qu.:520 1st Qu.:3.13 1st Qu.:1.00 1st Qu.:0.000   
## Median :0.000 Median :580 Median :3.40 Median :2.00 Median :0.000   
## Mean :0.318 Mean :588 Mean :3.39 Mean :1.99 Mean :0.475   
## 3rd Qu.:1.000 3rd Qu.:660 3rd Qu.:3.67 3rd Qu.:3.00 3rd Qu.:1.000   
## Max. :1.000 Max. :800 Max. :4.00 Max. :3.00 Max. :1.000   
## Race rank GreLevels Gender   
## Min. :1.00 Min. :1.00 Length:400 Length:400   
## 1st Qu.:1.00 1st Qu.:2.00 Class :character Class :character   
## Median :2.00 Median :2.00 Mode :character Mode :character   
## Mean :1.96 Mean :2.48   
## 3rd Qu.:3.00 3rd Qu.:3.00   
## Max. :3.00 Max. :4.00   
## Demo Socioeco   
## Length:400 Length:400   
## Class :character Class :character   
## Mode :character Mode :character   
##   
##   
##

str(College\_clean)

## 'data.frame': 400 obs. of 11 variables:  
## $ admit : int 0 1 1 1 0 1 1 0 1 0 ...  
## $ gre : int 380 660 800 640 520 760 560 400 540 700 ...  
## $ gpa : num 3.61 3.67 4 3.19 2.93 3 2.98 3.08 3.39 3.92 ...  
## $ ses : int 1 2 2 1 3 2 2 2 1 1 ...  
## $ Gender\_Male: int 0 0 0 1 1 1 1 0 1 0 ...  
## $ Race : int 3 2 2 2 2 1 2 2 1 2 ...  
## $ rank : int 3 3 1 4 4 2 1 2 3 2 ...  
## $ GreLevels : chr "Low" "High" "High" "High" ...  
## $ Gender : chr "female" "female" "female" "male" ...  
## $ Demo : chr "African-American" "Asian" "Asian" "Asian" ...  
## $ Socioeco : chr "Low" "Medium" "Medium" "Low" ...

#Data type as numeric (admit, ses, race, gender, and rank), GreLevels as character variable

#Transforming categorical variables (admit, ses, gender, rank)

College\_clean$admit <- as.factor(College\_clean$admit)  
College\_clean$ses <- as.factor(College\_clean$ses)  
College\_clean$Race <- as.factor(College\_clean$Race)  
College\_clean$Gender\_Male <- as.factor(College\_clean$Gender\_Male)  
College\_clean$rank <- as.factor(College\_clean$rank)  
College\_clean$GreLevels <- as.factor(College\_clean$GreLevels)  
College\_clean$Gender <- as.factor(College\_clean$Gender)  
College\_clean$Demo <- as.factor(College\_clean$Demo)  
College\_clean$Socioeco <- as.factor(College\_clean$Socioeco)

#Finding missing data

is.na(College\_clean)

## admit gre gpa ses Gender\_Male Race rank GreLevels Gender Demo  
## [1,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [2,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [3,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [4,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [5,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [6,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [7,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [8,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [9,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [10,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [11,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [12,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [13,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [14,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [15,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [16,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [17,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [18,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [19,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [20,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [21,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [22,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [23,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [24,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [25,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [26,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [27,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [28,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [29,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [30,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [31,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [32,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [33,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [34,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [35,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [36,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [37,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [38,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [39,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [40,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [41,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [42,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [43,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [44,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [45,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [46,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [47,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [48,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [49,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [50,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [51,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [52,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [53,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [54,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [55,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [56,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [57,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [58,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [59,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [60,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [61,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [62,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [63,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [64,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [65,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [66,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [67,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [68,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [69,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [70,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [71,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [72,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [73,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [74,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [75,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [76,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [77,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [78,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [79,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [80,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [81,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [82,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [83,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [84,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [85,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [86,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [87,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [88,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [89,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [90,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [91,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [92,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [93,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [94,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [95,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [96,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [97,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [98,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [99,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [100,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [101,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [102,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [103,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [104,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [105,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [106,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [107,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [108,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [109,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [110,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [111,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [112,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [113,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [114,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [115,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [116,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [117,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [118,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [119,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [120,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [121,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [122,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [123,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [124,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [125,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [126,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [127,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [128,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [129,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [130,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [131,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [132,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [133,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [134,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [135,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [136,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [137,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [138,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [139,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [140,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [141,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [142,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [143,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [144,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [145,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [146,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [147,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [148,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [149,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [150,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [151,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [152,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [153,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [154,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [155,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [156,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [157,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [158,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [159,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [160,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [161,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [162,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [163,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [164,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [165,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [166,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [167,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [168,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [169,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [170,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [171,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [172,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [173,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [174,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [175,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [176,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [177,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [178,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [179,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [180,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [181,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [182,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [183,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [184,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [185,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [186,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [187,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [188,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [189,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [190,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [191,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [192,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [193,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [194,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [195,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [196,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [197,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [198,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [199,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [200,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [201,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [202,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [203,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [204,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [205,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [206,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [207,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [208,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [209,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [210,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [211,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [212,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [213,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [214,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [215,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [216,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [217,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [218,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [219,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [220,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [221,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [222,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [223,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [224,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [225,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [226,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [227,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [228,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [229,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [230,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [231,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [232,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [233,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [234,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [235,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [236,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [237,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [238,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [239,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [240,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [241,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [242,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [243,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [244,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [245,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [246,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [247,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [248,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [249,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [250,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [251,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [252,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [253,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [254,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [255,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [256,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [257,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [258,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [259,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [260,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [261,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [262,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [263,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [264,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [265,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [266,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [267,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [268,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [269,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [270,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [271,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [272,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [273,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [274,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [275,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [276,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [277,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [278,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [279,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [280,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [281,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [282,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [283,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [284,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [285,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [286,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [287,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [288,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [289,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [290,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [291,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [292,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [293,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [294,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [295,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [296,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [297,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [298,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [299,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [300,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [301,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [302,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [303,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [304,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [305,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [306,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [307,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [308,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [309,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [310,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [311,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [312,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [313,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [314,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [315,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [316,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [317,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [318,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [319,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [320,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [321,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [322,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [323,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [324,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [325,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [326,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [327,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [328,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [329,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [330,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [331,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [332,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [333,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [334,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [335,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [336,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [337,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [338,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [339,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [340,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [341,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [342,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [343,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [344,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [345,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [346,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [347,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [348,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [349,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [350,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [351,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [352,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [353,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [354,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [355,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [356,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [357,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [358,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [359,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [360,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [361,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [362,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [363,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [364,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [365,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [366,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [367,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [368,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [369,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [370,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [371,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [372,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [373,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [374,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [375,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [376,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [377,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [378,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [379,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [380,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [381,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [382,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [383,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [384,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [385,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [386,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [387,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [388,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [389,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [390,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [391,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [392,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [393,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [394,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [395,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [396,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [397,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [398,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [399,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [400,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## Socioeco  
## [1,] FALSE  
## [2,] FALSE  
## [3,] FALSE  
## [4,] FALSE  
## [5,] FALSE  
## [6,] FALSE  
## [7,] FALSE  
## [8,] FALSE  
## [9,] FALSE  
## [10,] FALSE  
## [11,] FALSE  
## [12,] FALSE  
## [13,] FALSE  
## [14,] FALSE  
## [15,] FALSE  
## [16,] FALSE  
## [17,] FALSE  
## [18,] FALSE  
## [19,] FALSE  
## [20,] FALSE  
## [21,] FALSE  
## [22,] FALSE  
## [23,] FALSE  
## [24,] FALSE  
## [25,] FALSE  
## [26,] FALSE  
## [27,] FALSE  
## [28,] FALSE  
## [29,] FALSE  
## [30,] FALSE  
## [31,] FALSE  
## [32,] FALSE  
## [33,] FALSE  
## [34,] FALSE  
## [35,] FALSE  
## [36,] FALSE  
## [37,] FALSE  
## [38,] FALSE  
## [39,] FALSE  
## [40,] FALSE  
## [41,] FALSE  
## [42,] FALSE  
## [43,] FALSE  
## [44,] FALSE  
## [45,] FALSE  
## [46,] FALSE  
## [47,] FALSE  
## [48,] FALSE  
## [49,] FALSE  
## [50,] FALSE  
## [51,] FALSE  
## [52,] FALSE  
## [53,] FALSE  
## [54,] FALSE  
## [55,] FALSE  
## [56,] FALSE  
## [57,] FALSE  
## [58,] FALSE  
## [59,] FALSE  
## [60,] FALSE  
## [61,] FALSE  
## [62,] FALSE  
## [63,] FALSE  
## [64,] FALSE  
## [65,] FALSE  
## [66,] FALSE  
## [67,] FALSE  
## [68,] FALSE  
## [69,] FALSE  
## [70,] FALSE  
## [71,] FALSE  
## [72,] FALSE  
## [73,] FALSE  
## [74,] FALSE  
## [75,] FALSE  
## [76,] FALSE  
## [77,] FALSE  
## [78,] FALSE  
## [79,] FALSE  
## [80,] FALSE  
## [81,] FALSE  
## [82,] FALSE  
## [83,] FALSE  
## [84,] FALSE  
## [85,] FALSE  
## [86,] FALSE  
## [87,] FALSE  
## [88,] FALSE  
## [89,] FALSE  
## [90,] FALSE  
## [91,] FALSE  
## [92,] FALSE  
## [93,] FALSE  
## [94,] FALSE  
## [95,] FALSE  
## [96,] FALSE  
## [97,] FALSE  
## [98,] FALSE  
## [99,] FALSE  
## [100,] FALSE  
## [101,] FALSE  
## [102,] FALSE  
## [103,] FALSE  
## [104,] FALSE  
## [105,] FALSE  
## [106,] FALSE  
## [107,] FALSE  
## [108,] FALSE  
## [109,] FALSE  
## [110,] FALSE  
## [111,] FALSE  
## [112,] FALSE  
## [113,] FALSE  
## [114,] FALSE  
## [115,] FALSE  
## [116,] FALSE  
## [117,] FALSE  
## [118,] FALSE  
## [119,] FALSE  
## [120,] FALSE  
## [121,] FALSE  
## [122,] FALSE  
## [123,] FALSE  
## [124,] FALSE  
## [125,] FALSE  
## [126,] FALSE  
## [127,] FALSE  
## [128,] FALSE  
## [129,] FALSE  
## [130,] FALSE  
## [131,] FALSE  
## [132,] FALSE  
## [133,] FALSE  
## [134,] FALSE  
## [135,] FALSE  
## [136,] FALSE  
## [137,] FALSE  
## [138,] FALSE  
## [139,] FALSE  
## [140,] FALSE  
## [141,] FALSE  
## [142,] FALSE  
## [143,] FALSE  
## [144,] FALSE  
## [145,] FALSE  
## [146,] FALSE  
## [147,] FALSE  
## [148,] FALSE  
## [149,] FALSE  
## [150,] FALSE  
## [151,] FALSE  
## [152,] FALSE  
## [153,] FALSE  
## [154,] FALSE  
## [155,] FALSE  
## [156,] FALSE  
## [157,] FALSE  
## [158,] FALSE  
## [159,] FALSE  
## [160,] FALSE  
## [161,] FALSE  
## [162,] FALSE  
## [163,] FALSE  
## [164,] FALSE  
## [165,] FALSE  
## [166,] FALSE  
## [167,] FALSE  
## [168,] FALSE  
## [169,] FALSE  
## [170,] FALSE  
## [171,] FALSE  
## [172,] FALSE  
## [173,] FALSE  
## [174,] FALSE  
## [175,] FALSE  
## [176,] FALSE  
## [177,] FALSE  
## [178,] FALSE  
## [179,] FALSE  
## [180,] FALSE  
## [181,] FALSE  
## [182,] FALSE  
## [183,] FALSE  
## [184,] FALSE  
## [185,] FALSE  
## [186,] FALSE  
## [187,] FALSE  
## [188,] FALSE  
## [189,] FALSE  
## [190,] FALSE  
## [191,] FALSE  
## [192,] FALSE  
## [193,] FALSE  
## [194,] FALSE  
## [195,] FALSE  
## [196,] FALSE  
## [197,] FALSE  
## [198,] FALSE  
## [199,] FALSE  
## [200,] FALSE  
## [201,] FALSE  
## [202,] FALSE  
## [203,] FALSE  
## [204,] FALSE  
## [205,] FALSE  
## [206,] FALSE  
## [207,] FALSE  
## [208,] FALSE  
## [209,] FALSE  
## [210,] FALSE  
## [211,] FALSE  
## [212,] FALSE  
## [213,] FALSE  
## [214,] FALSE  
## [215,] FALSE  
## [216,] FALSE  
## [217,] FALSE  
## [218,] FALSE  
## [219,] FALSE  
## [220,] FALSE  
## [221,] FALSE  
## [222,] FALSE  
## [223,] FALSE  
## [224,] FALSE  
## [225,] FALSE  
## [226,] FALSE  
## [227,] FALSE  
## [228,] FALSE  
## [229,] FALSE  
## [230,] FALSE  
## [231,] FALSE  
## [232,] FALSE  
## [233,] FALSE  
## [234,] FALSE  
## [235,] FALSE  
## [236,] FALSE  
## [237,] FALSE  
## [238,] FALSE  
## [239,] FALSE  
## [240,] FALSE  
## [241,] FALSE  
## [242,] FALSE  
## [243,] FALSE  
## [244,] FALSE  
## [245,] FALSE  
## [246,] FALSE  
## [247,] FALSE  
## [248,] FALSE  
## [249,] FALSE  
## [250,] FALSE  
## [251,] FALSE  
## [252,] FALSE  
## [253,] FALSE  
## [254,] FALSE  
## [255,] FALSE  
## [256,] FALSE  
## [257,] FALSE  
## [258,] FALSE  
## [259,] FALSE  
## [260,] FALSE  
## [261,] FALSE  
## [262,] FALSE  
## [263,] FALSE  
## [264,] FALSE  
## [265,] FALSE  
## [266,] FALSE  
## [267,] FALSE  
## [268,] FALSE  
## [269,] FALSE  
## [270,] FALSE  
## [271,] FALSE  
## [272,] FALSE  
## [273,] FALSE  
## [274,] FALSE  
## [275,] FALSE  
## [276,] FALSE  
## [277,] FALSE  
## [278,] FALSE  
## [279,] FALSE  
## [280,] FALSE  
## [281,] FALSE  
## [282,] FALSE  
## [283,] FALSE  
## [284,] FALSE  
## [285,] FALSE  
## [286,] FALSE  
## [287,] FALSE  
## [288,] FALSE  
## [289,] FALSE  
## [290,] FALSE  
## [291,] FALSE  
## [292,] FALSE  
## [293,] FALSE  
## [294,] FALSE  
## [295,] FALSE  
## [296,] FALSE  
## [297,] FALSE  
## [298,] FALSE  
## [299,] FALSE  
## [300,] FALSE  
## [301,] FALSE  
## [302,] FALSE  
## [303,] FALSE  
## [304,] FALSE  
## [305,] FALSE  
## [306,] FALSE  
## [307,] FALSE  
## [308,] FALSE  
## [309,] FALSE  
## [310,] FALSE  
## [311,] FALSE  
## [312,] FALSE  
## [313,] FALSE  
## [314,] FALSE  
## [315,] FALSE  
## [316,] FALSE  
## [317,] FALSE  
## [318,] FALSE  
## [319,] FALSE  
## [320,] FALSE  
## [321,] FALSE  
## [322,] FALSE  
## [323,] FALSE  
## [324,] FALSE  
## [325,] FALSE  
## [326,] FALSE  
## [327,] FALSE  
## [328,] FALSE  
## [329,] FALSE  
## [330,] FALSE  
## [331,] FALSE  
## [332,] FALSE  
## [333,] FALSE  
## [334,] FALSE  
## [335,] FALSE  
## [336,] FALSE  
## [337,] FALSE  
## [338,] FALSE  
## [339,] FALSE  
## [340,] FALSE  
## [341,] FALSE  
## [342,] FALSE  
## [343,] FALSE  
## [344,] FALSE  
## [345,] FALSE  
## [346,] FALSE  
## [347,] FALSE  
## [348,] FALSE  
## [349,] FALSE  
## [350,] FALSE  
## [351,] FALSE  
## [352,] FALSE  
## [353,] FALSE  
## [354,] FALSE  
## [355,] FALSE  
## [356,] FALSE  
## [357,] FALSE  
## [358,] FALSE  
## [359,] FALSE  
## [360,] FALSE  
## [361,] FALSE  
## [362,] FALSE  
## [363,] FALSE  
## [364,] FALSE  
## [365,] FALSE  
## [366,] FALSE  
## [367,] FALSE  
## [368,] FALSE  
## [369,] FALSE  
## [370,] FALSE  
## [371,] FALSE  
## [372,] FALSE  
## [373,] FALSE  
## [374,] FALSE  
## [375,] FALSE  
## [376,] FALSE  
## [377,] FALSE  
## [378,] FALSE  
## [379,] FALSE  
## [380,] FALSE  
## [381,] FALSE  
## [382,] FALSE  
## [383,] FALSE  
## [384,] FALSE  
## [385,] FALSE  
## [386,] FALSE  
## [387,] FALSE  
## [388,] FALSE  
## [389,] FALSE  
## [390,] FALSE  
## [391,] FALSE  
## [392,] FALSE  
## [393,] FALSE  
## [394,] FALSE  
## [395,] FALSE  
## [396,] FALSE  
## [397,] FALSE  
## [398,] FALSE  
## [399,] FALSE  
## [400,] FALSE

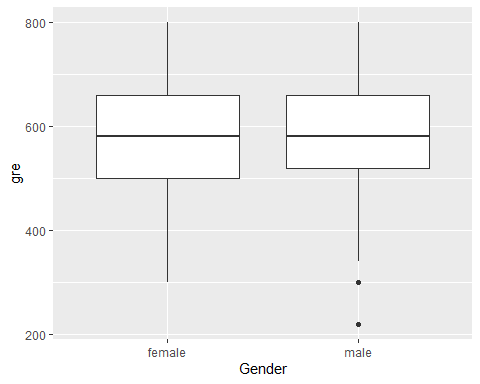
sum(is.na(College\_clean))

## [1] 0

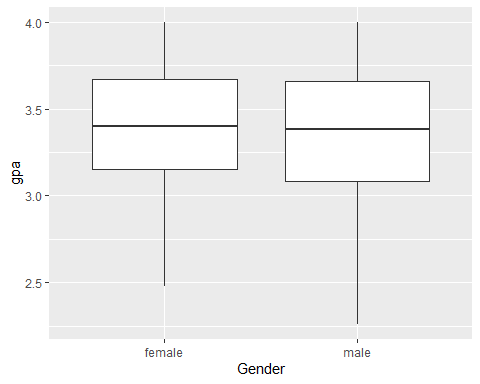
#there is no missing values in dataset

#Distribution of gre & gpa scores among gender

College\_clean %>%  
 ggplot(aes(Gender,gre))+  
 geom\_boxplot()

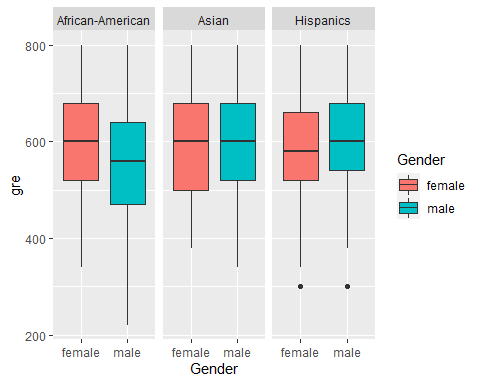


College\_clean %>%  
 ggplot(aes(Gender,gpa))+  
 geom\_boxplot()

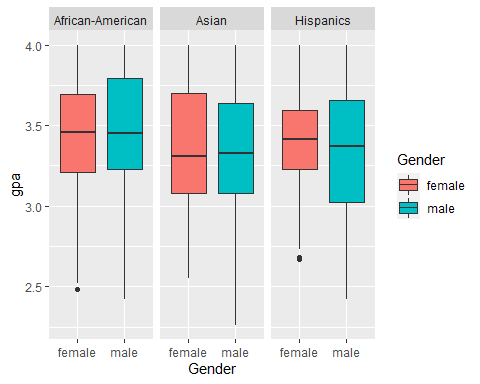


#Variation in gre & gpa by gender, race and socioenomic status

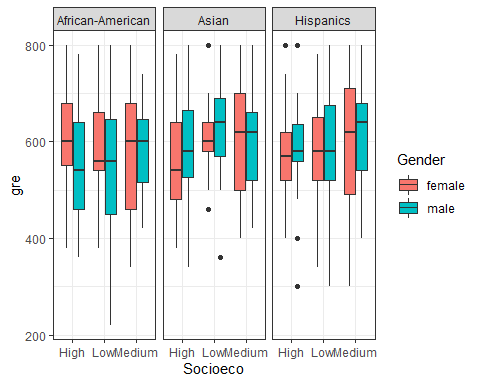
College\_clean %>%  
 ggplot(aes(Gender,gre, fill = Gender))+  
 geom\_boxplot()+  
 facet\_grid(~Demo)



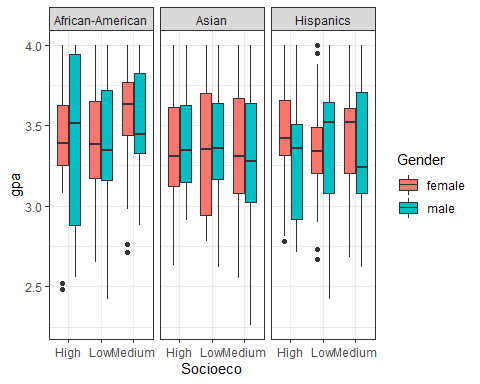
College\_clean %>%  
 ggplot(aes(Gender,gpa, fill = Gender))+  
 geom\_boxplot()+  
 facet\_grid(~Demo)



College\_clean %>%  
 ggplot(aes(Socioeco, gre, fill = Gender))+  
 #geom\_density(alpha = 0.5)+  
 geom\_boxplot()+  
 facet\_wrap(~Demo)+  
 theme\_bw()

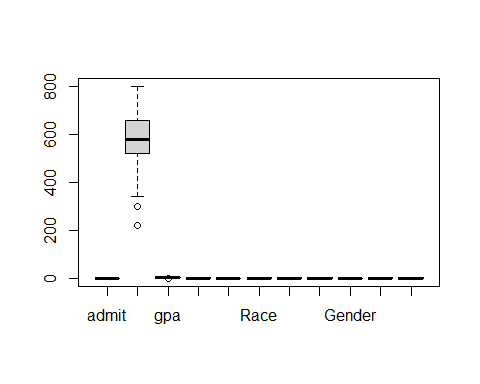


College\_clean %>%  
 ggplot(aes(Socioeco, gpa, fill = Gender))+  
 #geom\_density(alpha = 0.5)+  
 geom\_boxplot()+  
 facet\_wrap(~Demo)+  
 theme\_bw()

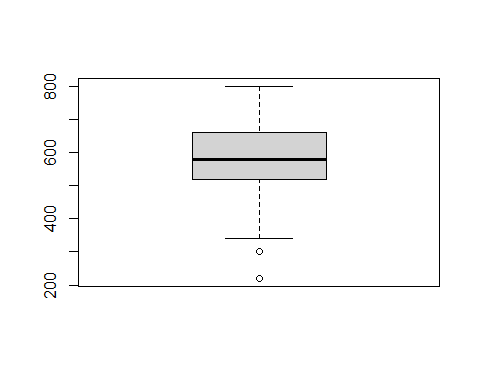


#Find outlers in data set

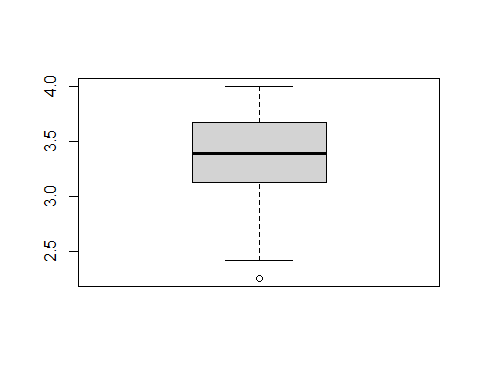
boxplot(College\_clean)



boxplot(College\_clean$gre)



boxplot(College\_clean$gpa)



#Identifying of outliers

boxplot.stats(College\_clean$gpa)$out

## [1] 2.26

boxplot.stats(College\_clean$gre)$out

## [1] 300 300 220 300

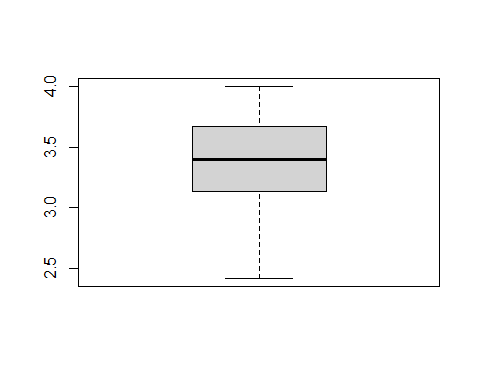
#GPA outlier is 2.26  
#GRE outlier are 300 300 220 300

#Removing outliers

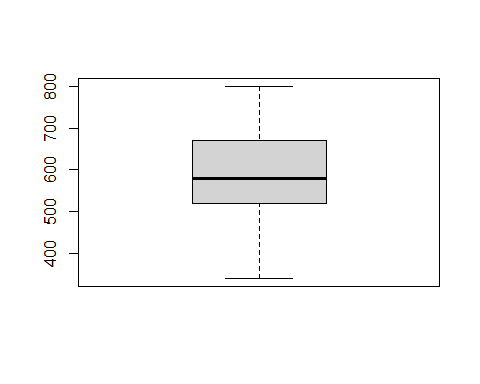
College\_clean <- subset(College\_clean, gpa != 2.26)  
College\_clean <- subset(College\_clean, gre != 300 & gre !=220)

#Confirmation outliers removed

boxplot(College\_clean$gpa)



boxplot(College\_clean$gre)



#Creating train and test sets

set.seed(42, sample.kind = "Rounding")

## Warning in set.seed(42, sample.kind = "Rounding"): non-uniform 'Rounding'  
## sampler used

test\_index <- createDataPartition(College\_clean$admit, times = 1, p = 0.2, list = FALSE)  
College\_test <- College\_clean[test\_index,]  
College\_train <- College\_clean[-test\_index,]  
  
nrow(College\_train)

## [1] 315

nrow(College\_test)

## [1] 80

#Proportion of individuals in the training set admitted

mean(College\_train$admit == 1)

## [1] 0.317

#Accuracy of guessing method

set.seed(3, sample.kind = "Rounding")

## Warning in set.seed(3, sample.kind = "Rounding"): non-uniform 'Rounding' sampler  
## used

guess <- sample(c(0,1), nrow(College\_test), replace = TRUE)  
mean(guess == College\_test$admit)

## [1] 0.425

#Proportion of females admitted

College\_train %>%  
 group\_by(Gender\_Male) %>%  
 summarize(Admit = mean(admit == 1)) %>%  
 filter(Gender\_Male == "0") %>%  
 pull(Admit)

## [1] 0.327

#Proportion of males admitted

College\_train %>%  
 group\_by(Gender\_Male) %>%  
 summarize(Admit = mean(admit == 1)) %>%  
 filter(Gender\_Male == "1") %>%  
 pull(Admit)

## [1] 0.307

#Predicting admission by sex # predict admission = 1 if male, 0 if female

sex\_model <- ifelse(College\_test$Gender == "female", 0, 1)

#Calculate accuracy

mean(sex\_model == College\_test$admit)

## [1] 0.5

#Predicting admission by socioeconomic class

College\_train %>%  
 group\_by(Socioeco) %>%  
 summarize(Admit = mean(admit == 1))

## # A tibble: 3 × 2  
## Socioeco Admit  
## <fct> <dbl>  
## 1 High 0.287  
## 2 Low 0.343  
## 3 Medium 0.321

#Accuracy of class-based prediction method # predict admission only if socioeconomic class is low

class\_model <- ifelse(College\_test$ses == 1, 1, 0)

#Calculate accuracy

mean(class\_model == College\_test$admit)

## [1] 0.625

#Prediction of admission based on class and sex

College\_train %>%  
 group\_by(Gender, Socioeco) %>%  
 summarize(admit = mean(admit == 1))

## `summarise()` has grouped output by 'Gender'. You can override using the  
## `.groups` argument.

## # A tibble: 6 × 3  
## # Groups: Gender [2]  
## Gender Socioeco admit  
## <fct> <fct> <dbl>  
## 1 female High 0.263  
## 2 female Low 0.368  
## 3 female Medium 0.353  
## 4 male High 0.318  
## 5 male Low 0.314  
## 6 male Medium 0.291

#filter(admit > 0.5)

#Prediction of admission based on class and sex

College\_train %>%  
 group\_by(Gender, Race) %>%  
 summarize(admit = mean(admit == 1))

## `summarise()` has grouped output by 'Gender'. You can override using the  
## `.groups` argument.

## # A tibble: 6 × 3  
## # Groups: Gender [2]  
## Gender Race admit  
## <fct> <fct> <dbl>  
## 1 female 1 0.321  
## 2 female 2 0.283  
## 3 female 3 0.385  
## 4 male 1 0.406  
## 5 male 2 0.25   
## 6 male 3 0.217

#Prediction of admission based on race and sex

College\_train %>%  
 group\_by(Gender, Race) %>%  
 summarize(admit = mean(admit == 1))

## `summarise()` has grouped output by 'Gender'. You can override using the  
## `.groups` argument.

## # A tibble: 6 × 3  
## # Groups: Gender [2]  
## Gender Race admit  
## <fct> <fct> <dbl>  
## 1 female 1 0.321  
## 2 female 2 0.283  
## 3 female 3 0.385  
## 4 male 1 0.406  
## 5 male 2 0.25   
## 6 male 3 0.217

#filter(admit > 0.5)

#Confusion matrix

confusionMatrix(data = factor(sex\_model), reference = factor(College\_test$admit))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 29 15  
## 1 25 11  
##   
## Accuracy : 0.5   
## 95% CI : (0.386, 0.614)  
## No Information Rate : 0.675   
## P-Value [Acc > NIR] : 1.000   
##   
## Kappa : -0.036   
##   
## Mcnemar's Test P-Value : 0.155   
##   
## Sensitivity : 0.537   
## Specificity : 0.423   
## Pos Pred Value : 0.659   
## Neg Pred Value : 0.306   
## Prevalence : 0.675   
## Detection Rate : 0.362   
## Detection Prevalence : 0.550   
## Balanced Accuracy : 0.480   
##   
## 'Positive' Class : 0   
##

confusionMatrix(data = factor(class\_model), reference = factor(College\_test$admit))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 41 17  
## 1 13 9  
##   
## Accuracy : 0.625   
## 95% CI : (0.51, 0.731)  
## No Information Rate : 0.675   
## P-Value [Acc > NIR] : 0.858   
##   
## Kappa : 0.11   
##   
## Mcnemar's Test P-Value : 0.584   
##   
## Sensitivity : 0.759   
## Specificity : 0.346   
## Pos Pred Value : 0.707   
## Neg Pred Value : 0.409   
## Prevalence : 0.675   
## Detection Rate : 0.512   
## Detection Prevalence : 0.725   
## Balanced Accuracy : 0.553   
##   
## 'Positive' Class : 0   
##

#F means score

F\_meas(data = factor(sex\_model), reference = College\_test$admit)

## [1] 0.592

F\_meas(data = factor(class\_model), reference = College\_test$admit)

## [1] 0.732

#Admission by gre and gpa using LDA and QDA #The accuracy on the test set for the LDA model

set.seed(1, sample.kind = "Rounding")

## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' sampler  
## used

train\_lda <- train(admit ~ gpa, method = "lda", data = College\_train)  
lda\_preds <- predict(train\_lda, College\_test)  
mean(lda\_preds == College\_test$admit)

## [1] 0.675

train\_lda\_a <- train(admit ~ gre, method = "lda", data = College\_train)  
lda\_preds\_a <- predict(train\_lda\_a, College\_test)  
mean(lda\_preds\_a == College\_test$admit)

## [1] 0.675

#The accuracy on the test set for the QDA model

train\_qda <- train(admit ~ gpa, method = "qda", data = College\_train)  
qda\_preds <- predict(train\_qda, College\_test)  
mean(qda\_preds == College\_test$admit)

## [1] 0.675

train\_qda\_a <- train(admit ~ gre, method = "qda", data = College\_train)  
qda\_preds\_a <- predict(train\_qda\_a, College\_test)  
mean(qda\_preds\_a == College\_test$admit)

## [1] 0.675

#The accuracy of your model (using gre as the only predictor) on the test set

train\_glm\_gre <- train(admit ~ gre, method = "glm", data = College\_train)  
glm\_preds\_gre <- predict(train\_glm\_gre, College\_test)  
mean(glm\_preds\_gre == College\_test$admit)

## [1] 0.675

#The accuracy of your model (using these four predictors) on the test

train\_glm <- train(admit ~ gre + gpa + ses + Race, method = "glm", data = College\_train)  
glm\_preds <- predict(train\_glm, College\_test)  
mean(glm\_preds == College\_test$admit)

## [1] 0.725

#The accuracy of your model (using all predictors) on the test set

str(College\_train)

## 'data.frame': 315 obs. of 11 variables:  
## $ admit : Factor w/ 2 levels "0","1": 2 2 2 1 2 1 1 1 2 1 ...  
## $ gre : int 800 640 760 400 540 700 800 440 760 700 ...  
## $ gpa : num 4 3.19 3 3.08 3.39 3.92 4 3.22 4 3.08 ...  
## $ ses : Factor w/ 3 levels "1","2","3": 2 1 2 2 1 1 1 3 3 2 ...  
## $ Gender\_Male: Factor w/ 2 levels "0","1": 1 2 2 1 2 1 2 1 2 1 ...  
## $ Race : Factor w/ 3 levels "1","2","3": 2 2 1 2 1 2 1 2 2 2 ...  
## $ rank : Factor w/ 4 levels "1","2","3","4": 1 4 2 2 3 2 4 1 1 2 ...  
## $ GreLevels : Factor w/ 3 levels "High","Low","Medium": 1 1 1 2 3 1 1 3 1 1 ...  
## $ Gender : Factor w/ 2 levels "female","male": 1 2 2 1 2 1 2 1 2 1 ...  
## $ Demo : Factor w/ 3 levels "African-American",..: 2 2 3 2 3 2 3 2 2 2 ...  
## $ Socioeco : Factor w/ 3 levels "High","Low","Medium": 3 2 3 3 2 2 2 1 1 3 ...

college\_train\_all <- College\_train %>% select (admit:rank)  
college\_test\_all <- College\_test %>% select (admit:rank)  
  
train\_glm\_all <- train(admit ~ ., method = "glm", data = college\_train\_all)  
train\_glm\_all

## Generalized Linear Model   
##   
## 315 samples  
## 6 predictor  
## 2 classes: '0', '1'   
##   
## No pre-processing  
## Resampling: Bootstrapped (25 reps)   
## Summary of sample sizes: 315, 315, 315, 315, 315, 315, ...   
## Resampling results:  
##   
## Accuracy Kappa  
## 0.692 0.17

glm\_all\_preds <- predict(train\_glm\_all, college\_test\_all)  
glm\_all\_preds

## [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [39] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0  
## [77] 1 0 0 0  
## Levels: 0 1

mean(glm\_all\_preds == college\_test\_all$admit)

## [1] 0.7

#kNN model

set.seed(6, sample.kind = "Rounding")

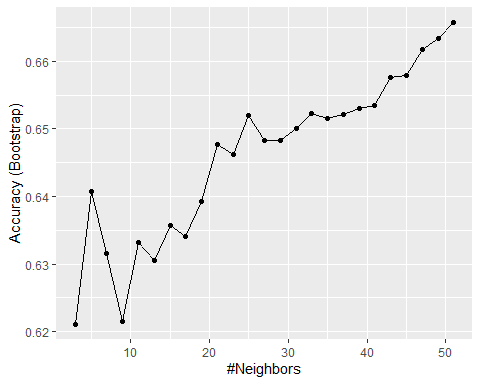
## Warning in set.seed(6, sample.kind = "Rounding"): non-uniform 'Rounding' sampler  
## used

train\_knn <- train(admit ~ .,  
 method = "knn",  
 data = college\_train\_all,  
 tuneGrid = data.frame(k = seq(3, 51, 2)))  
train\_knn$bestTune

## k  
## 25 51

#Highest Knn accuracy

ggplot(train\_knn)



#Accuracy of the kNN model

knn\_preds <- predict(train\_knn, college\_test\_all)  
mean(knn\_preds == college\_test\_all$admit)

## [1] 0.675

#Cross-validation of Knn

set.seed(8, sample.kind = "Rounding")

## Warning in set.seed(8, sample.kind = "Rounding"): non-uniform 'Rounding' sampler  
## used

train\_knn\_cv <- train(admit ~ .,  
 method = "knn",  
 data = college\_train\_all,  
 tuneGrid = data.frame(k = seq(3, 51, 2)),  
 trControl = trainControl(method = "cv", number = 10, p = 0.9))  
train\_knn\_cv$bestTune

## k  
## 25 51

#The accuracy of cross-validated kNN model

knn\_cv\_preds <- predict(train\_knn\_cv, college\_test\_all)  
mean(knn\_cv\_preds == college\_test\_all$admit)

## [1] 0.675

#Classification tree model

set.seed(10, sample.kind = "Rounding")

## Warning in set.seed(10, sample.kind = "Rounding"): non-uniform 'Rounding'  
## sampler used

train\_rpart <- train(admit ~ .,   
 method = "rpart",  
 tuneGrid = data.frame(cp = seq(0, 0.05, 0.002)),  
 data = college\_train\_all)  
train\_rpart$bestTune

## cp  
## 25 0.048

#The accuracy of the decision tree model

rpart\_preds <- predict(train\_rpart, college\_test\_all)  
mean(rpart\_preds == college\_test\_all$admit)

## [1] 0.675

#Inspect final model

train\_rpart$finalModel

## n= 315   
##   
## node), split, n, loss, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 315 100 0 (0.683 0.317) \*

#Random forest model

set.seed(14, sample.kind = "Rounding")

## Warning in set.seed(14, sample.kind = "Rounding"): non-uniform 'Rounding'  
## sampler used

train\_rf <- train(admit ~ .,  
 data = college\_train\_all,  
 method = "rf",  
 ntree = 100,  
 tuneGrid = data.frame(mtry = seq(1:7)))  
train\_rf$bestTune

## mtry  
## 1 1

#The accuracy of the random forest model

rf\_preds <- predict(train\_rf, college\_test\_all)  
mean(rf\_preds == college\_test\_all$admit)

## [1] 0.675

#The most important variable

varImp(train\_rf)

## rf variable importance  
##   
## Overall  
## gpa 100.000  
## gre 62.006  
## rank3 10.580  
## rank4 7.930  
## ses3 5.980  
## Race3 4.321  
## rank2 2.706  
## Race2 1.863  
## ses2 0.688  
## Gender\_Male1 0.000