# Homework 3

## 4375 Machine Learning with Dr. Mazidi

### Garrett Strealy

### 9/12/2021

This homework runs logistic regression to predict the binary feature of whether or not a person was admitted to graduate school, based on a set of predictors: GRE score, TOEFL score, rating of undergrad university attended, SOP statement of purpose, LOR letter or recommendation, Undergrad GPA, Research experience (binary).

The data set was downloaded from Kaggle: https://www.kaggle.com/mohansacharya/graduate-admissions (https://www.kaggle.com/mohansacharya/graduate-admissions)

The data is available in Piazza.

# Step 1 Load the data

- Load the data
- Examine the first few rows with head()

```
df <- read.csv("Admission_Predict.csv") # read in csv to data frame

head(df)
```

| | Serial.No.<br><int> | GRE.Sc…<br><int> | TOEFL.Score<br><int> | University.Rating<br><int> | …<br><dbl> | …<br><dbl> | C…<br><dbl> | Resear…<br><int> | Chance.of.Ac<br>< |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 337 | 118 | 4 | 4.5 | 4.5 | 9.65 | 1 | ( |
| 2 | 2 | 324 | 107 | 4 | 4.0 | 4.5 | 8.87 | 1 | ( |
| 3 | 3 | 316 | 104 | 3 | 3.0 | 3.5 | 8.00 | 1 | ( |
| 4 | 4 | 322 | 110 | 3 | 3.5 | 2.5 | 8.67 | 1 | ( |
| 5 | 5 | 314 | 103 | 2 | 2.0 | 3.0 | 8.21 | 0 | ( |
| 6 | 6 | 330 | 115 | 5 | 4.5 | 3.0 | 9.34 | 1 | ( |

6 rows

# Step 2 Data Wrangling

Perform the following steps:

- Make Research a factor
- Get rid of the Serial No column
- Make a new column that is binary factor based on if Chance.of.Admit > 0.5. Hint: See p. 40 in the book.
- Output column names with names() function

- Output a summary of the data
- Is the data set unbalanced? Why or why not?

Your commentary here:

The data is unbalanced. In Admit, there are many more True values than False, meaning many more students are Admit to be admitted than not.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

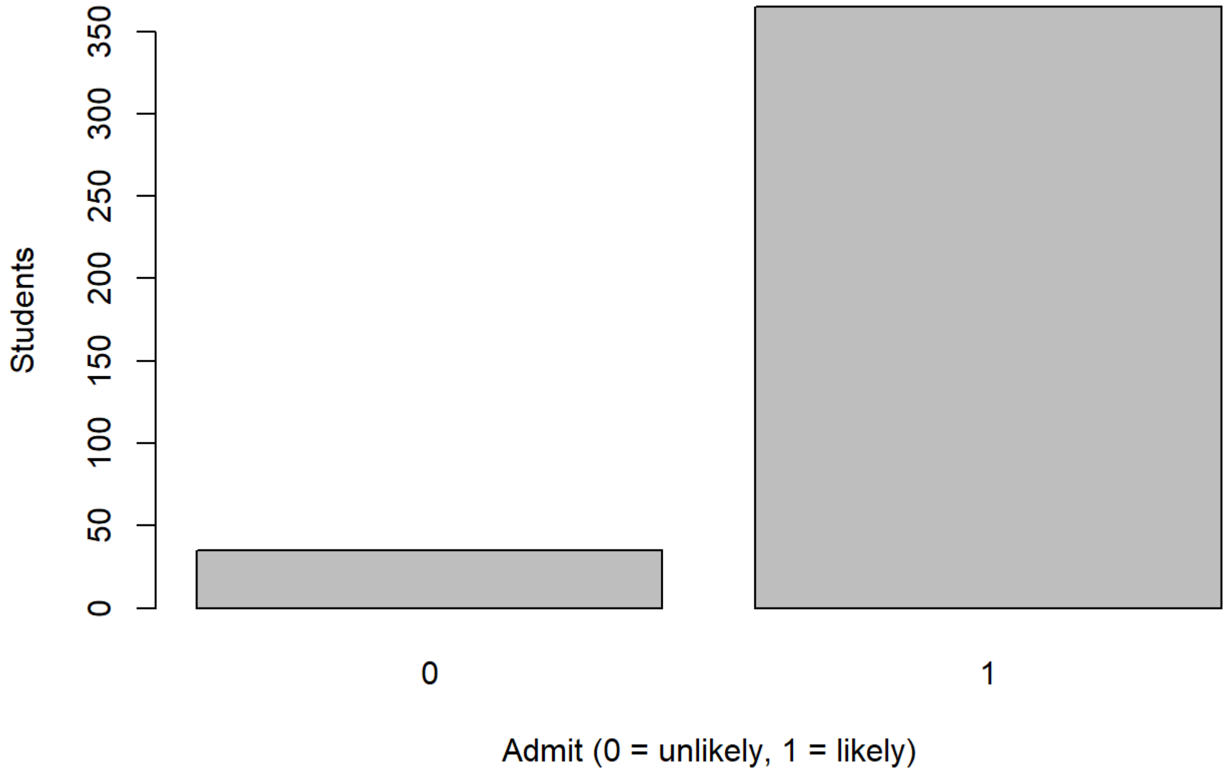
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

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

```
# Make Research a factor
df$Research <- factor(df$Research, levels = c("0", "1"), labels = c("No", "Yes"))

# get rid of the Serial No column
df <- select(df, -Serial.No.)

# make new column Admit based on if Chance.of.Admit > 0.5
# 1 = likely to be admitted, 0 = unlikely
df$Admit <- 0 # create column
df$Admit[df$Chance.of.Admit > 0.5] <- 1 # assign values to each observation
df$Admit <- factor(df$Admit) # convert column to factor

# output column names
names(df)
```

```
## [1] "GRE.Score"         "TOEFL.Score"       "University.Rating"
## [4] "SOP"               "LOR"               "CGPA"
## [7] "Research"          "Chance.of.Admit"   "Admit"
```

```
# check for balance
plot(df$Admit, xlab = "Admit (0 = unlikely, 1 = likely)", ylab = "Students")
```



```
summary(df)
```

```
##    GRE.Score      TOEFL.Score    University.Rating      SOP     
##  Min.   :290.0   Min.   : 92.0   Min.   :1.000     Min.   :1.0  
##  1st Qu.:308.0   1st Qu.:103.0   1st Qu.:2.000     1st Qu.:2.5  
##  Median :317.0   Median :107.0   Median :3.000     Median :3.5  
##  Mean   :316.8   Mean   :107.4   Mean   :3.087     Mean   :3.4  
##  3rd Qu.:325.0   3rd Qu.:112.0   3rd Qu.:4.000     3rd Qu.:4.0  
##  Max.   :340.0   Max.   :120.0   Max.   :5.000     Max.   :5.0  
##       LOR             CGPA       Research  Chance.of.Admit  Admit  
##  Min.   :1.000   Min.   :6.800   No :181   Min.   :0.3400   0: 35  
##  1st Qu.:3.000   1st Qu.:8.170   Yes:219   1st Qu.:0.6400   1:365  
##  Median :3.500   Median :8.610             Median :0.7300          
##  Mean   :3.453   Mean   :8.599             Mean   :0.7244          
##  3rd Qu.:4.000   3rd Qu.:9.062             3rd Qu.:0.8300          
##  Max.   :5.000   Max.   :9.920             Max.   :0.9700          
```

# Step 3 Data Visualization

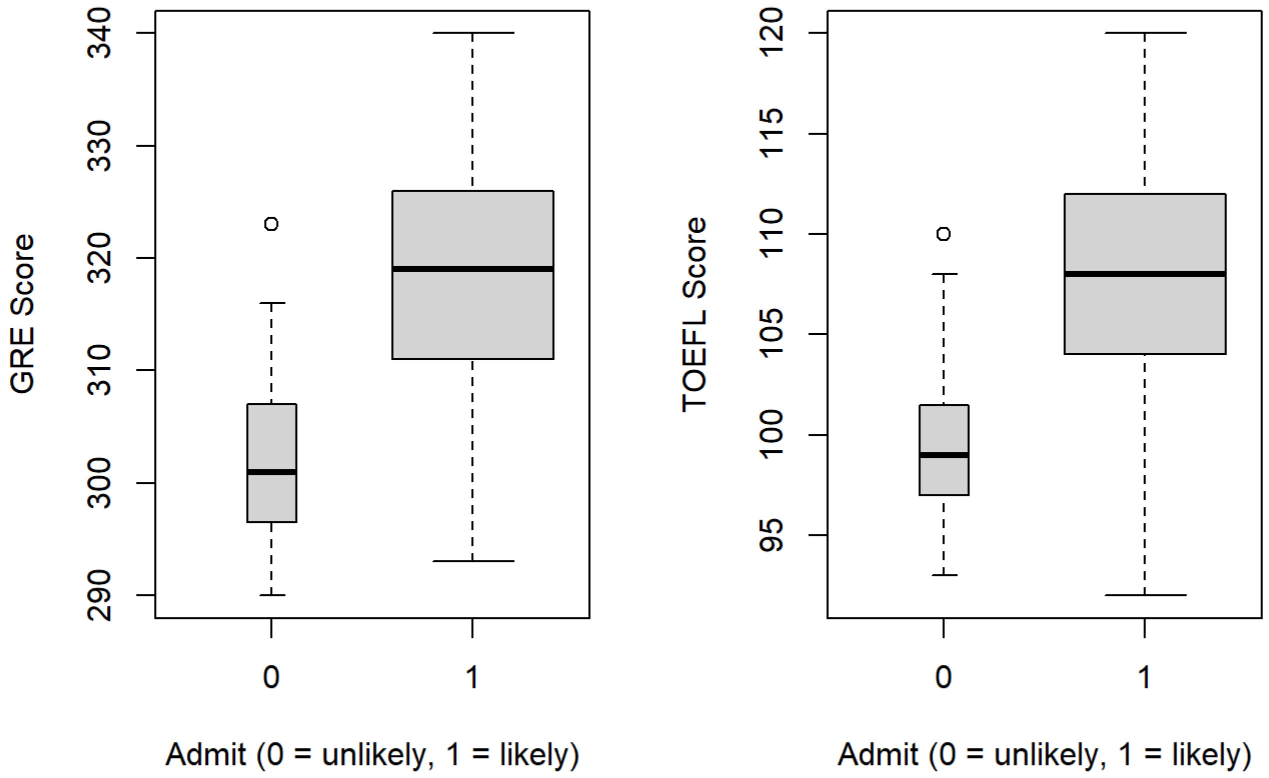
- Create a side-by-side graph with Admit on the x axis of both graphs, GRE score on the y axis of one graph and TOEFL score on the y axis of the other graph; save/restore the original graph parameters
- Comment on the graphs and what they are telling you about whether GRE and TOEFL are good predictors
- You will get a lot of warnings, you can suppress them with disabling warnings as shown below:

```
{r,warning=FALSE}
```

Your commentary here:

These graphs tell me that there is a strong positive correlation between GRE and Admit and between TOEFL and Admit. GRE and TOEFL are both good predictors of Admit.

```
par(mfrow=c(1,2))
plot(df$Admit, df$GRE.Score, data=df, xlab="Admit (0 = unlikely, 1 = likely)", ylab="GRE Score",
varwidth=TRUE)
plot(df$Admit, df$TOEFL.Score, data = df, xlab="Admit (0 = unlikely, 1 = likely)", ylab="TOEFL S
core", varwidth=TRUE)
```



# Step 4 Divide train/test

- Divide into 75/25 train/test, using seed 1234

```
# enable reproduction
set.seed(1234)

# create data frame i, consisting of the row indexes of -
# 75% of the rows in df, randomly sampled
i <- sample(1:nrow(df), 0.75*nrow(df), replace=FALSE)

train <- df[i,] # 75% train
test <- df[-i,] # 25% test
```

# Step 5 Build a Model with all predictors

- Build a model, predicting Admit from all predictors
- Output a summary of the model
- Did you get an error? Why? Hint: see p. 120 Warning

Your commentary here:

I received two error messages because the training data is perfectly or nearly perfectly linearly separable. R gave these warnings due to the inability to maximize the likelihood which already has separated the data perfectly.

```
glm1 <- glm(Admit ~ ., data=train, family=binomial)
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(glm1)
```

```
## 
## Call:
## glm(formula = Admit ~ ., family = binomial, data = train)
## 
## Deviance Residuals: 
##        Min          1Q      Median          3Q         Max  
## -9.801e-05   2.100e-08   2.100e-08   2.100e-08   1.123e-04  
## 
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -6.465e+02  2.921e+05  -0.002    0.998
## GRE.Score         -3.617e-01  9.554e+02   0.000    1.000
## TOEFL.Score        3.551e+00  3.562e+03   0.001    0.999
## University.Rating -5.000e+00  1.511e+04   0.000    1.000
## SOP               -7.867e+00  1.262e+04  -0.001    1.000
## LOR               -4.673e+00  1.970e+04   0.000    1.000
## CGPA               3.605e+00  1.897e+04   0.000    1.000
## ResearchYes       -1.109e+01  1.199e+04  -0.001    0.999
## Chance.of.Admit    7.993e+02  1.610e+05   0.005    0.996
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 1.7685e+02  on 299  degrees of freedom
## Residual deviance: 5.7812e-08  on 291  degrees of freedom
## AIC: 18
## 
## Number of Fisher Scoring iterations: 25
```

# Step 6 Build a Model with all predictors except Chance.of.Admit

- Build another model, predicting Admit from all predictors *except* Chance.of.Admit
- Output a summary of the model
- Did you get an error? Why or why not?

There was no error when building this model because the training data is linearly separable.

```
glm2 <- glm(Admit ~ . - Chance.of.Admit, data=train, family=binomial)

summary(glm2)
```

```
##
## Call:
## glm(formula = Admit ~ . - Chance.of.Admit, family = binomial,
##     data = train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.98738   0.02404   0.08347   0.25965   1.79020
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -52.42714   12.25908  -4.277  1.9e-05 ***
## GRE.Score           0.01685    0.04566   0.369 0.712200
## TOEFL.Score         0.17305    0.10614   1.630 0.103027
## University.Rating  -0.66933    0.40166  -1.666 0.095631 .
## SOP                -0.81828    0.45026  -1.817 0.069161 .
## LOR                 1.22762    0.54752   2.242 0.024951 *
## CGPA                3.94613    1.07273   3.679 0.000235 ***
## ResearchYes         0.10073    0.73916   0.136 0.891600
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 176.854  on 299  degrees of freedom
## Residual deviance:  89.024  on 292  degrees of freedom
## AIC: 105.02
##
## Number of Fisher Scoring iterations: 8
```

# Step 7 Predict probabilities

- Predict the probabilities using type="response"
- Examine a few probabilities and the corresponding Chance.of.Admit values
- Run cor() on the predicted probs and the Chance.of.Admit, and output the correlation
- What do you conclude from this correlation.

Your commentary here:

I conclude that glm2 has a slightly positive correlation with

```
# predict probabilities using glm1
#probs1 <-  predict(glm1, newdata=test, type="response")
#pred1 <-  ifelse(probs1>0.5, 1, 0)

# examine glm1 probabilities and corresponding Chance.of.Admit
#print(paste("head of probs1:"))
#head(probs1)
#print(paste("head of test$Chance.of.Admit:"))
#head(test$Chance.of.Admit)

# predict probabilities using glm2
probs2 <-  predict(glm2, newdata=test, type="response")
pred2 <-  ifelse(probs2>0.5, 1, 0)

# examine glm2 probabilities and corresponding Chance.of.Admit
print(paste("head of probs2:"))
```

```
## [1] "head of probs2:"
```

```
head(probs2)
```

```
##         1         3         5         8         9        11
## 0.9999835 0.9165608 0.9779368 0.9262458 0.8046423 0.9907820
```

```
print(paste("head of test$Chance.of.Admit:"))
```

```
## [1] "head of test$Chance.of.Admit:"
```

```
head(test$Chance.of.Admit)
```

```
## [1] 0.92 0.72 0.65 0.68 0.50 0.52
```

```
# find correlation between predicted probabilities and Chance.of.Admit
cor.probs.2 <- cor(probs2, test$Chance.of.Admit)
print(paste("glm2 correlation with probs2 =", cor.probs.2))
```

```
## [1] "glm2 correlation with probs2 = 0.648545031338275"
```

```
#cor.probs.1 <- cor(probs1, test$Chance.of.Admit)
#print(paste("glm1 correlation with probs1 =", cor.probs.1))
```

# Step 8 Make binary predictions, print table and accuracy

- Run predict() again, this time making binary predictions
- Output a table comparing the predictions and the binary Admit column
- Calculate and output accuracy
- Was the model able to generalize well to new data?

Your commentary here:

The model generalized well to the new data with an accuracy of 92%. Testing this model on a a larger and more balanced data set would help remove skepticism about this accuracy result.

```
# make binary prediction on the test data using glm2
probs2 <- predict(glm2, newdata=test)
pred2 <- ifelse(probs2>0.5, 1, 0)
table(pred2, test$Admit)
```

```
##
## pred2  0  1
##     0  5  4
##     1  4 87
```

```
acc2 <- mean(pred2==test$Admit)
print(paste("accuracy of glm2 predictions: ", acc2))
```

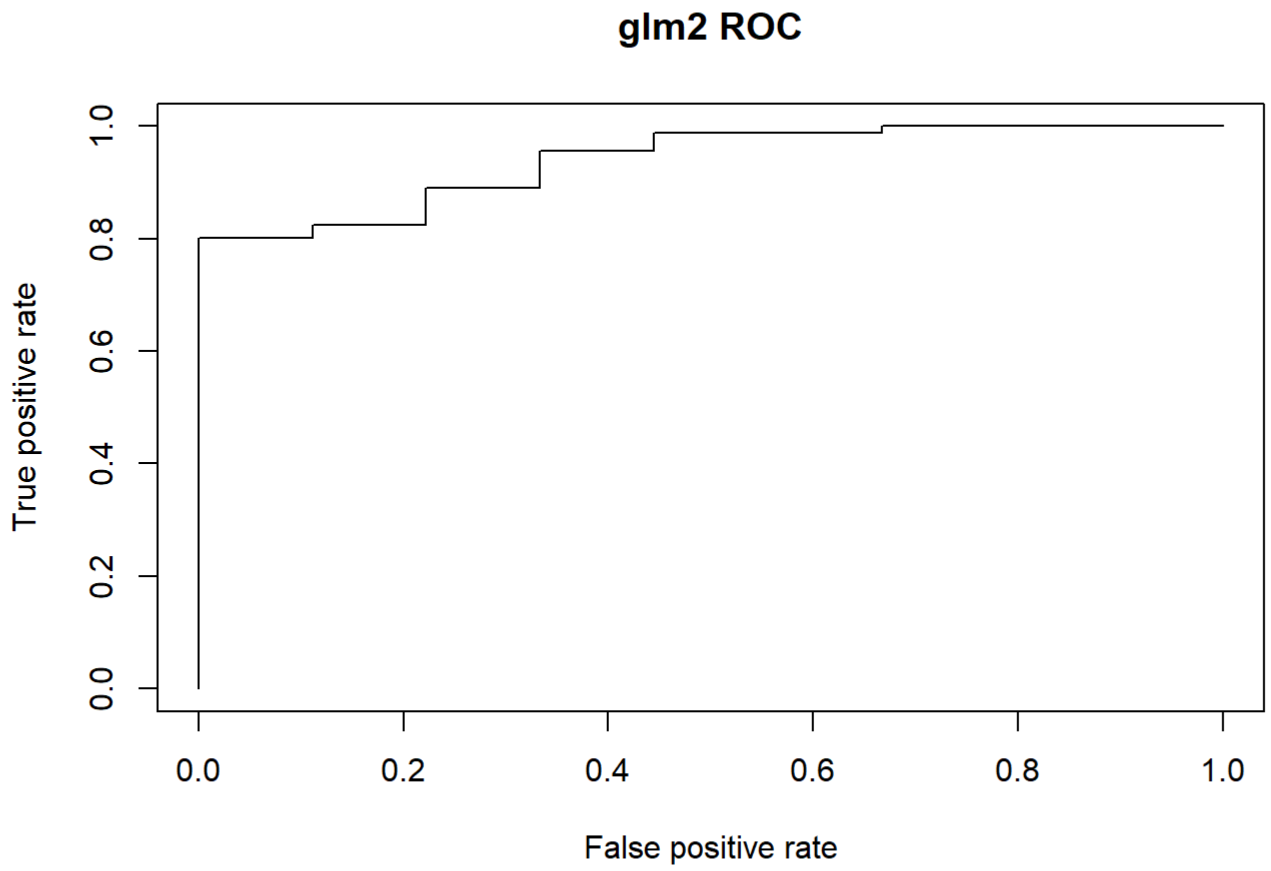
```
## [1] "accuracy of glm2 predictions:  0.92"
```

```
# make binary prediction on the test data using glm1
#probs1 <- predict(glm1, newdata=test)
#pred1 <- ifelse(probs1>0.5, 1, 0)
#table(pred1, test$Admit)

#acc1 <- mean(pred1==test$Admit)
#print(paste("accuracy of glm1 predictions: ", acc1))
```

# Step 9 Output ROCR and AUC

- Output a ROCR graph
- Extract and output the AUC metric

```
# your code here
library(ROCR)

p2 <- predict(glm2, newdata=test, type="response")
pr2 <- prediction(p2, test$Admit)
# TPR = sensitivity, FPR=specificity
prf2 <- performance(pr2, measure = "tpr", x.measure = "fpr")
plot(prf2, main="glm2 ROC")
```



```
auc2 <- performance(pr2, measure = "auc")
auc2 <- auc2@y.values[[1]]
print(paste("glm2 AUC: ", auc2))
```

```
## [1] "glm2 AUC:  0.938949938949939"
```

```
#p <- predict(glm1, newdata=test, type="response")
#pr <- prediction(p, test$Admit)
# TPR = sensitivity, FPR = specificity
#prf <- performance(pr, measure = "tpr", x.measure = "fpr")
#plot(prf, main="glm1 ROC")

#auc <- performance(pr, measure = "auc")
#auc <- auc@y.values[[1]]
#print(paste("glm1 AUC: ", auc))
```

# Step 10

- Make two more graphs and comment on what you learned from each graph:
    - Admit on x axis, SOP on y axis
    - Research on x axis, SOP on y axis

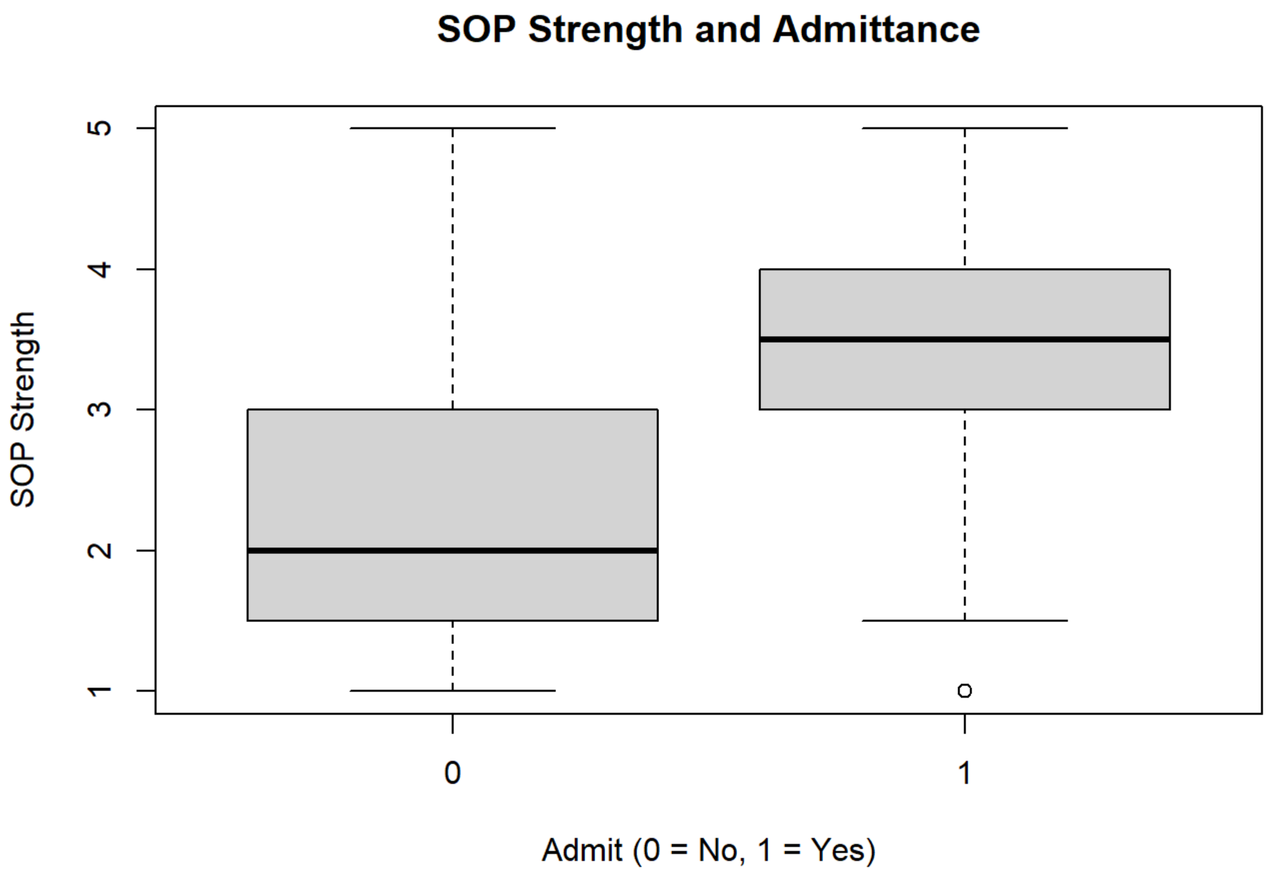
Your commentary here:

Plot 1 shows that students likely to be admitted have a significantly higher median SOP than students unlikely to be admitted (approximately 3.5 vs 2). This shows the importance of SOP in applicants' chances of admission.

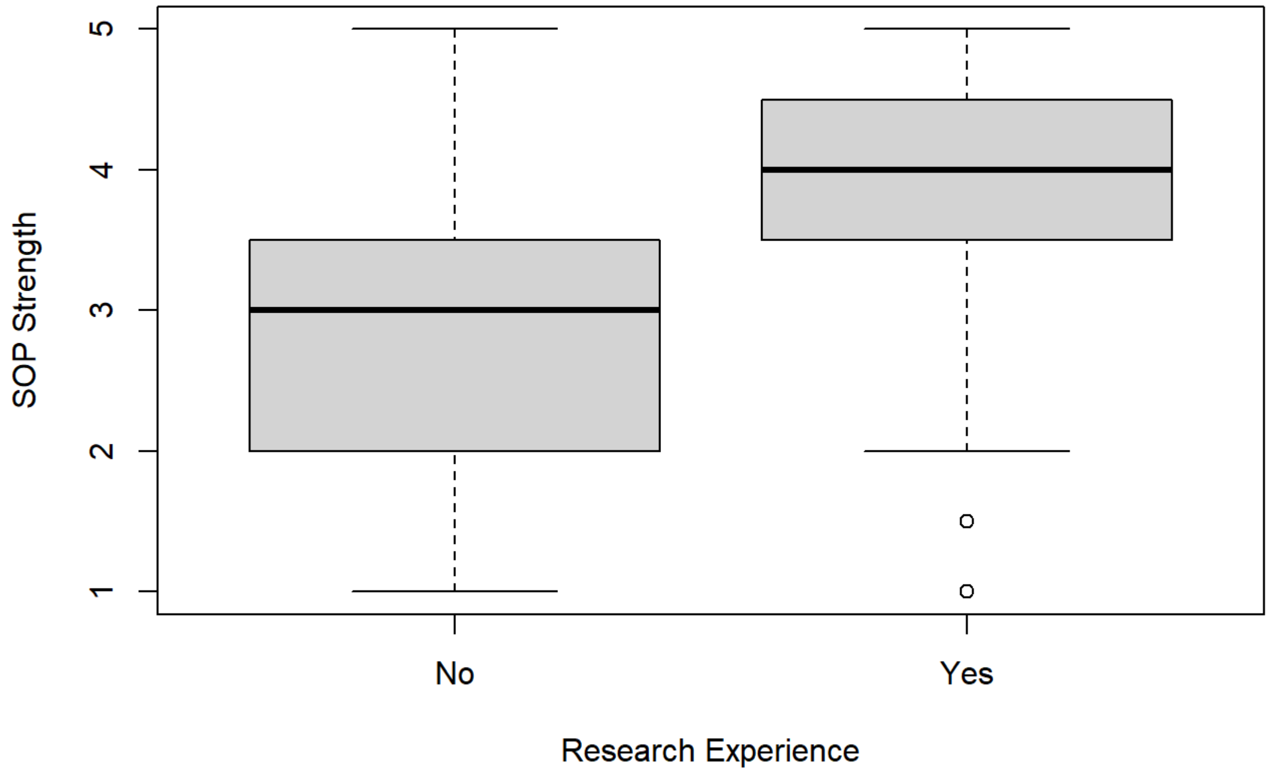
Plot 2 shows that students with research experience have a significantly higher median SOP than students without research experience (approximately 4 vs 3).This shows that students with research experience are likely to have a higher SOP.

My conclusion is that future applicants should consider research as a means to improve their chances of admittance.

```
# plot 1
plot(df$Admit, df$SOP, main="SOP Strength and Admittance",  xlab="Admit (0 = No, 1 = Yes)", ylab
="SOP Strength")
```



```
# plot 2
plot(df$Research, df$SOP, xlab="Research Experience", ylab="SOP Strength")
```


SOP Strength
5
4
3
2
1
No
Yes
Research Experience