Can Medical School Acceptance be Predicted Based on Your Application Statistics?

Contents

1. Introduction

The medical school admissions process is a topic of concern for many pre-med students, as medical school acceptances are very competitive and can be unpredictable. There are many parts to a medical school application, and discerning which sections are the most impactful for a successful applicant is widely debated. Therefore, data was collected on different parts of the application to evaluate which sections of a medical school application are the most impactful, to ultimately determine if medical school admissions can be predicted.

First a correlation matrix was visualized to observe which pairs of variables are the most (or least) correlated. Next, linear regression and k-nearest neighbors (kNN) prediction models were created, calculating the AUC values for each ROC curve to get a measure of how accurate the models are. In addition, 10-fold cross validation analysis was performed to determine potential signs of overfitting/underfitting. Then, dimensionality reduction was conducted through principle component analysis (PCA). Lastly, in order to observe clustering, the k-means method was utilized.

Data of the application statistics was obtained from Kaggle (Link), which contained data of 55 applicants. The variables it contained were the application result (accepted or denied), sex, BCPM (biology, chemistry, physics, math GPA), GPA (cumulative GPA), VR, PS, WS, BS (4 sections of the "old MCAT"), MCAT (scaled on "old" MCAT scores), and number of applications the individual submitted.

In order to make the results more relevant, the "old" MCAT score scale (3-45) was converted to the current MCAT score scale (472-528). In order to do this, an MCAT score converter was used (Link), which contained the old MCAT score and what the equivalent new MCAT score would be. After merging and cleaning, the final dataset used for analysis contained 53 observations with the variables sex, sGPA (science GPA), cGPA (cumulative GPA), MCAT ("new" MCAT score), number of applications, and the application result, which was used as the outcome variable.

This report aims to address two research questions 1) Which factor(s) are the most impactful in determining acceptance into medical school 2) Can medical school acceptance be accurately predicted based on applicant statistics?

It is important to identify that there are notable limitations with this report. The dataset used for analysis did not contain data on other important parts of the medical school application such as recommendation

letters, personal statement, whether they are an in-state applicant, etc. In addition, it does not include and would be difficult to measure the results of the interviews. Nevertheless, the variables used for analysis in this report are still important factors for an applicant, with the reminder that this report is not conclusive.

a. Set up

```
# Load packages
library(tidyverse)
library(readxl)
library(ade4)
library(plotROC)
library(caret)
library(rpart)
library(rpart.plot)
library(ggcorrplot)
library(factoextra)
library(cluster)
```

b. Dataset Creation

```
# import data
med_data <- read.csv("~/Desktop/UT CS:DS/SDS322E/Project/Project2/MedGPA.csv")

# clean data
med_data_clean <- med_data %>%
  select(-X, -Accept, -VR, -PS, -WS, -BS) %>%
  filter(MCAT <= 45) %>%
  na.omit()

# import data to convert old mcat scores to new mcat scores
MCAT_Conversion <- read_excel("~/Desktop/UT CS:DS/SDS322E/Project/Project2/MCAT_Conversion.xlsx")

# join data
admissions_data <- left_join(med_data_clean, MCAT_Conversion, by = c("MCAT" = "old_MCAT"))

# clean data
admissions_data <- admissions_data %>%
  select(-MCAT) %>%
  rename(sGPA = BCPM,
         cGPA = GPA,
         MCAT = new_MCAT) %>%
  relocate(MCAT, .after = cGPA)
```

2. Exploratory Data Analysis

a. Correlation Matrix

```r
# create dataframe with only numeric variables
admissions_data_numeric <- admissions_data %>%
  select(-Sex)

# correlation matrix
ggcorrplot(cor(admissions_data_numeric),
           type = "upper", # upper diagonal
           lab = TRUE, # print values
           method = "circle") # use circles with different sizes
```

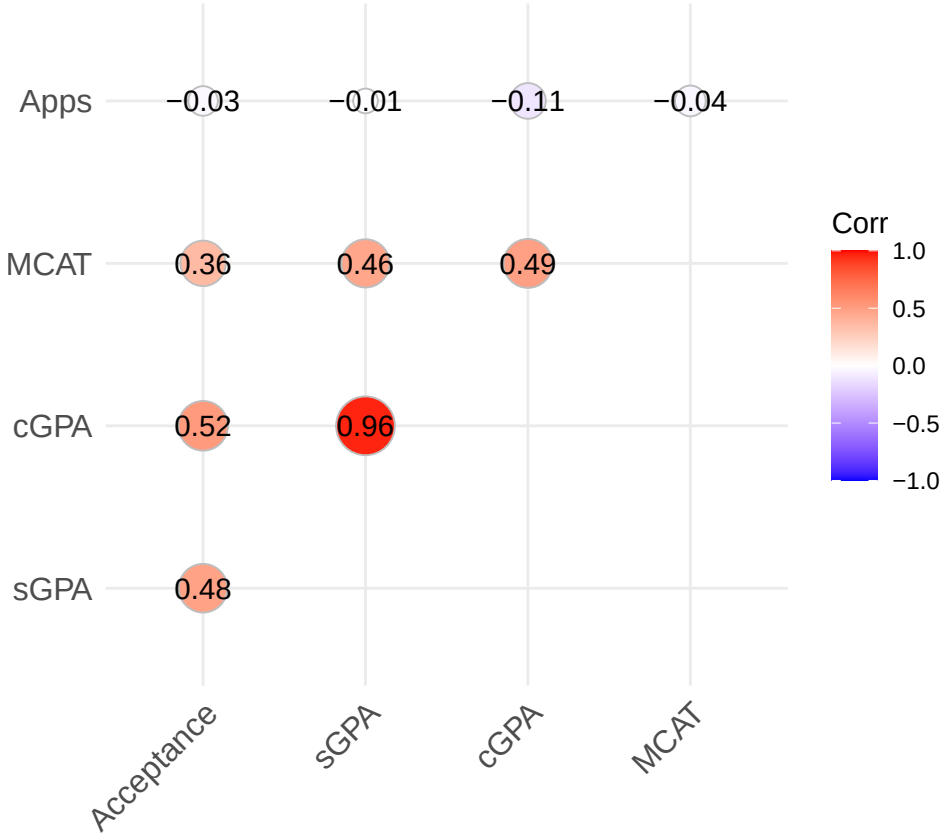
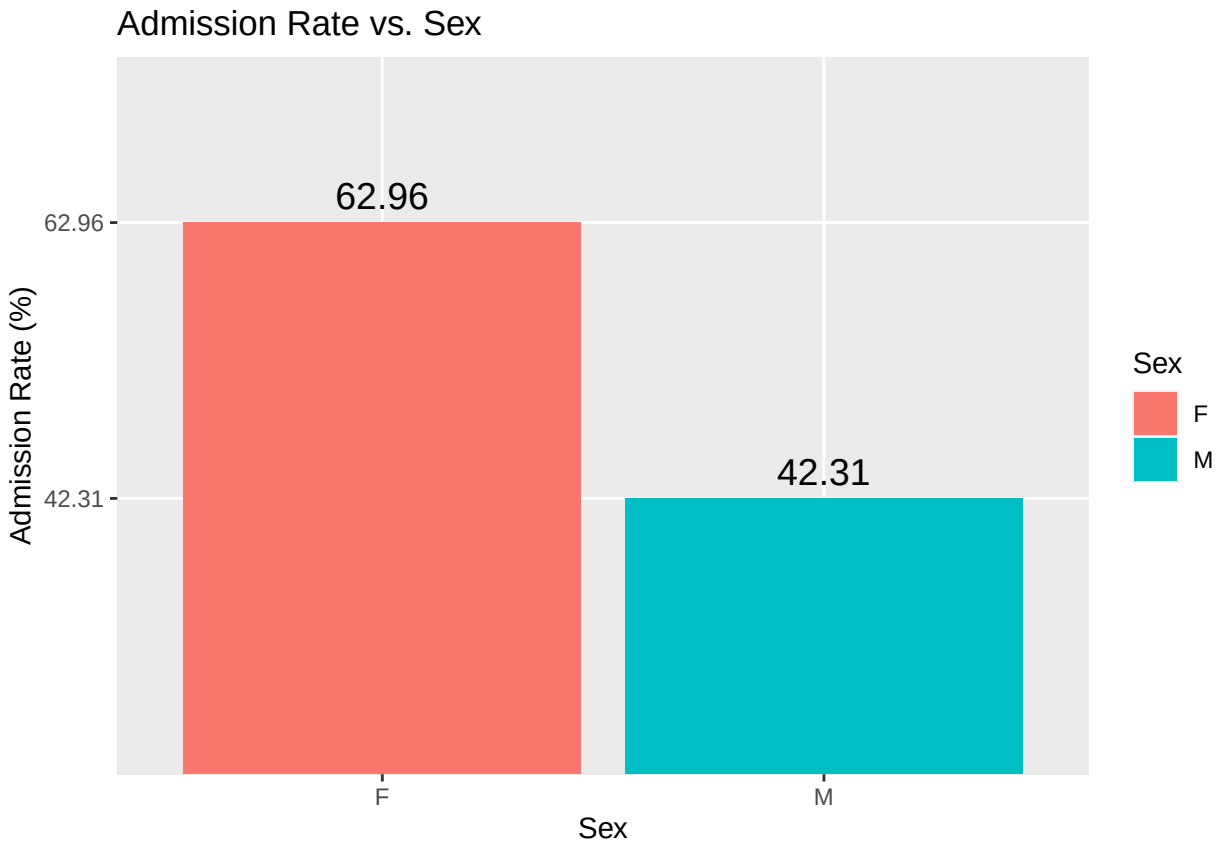


Figure 1: Correlation matrix of numeric variables in the 'admissions_data' dataset. sGPA and cGPA had the highest correlation value at 0.96, which is not surprising, as sGPA and cGPA often overlap. There was not much of a correlation between number of applications submitted and the other variables. All the other pairs of variables had a moderate correlation around 0.50. cGPA had the highest correlation to the acceptance outcome with a value of 0.52.

b. Acceptance Rates of Males and Females

```r
# relationship between sex and acceptance rate
admissions_data %>%
  group_by(Sex) %>%
  summarize(admission_rate = sprintf((mean(Acceptance) * 100), fmt = "%#.2f")) %>%
  ggplot(aes(x = Sex, y = admission_rate, fill = Sex)) +
  geom_bar(stat = "summary", fun = "mean") +
```

```
  geom_text(aes(label = admission_rate), size = 5, hjust = 0.5, vjust = -0.5) +
  labs(x = "Sex", y = "Admission Rate (%)", title = "Admission Rate vs. Sex")
```



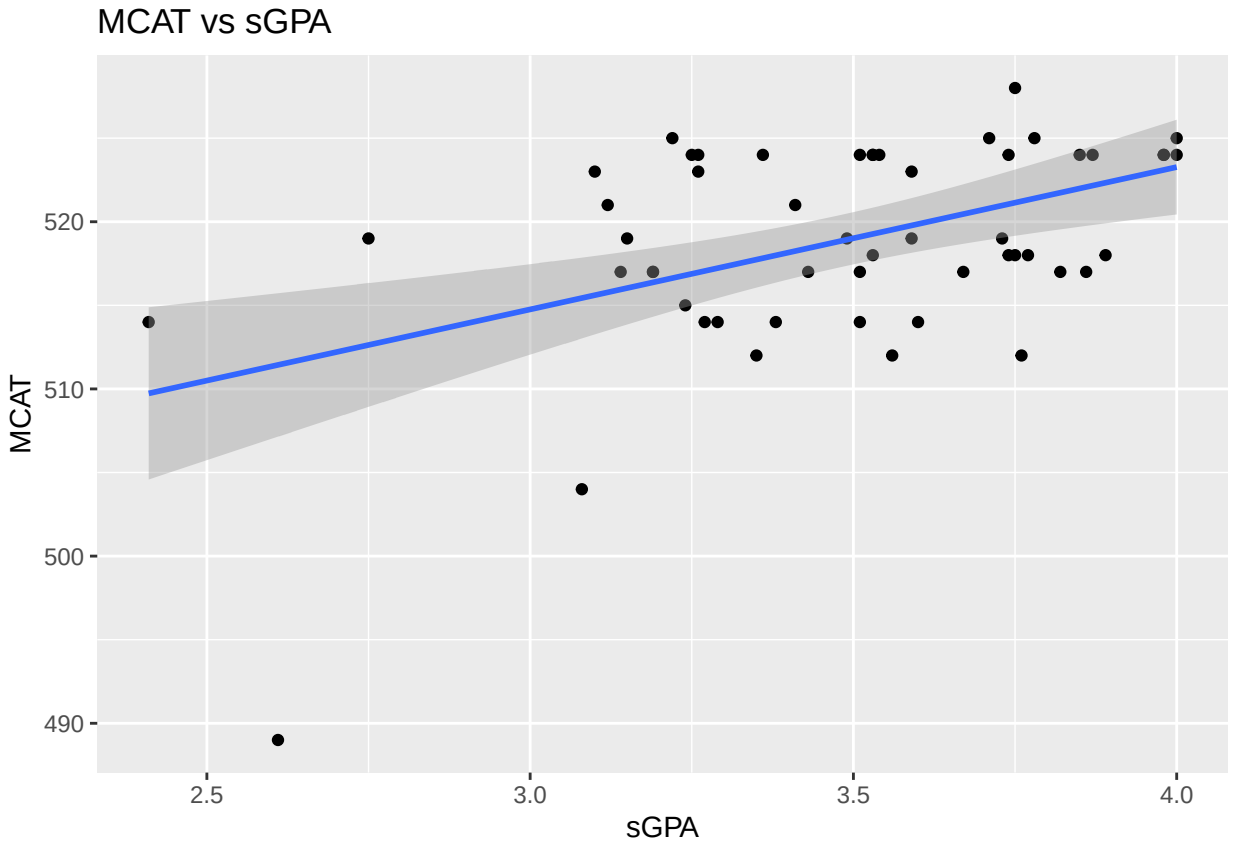
Source referenced to format: Link

Figure 2: Admission rate between males and females. In the dataset used, more females were accepted compared to males at 62.96% and 42.31%, respectively.

c. Relationship Between sGPA and MCAT

```
# ggplot of MCAT vs sGPA
admissions_data %>%
  ggplot(aes(x = sGPA, y = MCAT)) +
  geom_point() +
  geom_smooth(method = "lm") +
  labs(title = "MCAT vs sGPA")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
# calculate correlation coefficient
cor(x = admissions_data$sGPA, y = admissions_data$MCAT)
```

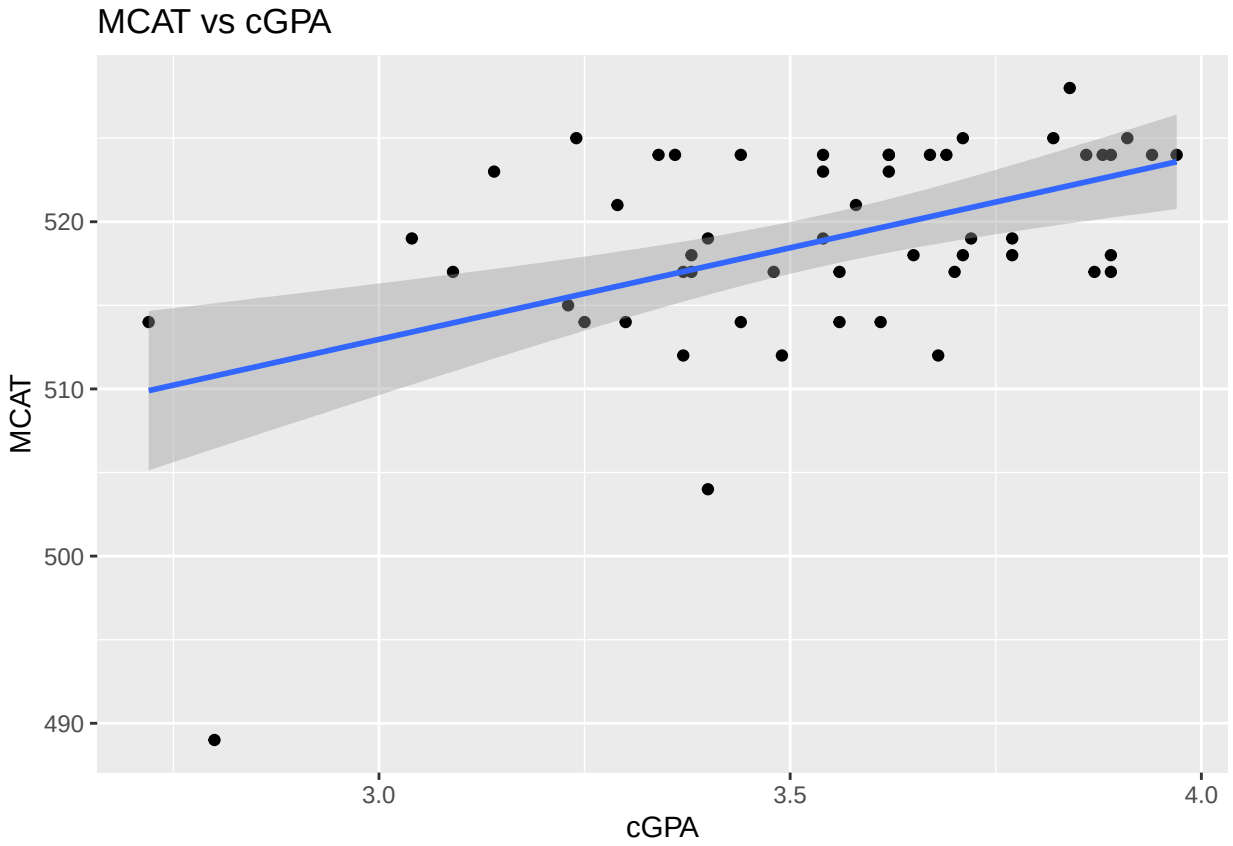
```
## [1] 0.4635634
```

Figure 3: Linear correlation between MCAT and sGPA with a correlation coefficient of 0.46.

d. Relationship Between cGPA and MCAT

```
# ggplot of MCAT vs cGPA
admissions_data %>%
  ggplot(aes(x = cGPA, y = MCAT)) +
  geom_point() +
  geom_smooth(method = "lm") +
  labs(title = "MCAT vs cGPA")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
# calculate correlation coefficient
cor(x = admissions_data$cGPA, y = admissions_data$MCAT)
```

```
## [1] 0.4882672
```

Figure 4: Linear correlation between MCAT and sGPA with a correlation coefficient of 0.49, which is slightly higher than the correlation coefficient between MCAT and cGPA.

e. MCAT Score vs Sex

```
# bar graph
admissions_data %>%
  group_by(Sex) %>%
  summarize(avg_MCAT = round(mean(MCAT), digits = 2)) %>%
  ggplot(aes(x = Sex, y = avg_MCAT, fill = Sex)) +
  geom_bar(stat = "summary", fun = "mean") +
  geom_text(aes(label = avg_MCAT), size = 5, hjust = 0.5, vjust = 1.5) +
  labs(x = "Sex", y = "MCAT Score", title = "MCAT Score vs. Sex")
```

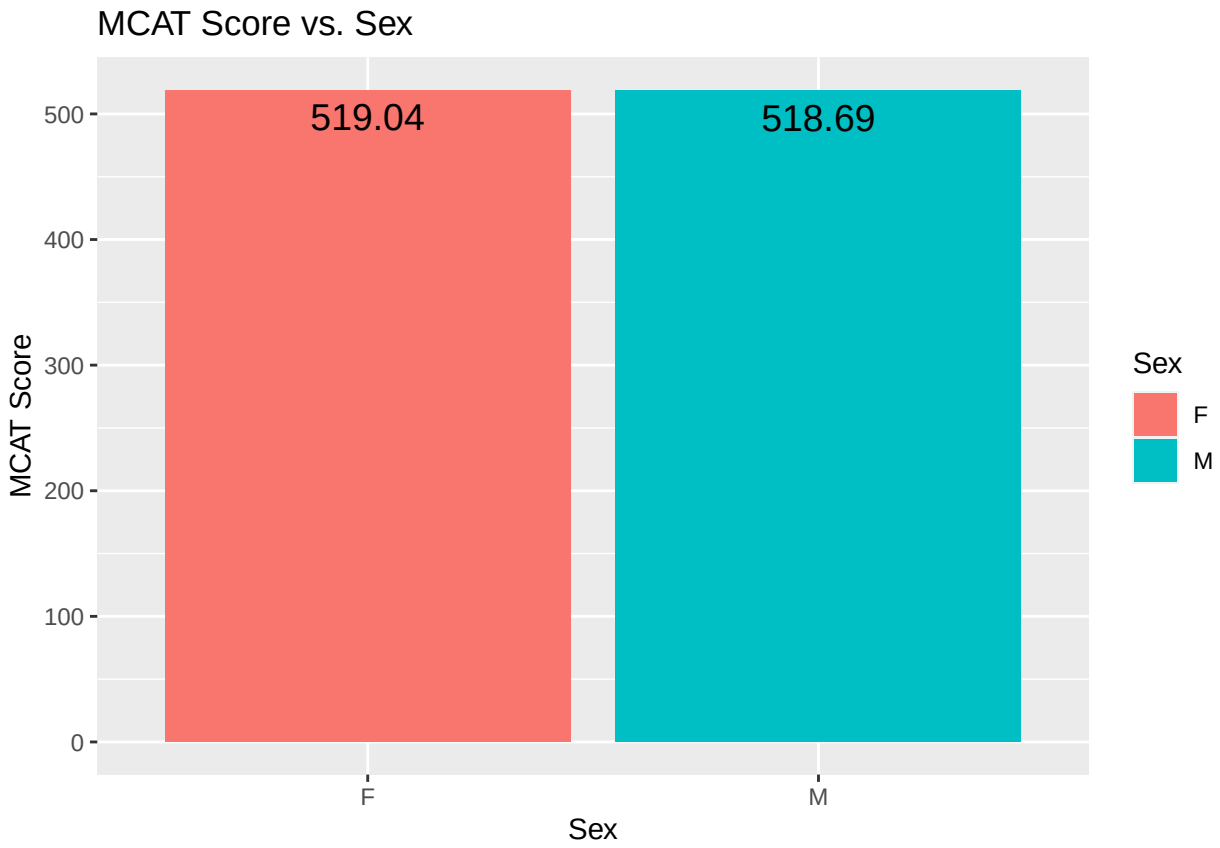


Figure 5: Plot of MCAT score vs. sex. The plot shows that there was little differentiation on MCAT score between males and females. Males had an average MCAT score of 518.69, while females had an average MCAT score of 519.04.

3. Prediction and Cross Validation

a. Logistic Regression

I. Model

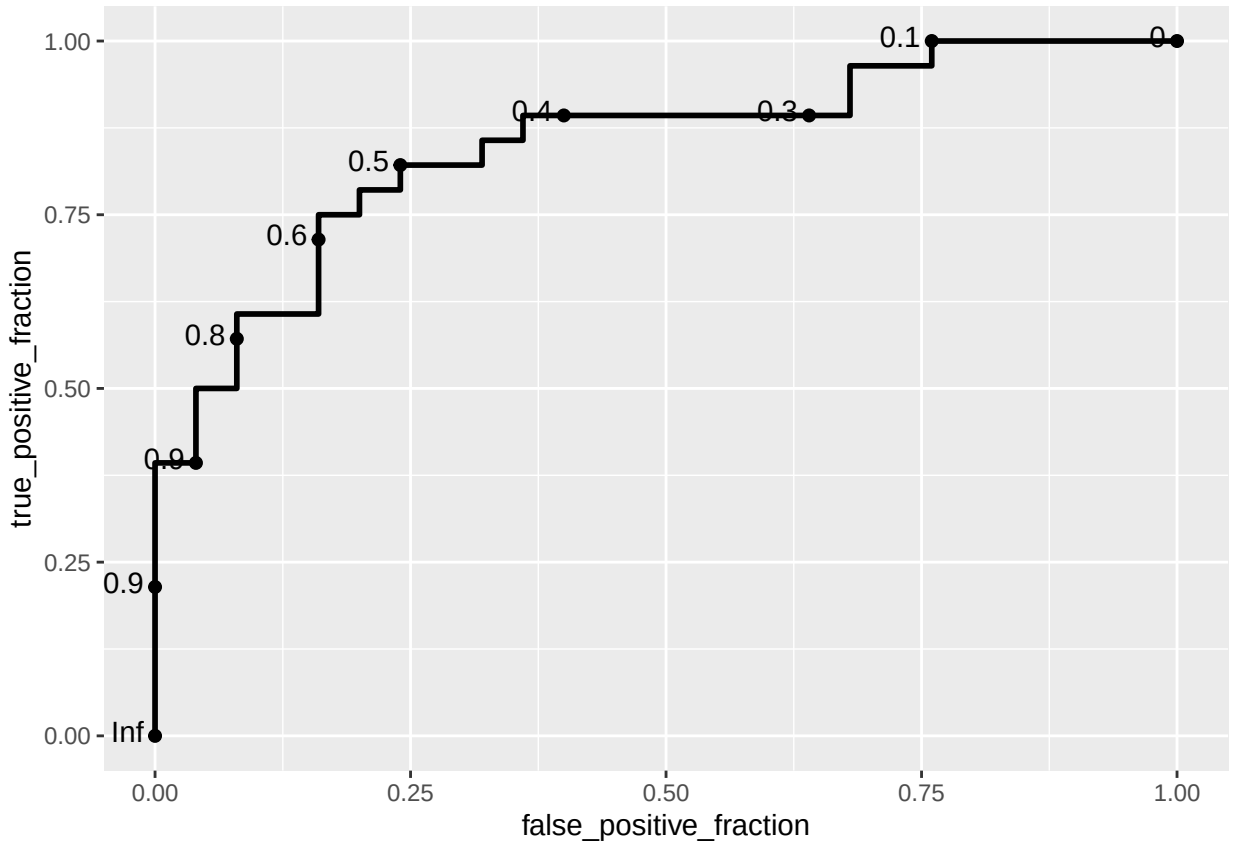
```
# fit logistic regression model
fit_log <- glm(Acceptance ~ ., data = admissions_data, family = "binomial")

# summary of logistic regression model
summary(fit_log)
```

```
##
## Call:
## glm(formula = Acceptance ~ ., family = "binomial", data = admissions_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0156  -0.8585   0.2342   0.6611   2.0848
##
```

```
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -85.68993   42.52494  -2.015   0.0439 *
## SexM         -1.27335    0.75744  -1.681   0.0927 .
## sGPA          0.70095    3.66605   0.191   0.8484
## cGPA          4.54281    4.48885   1.012   0.3115
## MCAT          0.13028    0.08264   1.576   0.1149
## Apps          0.02830    0.07759   0.365   0.7153
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 73.304  on 52  degrees of freedom
## Residual deviance: 50.572  on 47  degrees of freedom
## AIC: 62.572
##
## Number of Fisher Scoring iterations: 5
```

II. ROC Plot

```
# ROC
ROC <- admissions_data %>%
  # Make predictions
  mutate(predictions = predict(fit_log, type = "response")) %>%
  ggplot() +
  geom_roc(aes(d = Acceptance, m = predictions), n.cuts = 10)
ROC
```




```
# AUC value
calc_auc(ROC)$AUC
```

```
## [1] 0.8485714
```

Figure 6: ROC plot of the logistic regression model. The AUC value of the ROC plot was calculated to be 0.85.

III. k-fold Cross Validation

```
# set seed
set.seed(15)

# define k
k = 10

# Randomly order rows in the dataset
data <- admissions_data[sample(nrow(admissions_data)), ]

# Create k folds from the dataset (break rows into k parts)
folds <- cut(seq(1:nrow(data)), breaks = k, labels = FALSE)

# Initialize a vector to keep track of the performance
perf_k <- NULL
```

```r
# Use a for loop to get diagnostics for each test set
for(i in 1:k){
  # Create train and test sets
  train_not_i <- data[folds != i, ] # all observations except in fold i
  test_i <- data[folds == i, ]  # observations in fold i

  # Train model on train set (all but fold i)
  fit_log <- glm(Acceptance ~ ., data = train_not_i, family = "binomial")

  # Test model on test set (fold i)
  predict_i <- data.frame(
    predictions = predict(fit_log, newdata = test_i, type = "response"),
    outcome = test_i$Acceptance)

  # Consider the ROC curve for the test dataset
  ROC <- ggplot(predict_i) +
    geom_roc(aes(d = outcome, m = predictions))

  # Get diagnostics for fold i (AUC)
  perf_k[i] <- calc_auc(ROC)$AUC
}

# get value
mean(perf_k)
```

```
## [1] 0.8097222
```

The 10-fold cross validation resulted in a mean value of 0.81.

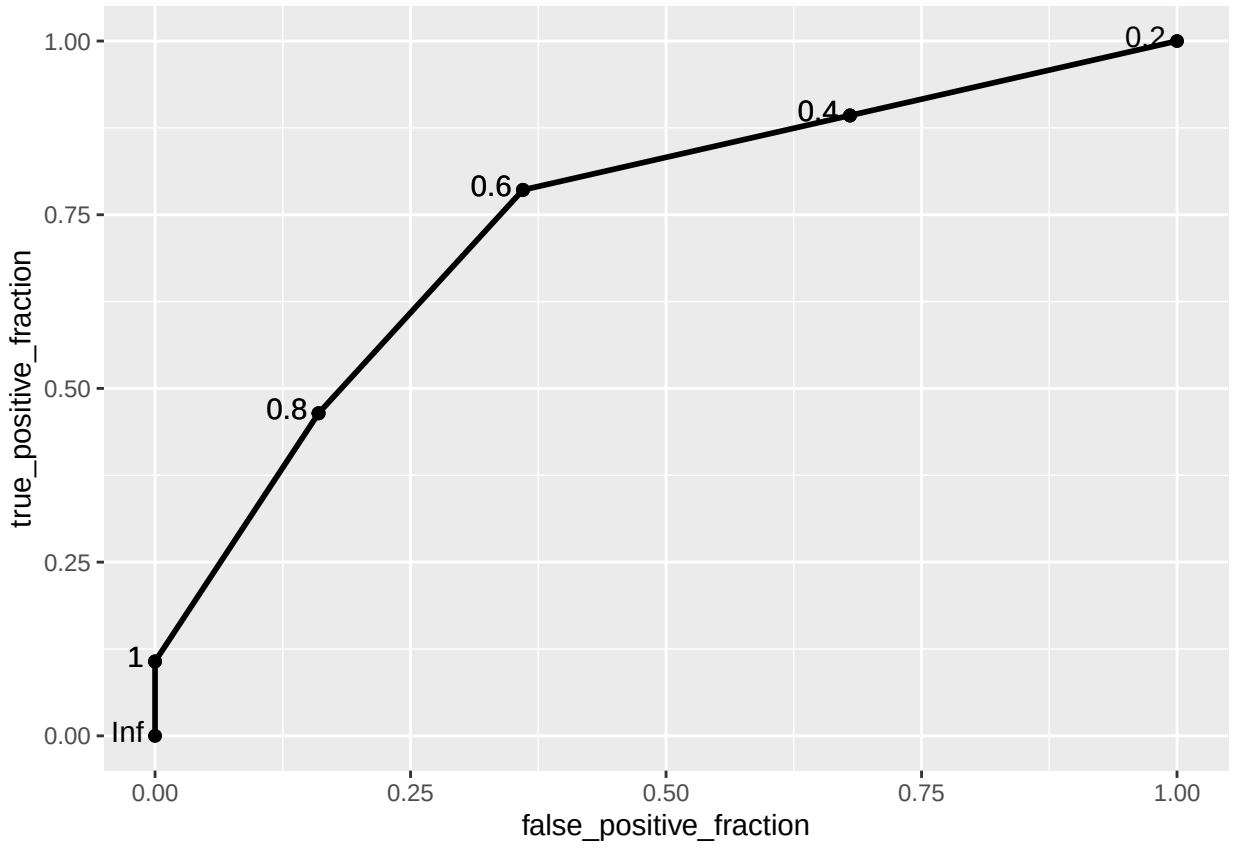
b. kNN Analysis

I. Model

```r
# fit kNN model
fit_kNN <- knn3(Acceptance ~ ., data = admissions_data, k = 5)
```

II. ROC Plot

```r
# ROC
ROC <- admissions_data %>%
  # Make predictions
  mutate(predictions = predict(fit_kNN, admissions_data)[,2]) %>%
  ggplot() +
  geom_roc(aes(d = Acceptance, m = predictions), n.cuts = 10)
ROC
```




```
# AUC value
calc_auc(ROC)$AUC
```

```
## [1] 0.7421429
```

Figure 7: ROC plot of the kNN model. The AUC value of the ROC plot was calculated to be 0.74.

III. k-fold Cross Validation

```
# set seed
set.seed(1)

# define k
k = 10

# Randomly order rows in the dataset
data <- admissions_data[sample(nrow(admissions_data)), ]

# Create k folds from the dataset (break rows into k parts)
folds <- cut(seq(1:nrow(data)), breaks = k, labels = FALSE)

# Initialize a vector to keep track of the performance
perf_k <- NULL
```

```r
# Use a for loop to get diagnostics for each test set
for(i in 1:k){
  # Create train and test sets
  train_not_i <- data[folds != i, ] # all observations except in fold i
  test_i <- data[folds == i, ]  # observations in fold i

  # Train model on train set (all but fold i)
  fit_kNN <- knn3(Acceptance ~ ., data = admissions_data, k = 5)

  # Test model on test set (fold i)
  predict_i <- data.frame(
    predictions = predict(fit_kNN, newdata = test_i)[,2],
    outcome = test_i$Acceptance)

  # Consider the ROC curve for the test dataset
  ROC <- ggplot(predict_i) +
    geom_roc(aes(d = outcome, m = predictions))

  # Get diagnostics for fold i (AUC)
  perf_k[i] <- calc_auc(ROC)$AUC
}
mean(perf_k)
```

```
## [1] 0.7361111
```

The 10-fold cross validation resulted in a mean value of 0.74.

c. Results

The AUC value of the ROC plot for the logistic regression model was 0.85, which indicates that the regression model is a satisfactory classifier. In addition, the 10-fold cross validation test resulted in a mean value of 0.81. Because this value is close to the AUC value, it is a good result, indicating that the regression model is generally repeatable with new data. In addition, the similar AUC and cross-validation values indicate that overfitting was limited.

The AUC value of the ROC plot for the kNN model was 0.74, which indicates that the kNN model is a moderate classifier. Because this value is close to the AUC value, it is a good result, indicating that the kNN model is generally repeatable with new data. In addition, the nearly identical AUC and cross-validation values indicate that overfitting was limited.

Overall, these results indicate that the logistic regression model was a better classifier for predicting medical school admission outcomes, as the logistic regression model had a higher AUC value compared to the kNN model.

4. Dimensionality Reduction - PCA Analysis

a. Prepare Data

```r
# scale data
admissions_data_scaled <- admissions_data %>%
```

```
  select(-Sex) %>% # remove categorical variable
  scale %>% # Scale the variables (find how many sd from mean)
  as.data.frame # Save as a data frame
```

b. Identify Principle Components

```
# PCA performed with the function prcomp()
pca <- admissions_data_scaled %>%
  prcomp

# output gives the principle components
names(pca)
```

```
## [1] "sdev"     "rotation" "center"   "scale"    "x"
```

c. Scree Plot

```
# Visualize percentage of variance explained for each PC in a scree plot
fviz_eig(pca, addlabels = TRUE)
```

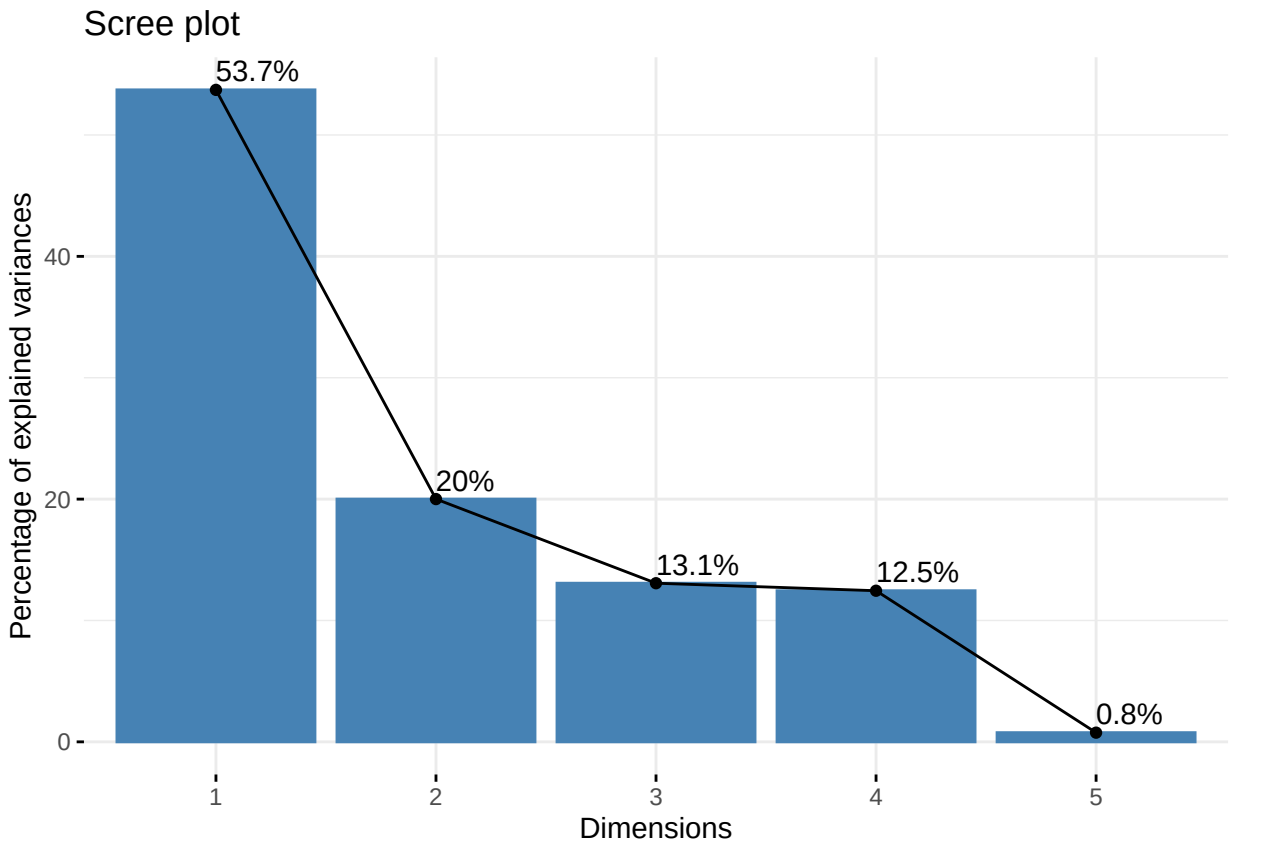


Figure 8: Scree plot showing the percentage of variance explained by each of the 5 principle components (PC) identified. Based on these results, retaining the first 2 PCs would explain 73.7% of variance, which although is slightly below 80%, is acceptable.

d. Visualize

```
# Visualize the contributions of the variables to the PCs in a table
get_pca_var(pca)$coord %>% as.data.frame
```

```
##                  Dim.1       Dim.2         Dim.3       Dim.4        Dim.5
## Acceptance  0.70047105 -0.03797016  1.143563e-01 -0.70340431 -0.006601183
## sGPA        0.91986875 -0.07839835  2.358225e-01  0.27216929 -0.134189450
## cGPA        0.93949010  0.02535682  2.129446e-01  0.22740111  0.140209569
## MCAT        0.67675665 -0.02379352 -7.345095e-01  0.04379688 -0.003461140
## Apps       -0.09138991 -0.99565841  4.614009e-05  0.01013886  0.014471330
```

```
# Visualize observations using the first 2 PCs
fviz_pca_biplot(pca, repel = TRUE) # Avoid text overlapping of the names
```

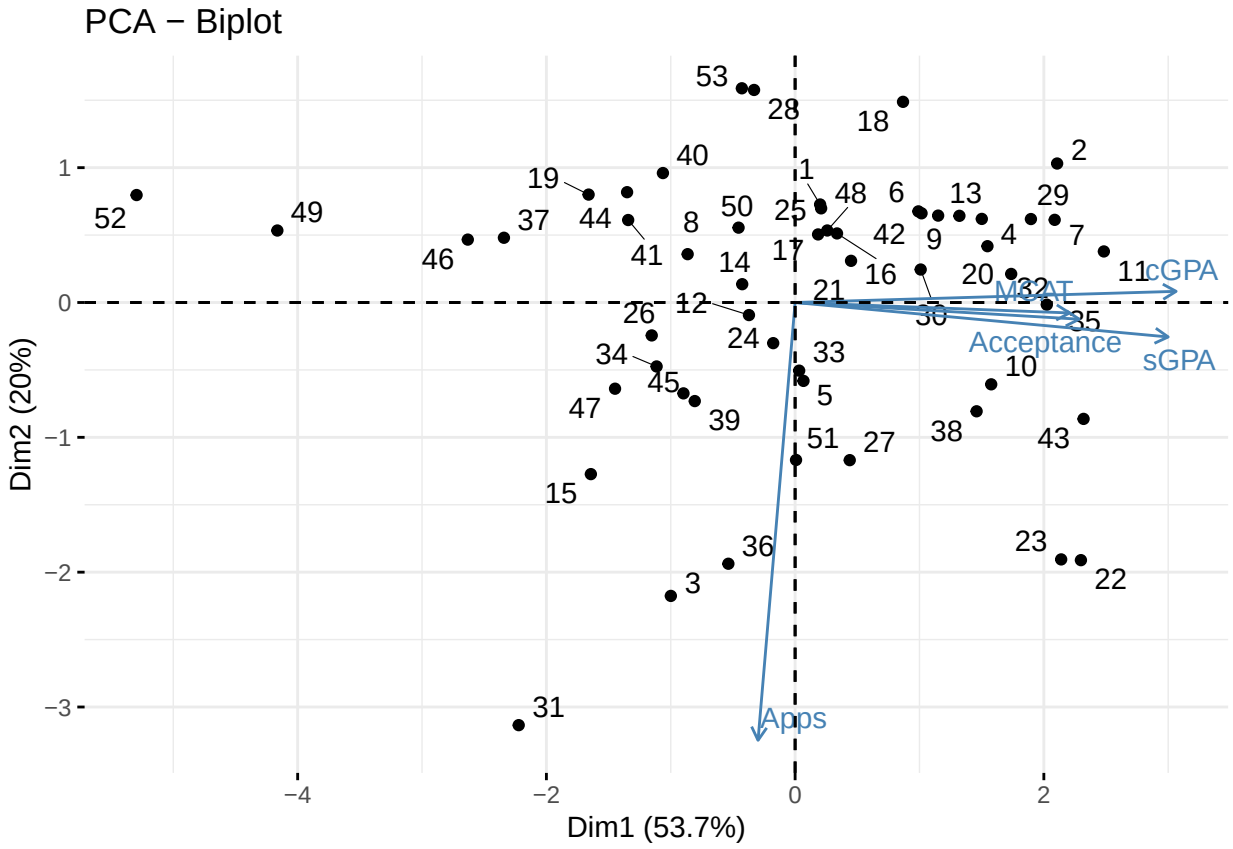


Figure 9: Dataframe of each PC's contribution to variance and biplot of PC1 and PC2 on the dataset's variables and individual observations. In the biplot, since only PC1 and PC2 were used, the total variation explained by these variable is 73.7%. A high value in the PCs indicate that the variance for the given variable is heavily influenced by the PC. For example, cGPA had the highest value for PC1 (0.939), indicating that variance for cGPA is mostly explained by PC1. Conversely, a low (close to 0) value indicates that the variance for the given variable is not influenced much by the PC. In addition, a high negative value, as seen by the 'Apps' variable in PC2 (-0.996) indicates that PC2 has a very strong inverse relationship in explaining variance for 'Apps'.

5. Clustering

a. kMeans

i. Silhouette Analysis

```
# Silhouette analysis to find optimal number of clusters
fviz_nbclust(admissions_data_scaled, kmeans, method = "silhouette")
```

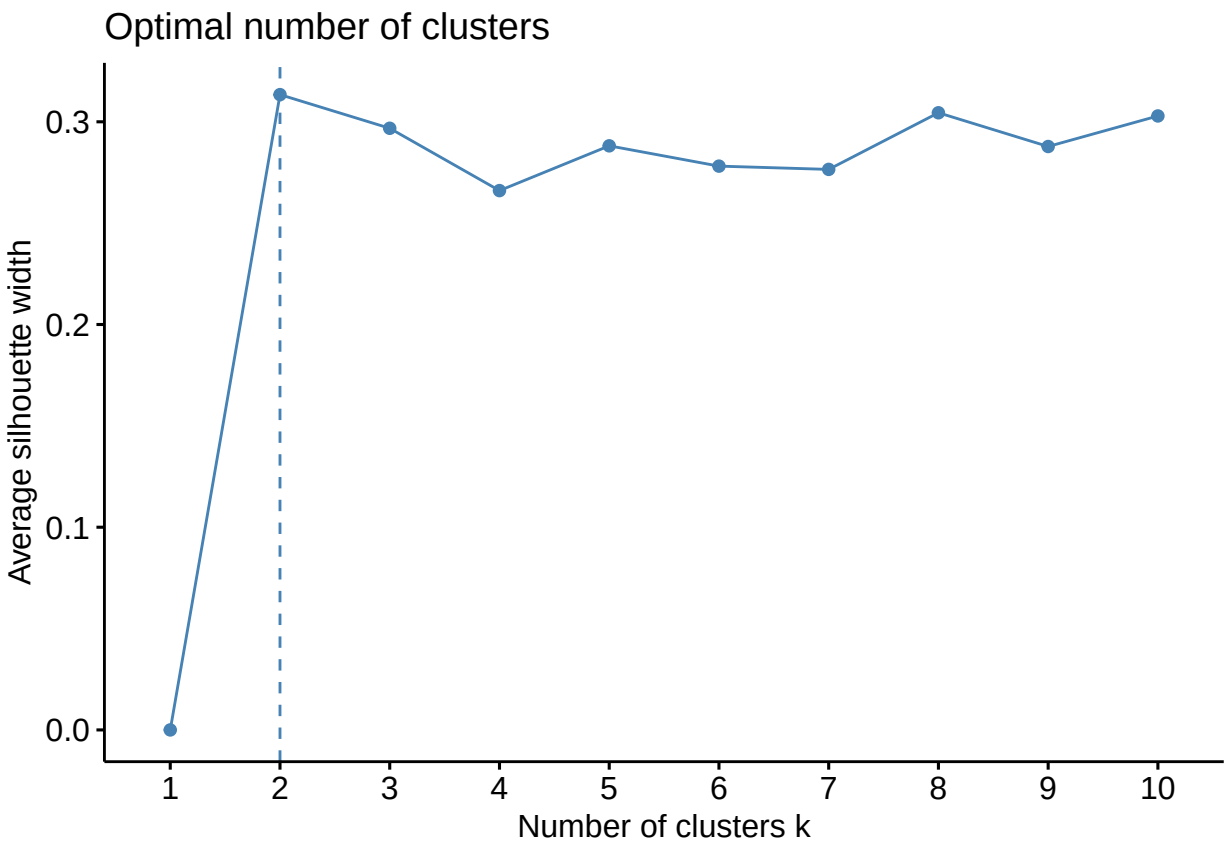


Figure 9: This resulting plot of Silhoutte analysis indicated that the optimal number of clusters is 2.

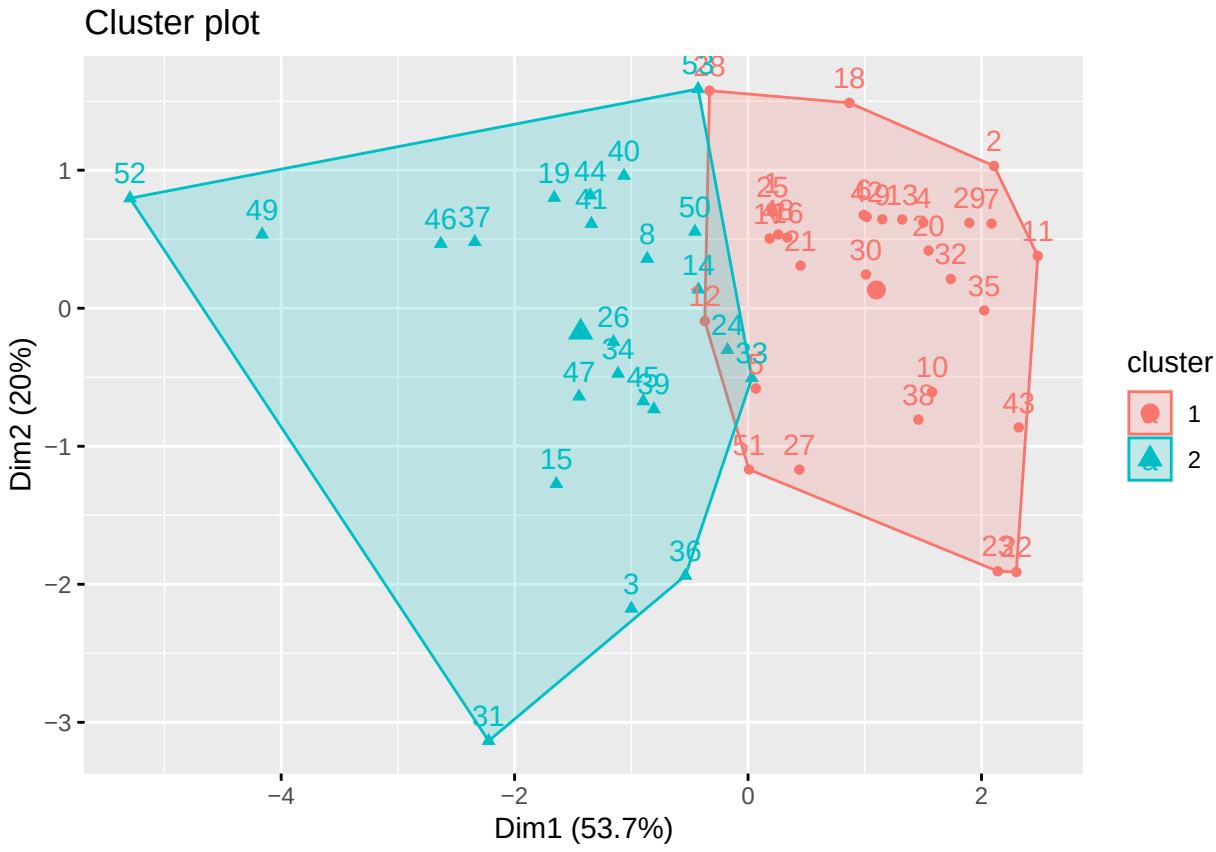
ii. kMeans Results

```
# use kmeans() to find clusters, using 2 clusters as defined by silhoutte analysis
kmeans_results <- admissions_data_scaled %>%
  kmeans(centers = 2) # centers sets the number of clusters to find

# Resulting object
kmeans_results
```

```
## K-means clustering with 2 clusters of sizes 30, 23
##
## Cluster means:
```

```
##   Acceptance       sGPA       cGPA       MCAT       Apps
## 1  0.6713913  0.5478144  0.6039639  0.3581861 -0.1954007
## 2 -0.8757278 -0.7145405 -0.7877790 -0.4671992  0.2548705
##
## Clustering vector:
##  [1] 1 1 2 1 1 1 1 2 1 1 1 1 1 2 2 1 1 1 2 1 1 1 1 2 1 2 1 1 1 1 2 1 2 2 1 2 2 1
## [39] 2 2 2 1 1 2 2 2 2 1 2 2 1 2 2
##
## Within cluster sum of squares by cluster:
## [1]  68.60827 102.75835
##  (between_SS / total_SS =  34.1 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```r
# set seed
set.seed(5)

# resulting cluster plot
fviz_cluster(kmeans_results, data = admissions_data_scaled)
```




```r
# cluster statistics
admissions_data %>%
  mutate(cluster = as.factor(kmeans_results$cluster)) %>%
```

```
  group_by(cluster) %>%
  summarize(sGPA_mean = mean(sGPA),
            cGPA_mean = mean(cGPA),
            MCAT_mean = mean(MCAT))
```

```
## # A tibble: 2 x 4
##   cluster sGPA_mean cGPA_mean MCAT_mean
##   <fct>       <dbl>     <dbl>     <dbl>
## 1 1            3.67      3.71      521.
## 2 2            3.24      3.32      516.
```

Figure 10: Resulting cluster plot using the kMeans method and summary statistics of clusters. The plot indicates that observations with a positive or slightly negative value on PC1 tends to be classified into cluster 1. Conversely, observations with a slightly negative to largely negative value on PC1 tends to be classified into cluster 2. PC2 did not seem to have much effect on the cluster classification.

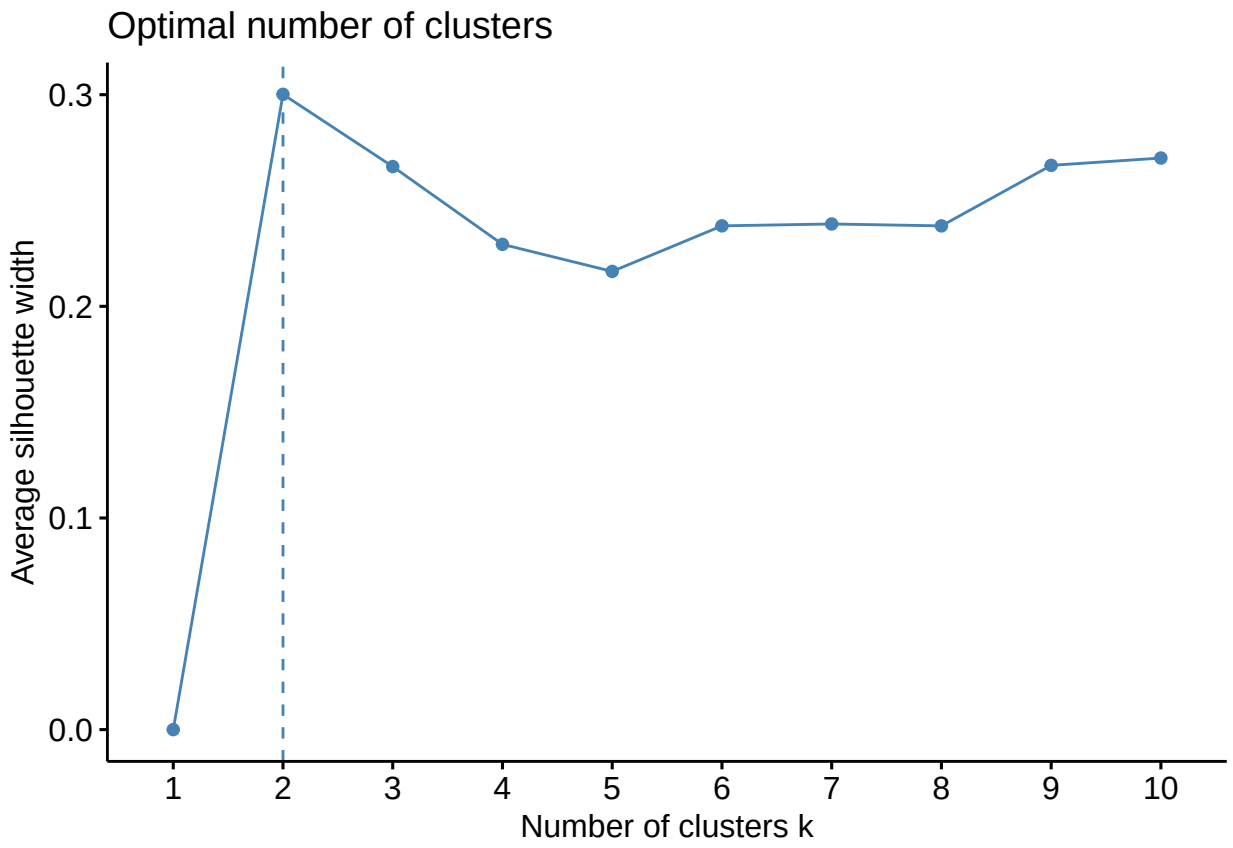
Looking at the summary statistics, unexpectedly, those who were accepted (cluster 1) had higher average sGPA, cGPA, and MCAT score. The average cGPA and MCAT of cluster 1 was 3.69 and 521, respectively. On the other hand, the average cGPA and MCAT of cluster 2 was 3.24 and 516, respectively.

An observation representative of cluster 1 would be observation 30, as it is closest to the center of the cluster.. This individual had a cGPA and MCAT of 3.62 and 524, respectively. An observation representative of cluster 2 would be observation 47, as it is closest to the center of the cluster. This individual had a cGPA and MCAT of 3.29 and 521. For both observations, their cGPA is closer to its expected value compared to the MCAT score. However, this is expected because the correlation matrix indicated that cGPA had a higher correlation to admissions than MCAT. In addition, the PCA showed that for PC1, which was the PC that explained most variance, cGPA had a higher value than MCAT.

b. PAM

i. Silhouette Width

```
# Silhouette width to find optimal number of clusters
fviz_nbclust(admissions_data_scaled, pam, method = "silhouette")
```

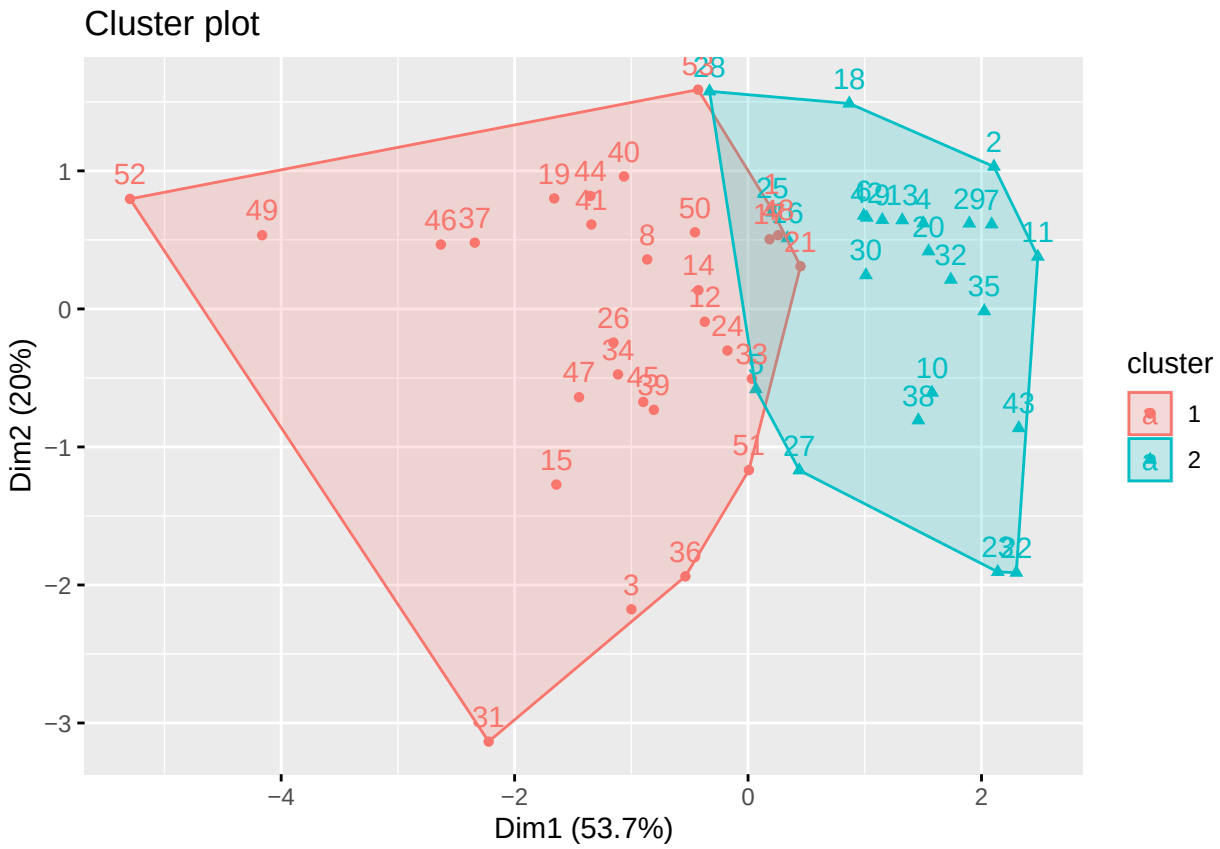


ii. PAM Results

```
# Use the function pam() to find clusters
pam_results <- admissions_data_scaled %>%
  pam(k = 2) # k is the number of clusters

# Take a look at the resulting object
pam_results
```

```
## Medoids:
##      ID Acceptance       sGPA       cGPA       MCAT       Apps
## [1,] 26 -1.0482690 -0.2991565 -0.3513493 -0.7696725  0.3182465
## [2,] 20  0.9359545  0.6591306  0.6063285  0.9695487 -0.5146950
## Clustering vector:
##  [1] 1 2 1 2 2 2 2 1 2 2 2 1 2 1 1 2 1 2 1 2 1 2 2 1 2 1 2 2 2 2 1 2 1 1 2 1 1 2
## [39] 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1
## Objective function:
##    build     swap
## 1.699278 1.675793
##
## Available components:
##  [1] "medoids"    "id.med"     "clustering" "objective"  "isolation"
##  [6] "clusinfo"   "silinfo"    "diss"       "call"       "data"
```

```
# Let's visualize the clusters after dimension reduction
fviz_cluster(pam_results, data = admissions_data_scaled)
```




```
# cluster statistics
admissions_data %>%
  mutate(cluster = as.factor(kmeans_results$cluster)) %>%
  group_by(cluster) %>%
  summarize(sGPA_mean = mean(sGPA),
            cGPA_mean = mean(cGPA),
            MCAT_mean = mean(MCAT))
```

```
## # A tibble: 2 x 4
##   cluster sGPA_mean cGPA_mean MCAT_mean
##   <fct>       <dbl>     <dbl>     <dbl>
## 1 1            3.67      3.71      521.
## 2 2            3.24      3.32      516.
```

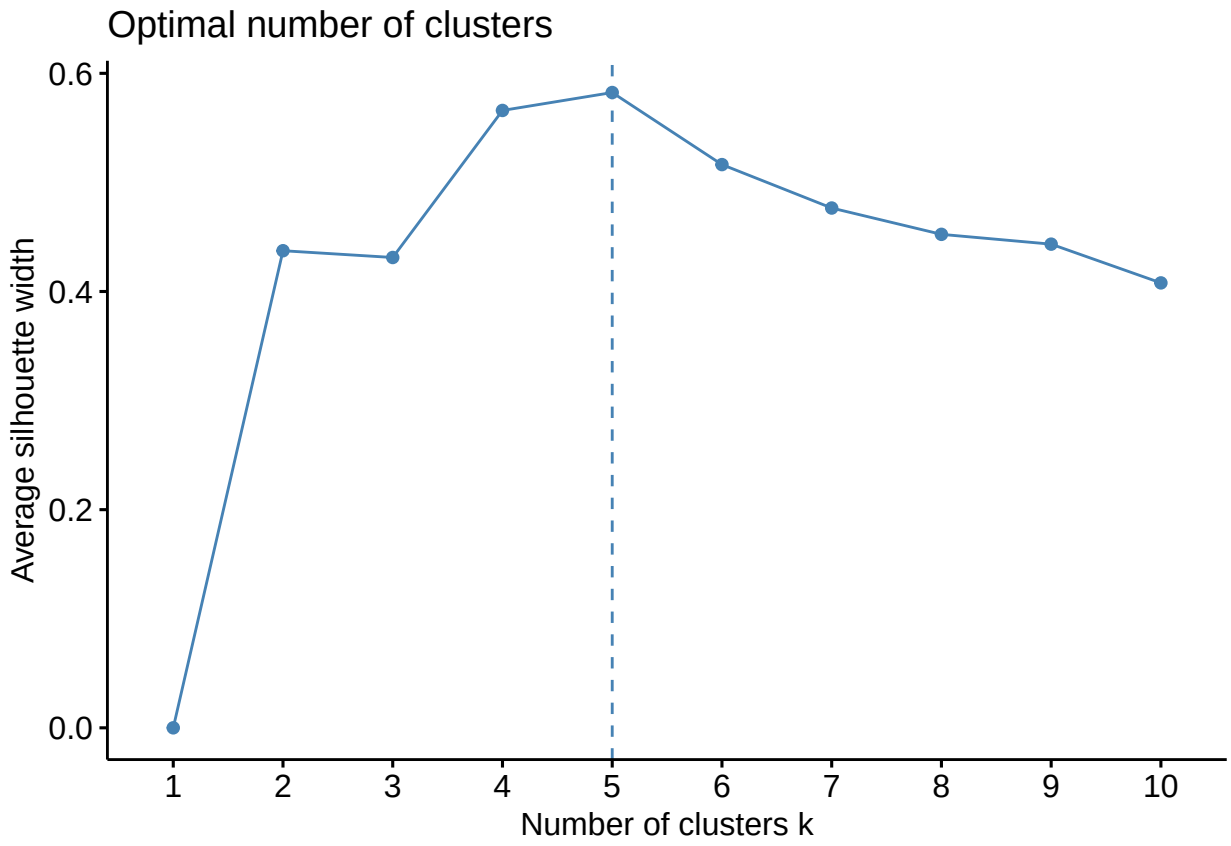
c. Gower

```
admissions_data_factored <- admissions_data %>%
  # Consider categorical variables as factors
  mutate_if(is.character, as.factor) %>%
  # Ignore missing values
```

```
  na.omit

# Calculate Gower distances between observations
admission_data_gower <- admissions_data_factored %>%
  # No need to scale when calculating the Gower's distance
  daisy(metric = "gower") %>%
  # Save as a matrix
  as.matrix
```

```
## Warning in daisy(., metric = "gower"): binary variable(s) 1 treated as interval
## scaled
```

```
fviz_nbclust(admission_data_gower, pam, method = "silhouette")
```




```
# Apply PAM on the dissimilarity object (specify diss = TRUE)
pam_results <- pam(admission_data_gower, k = 5, diss = TRUE)

# Look at the final medoids
admissions_data[pam_results$id.med,]
```

```
##    Acceptance Sex sGPA cGPA MCAT Apps
## 41          0   F 3.19 3.38  517    6
## 7           1   M 3.85 3.89  524    5
## 9           1   F 3.74 3.71  518    5
## 50          0   M 3.51 3.56  517    6
## 49          0   M 2.41 2.72  514    7
```

3. Discussion

a. Research Question 1: Which factor(s) are the most impactful in determining acceptance into medical school

The results of the correlation matrix indicated that GPA was the most impactful in determining acceptance into medical school. sGPA and cGPA had similar correlation coefficients of 0.48 and 0.52, respectively. MCAT score had a less significant impact, with a correlation coefficient of 0.36. Surprisingly, the number of applications submitted had virtually no impact on acceptance, as is had a correlation coefficient of -0.03.

b. Research Question 2: Can medical school acceptance be accurately predicted based on applicant statistics?

The linear regression model resulted in an AUC value of 0.85, which indicates that given an applicant's statistics, medical school acceptance can generally be predicted. In addition, the 10-fold cross validation resulted in a value of 0.81. Therefore, these results indicated that the linear regression model is generally accurate and reliable/repeatable.

However, as discussed in the introduction, it is still important to understand the limitations of these results. The dataset used for modelling only contained some variables that go into a medical school application. Other equally, if not more, important parts of the application, such as interviews and personal statement, were not analyzed in this report.

c. Reflections

In completing this project, I gained a deeper understanding of what the process for classification and modelling is like. Each step built onto the next, and each part of the project represented an important aspect in creating a classification/predictive model. In addition, interpreting the results of the analyses allowed me to gain a deeper understanding the concepts. The most challenging aspects of this project were conceiving of the project idea and finding a suitable dataset for analysis.