CBB Data Analysis

Tom Allen

1/13/2020

Download Common Libraries

```
library(psych)
library(ggplot2)
library(dplyr)
library(caTools)
```

Import Dataset: cbb.csv

cbb.csv contains data from Division 1 college basketball teams from 2015-2019.

```
cbb <- read.csv('cbb.csv')
head(cbb)
```

```
##             TEAM CONF  G  W ADJOE ADJDE BARTHAG EFG_O EFG_D  TOR TORD  ORB  DRB
## 1 North Carolina  ACC 40 33 123.3  94.9  0.9531  52.6  48.1 15.4 18.2 40.7 30.0
## 2      Wisconsin  B10 40 36 129.1  93.6  0.9758  54.8  47.7 12.4 15.8 32.1 23.7
## 3       Michigan  B10 40 33 114.4  90.4  0.9375  53.9  47.7 14.0 19.5 25.5 24.9
## 4     Texas Tech  B12 38 31 115.2  85.2  0.9696  53.5  43.0 17.7 22.8 27.4 28.7
## 5        Gonzaga  WCC 39 37 117.8  86.3  0.9728  56.6  41.1 16.2 17.1 30.0 26.2
## 6           Duke  ACC 39 35 125.2  90.6  0.9764  56.6  46.5 16.3 18.6 35.8 30.2
##    FTR FTRD X2P_O X2P_D X3P_O X3P_D ADJ_T  WAB POSTSEASON YEAR
## 1 32.3 30.4  53.9  44.6  32.7  36.2  71.7  8.6        2ND 2016
## 2 36.2 22.4  54.8  44.7  36.5  37.5  59.3 11.3        2ND 2015
## 3 30.7 30.0  54.7  46.8  35.2  33.2  65.9  6.9        2ND 2018
## 4 32.9 36.6  52.8  41.9  36.5  29.7  67.5  7.0        2ND 2019
## 5 39.0 26.9  56.3  40.0  38.2  29.0  71.5  7.7        2ND 2017
## 6 39.8 23.9  55.9  46.3  38.7  31.4  66.4 10.7  Champions 2015
```

Variables

G: Games played in that season
W: Number of wins in that season
ADJOE: Adjusted Offensive Efficiency (points scored per 100 possessions)
ADJDE: Adjusted Defensive Efficiency (points allowed per 100 possessions)
BATHAG: Power Rating (chance of beating an average Division 1 team)
EFG_O: Effective Field Goal Percentage Shot
EFG_D: Effective Field Goal Percentage Allowed
TOR: Turnover Percentage Allowed (Turnover Rate)
TORD: Turnover Percentage Committed (Steal Rate)
ORB: Offensive Rebound Percentage
DRB: Defensive Rebound Percentage
FTR: Free Throw Rate (how often the team shoots Free Throws)
FTRD: Free Throw Rate Allowed
X2P_O: Two-Point Shooting Percentage
X2P_D: Two-Point Shooting Percentage Allowed
X3P_O: Three-Point Shooting Percentage
X3P_D: Three Point Shooting Percentage Allowed
ADJ_T: Adjusted Tempo (possesions per 40 minutes)
WAB: Wins Above Bubble (bubble refers to the cut off between making NCAA Tournament and not)

POSTSEASON: Round of the NCAA Tournament the team was eliminated
Year: Season

Replace NA Values

Replaces NA values in the POSTSEASON column with the string 'None'. These teams did not reach the final NCAA Tournament in that year. Not going to delete these NA values since it may be an important target feature to see what teams did or did not make the tournament.

```
library(forcats)

cbb$POSTSEASON <- fct_explicit_na(cbb$POSTSEASON, na_level = 'None')
```

Create new Win Percentage Column

```
cbb$Win_Perc <- cbb$W/cbb$G
```

Reorder Postseason Finishes

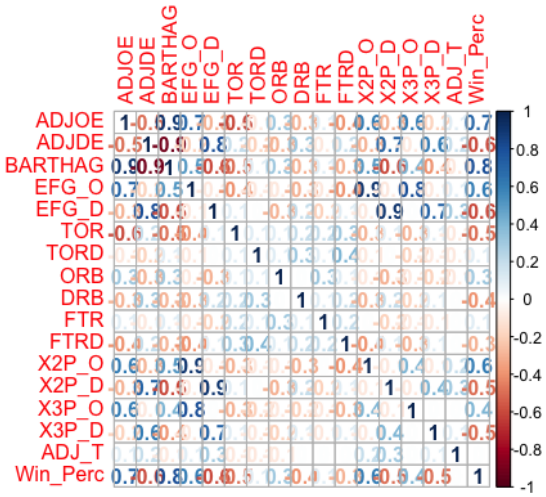
This factor was previosuly out of order.

```
cbb$POSTSEASON <- factor(cbb$POSTSEASON, levels(cbb$POSTSEASON)[c(2,1,4,3,8,6,7,9,5)])
```

Explore the Correlations Between Variables

Will only look at the variables with numeric values.

```
library(corrplot)

cor.data <- round(cor(cbb[,c(5:20,24)]),1)
cor.plot <- corrplot(cor.data, method = 'number')
```



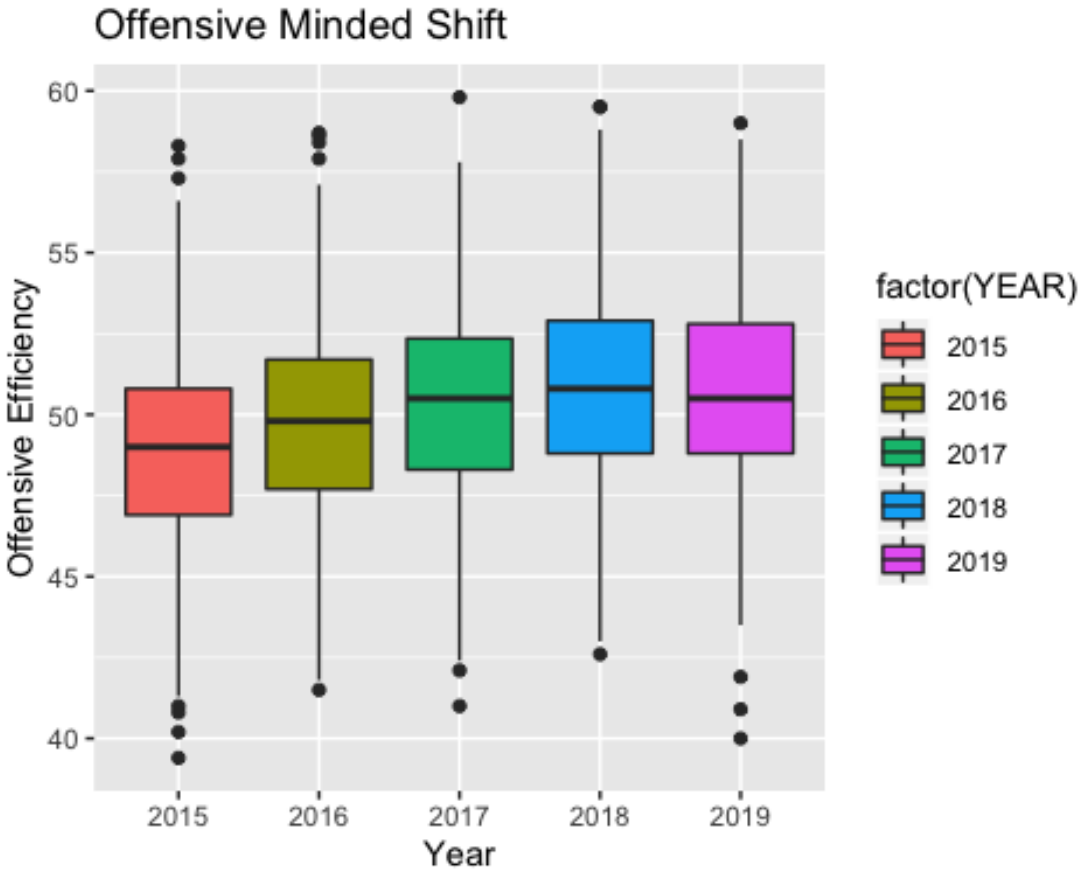
One observation from the correlation plot is:
X2P_D has a higher correlation to EFG_D than X3P_D. Teams should focus on defending inside the 3-point line than worrying about the 3-point shooters.

Simple Plots

Offensive Minded Shift

Boxplot showing team's shift to being more efficient offensivley over the years.

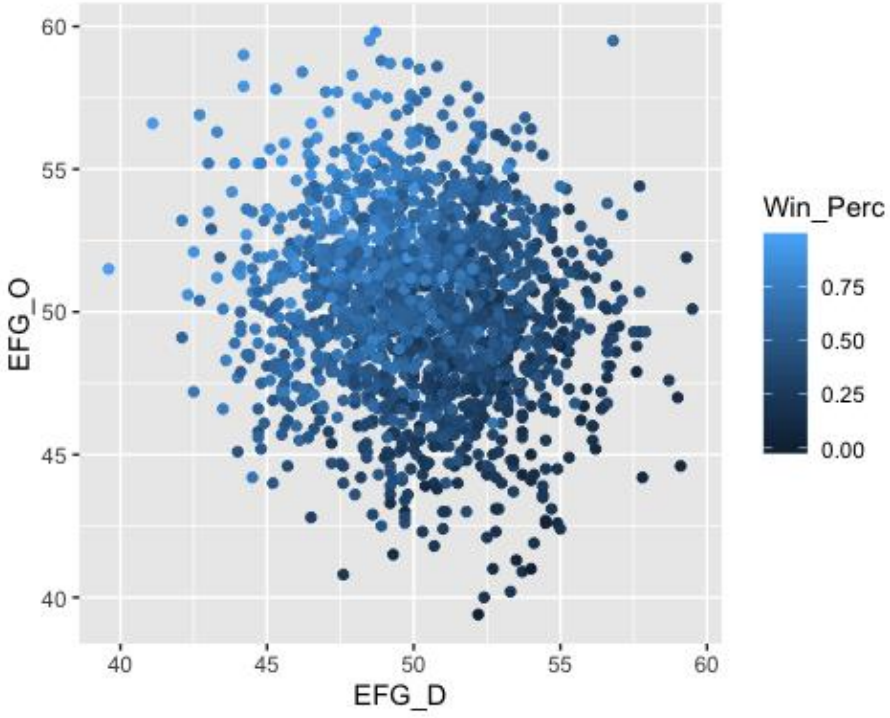
```
Eff.O <- ggplot(cbb, aes(x = factor(YEAR), y = EFG_O)) + geom_boxplot(aes(fill=factor(YEAR)))
+ xlab('Year') + ylab('Offensive Efficiency') + ggtitle('Offensive Minded Shift')
plot(Eff.O)
```



Offensive vs. Defensive Efficiency

Shows scatter plot comparing offensive and defensive efficiency and the respective win percenage. Looks like teams who are more efficient offensivley yield a better win percentage.

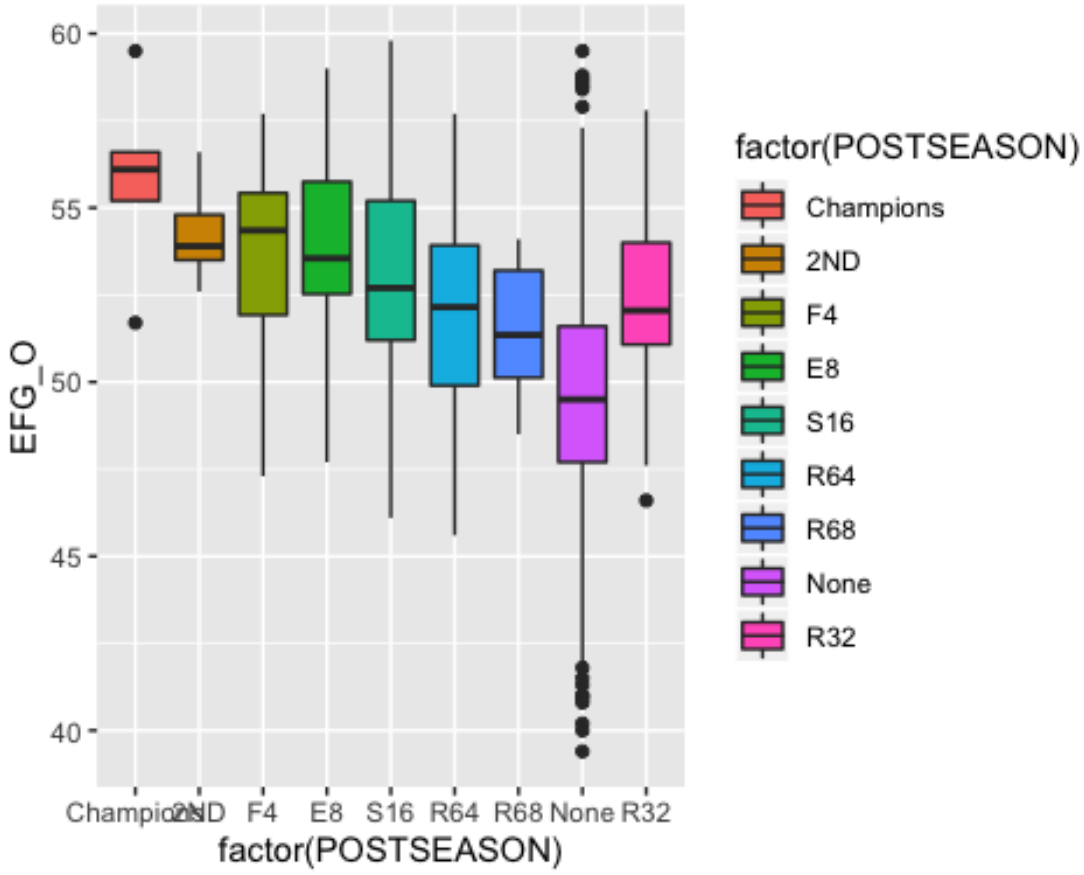
```
O.Vs.D <- ggplot(cbb, aes(x = EFG_D, y = EFG_O, color = Win_Perc)) + geom_point()
plot(O.Vs.D)
```



Offensive Efficiency Impact to Postseason Finish

If teams want to make a run in the postseason, they should focus on becoming more efficient offensivley. This could be taking higher percentage shots (2-pointers) or limiting isolation plays.

```
Postseason.O<- ggplot(cbb, aes(x = factor(POSTSEASON), y = EFG_O)) + geom_boxplot(aes(fill =
factor(POSTSEASON)))
plot(Postseason.O)
```



Machine Learning

Decision Tree Classification

Checks the optimal splits in the data to predict if a team will make the NCAA Tournament. First I create the Made.Tourney column which is a binary variable checking if a team made the tournament ('YES') or not ('NO').

Encoding Target Feature as Factor

```
cbb$Made.Tourney = factor(cbb$Made.Tourney, levels = c('YES','NO'))
```

Splitting Data into Training and Testing Sets

sample.split splits the data by randomly assigning the Boolean values to new column 'split'. 70% to the training set and 30% to the test set.

```
split = sample.split(cbb$Made.Tourney, SplitRatio = .7)
dtree_train = subset(cbb, select = c(ADJOE, ADJDE,EFG_O:ADJ_T,Made.Tourney), split == TRUE)
dtree_test = subset(cbb, select = c(ADJOE, ADJDE,EFG_O:ADJ_T,Made.Tourney), split == FALSE)
```

Fitting Decision Tree Classifier to Training Set

Uses Made.Tourney as the target feature.

```
library(rpart)

classifier = rpart(formula = Made.Tourney~.,data = dtree_train)
```

Predicting Test Set Results

Creates a vector of predictions as 'YES' or 'NO' to determine if the model predicted that observation to make the NCAA Tournament.

```
dtree_y_pred = predict(classifier, newdata = dtree_test[,-16], type = 'class')
```

Confusion Matrix

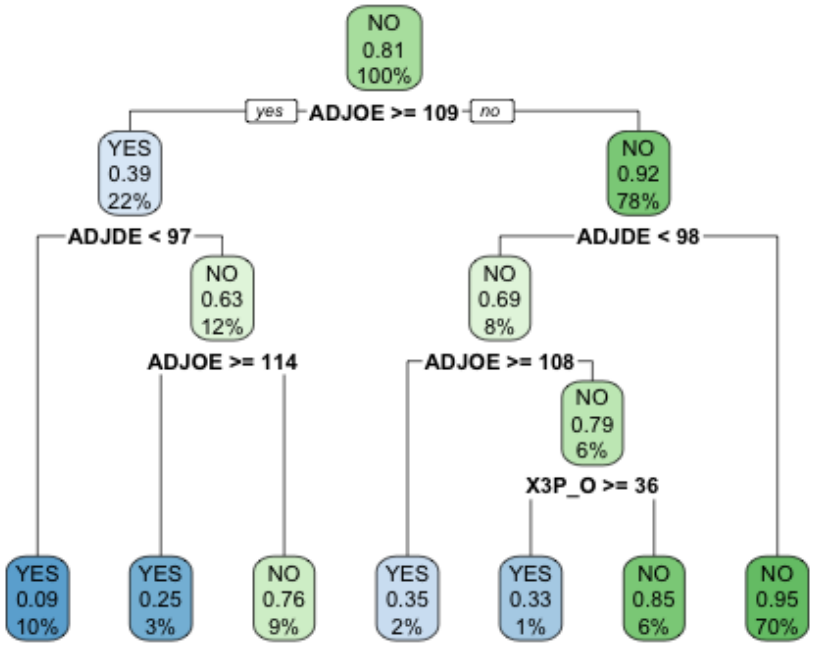
Confusion matrix displayed a 89% accuracy in predictions (470 Correct / 527 Total)

```
library(broom) # library containing tidy() to create tables
library(pander) # pander() to view tables

conf_mat = tidy(table(dtree_test[,16],dtree_y_pred))
pander(conf_mat)
```

Var1	dtree_y_pred	n
YES	YES	60
NO	YES	18
YES	NO	42
NO	NO	407

Plotting the Decision Tree

```
library(rpart.plot)

rpart.plot(classifier)
```



Displays which splits in the data are most optimal to predict if the team makes the NCAA Tournament

Let's see if it predicts if Villanova University will make the NCAA Tournament using their current stats: ADJOE = 112.6, ADJDE = 95.9, EFG_O = 53.1. According the decision tree, Villanova WILL make the 2020 NCAA Tournament.

Exploring the Big 5 Conferences

The "Big 5" conferences are the ACC, SEC, Big-10, Big-12 and Pac-12.

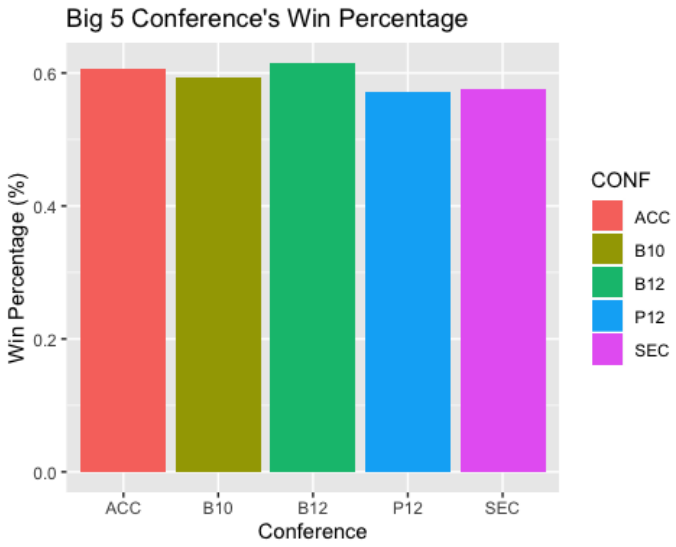
```
Big5 <- filter(cbb,CONF %in% c('ACC','SEC','B10','B12','P12'))
head(Big5)
```

```
##             TEAM CONF  G  W ADJOE ADJDE BARTHAG EFG_O EFG_D  TOR TORD  ORB  DRB
## 1 North Carolina  ACC 40 33 123.3  94.9  0.9531  52.6  48.1 15.4 18.2 40.7 30.0
## 2      Wisconsin  B10 40 36 129.1  93.6  0.9758  54.8  47.7 12.4 15.8 32.1 23.7
## 3       Michigan  B10 40 33 114.4  90.4  0.9375  53.9  47.7 14.0 19.5 25.5 24.9
## 4     Texas Tech  B12 38 31 115.2  85.2  0.9696  53.5  43.0 17.7 22.8 27.4 28.7
## 5           Duke  ACC 39 35 125.2  90.6  0.9764  56.6  46.5 16.3 18.6 35.8 30.2
## 6       Virginia  ACC 38 35 123.0  89.9  0.9736  55.2  44.7 14.7 17.5 30.4 25.4
##    FTR FTRD X2P_O X2P_D X3P_O X3P_D ADJ_T  WAB POSTSEASON YEAR  Win_Perc
## 1 32.3 30.4  53.9  44.6  32.7  36.2  71.7  8.6        2ND 2016 0.8250000
## 2 36.2 22.4  54.8  44.7  36.5  37.5  59.3 11.3        2ND 2015 0.9000000
## 3 30.7 30.0  54.7  46.8  35.2  33.2  65.9  6.9        2ND 2018 0.8250000
## 4 32.9 36.6  52.8  41.9  36.5  29.7  67.5  7.0        2ND 2019 0.8157895
## 5 39.8 23.9  55.9  46.3  38.7  31.4  66.4 10.7  Champions 2015 0.8974359
## 6 29.1 26.3  52.5  45.7  39.5  28.9  60.7 11.1  Champions 2019 0.9210526
##   Made.Tourney
## 1          YES
## 2          YES
## 3          YES
## 4          YES
## 5          YES
## 6          YES
```

Big 5 Win Percentage

Creates a bar chart with the average win percetnages for the Big 5 conferences.

```
Big5_Success <- ggplot(Big5, aes(x = CONF, y = Win_Perc, fill = CONF)) +
stat_summary(fun.y='mean',geom = 'bar') + ylab('Win Percentage (%)') + xlab('Conference') +
ggtitle("Big 5 Conference's Win Percentage")
plot(Big5_Success)
```



K-Means Clustering

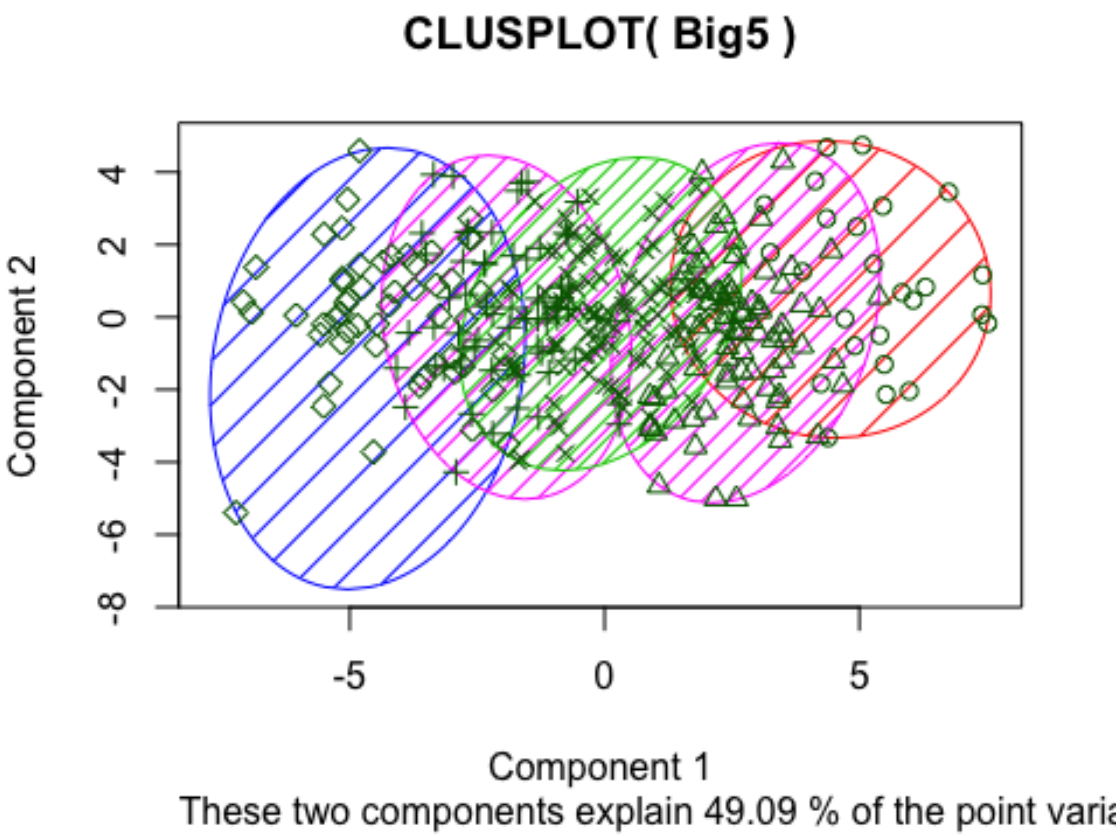
Creating the K Means Algorithm

In this instance, I know how many clusters there should be since there are 5 conferences (usually don't know this). If features like annual revenue, ticket prices, average attendance or location were available, they might be able to better predict the conference a team belongs to.

```
Big5.Cluster <- kmeans(x = Big5[,c('BARTHAG','X3P_O','WAB')], center = 5, nstart = 20)
```

Creating the Cluster Plot

```
library(cluster)

clusplot(Big5, Big5.Cluster$cluster, color = T, shade = T, labels = 0, lines = 0)
```



Multiple Linear Regression

Splitting the Data into Training and Testing Sets

sample.split splits the data by randomly assigning the Boolean values to new column 'sample'. 70% to the training set and 30% to the test set.

```
sample = sample.split(cbb$Win_Perc, SplitRatio = 0.7)
l.train = subset(cbb, select = c(TOR:X3P_D,Win_Perc), sample == TRUE)
l.test = subset(cbb, select = c(TOR:X3P_D,Win_Perc), sample == FALSE)
```

Fitting Regressor to Training Set

Want to check to make sure each variable has a p-value of < Significance Level (0.05). After checking the regressor, luckily all variables had a p-value of < 0.05.

```
reg = lm(Win_Perc~., data = l.train)
summary(reg)
```

```
##
## Call:
## lm(formula = Win_Perc ~ ., data = l.train)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.241424 -0.047051 -0.002827  0.046898  0.220282
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.4459694  0.0746892   5.971 3.08e-09 ***
## TOR         -0.0217653  0.0011219 -19.401  < 2e-16 ***
## TORD         0.0237703  0.0010803  22.004  < 2e-16 ***
## ORB          0.0095123  0.0005410  17.582  < 2e-16 ***
## DRB         -0.0118465  0.0007248 -16.346  < 2e-16 ***
## FTR          0.0026992  0.0004499   5.999 2.60e-09 ***
## FTRD        -0.0031699  0.0004095  -7.741 2.05e-14 ***
## X2P_O        0.0167060  0.0007031  23.761  < 2e-16 ***
## X2P_D       -0.0119087  0.0006939 -17.162  < 2e-16 ***
## X3P_O        0.0146799  0.0008295  17.698  < 2e-16 ***
## X3P_D       -0.0178301  0.0009032 -19.740  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06952 on 1231 degrees of freedom
## Multiple R-squared:  0.8516, Adjusted R-squared:  0.8504
## F-statistic: 706.4 on 10 and 1231 DF,  p-value: < 2.2e-16
```

Getting Predictions off of the Test Set

```
Win.Pred <- predict(reg, newdata = l.test)
```

Viewing the Results in a Data Frame: 'results'

```
results <- cbind(Win.Pred, l.test$Win_Perc)
colnames(results) <- c('Predictions','Actual')
results <- as.data.frame(results)
pander(head(results))
```

	Predictions	Actual
1	0.7699	0.825
27	0.8365	0.8421
32	0.7513	0.6923
45	0.5374	0.5152
47	0.7167	0.6875
48	0.6969	0.6

Evaluating Prediction Results

R^2 is 0.824 which means the model was pretty good at using the independent features to predict the target variable.

```
MSE <- mean((results$Actual - results$Predictions)^2)
RMSE <- MSE^0.5
SSE <- sum((results$Predictions - results$Actual)^2)
```

```
SST <- sum((mean(cbb$Win_Perc) - results$Actual)^2)
R2 <- 1 - (SSE/SST)
stat.names <- c('MSE','RMSE','SSE','SST','R2')
stat.values <- c(MSE,RMSE,SSE,SST,R2)
stats <- data.frame(stat.names,stat.values)
pander(stats)
```

stat.names	stat.values
MSE	0.005243
RMSE	0.07241
SSE	2.7
SST	15.8
R2	0.8291

Using the Linear Regression Model to Predict

Predicting 2020 Villanova Basektball Team Win % (Current Win % = 78.5%)

```
villanova_pred <- predict(reg, data.frame(
  TOR = 16.7,
  TORD = 17.8,
  ORB = 29.2,
  DRB = 24,
  FTR = 25.2,
  FTRD = 24.5,
  X2P_O = 55.4,
  X2P_D = 49.8,
  X3P_O = 32.9,
  X3P_D = 31.9))
print(paste('Model predicts Villanova will win',round(100*villanova_pred,2),'% of their games in the 2020 season'))
```

```
## [1] "Model predicts Villanova will win 73.6 % of their games in the 2020 season"
```

Conclusion

This was a very interesting dataset to work with, especially because I follow college basketball and never watched with analytics in mind. Being new to R, this was good practice to manipulate, analyze and visualize data that I was curious in.

One key observations I made was the impact of being efficient offensively had on winning and making a deep postseason run. Usually you hear on TV that “Defense wins games”, but this analysis, to a degree, says otherwise. Teams should look to focus on making every offensive possession count with high-percentage shots. There is definitley a lot more to explore with this data and I look forward to analyzing it more.