# NBA Data Science Project

*Dennis Li*

*2018-07-28*

## Intro

The 2 main questions I hoped to answer that motivated me to do this project were:

- How well I could detect certain clusters of players (superstars, starters, role players, scrubs etc.) from the 2017-18 regular season using dimensionality reduction and k-means cluster analysis
- How strong of a predictive model I could create to predict an NBA player's position (traditional PG/SG/SF/PF/C) based on per-game and per-36 features and other attributes from the past 4 regular seasons, given that the NBA is transitioning to a more "positionless" league. Also would like to see how strong of a model I could make to predict a binary position classification (guard/wing, big)

## Loading Packages

```r
library(ggplot2)
library(kableExtra)
library(corrplot)
library(tidyverse)
library(cluster)
library(factoextra)
library(gridExtra)
library(dplyr)
library(tree)
library(randomForest)
library(xgboost)
library(caret)
library(glmnet)
```

## Data Collection/Cleaning

For each of the past 4 seasons, the "seasonStats" dataset contains individual per-36 minute and per-game player statistics for every player who appeared in a regular season game. Aggregate statistics are used for players traded during that season and the team that the traded player is assigned to is whichever he played more minutes for. Only players who played over 20 games are used. Per-36 stats are included to minimize bias against players whose stats may be inflated due to high playing time.

Additional data, such as height, weight, nationality and experience are also included.

Pos2 is the binary position classification. Guards/SF's are separated from PF/C's.

All data comes from basketball-reference.com.

### 2017-2018 Regular Season

```r
perGame_17 <- read.csv("perGame17.csv")
perGame_17$Player <- gsub("\\\\.*", "", perGame_17$Player)
```

```r
perGame_17 <- perGame_17[perGame_17$G >= 20,]

per36_17<- read.csv("per36_17.csv")
per36_17$Player <- gsub("\\\\.*", "", per36_17$Player)

seasonStats17 <- merge(x = perGame_17, y = per36_17, by = "Player")


body17 <- read.csv("nbaBody17.csv")
body17$Player <- gsub("\\\\.*", "", body17$Player)
names(body17)[7] <- "Nationality"
body17 <- body17[,c(2,4,5,7,8)]
body17 <- unique(body17)

#Assign rookies 0 years of experience rather than "R"
body17$Exp <- as.numeric(levels(body17$Exp))[body17$Exp]
for(i in 1:nrow(body17)){
  if(is.na(body17$Exp[i]) == TRUE){
    body17$Exp[i] <- 0
  }
}

#Convert height from feet-inches to inches
input <- body17$Ht
feet <- substr(input, start = 1, stop = 1)
inches <- substr(input, start = 3, stop = 4)
feet <- as.integer(feet)
inches <- as.integer(inches)
output <- feet*12 + inches
body17$Ht <- output


#Joining data
nba17 <- merge(x = seasonStats17, y = body17, by = "Player")



#Add Pos2 column
nba17$Pos2 <- with(nba17, ifelse(nba17$Pos == "PF" | nba17$Pos == "C", "Big", "Guard/Wing"))

#Add US column
nba17$USA <- with(nba17, ifelse(nba17$Nationality == "us", "Yes", "No"))

nba17[, "Pos2"] <- as.factor(nba17[, "Pos2"])
nba17[, "USA"] <- as.factor(nba17[, "USA"])
```

### 2016-2017 Regular Season

```r
perGame_16 <- read.csv("perGame16.csv")
perGame_16$Player <- gsub("\\\\.*", "", perGame_16$Player)
perGame_16 <- perGame_16[perGame_16$G >= 20,]
```

```r
per36_16<- read.csv("per36_16.csv")
per36_16$Player <- gsub("\\\\.*", "", per36_16$Player)

seasonStats16 <- merge(x = perGame_16, y = per36_16, by = "Player")


body16 <- read.csv("nbaBody16.csv")
body16$Player <- gsub("\\\\.*", "", body16$Player)
names(body16)[7] <- "Nationality"
body16 <- body16[,c(2,4,5,7,8)]
body16 <- unique(body16)

#Assign rookies 0 years of experience rather than "R"
body16$Exp <- as.numeric(levels(body16$Exp))[body16$Exp]
for(i in 1:nrow(body16)){
  if(is.na(body16$Exp[i]) == TRUE){
    body16$Exp[i] <- 0
  }
}

#Convert height from feet-inches to inches
input <- body16$Ht
feet <- substr(input, start = 1, stop = 1)
inches <- substr(input, start = 3, stop = 4)
feet <- as.integer(feet)
inches <- as.integer(inches)
output <- feet*12 + inches
body16$Ht <- output


#Joining data
nba16 <- merge(x = seasonStats16, y = body16, by = "Player")


#Add Pos2 column
nba16$Pos2 <- with(nba16, ifelse(nba16$Pos == "PF" | nba16$Pos == "C", "Big", "Guard/Wing"))

#Add US column
nba16$USA <- with(nba16, ifelse(nba16$Nationality == "us", "Yes", "No"))
nba16[, "Pos2"] <- as.factor(nba16[, "Pos2"])
nba16[, "USA"] <- as.factor(nba16[, "USA"])
```

**2015-2016 Regular Season**

```r
perGame_15 <- read.csv("perGame15.csv")
perGame_15$Player <- gsub("\\\\.*", "", perGame_15$Player)
perGame_15 <- perGame_15[perGame_15$G >= 20,]

per36_15<- read.csv("per36_15.csv")
per36_15$Player <- gsub("\\\\.*", "", per36_15$Player)

seasonStats15 <- merge(x = perGame_15, y = per36_15, by = "Player")
```

```r
body15 <- read.csv("nbaBody15.csv")
body15$Player <- gsub("\\\\.*", "", body15$Player)
names(body15)[7] <- "Nationality"
body15 <- body15[,c(2,4,5,7,8)]
body15 <- unique(body15)

#Assign rookies 0 years of experience rather than "R"
body15$Exp <- as.numeric(levels(body15$Exp))[body15$Exp]
for(i in 1:nrow(body15)){
  if(is.na(body15$Exp[i]) == TRUE){
    body15$Exp[i] <- 0
  }
}

#Convert height from feet-inches to inches
input <- body15$Ht
feet <- substr(input, start = 1, stop = 1)
inches <- substr(input, start = 3, stop = 4)
feet <- as.integer(feet)
inches <- as.integer(inches)
output <- feet*12 + inches
body15$Ht <- output


#Joining data
nba15 <- merge(x = seasonStats15, y = body15, by = "Player")


#Add Pos2 column
nba15$Pos2 <- with(nba15, ifelse(nba15$Pos == "PF" | nba15$Pos == "C", "Big", "Guard/Wing"))

#Add US column
nba15$USA <- with(nba15, ifelse(nba15$Nationality == "us", "Yes", "No"))
nba15[, "Pos2"] <- as.factor(nba15[, "Pos2"])
nba15[, "USA"] <- as.factor(nba15[, "USA"])
```

### 2014-2015 Regular Season

```r
perGame_14 <- read.csv("perGame14.csv")
perGame_14$Player <- gsub("\\\\.*", "", perGame_14$Player)
perGame_14 <- perGame_14[perGame_14$G >= 20,]

per36_14<- read.csv("per36_14.csv")
per36_14$Player <- gsub("\\\\.*", "", per36_14$Player)

seasonStats14 <- merge(x = perGame_14, y = per36_14, by = "Player")


body14 <- read.csv("nbaBody14.csv")
body14$Player <- gsub("\\\\.*", "", body14$Player)
names(body14)[7] <- "Nationality"
```

```
body14 <- body14[,c(2,4,5,7,8)]
body14 <- unique(body14)

#Assign rookies 0 years of experience rather than "R"
body14$Exp <- as.numeric(levels(body14$Exp))[body14$Exp]
for(i in 1:nrow(body14)){
  if(is.na(body14$Exp[i]) == TRUE){
    body14$Exp[i] <- 0
  }
}

#Convert height from feet-inches to inches
input <- body14$Ht
feet <- substr(input, start = 1, stop = 1)
inches <- substr(input, start = 3, stop = 4)
feet <- as.integer(feet)
inches <- as.integer(inches)
output <- feet*12 + inches
body14$Ht <- output


#Joining data
nba14 <- merge(x = seasonStats14, y = body14, by = "Player")


#Add Pos2 column
nba14$Pos2 <- with(nba14, ifelse(nba14$Pos == "PF" | nba14$Pos == "C", "Big", "Guard/Wing"))

#Add US column
nba14$USA <- with(nba14, ifelse(nba14$Nationality == "us", "Yes", "No"))
nba14[, "Pos2"] <- as.factor(nba14[, "Pos2"])
nba14[, "USA"] <- as.factor(nba14[, "USA"])
```

### Final 4-season data

```
nba1 <- rbind(nba14, nba15)
nba2 <- rbind(nba1, nba16)
nba <- rbind(nba2, nba17)
nba$Pos <- factor(nba$Pos,c("PG","SG","SF","PF","C"))
#Add BMI variable
nba <- nba %>%
  mutate(bmi = Wt / (Ht^2)*703)
```

## Exploratory Data Analysis

Some basic summary statistics and visuals to help get a high level understanding of what features are important.

```
head(nba)[,1:10]
```

```
##          Player Pos Age  Tm  G GS   MP  FG  FGA FGpct
## 1    A.J. Price  PG  28 IND 26  0 12.5 2.0  5.3 0.372
## 2   Aaron Brooks  PG  30 CHI 82 21 23.0 4.2 10.0 0.421
## 3   Aaron Gordon  PF  19 ORL 47  8 17.0 2.0  4.4 0.447
## 4 Adreian Payne  PF  23 MIN 32 22 23.1 2.8  6.9 0.414
## 5     Al Horford   C  28 ATL 76 76 30.5 6.8 12.7 0.538
## 6   Al Jefferson   C  30 CHO 65 61 30.6 7.5 15.5 0.481
```

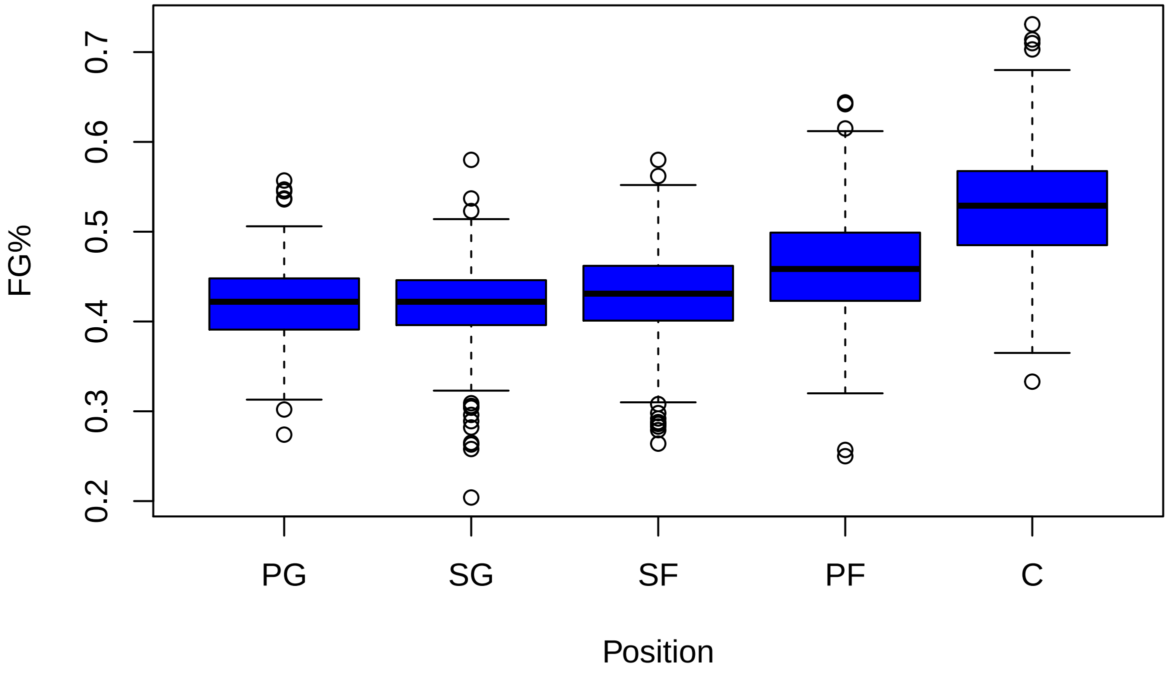
```
dim(nba)
```

```
## [1] 1664   53
```

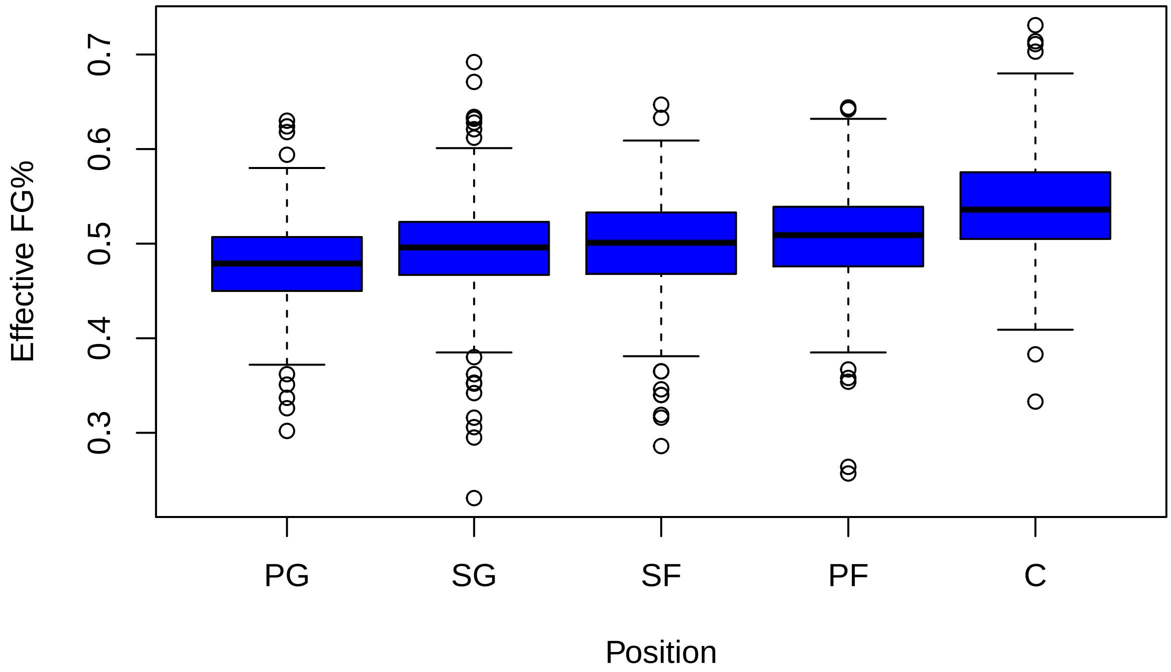
```
summary(nba$Pos)
```

```
##  PG  SG  SF  PF   C
## 334 359 305 342 324
```

```
boxplot(nba$FGpct~nba$Pos, col = "blue", ylab = "FG%", xlab = "Position")
```



```
boxplot(nba$eFGpct~nba$Pos, col = "blue", ylab = "Effective FG%", xlab = "Position")
```

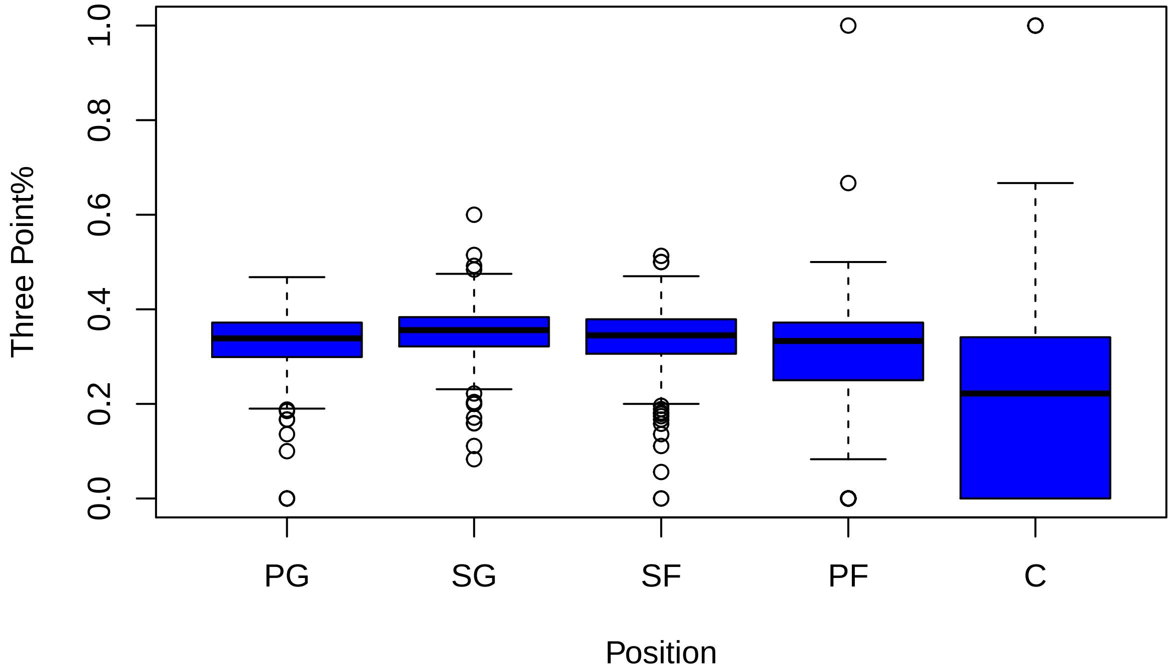


Effective FG% is a metric that takes into account that three pointers are worth more than 2 pointers (used heavily in the modern "moneyball" approach pioneered by Daryl Morey):

eFG% = (FGM + (0.5 x 3PTM)) / FG

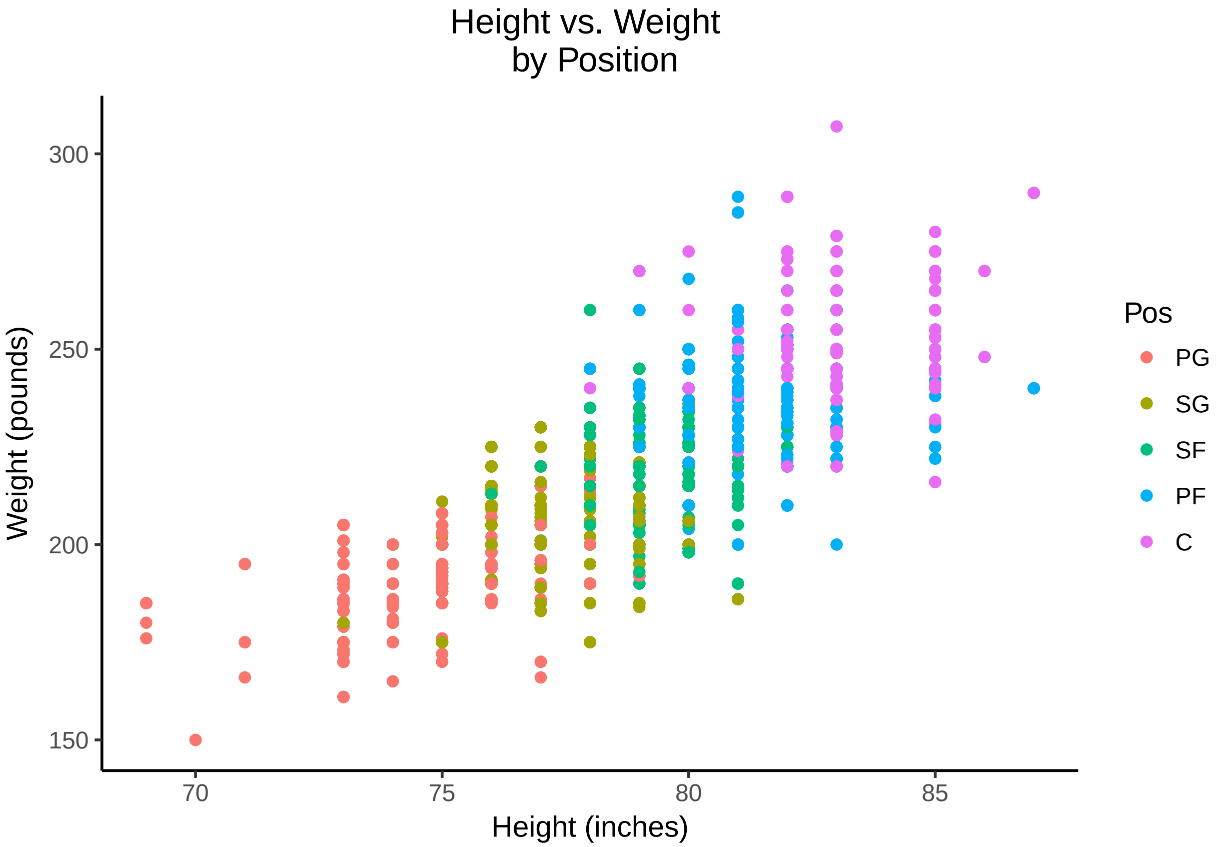
As expected, centers on average have the highest FG% and eFG%, but the edge is reduced when weighing 3's into the metric.

```r
boxplot(nba$threePct~nba$Pos, col = "blue", ylab = "Three Point%", xlab = "Position")
```

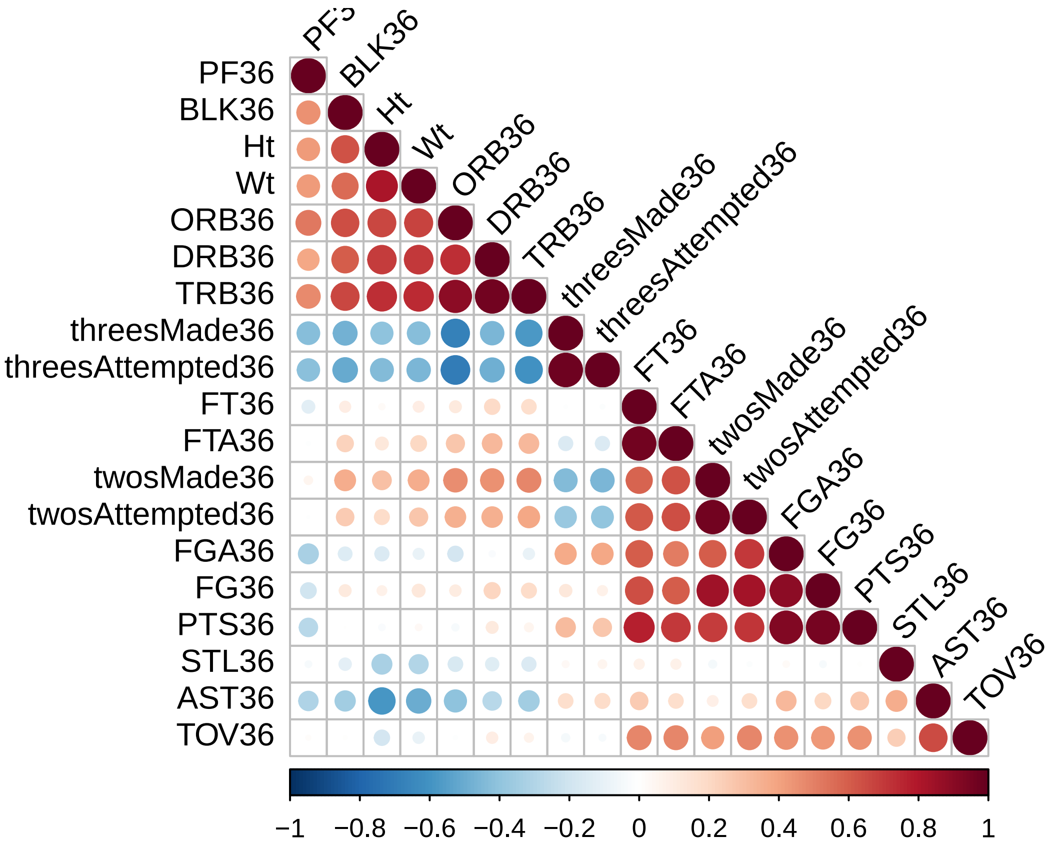


The stretch 4 is in full effect, although the variability is still larger than PG/SG/SF's as seen in the IQR. On average, it is hard to distinguish positions 1 through 4 based on 3P% in today's NBA.

```
p <- ggplot(nba, aes(x = Ht, y = Wt, color = Pos)) + geom_point() +
  labs(title="Height vs. Weight \n by Position",
       x="Height (inches)", y = "Weight (pounds)")
p + theme_classic() + theme(plot.title = element_text(hjust=0.5))
```



```
nbaCorr <- nba[,30:48]
source("http://www.sthda.com/upload/rquery_cormat.r")
rquery.cormat(nbaCorr)
```



An easy-to-interpret correlation plot of the per-36 features. Certain features are highly correlated with many other features (such as TRB36), while others like STL36 are not highly correlated with any other features. As you can tell from this subset of the data, high correlation between variables exist. Let's move to PCA to try to transform this dataset.

## K-Means Cluster Analysis and PCA

Using data from the 2017-2018 regular season, I want to see how well I can cluster the players into different categories to judge skill level.

The intuition behind using PCA and k-means clustering is that principal component analysis is a dimensionality reduction technique which does not actually perform feature selection/removal, but rather performs orthogonal transformations to project the data on a lower dimension, resulting in less multicollinearity and more interpretability without loss of information.

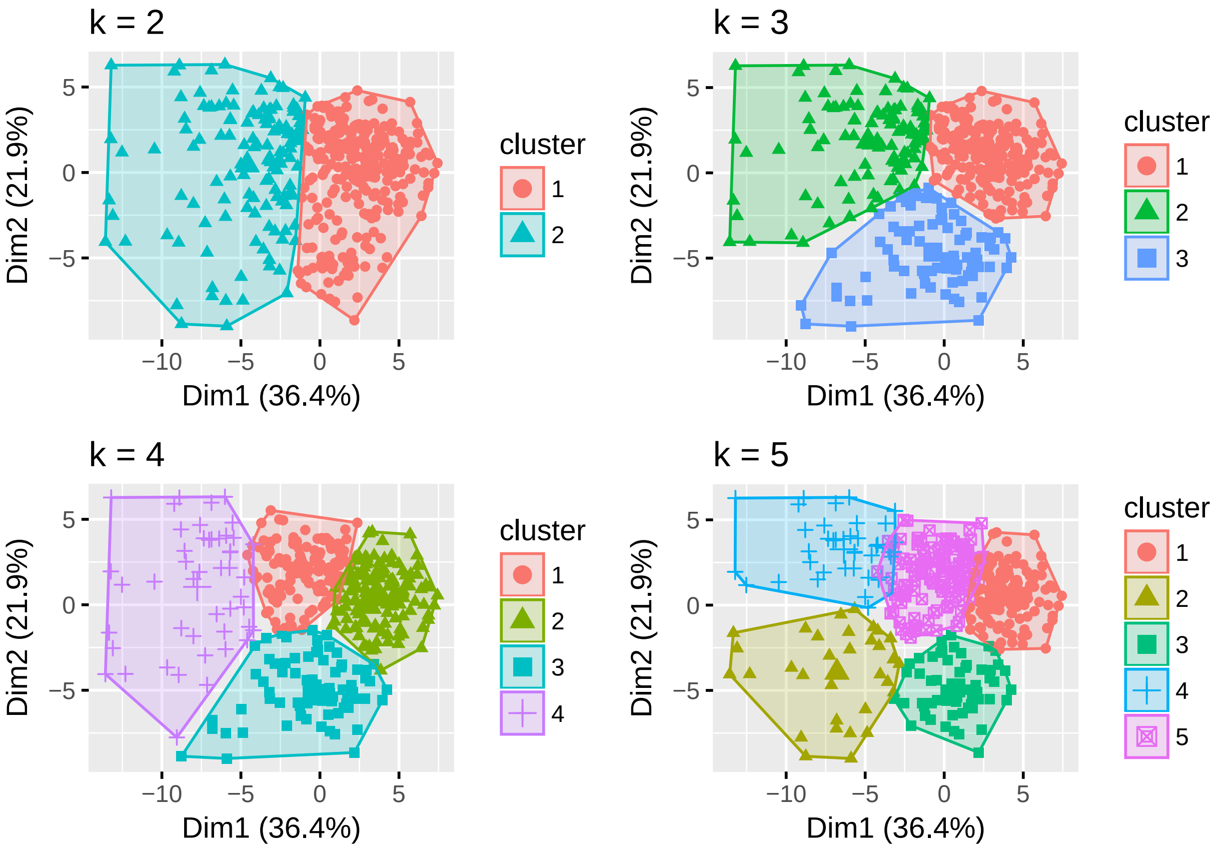
In a similar dynamic, k-means is a non-parametric algorithm to represent the data into a small number of cluster centroids. Applying PCA before k-means is (hopefully) a way to reduce the noise and improve clustering results.

I'll be using the "cluster" and "factoextra" packages to apply these techniques and cluster with 2 principal components.

```r
#remove categorical predictors
nbaCluster <- nba17[,-c(2,4,49,51,52)]
nbaCluster <- na.omit(nbaCluster)
index <- sapply(nbaCluster, is.numeric)
nbaCluster[index] <- lapply(nbaCluster[index], scale)
#we scale because k-means and other cluster algorithms are based on distances between points (euclidean
k2 <- kmeans(nbaCluster[,-1], centers = 2, nstart = 25)
k3 <- kmeans(nbaCluster[,-1], centers = 3, nstart = 25)
k4 <- kmeans(nbaCluster[,-1], centers = 4, nstart = 25)
k5 <- kmeans(nbaCluster[,-1], centers = 5, nstart = 25)

# Roughly comparing the clusters, seeing if there's any delinations
p1 <- fviz_cluster(k2, geom = "point", data = nbaCluster[,-1]) + ggtitle("k = 2")
p2 <- fviz_cluster(k3, geom = "point",  data = nbaCluster[,-1]) + ggtitle("k = 3")
p3 <- fviz_cluster(k4, geom = "point",  data = nbaCluster[,-1]) + ggtitle("k = 4")
p4 <- fviz_cluster(k5, geom = "point",  data = nbaCluster[,-1]) + ggtitle("k = 5")

grid.arrange(p1, p2, p3, p4, nrow = 2)
```



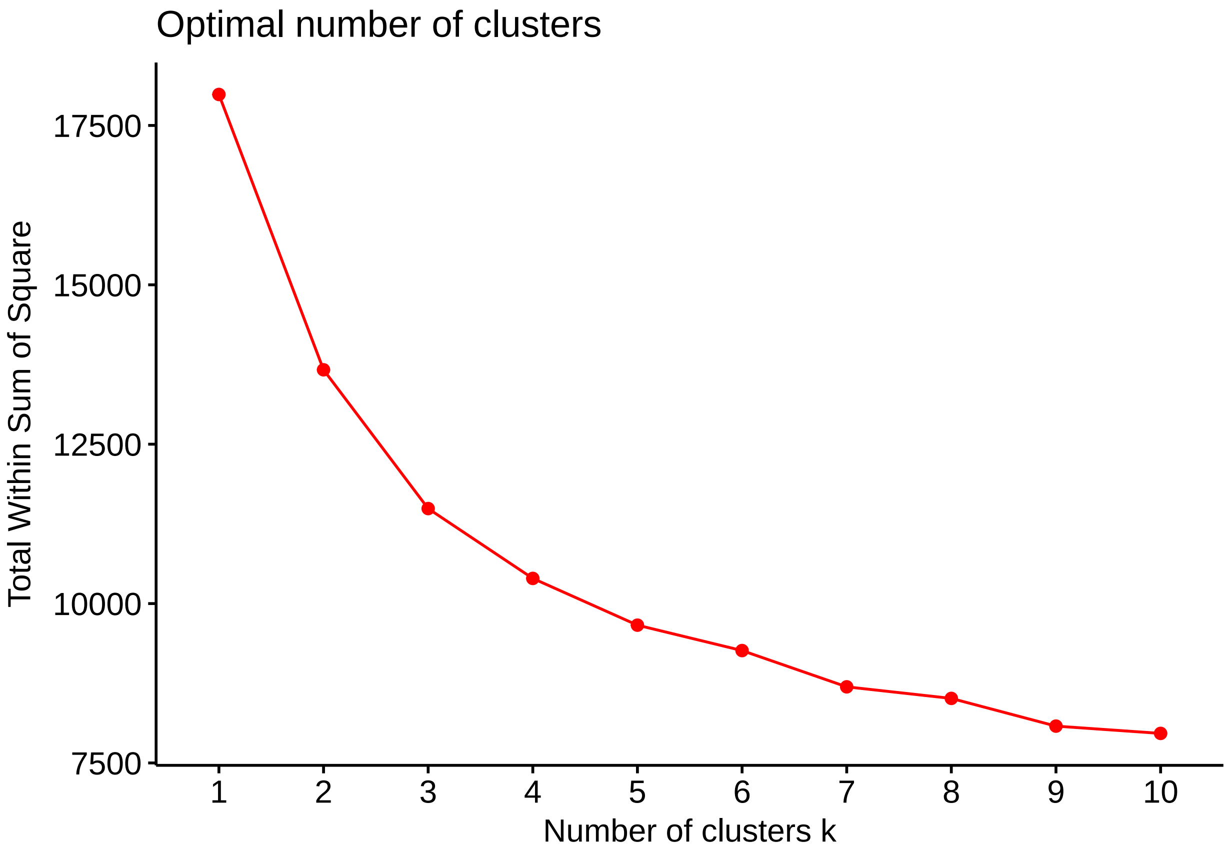
This visual shows us that k = 2, 3, 4 and 5 all do a pretty good job at making distinct clusters. However, if we want to see what the optimal number of clusters is, we can use the elbow method. This is a good way to achieve the goal of clustering the data, which is to minimize the within-cluster sum of squares variation.

```
set.seed(999)

#calculates within cluster variation
# withinVariation <- function(k) {
#   kmeans(nbaCluster[,-1], k, nstart = 25 )$tot.withinss #tot.withinss is the value we are extracting
# }
#
# #Consider values 1 through 10
# kValues <- 1:10
#
```

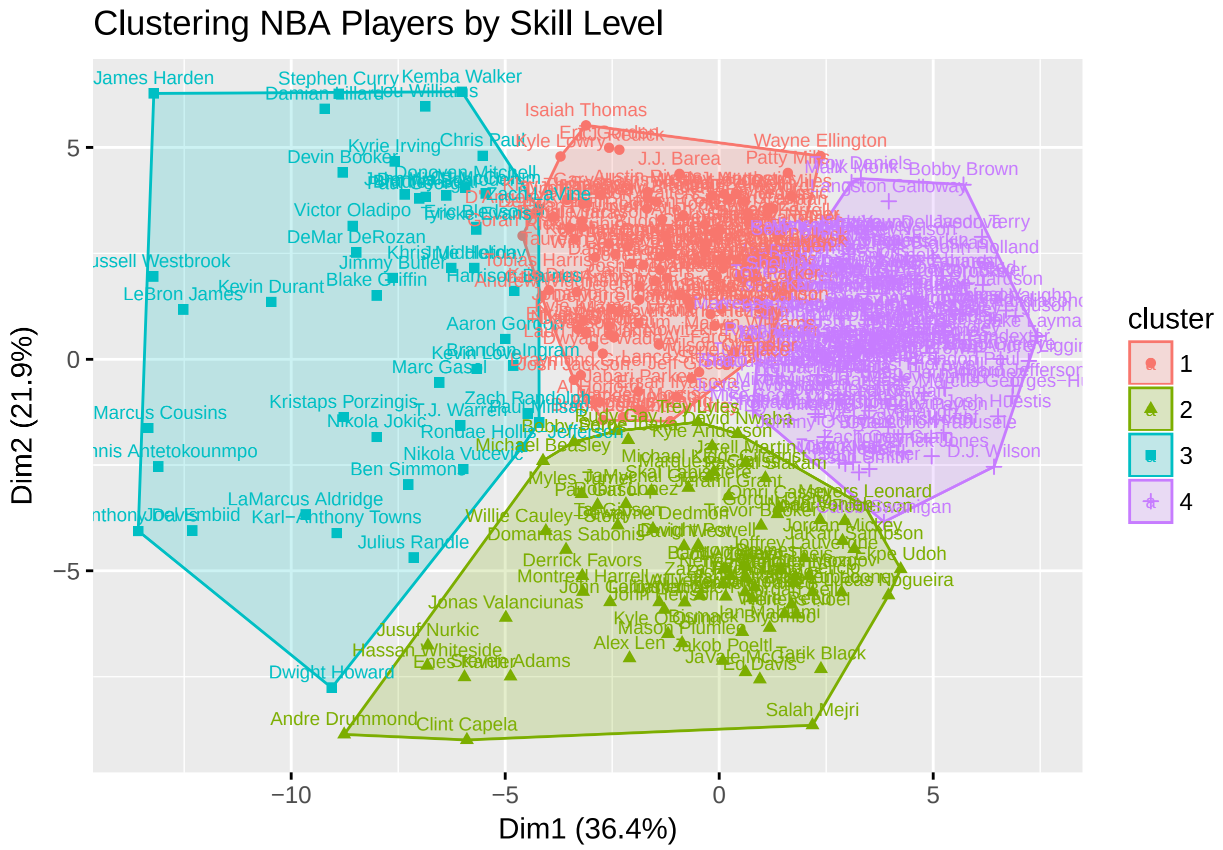
```
# # maps the value of K to the corresponding within-cluster variation value
# withinSSvalues <- map_dbl(kValues, withinVariation)
#
# plot(kValues, withinSSvalues,
#        type="b", pch = 19, frame = TRUE,
#        xlab="Number of Clusters",
#        ylab="Total Within-Clusters SS")

fviz_nbclust(nbaCluster[,-1], kmeans, method = "wss", k.max = 10, linecolor = "red")
```



The elbow plot suggests that a value around k = 4 is a good (yet arbitrary) cut-off point. This uses factoextra's build-in function but I provided code that does it manually.

```
optimalCluster <- kmeans(nbaCluster[,-1], 4, nstart = 25)
nbaCluster$Cluster <- optimalCluster$cluster
rownames(nbaCluster) <- nbaCluster$Player
fviz_cluster(optimalCluster, data = nbaCluster[,-c(1,48)], labelsize = 7, show.clust.cent = FALSE, main
```



The clustering results show some interesting patterns in the data for the 2017-2018 regular season. Just from a rough visual analysis of where the players are placed, it appears that the first principal component (Dim 1, which explains 36% of the total variance) is a measure relating to **scoring prowess** of the players. The second principal component (Dim 2, which explains 21.4% of the total variance) is a little less clear, but it seems that guards populate the bottom half of the plot while big men dominate the top half. The k-means algorithm does not take Position (or any categorical variable) into consideration so this is probably some sort of linear combination that represents **ball-handling ability**.

Although the plot is quite crowded, some big names pop up when examining the  blue. Most of the players here are superstars and all-star level players. The few exceptions that jump out are Rondae Hollis-Jefferson, TJ Warren (although he has proven to be a very effective scorer, but not much else) and Dwight Howard (lol). Some others, like Julius Randle and Brandon Ingram, are probable to prove their worth as they get more minutes and usage. You could make the argument that Howard still belongs in that category when just looking at numbers, though. That hurt to say. Sorry Carmelo and Draymond. Lets call this cluster **"Studs"**.

The purple cluster is filled (apologies for the spacing) with players who spent most of their time during the 2017-18 regular season riding the bench. Many of these players are young players who have a lot of room to grow, but for the time being, players like D.J. Wilson have a lot to prove. Let's call these players **"Scrubs"**.

The red and green clusters which occupy the middle section of the plot are filled with role players. Specifically, the red cluster is populated with ball-handling role players such as Lonzo Ball and the green cluster is non-ball handling role players such as Robin Lopez. Some players who were categorized in these 2 clusters may not come to mind as role players, but also probably don't deserve to be in the blue cluster (Andrew Wiggins, Isaiah Thomas, Jusuf Nurkic). Others, such as Andre Drummond and Kyle Lowry are borderline all-stars, but can't carry the team by themselves. Let's call these players **"Ball Handling Role Players"** and **"Off Ball Role Players"**.

Let's take a deeper look into how last year's actual all-stars were clustered, and if some of them didn't deserve to be in the game (based on clustering results). All-star replacements are considered.

```
allStarList <- c("LeBron James", "Kevin Durant", "Russell Westbrook", "Kyrie Irving", "Anthony Davis",
nbaCluster$AllStar <- ifelse(nbaCluster$Player %in% allStarList,"Yes", "No")
nbaCluster$Cluster <- as.factor(nbaCluster$Cluster)
#Rename clusters 1,2,3,4 into "BallHandlingRolePlayer", "OffBallRolePlayer", "Studs", "Scrubs"
levels(nbaCluster$Cluster)[levels(nbaCluster$Cluster)=="1"] <- "BallHandlingRolePlayer"
levels(nbaCluster$Cluster)[levels(nbaCluster$Cluster)=="2"] <- "OffBallRolePlayer"
levels(nbaCluster$Cluster)[levels(nbaCluster$Cluster)=="3"] <- "Stud"
levels(nbaCluster$Cluster)[levels(nbaCluster$Cluster)=="4"] <- "Scrub"

ClusterTable <- nbaCluster[,c(1, 48, 49)]
AllStarClusterTable <- ClusterTable[ClusterTable$AllStar == "Yes",]


rownames(AllStarClusterTable) <- NULL
kable(AllStarClusterTable[,c(1,2)]) %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"))
```

| Player | Cluster |
|---|---|
| Al Horford | BallHandlingRolePlayer |
| Andre Drummond | OffBallRolePlayer |
| Anthony Davis | Stud |
| Bradley Beal | Stud |
| Damian Lillard | Stud |
| DeMar DeRozan | Stud |
| DeMarcus Cousins | Stud |
| Draymond Green | BallHandlingRolePlayer |
| Giannis Antetokounmpo | Stud |
| Goran Dragic | BallHandlingRolePlayer |
| James Harden | Stud |
| Jimmy Butler | Stud |
| Joel Embiid | Stud |
| John Wall | Stud |
| Karl-Anthony Towns | Stud |
| Kemba Walker | Stud |
| Kevin Durant | Stud |
| Kevin Love | Stud |
| Klay Thompson | BallHandlingRolePlayer |
| Kristaps Porzingis | Stud |
| Kyle Lowry | BallHandlingRolePlayer |
| Kyrie Irving | Stud |
| LaMarcus Aldridge | Stud |
| LeBron James | Stud |
| Paul George | Stud |
| Russell Westbrook | Stud |
| Stephen Curry | Stud |
| Victor Oladipo | Stud |

Out of the 28 all-stars, 22 of them were placed in the stud cluster using k-means analysis combined with PCA. Kyle Lowry, Klay Thompson, Goran Dragic, Draymond Green, Andre Drummond and Al Horford (who was clustered as a Ball Handling Role Player which further supports his heavier point-forward role with the Celtics) were snubbed. However, 4 of the 6 (Lowry, Dragic, Green and Horford) arguably got in for team contributions/success rather than statistical prowess.