

Contributions:

---Yuhong Fang: R Coding, Structure of Analysis

---Yiqi Ren: Introduction, Analysis plan summary, Conclusion

---Jung-Cheng Chang: Methodologies, Role Categories Description, Lineups Summary

---Zhongrui(Harry) Wang: Attributes Analysis and Player Role Categorizing

NBA Playoff Teams Lineup Analysis Project

1. Introduction:

1.1 Goals of this project & Background

Sports analysis is playing a larger role in the basketball industry these years. In this project, We are interested in what type of roster combination is more likely to make an NBA team more competitive in qualifying for the playoffs and even become an NBA champion team. We will find the relationship between specific team roster combinations of the playoff teams in one season and the number of team wins of respective teams in that season, then we will compare different teams among various ranking and predict a reasonable teammates combination in general. Once we defined what a playoff team looks like, we will be able to draw out a pattern of a champion team and give suggestions to the future playoff teams.

1.2 Dataset information

We grab data from the official stats from NBA.com. The dataset contains 2524 basketball players among the past 5 seasons, including 2014-2019, and each player was evaluated by 42 (variables) attributes. For example, age, scores, games play, win rate, offensive rating, minutes play, etc. In these variables, some of them are collected from player's game play directly like scoring and games play. But some variables are combined by others, for example, offensive rating and defensive rating. This represents that some of variables are independent and some of them are correlated.

1.3 Key Questions

- What attributes (variables) among all technical skills are the most informative ones to analyse the player groupings?
- How to categorize different players and how many clusters there are?
- How are different types of players distributed in those successful teams?
- How is a good team lineup for a champion team?

2. Analysis Plan/Methodologies

The main methodologies will include standardization, PCA, and Clustering. Standardized data is important in PCA since it is a variance maximizing exercise. It projects your original data onto directions which maximize the variance. In our analysis, we will use *scale()* method to scale all of the data for each player. We will have more details in the next Standardization section. The methodology will be the principal components analysis, we will use PCA method to filter out several “essential” attributes, such as offensive ratings and defensive ratings, which can explain the majority of the total variance and summarize the majority of all the original basketball features. The last methodology is doing clustering by using k-means. Clustering helps us to define what type of player roles and how many types of players roles there are based on the essential attributes we obtained. We expect that each player grouping occupies similar functional roles. The next step will be to observe the number of each group of players in these successful teams and if there are some relationships between specific combinations of the team roster and the number of wins of that team in that season. In the end, we will construct an NBA champion team’s roster.

2.1 Standardization

We standardized the data to standard normal distribution prior to performing PCA to make the variables comparable and combinable. The procedure is essential when the parameters in a model do not have the same unit and scale. Additionally, this is required as the assumption of PCA and K-means clustering. In our analysis, for different aspects of NBA statistics, the value can differ by a factor of up to 100. For instance, defense parameters, such as block and steal, rarely exceed 3.0 and have the unit of “per game”, while rating parameters are mostly in 100s range and do not have a unit. Therefore, standardization has to be done to prevent unintentional weighting of parameters.

2.2 Principal Component Analysis (PCA)

PCA is a procedure that transforms a large number of correlated variables into a smaller number of uncorrelated variables, known as principal components (PCs). In other words, it determines the minimum number of variables that accounts for maximum variance. The brief algorithm is given by $PCA(X) = (X - \mu)P$, which normalize our data X to zero mean. The covariance matrix Σ_x is calculated by solving $(X - \mu_x)^T(X - \mu_x)$. Lastly, the orthogonal eigenvector is found by $\Sigma_x V = V\Lambda$. We hope the PCs can explain the grouping that we will later obtained well, and yet we would like to preserve as much information as possible. By examining the first 10 PCs, we are able to give each of them a reasoning. Starting at 11th PC, it became much harder to give each PC an explanation. Furthermore, the first 10 PCs already contain 88% information of the original data. We considered the amount of information loss by including only 10 PCs to be reasonable. Therefore, 10 PCs are included as variables for K-means clustering in the following step.

2.3 Clustering (K-Means)

K-means is a common unsupervised clustering method. By providing a value of k, the algorithm looks for a clustering that minimizes the within-cluster variation.

Mathematically, the following is solved:

$$\arg \min (s) \sum_{i=1}^k \sum_{x \in S_i} \|x - \mu_i\|^2 = \arg \min (s) \sum_{i=1}^k |S_i| \text{Var}(S_i)$$

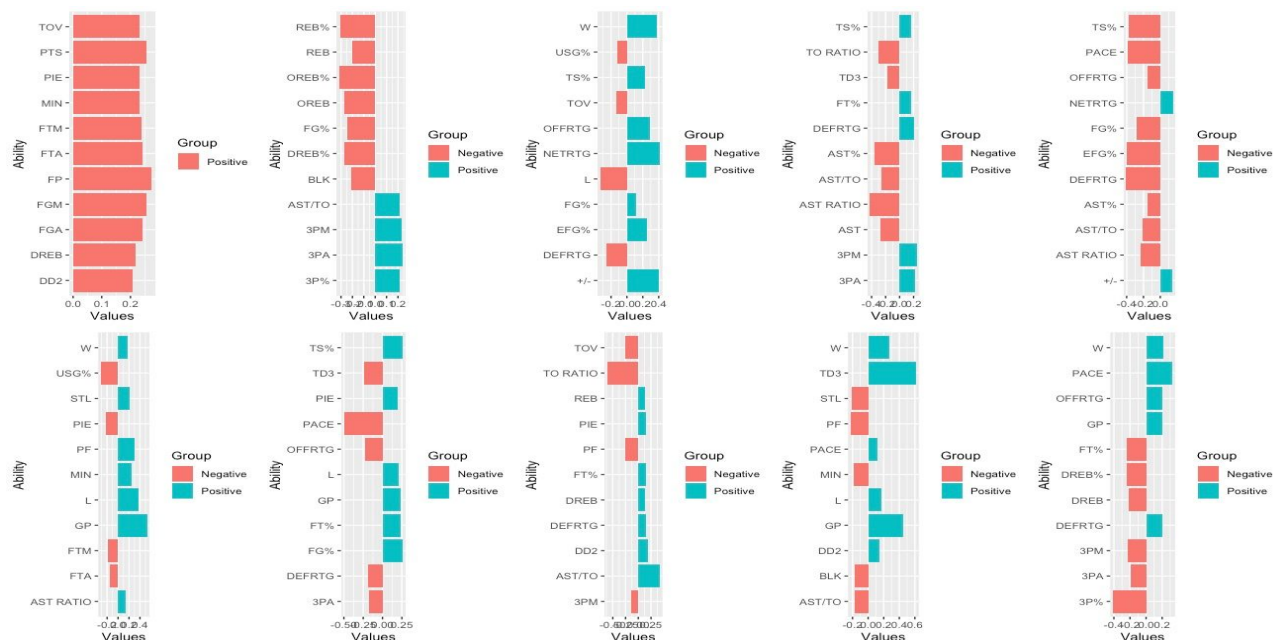
Since optimization for K-means is difficult to be solved precisely, we attempted to plot components pairwise to look for a pattern, but could not find an obvious pattern.

Based on our understanding of basketball, we know it is common to put players into 5 groups, of which each represents one position, but we are interested in further grouping of players for even more distinguishable characteristics. Therefore, we started with k = 6 and increased k incrementally to examine each grouping. We found the grouping most appropriate at k=11, at which each group gives a strong impression of a particular style of players. When k>11, the difference between groups becomes too little and has become difficult to explain the difference. As a result, K-means with k=11 is used in this analysis.

3. Analyze Results

3.1 Classification of the playing attributes

We used PCA to obtain top 10 components which can explain 88% of the total variance of the data and can be used as 10 basketball skill attributes. These 10 basketball skill attributes can help us predict the common characteristics of each player group we obtain later.



(The graph is a summary plot of the major properties of the top 10 components)

- Component 1: Scoring
Since most of the major positive properties in component 1 are related to scoring and field goals, such as points(PTS), field goals made(FGM), field goals attempted(FGA), field goals percentage(FP). The first component - the “scoring” trait is the most effective trait to assess a player’s ability.
- Component 2: 3-pointer Shooter+Passing Efficiency
Most of the major positive properties in component 2 are related to 3-pointer shooting and assists, such as assist to turnover ratio(AST/TO), 3-point field goals made(3PM), 3-point field goals attempted(3PA), 3-point field goals percentage(3P%). The second component is effective in predicting the assist efficiency and 3-pointer shooting ability of a player.
- Component 3: Dominance on the court
Most of the major positive properties in component 3 are related to efficiency and the number of wins, such as the number of wins(W), plus-minus value which counts the point difference when a player is on the floor(+/-), effective field goal percentage(EFG%), Net Rating(NETRTG). The third component is effective in predicting the scoring efficiency and on-court influence of a player.
- Component 4: 3D Player (Three-pointer Shooter + Defensive Player)
Most of the major positive properties in component 4 are related to 3-pointer shooting and defensive ability, such as defensive rating(DEFRTG), 3-point field goals made(3PM), 3-point field goals attempted(3PA). The fourth component is effective in predicting the defensive ability and 3-pointer shooting ability of a player.
- Component 5: +/- Value
The major positive properties in component 5 are net rating(NETRTG), plus-minus value which counts the point difference when a player is on the floor(+/-). The fifth component is effective in predicting the on-court influence of a player.
- Component 6: “Blue Collar” Player (Aggressive Defensive Player)
Most of the major positive properties in component 6 are related to minutes played and the number of fouls, such as personal fouls(PF), minutes played(MIN), Game Played(GP). The sixth component is effective in predicting the game involvement and the intention of making foul of a player. Players with such traits always follow the order of coach and do whatever defense that is asked to do.
- Component 7: Efficient Player
Most of the major positive properties in component 7 are related to the number of games played and the shooting percentage, such as Game Played(GP), True Shooting Percentage(TS%), Field Goal Percentage(FG%). The seventh component is effective in predicting the game involvement and the shooting percentage of a player. Players with such traits shoots with a good percentage and played with less injury.
- Component 8: Rim Protector(Rebounding+Defense)

Most of the major positive properties in component 8 are related to the rebounding ability and the defensive attributes, such as Rebounding(REB), Defensive Rebounding(DREB), Defensive Rating(DEFRTG). The eighth component is effective in predicting the defensive impact and the rebounding ability of a player. Players with such traits are important in defense and getting rebounds for a team.

- Component 9: Triple Double Machine

Most of the major positive properties in component 9 are related to the number of triple doubles and game played, such as Game Played(GP), the number of triple doubles(TD3). The ninth component is effective in predicting the versatility of a player. Triple double means a player has double-digit number total in three of the five main categories (points, rebounds, assists, steals, blocks) in a game. Players with such traits contribute in many aspects in a great amount for a team(such as LeBron James, Russel Westbrook).

- Component 10: Mid Range Monster

Most of the major negative properties in component 10 are related to the 3-pointer shooting attributes, such as 3-point field goals made(3PM), 3-point field goals attempted(3PA), 3-point field goal percentage(3P%). The tenth component is effective in predicting if a player is reluctant to shoot 3-pointers. Players with such traits usually prefer to shoot mid-range 2-pointer instead of 3-pointers.

3.2 Player Role Categories

We obtained 11 clusters by using K-means, and we named the 11 clusters by observing the mean player attribute values of these 11 groups.

- Group 1-“Role Players” (3-point shooter/Defensive specialist)
Players are good at executing one specific task, such as defense
Example: Andre Iguodala
- Group 2-Outstanding Wings/Big Men
Sub-elite players who have dominant height
Example: Paul Millsap
- Group 3-3&D Player
Players who are good at both shooting 3-pointers and defense
Example: Harrison Barnes
- Group 4-Scoring Guards
Guards who are good at scoring
Example: Eric Bledsoe
- Group 5-Elite Big Men
Players who have dominant height and good on the court in all aspects
Example: Anthony Davis
- Group 6-Non-scoring players (“Trash Players”)
Players who are not really relatively good at any aspect
Example: Mirza Teletovic
- Group 7-Defensive Big Men

Players who have dominant height and are also good at defense

Example: Myles Turner

- Group 8-Ball handler on the bench

Bench players who are valued for controlling the ball and flow of the game

Example: Dennis Schroder

- Group 9-Big Men who can't shoot 3-pointers

Players with dominant height but is bad at shooting 3-pointers

Example: Andrew Bogut

- Group 10-Elite guards

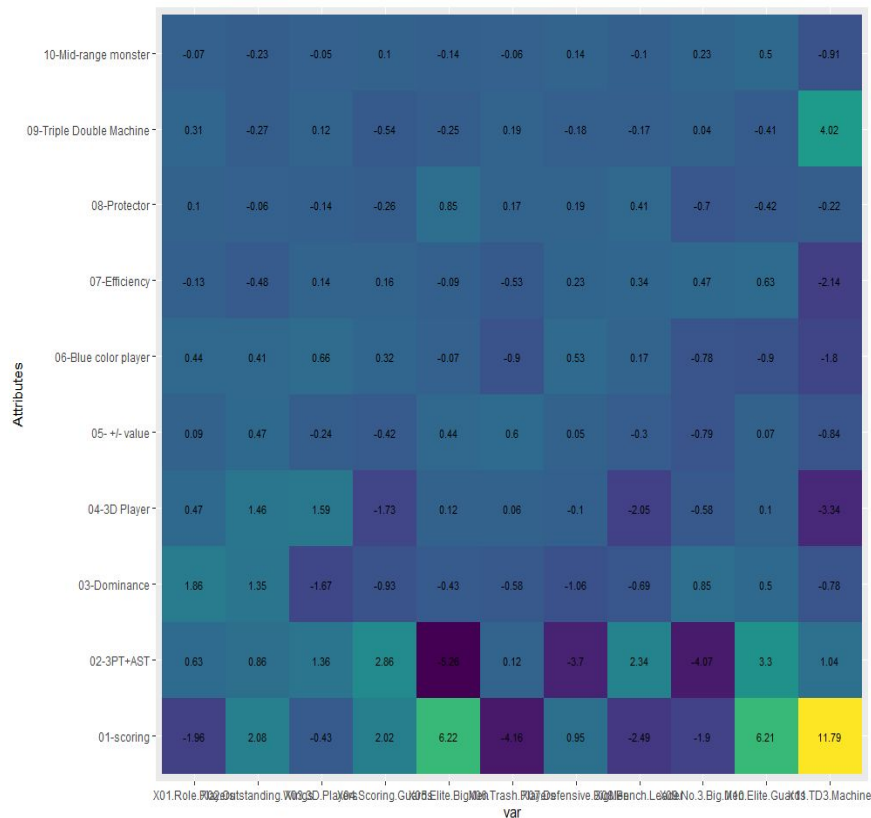
Guards who are good in all aspects in the league

Example: Stephen Curry

- Group 11-"Triple Double" machine

Overall superstar players who are dominant in all aspects in the league

Example: Lebron James



(The Graph of Player Roles Variety Versus Player Attributes)

Players in Group 5, 10, 11 are considered all-star players or players whose ability is close to all-star level.

3.3 Summary of Successful Team Lineups

Champion team Summary: (5 Teams in Total)

Season	1	2	3	4	5	6	7	8	9	10	11
2014-2015	2	2	0	0	0	1	0	1	4	2	0
2015-2016	3	1	0	0	1	2	1	2	1	2	0
2016-2017	4	1	1	1	0	0	0	0	5	2	0
2017-2018	3	1	0	1	0	1	0	0	6	2	0
2018-2019	2	3	1	1	1	0	0	0	0	2	0
Player/Team	2.8	1.6	0.4	0.6	0.4	0.8	0.2	0.6	3.2	2	0

Top 4 Team Summary: (20 Teams in Total)

Season	1	2	3	4	5	6	7	8	9	10	11
2014-2015	8	10	1	0	1	7	4	5	7	6	0
2015-2016	16	4	0	0	3	8	2	4	5	6	3
2016-2017	17	6	2	3	1	5	1	2	8	5	1
2017-2018	11	9	3	2	2	4	2	1	11	4	2
2018-2019	16	9	2	2	3	0	0	2	4	7	1
Player/Team	3.4	1.9	0.4	0.35	0.5	1.2	0.45	0.7	1.75	1.4	0.35

Top 16 Team Summary: (80 Teams in Total)

Season	1	2	3	4	5	6	7	8	9	10	11
2014-2015	31	26	8	12	10	23	21	20	20	16	0
2015-2016	50	23	5	14	9	25	17	13	13	15	2
2016-2017	43	21	8	8	8	22	17	19	22	21	3
2017-2018	39	29	8	9	12	21	10	9	24	17	5
2018-2019	49	33	6	12	11	14	6	11	23	18	6

Player/Team	2.65	1.65	0.44	0.69	0.63	1.31	0.89	0.9	1.3	1.09	0.2
-------------	------	------	------	------	------	------	------	-----	-----	------	-----

There are some trends from the tables above. Having overall superstar players is not needed for a team to gain championship. Role players, big men who cannot shoot 3-pointers and elite guards are much needed for teams to get a space in playoffs. Interestingly, outstanding wings and trash players are abundant in playoffs team, but these do not take a team to championship. On the other hand, defensive big men and ball handler on the bench are ample in top 16 teams, but do not take a team into the quarterfinals.

4.Conclusion

By so far, we have enough information to construct our champion team patterns. Let's review again how we reach this step, we first start with a PCA analysis to reduce our attributes dimension, and finally we reduce the attributes dimensions to 10 components. They are scoring, 3-pointer shooter, Dominance, 3D Player, +/- Value, "Blue Collar" player, Efficient Player, Rim protector, Triple Double Machine and Mid-Range Monster, and these 10 attributes are the most informative ones in the total (42). Based on these 10 attributes, we classified 11 types players by using clustering (K-means), and these 11 categories helps us to draw out a pattern of a final champion team. Through comparing the players' distribution of diverse teams and ranking. We conclude that a champion team may need at least two all-stars players (category 5&10&11), three 3D role players (category 1&3) and at least 1 Elite guard (category 10). For category 2&4, they are optional for a champion team. For category 6&7&9, we suggest our champion team reduce these players if it is possible. Finally, we hope our research will provide some help and insights for some NBA teams and inspire them to build a competitive and efficient playoff team.

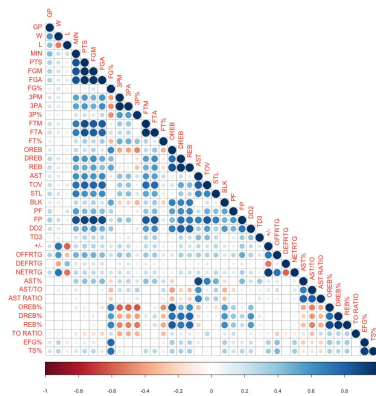
Appendix

Data source: <https://stats.nba.com/players/traditional/?sort=PTS&dir=-1>

NBA Stats Glossary: <https://stats.nba.com/help/glossary/>.

For each variable, here is the simple explanation (including some additional plots):

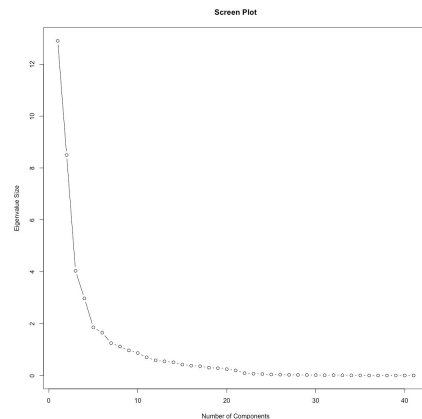
GP – Games Played	3PM – 3 Point Field Goals Made	DREB – Defensive Rebounds	FP – Fantasy Points
W – Wins	3PA – 3 Point Field Goals Attempted	REB – Rebounds	DD2 – Double doubles
L – Losses	3P% – 3 Point Field Goals Percentage	AST – Assists	TD3 – Triple doubles
MIN – Minutes Played	FTM – Free Throws Made	TOV – Turnovers	PTS – Points
FGM – Field Goals Made	FTA – Free Throws Attempted	STL – Steals	+/- – Plus Minus
FGA – Field Goals Attempted	FT% – Free Throw Percentage	BLK – Blocks	
FG% – Field Goal Percentage	OREB – Offensive Rebounds	PF – Personal Fouls	
GP – Games Played	DefRtg – Defensive Rating	OREB% – Offensive Rebound Percentage	TS% – True Shooting Percentage
W – Wins	NetRtg – Net Rating	DREB% – Defensive Rebound Percentage	USG% – Usage Percentage
L – Losses	AST% – Assist Percentage	REB% – Rebound Percentage	PACE – Pace
MIN – Minutes Played	AST/TO – Assist to Turnover Ratio	TO Ratio – Turnover Ratio	PIE – Player Impact Estimate
OffRtg – Offensive Rating	AST Ratio – Assist Ratio	eFG% – Effective Field Goal Percentage	



This is the correlation plot that show the pairwise correlations for all 42 variables in our data. This plot shows the general appearance of the relationship between variables. As we understand, we expect to see some variables have high correlation and some of them do not. This correlation plot do the same as we expected.

Importance of components:

	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5	Comp.6	Comp.7	Comp.8
Standard deviation	3.591699	2.9151151	2.00843210	1.72351393	1.36193265	1.2837984	1.11504822	1.0558427
Proportion of Variance	0.314824	0.2073859	0.09844239	0.07249323	0.04526673	0.0402218	0.03034276	0.0272061
Cumulative Proportion	0.314824	0.5222099	0.62065225	0.69314548	0.73841220	0.7786340	0.80897677	0.8361829
	Comp.9	Comp.10	Comp.11	Comp.12	Comp.13	Comp.14	Comp.15	Comp.16
Standard deviation	0.98146588	0.93020875	0.83632474	0.76315611	0.7386132	0.71235354	0.64632334	0.615076207
Proportion of Variance	0.02350814	0.02111683	0.01706938	0.01421329	0.0133138	0.01238394	0.01019454	0.009232636
Cumulative Proportion	0.85969101	0.88080783	0.89787721	0.91209050	0.9254043	0.93778824	0.94798278	0.957215416



The y-axis is the value of eigenvalue and the x-axis is the list of components that we get by using PCA. Larger eigenvalues represent this component can illustrate more information from original data. By looking at the summary table of the PCA, we can see that 12 components can explain more than 91% of information. By the plot, there is an elbow point at 6. These are all considered possible number of components to choose. But we finally decide to use ten components because it is the maximum number of components that can be

explained well by our knowledge and understanding in basketball.

Source code by using R:

```
# load data
library(readxl)
set.seed(1)
data <- read_excel("Downloads/data.xlsx",
  col_types = c("text", "text", "numeric",
    "numeric", "numeric", "numeric",
    "numeric", "numeric", "numeric",
    "numeric", "numeric", "numeric",
    "numeric", "numeric", "numeric",
    "numeric", "numeric", "numeric",
    "numeric", "numeric", "numeric",
    "numeric", "numeric", "numeric",
    "numeric", "numeric", "numeric",
    "numeric", "numeric", "numeric",
    "numeric", "numeric", "numeric",
    "numeric", "numeric", "text"))

# Data Cleaning
for(i in 1:nrow(data)){
  if(data$MIN[i]<5){ #minutes played per game lower than 5 minutes
    data[i,] = NA
  }
}
data = na.omit(data) #Delete players
for(i in 1:nrow(data)){
  if(data$GP[i]<40){ #played less than 40 games
    data[i,] = NA
  }
}
data = na.omit(data) #Delete players

#Describe data
#Highly correlated variables
library(corrplot)
par(mfrow=c(1,1))
corrplot(cor(as.matrix(data[,4:41]), use = "complete.obs"), method="circle", type="lower")
```

#boxplot, histogram...

```
plot(data$OFFRTG,data$DEFRTG)
abline(v=median(data$OFFRTG))
abline(h=median(data$DEFRTG))
```

#Standardize data

#Normalization is important in PCA since it is a variance maximizing exercise. It projects your
#original data onto directions which maximize the variance. The first plot below shows the
amount

#of total variance explained in the different principal components wher we have not normalized
the

#data.

```
data[,4:44] = as.data.frame(scale(data[,4:44]))
```

#PCA

```
data.pc <- princomp(as.data.frame(data[,4:44])) #also standardize the data
summary(data.pc)
```

#We choose 12 components to be the result of PCA becasue this is the lowest number of
components that

#contain more than 90 percent of the information of the original data.

```
par(mfrow = c(1,1))
```

#Elbow Plot decide the value of PC's

```
plot(1:(length(data.pc$sdev)), (data.pc$sdev)^2, type='b',
     main="Screen Plot", xlab="Number of Components", ylab="Eigenvalue Size")
```

```
pcdata = as.data.frame(data.pc$scores[,1:10])
```

```
pcdata$PLAYER = data$PLAYER
```

```
pcdata$TEAM = data$TEAM
```

```
pcdata$YEAR = data$YEAR
```

```
loading = data.pc$loadings[,1:10] #see the loadings of each components that we choose
```

#Plot the histograms of each components and show the main factors in it

```
library(ggplot2)
```

```
names = matrix('a',nrow=10,ncol=41)
```

```
values = matrix(0,nrow=10,ncol=41)
```

```
for(i in 1:10){
```

```
  values[i,] = as.numeric(loading[,i][order(abs(loading[,i]))])
```

```
  names[i,] = rownames(as.data.frame(loading[,i][order(abs(loading[,i]))]))
```

```
}
```

```
library(ggpubr)
```

```

ggarrange(ggplot(data.frame(Ability = names[1,31:41], Values = values[1,31:41],
Group=ifelse(values[1,31:41]>0,'Positive','Negative')),aes(x=Ability,y=Values,fill=Group))+geom
_bar(stat="identity")+coord_flip(),
  ggplot(data.frame(Ability = names[2,31:41], Values = values[2,31:41],
Group=ifelse(values[2,31:41]>0,'Positive','Negative')),aes(x=Ability,y=Values,fill=Group))+geom
_bar(stat="identity")+coord_flip(),
  ggplot(data.frame(Ability = names[3,31:41], Values = values[3,31:41],
Group=ifelse(values[3,31:41]>0,'Positive','Negative')),aes(x=Ability,y=Values,fill=Group))+geom
_bar(stat="identity")+coord_flip(),
  ggplot(data.frame(Ability = names[4,31:41], Values = values[4,31:41],
Group=ifelse(values[4,31:41]>0,'Positive','Negative')),aes(x=Ability,y=Values,fill=Group))+geom
_bar(stat="identity")+coord_flip(),
  ggplot(data.frame(Ability = names[5,31:41], Values = values[5,31:41],
Group=ifelse(values[5,31:41]>0,'Positive','Negative')),aes(x=Ability,y=Values,fill=Group))+geom
_bar(stat="identity")+coord_flip(),
  ggplot(data.frame(Ability = names[6,31:41], Values = values[6,31:41],
Group=ifelse(values[6,31:41]>0,'Positive','Negative')),aes(x=Ability,y=Values,fill=Group))+geom
_bar(stat="identity")+coord_flip(),
  ggplot(data.frame(Ability = names[7,31:41], Values = values[7,31:41],
Group=ifelse(values[7,31:41]>0,'Positive','Negative')),aes(x=Ability,y=Values,fill=Group))+geom
_bar(stat="identity")+coord_flip(),
  ggplot(data.frame(Ability = names[8,31:41], Values = values[8,31:41],
Group=ifelse(values[8,31:41]>0,'Positive','Negative')),aes(x=Ability,y=Values,fill=Group))+geom
_bar(stat="identity")+coord_flip(),
  ggplot(data.frame(Ability = names[9,31:41], Values = values[9,31:41],
Group=ifelse(values[9,31:41]>0,'Positive','Negative')),aes(x=Ability,y=Values,fill=Group))+geom
_bar(stat="identity")+coord_flip(),
  ggplot(data.frame(Ability = names[10,31:41], Values = values[10,31:41],
Group=ifelse(values[10,31:41]>0,'Positive','Negative')),aes(x=Ability,y=Values,fill=Group))+geo
m_bar(stat="identity")+coord_flip(),
  ncol = 5, nrow = 2)

```

```

par(mfrow=c(1,1))
plot(x=data.pc$scores[,2],y=data.pc$scores[,3])

```

```

kclustering = kmeans(pcddata[,1:10], centers = 11, nstart = 30)
pcdata$cluster = kclustering$cluster
data$cluster = kclustering$cluster
#Put players in to groups
group1=group2=group3=group4=group5=group6=group7=group8=group9=group10=group11=p
cdata
for(i in 1:nrow(pcddata)){

```

```
if(pcddata$cluster[i]!=1){
  group1[i,] = NA
}
if(pcddata$cluster[i]!=2){
  group2[i,] = NA
}
if(pcddata$cluster[i]!=3){
  group3[i,] = NA
}
if(pcddata$cluster[i]!=4){
  group4[i,] = NA
}
if(pcddata$cluster[i]!=5){
  group5[i,] = NA
}
if(pcddata$cluster[i]!=6){
  group6[i,] = NA
}
if(pcddata$cluster[i]!=7){
  group7[i,] = NA
}
if(pcddata$cluster[i]!=8){
  group8[i,] = NA
}
if(pcddata$cluster[i]!=9){
  group9[i,] = NA
}
if(pcddata$cluster[i]!=10){
  group10[i,] = NA
}
if(pcddata$cluster[i]!=11){
  group11[i,] = NA
}
}
group1 = na.omit(group1)
group2 = na.omit(group2)
group3 = na.omit(group3)
group4 = na.omit(group4)
group5 = na.omit(group5)
group6 = na.omit(group6)
group7 = na.omit(group7)
group8 = na.omit(group8)
group9 = na.omit(group9)
```

```
group10 = na.omit(group10)
group11 = na.omit(group11)
```

#Characteristics for each group

```
performance = data.frame('group1'=apply(group1[, 1:10], 2, mean),
                          'group2'=apply(group2[, 1:10], 2, mean),
                          'group3'=apply(group3[, 1:10], 2, mean),
                          'group4'=apply(group4[, 1:10], 2, mean),
                          'group5'=apply(group5[, 1:10], 2, mean),
                          'group6'=apply(group6[, 1:10], 2, mean),
                          'group7'=apply(group7[, 1:10], 2, mean),
                          'group8'=apply(group8[, 1:10], 2, mean),
                          'group9'=apply(group9[, 1:10], 2, mean),
                          'group10'=apply(group10[, 1:10], 2, mean),
                          'group11'=apply(group11[, 1:10], 2, mean))
rownames(performance) = c("","","","","","","","")
```

```
library(tidyverse)
library(viridis)
performance%>%
  rownames_to_column(var = "make") %>%
  gather(var, val, -make) %>%
  ggplot(aes(var, make)) +
  geom_tile(aes(fill = val)) +
  geom_text(aes(label = round(val, 2)), size = 3) +
  coord_fixed() +
  scale_fill_viridis() +
  guides(fill = FALSE)
```

```
sixteen89 =
data.frame(PPLAYER=data$PLAYER,TEAM=data$TEAM,YEAR=data$YEAR,cluster=data$cluster)
sixteen78 =
data.frame(PPLAYER=data$PLAYER,TEAM=data$TEAM,YEAR=data$YEAR,cluster=data$cluster)
sixteen67 =
data.frame(PPLAYER=data$PLAYER,TEAM=data$TEAM,YEAR=data$YEAR,cluster=data$cluster)
```

```

sixteen56 =
data.frame(PPLAYER=data$PLAYER,TEAM=data$TEAM,YEAR=data$YEAR,cluster=data$cluster)
sixteen45 =
data.frame(PPLAYER=data$PLAYER,TEAM=data$TEAM,YEAR=data$YEAR,cluster=data$cluster)
for(i in 1:nrow(sixteen89)){
  if(sixteen89$YEAR[i] == "2018-2019"){
    if(sum(data$TEAM[i]==c('MIL', 'TOR', 'PHI', 'BOS', 'IND', 'BKN', 'ORL', 'DET',
      'GSW', 'DEN', 'POR', 'HOU', 'UTA', 'OKC', 'SAS', 'LAC'))==0){
      sixteen89[i,] = NA
    }
  }
  else{
    sixteen89[i,] = NA
  }
}
sixteen89 = na.omit(sixteen89)
for(i in 1:nrow(sixteen78)){
  if(sixteen78$YEAR[i] == "2017-2018"){
    if(sum(data$TEAM[i]==c('TOR', 'BOS', 'PHI', 'CLE', 'IND', 'MIA', 'MIL', 'WAS',
      'HOU', 'GSW', 'POR', 'OKC', 'UTA', 'NOP', 'SAS', 'MIN'))==0){
      sixteen78[i,] = NA
    }
  }
  else{
    sixteen78[i,] = NA
  }
}
sixteen78 = na.omit(sixteen78)
for(i in 1:nrow(sixteen67)){
  if(sixteen67$YEAR[i] == "2016-2017"){
    if(sum(data$TEAM[i]==c('BOS', 'CLE', 'TOR', 'WAS', 'ATL', 'MIL', 'IND', 'CHI',
      'GSW', 'SAS', 'HOU', 'LAC', 'UTA', 'OKC', 'MEM', 'POR'))==0){
      sixteen67[i,] = NA
    }
  }
  else{
    sixteen67[i,] = NA
  }
}
sixteen67 = na.omit(sixteen67)
for(i in 1:nrow(sixteen56)){

```

```

if(sixteen56$YEAR[i] == "2015-2016"){
  if(sum(data$TEAM[i]==c('CLE', 'TOR', 'MIA', 'ATL', 'BOS', 'CHA', 'IND', 'DET',
    'GSW', 'SAS', 'OKC', 'LAC', 'POR', 'DAL', 'MEM', 'HOU'))==0){
    sixteen56[i,] = NA
  }
}
else{
  sixteen56[i,] = NA
}
}
sixteen56 = na.omit(sixteen56)
for(i in 1:nrow(sixteen45)){
  if(sixteen45$YEAR[i] == "2014-2015"){
    if(sum(data$TEAM[i]==c('ATL', 'CLE', 'CHI', 'TOR', 'WAS', 'MIL', 'BOS', 'NYK',
      'GSW', 'HOU', 'LAC', 'POR', 'MEM', 'SAS', 'DAL', 'NOP'))==0){
      sixteen45[i,] = NA
    }
  }
  else{
    sixteen45[i,] = NA
  }
}
sixteen45 = na.omit(sixteen45)
table(sixteen89$cluster)
table(sixteen78$cluster)
table(sixteen67$cluster)
table(sixteen56$cluster)
table(sixteen45$cluster)

```

```

playoff89 =
data.frame(PPLAYER=PLAYER,TEAM=TEAM,YEAR=YEAR,cluster=cluster)
playoff78 =
data.frame(PPLAYER=PLAYER,TEAM=TEAM,YEAR=YEAR,cluster=cluster)
playoff67 =
data.frame(PPLAYER=PLAYER,TEAM=TEAM,YEAR=YEAR,cluster=cluster)
playoff56 =
data.frame(PPLAYER=PLAYER,TEAM=TEAM,YEAR=YEAR,cluster=cluster)

```



```

playoff45 =
data.frame(PPLAYER=data$PPLAYER,TEAM=data$TEAM,YEAR=data$YEAR,cluster=data$cluster)
for(i in 1:nrow(playoff89)){
  if(playoff89$YEAR[i] == "2018-2019"){
    if(sum(data$TEAM[i]==c('GSW','TOR','MIL','POR'))==0){
      playoff89[i,] = NA
    }
  }
  else{
    playoff89[i,] = NA
  }
}
playoff89 = na.omit(playoff89)
for(i in 1:nrow(playoff78)){
  if(playoff78$YEAR[i] == "2017-2018"){
    if(sum(data$TEAM[i]==c('GSW','BOS','HOU','CLE'))==0){
      playoff78[i,] = NA
    }
  }
  else{
    playoff78[i,] = NA
  }
}
playoff78 = na.omit(playoff78)
for(i in 1:nrow(playoff67)){
  if(playoff67$YEAR[i] == "2016-2017"){
    if(sum(data$TEAM[i]==c('GSW','BOS','CLE','SAS'))==0){
      playoff67[i,] = NA
    }
  }
  else{
    playoff67[i,] = NA
  }
}
playoff67 = na.omit(playoff67)
for(i in 1:nrow(playoff56)){
  if(playoff56$YEAR[i] == "2015-2016"){
    if(sum(data$TEAM[i]==c('GSW','OKC','TOR','CLE'))==0){
      playoff56[i,] = NA
    }
  }
  else{

```

```

    playoff56[i,] = NA
  }
}
playoff56 = na.omit(playoff56)
for(i in 1:nrow(playoff45)){
  if(playoff45$YEAR[i] == "2014-2015"){
    if(sum(data$TEAM[i]==c('GSW','HOU','CLE','ATL'))==0){
      playoff45[i,] = NA
    }
  }
  else{
    playoff45[i,] = NA
  }
}
playoff45 = na.omit(playoff45)
table(playoff89$cluster)
table(playoff78$cluster)
table(playoff67$cluster)
table(playoff56$cluster)
table(playoff45$cluster)

champ89 =
data.frame(PAYER=data$PLAYER,TEAM=data$TEAM,YEAR=data$YEAR,cluster=data$cluster)
champ78 =
data.frame(PAYER=data$PLAYER,TEAM=data$TEAM,YEAR=data$YEAR,cluster=data$cluster)
champ67 =
data.frame(PAYER=data$PLAYER,TEAM=data$TEAM,YEAR=data$YEAR,cluster=data$cluster)
champ56 =
data.frame(PAYER=data$PLAYER,TEAM=data$TEAM,YEAR=data$YEAR,cluster=data$cluster)
champ45 =
data.frame(PAYER=data$PLAYER,TEAM=data$TEAM,YEAR=data$YEAR,cluster=data$cluster)
for(i in 1:nrow(champ89)){
  if(champ89$YEAR[i] == "2018-2019"){
    if(data$TEAM[i]!='TOR'){
      champ89[i,] = NA
    }
  }
  else{

```

```

    champ89[i,] = NA
  }
}
champ89 = na.omit(champ89)
for(i in 1:nrow(champ78)){
  if(champ78$YEAR[i] == "2017-2018"){
    if(data$TEAM[i]!='GSW'){
      champ78[i,] = NA
    }
  }
  else{
    champ78[i,] = NA
  }
}
champ78 = na.omit(champ78)
for(i in 1:nrow(champ67)){
  if(champ67$YEAR[i] == "2016-2017"){
    if(data$TEAM[i]!='GSW'){
      champ67[i,] = NA
    }
  }
  else{
    champ67[i,] = NA
  }
}
champ67 = na.omit(champ67)
for(i in 1:nrow(champ56)){
  if(champ56$YEAR[i] == "2015-2016"){
    if(data$TEAM[i]!='CLE'){
      champ56[i,] = NA
    }
  }
  else{
    champ56[i,] = NA
  }
}
champ56 = na.omit(champ56)
for(i in 1:nrow(champ45)){
  if(champ45$YEAR[i] == "2014-2015"){
    if(data$TEAM[i]!='GSW'){
      champ45[i,] = NA
    }
  }
}

```

```
else{  
  champ45[i,] = NA  
}  
}  
champ45 = na.omit(champ45)
```

```
table(champ89$cluster)  
table(champ78$cluster)  
table(champ67$cluster)  
table(champ56$cluster)  
table(champ45$cluster)
```