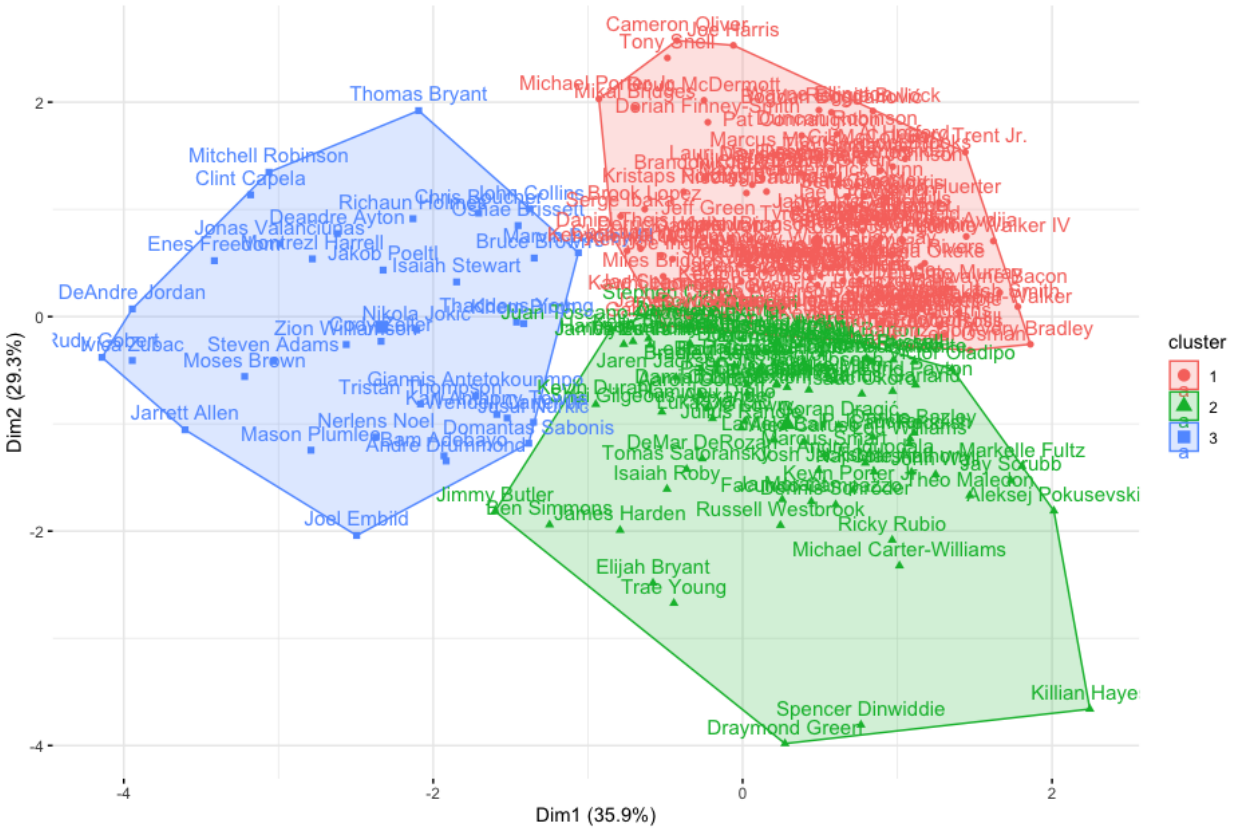
Clustering NBA players based on perfomance

Alfonso Marino

2024-04-08



Introduction

The NBA has a rich history spanning decades, with players of various skills, styles, and physical attributes gracing the courts. In this project, we leverage data spanning from 1950 to 2021 to gain deeper insights into player performance. Unlike traditional approaches that categorize players based solely on their positions (such as point guards, shooting guards, etc.), we employ cluster analysis to group players based on their overall performance metrics. This allows us to uncover hidden similarities and differences among players that may not be evident when considering positions alone.

Data Preparation

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.0     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
library(measurements)
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
nba = read.csv("/Users/alfonsomarino/Desktop/Progetti/nba/seasons_stats.csv")
player_stat = read.csv("/Users/alfonsomarino/Desktop/Progetti/nba/player_data.csv")
```

```
nba %>%
  glimpse()
```

```
## Rows: 28,057
## Columns: 51
## $ X      <int> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 1~
## $ Year   <int> 1950, 1950, 1950, 1950, 1950, 1950, 1950, 1950, 1950, 1950, 195~
## $ Player <chr> "Curly Armstrong", "Cliff Barker", "Leo Barnhorst", "Ed Bartels~
## $ Pos    <chr> "G-F", "SG", "SF", "F", "F", "F", "G", "G-F", "F-C", "F-C", "F-~
## $ Age    <int> 31, 29, 25, 24, 24, 24, 22, 23, 28, 28, 28, 25, 22, 22, 24, 27,~
## $ Tm     <chr> "FTW", "INO", "CHS", "TOT", "DNN", "NYK", "INO", "TRI", "TOT", ~
## $ G      <int> 63, 49, 67, 15, 13, 2, 60, 3, 65, 36, 29, 57, 60, 59, 62, 61, 4~
## $ GS     <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ MP     <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ FG     <int> 144, 102, 174, 22, 21, 1, 340, 5, 226, 125, 101, 80, 88, 204, 2~
## $ FGA    <int> 516, 274, 499, 86, 82, 4, 936, 16, 813, 435, 378, 248, 305, 600~
## $ FG.    <dbl> 0.279, 0.372, 0.349, 0.256, 0.256, 0.250, 0.363, 0.313, 0.278, ~
## $ X3P    <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ X3PA   <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ X3P.   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ X2P    <int> 144, 102, 174, 22, 21, 1, 340, 5, 226, 125, 101, 80, 88, 204, 2~
## $ X2PA   <int> 516, 274, 499, 86, 82, 4, 936, 16, 813, 435, 378, 248, 305, 600~
## $ X2P.   <dbl> 0.279, 0.372, 0.349, 0.256, 0.256, 0.250, 0.363, 0.313, 0.278, ~
## $ eFG.   <dbl> 0.279, 0.372, 0.349, 0.256, 0.256, 0.250, 0.363, 0.313, 0.278, ~
## $ FT     <int> 170, 75, 90, 19, 17, 2, 215, 0, 209, 132, 77, 82, 78, 204, 240,~
## $ FTA    <int> 241, 106, 129, 34, 31, 3, 282, 5, 321, 209, 112, 131, 117, 267,~
```

```
## $ FT.    <dbl> 0.705, 0.708, 0.698, 0.559, 0.548, 0.667, 0.762, 0.000, 0.651, ~
## $ ORB    <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ DRB    <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ TRB    <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ AST    <int> 176, 109, 140, 20, 20, 0, 233, 2, 163, 75, 88, 46, 40, 95, 137,~
## $ STL    <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ BLK    <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ TOV    <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ PF     <int> 217, 99, 192, 29, 27, 2, 132, 6, 273, 140, 133, 97, 111, 203, 2~
## $ PTS    <int> 458, 279, 438, 63, 59, 4, 895, 10, 661, 382, 279, 242, 254, 612~
## $ PER    <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ TS.    <dbl> 0.368, 0.435, 0.394, 0.312, 0.308, 0.376, 0.422, 0.275, 0.346, ~
## $ X3PAr  <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ FTr    <dbl> 0.467, 0.387, 0.259, 0.395, 0.378, 0.750, 0.301, 0.313, 0.395, ~
## $ ORB.   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ DRB.   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ TRB.   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ AST.   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ STL.   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ BLK.   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ TOV.   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ USG.   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ OWS    <dbl> -0.1, 1.6, 0.9, -0.5, -0.5, 0.0, 3.6, -0.1, -2.2, -0.7, -1.5, 0~
## $ DWS    <dbl> 3.6, 0.6, 2.8, -0.1, -0.1, 0.0, 1.2, 0.0, 5.0, 2.2, 2.8, 1.3, 1~
## $ WS     <dbl> 3.5, 2.2, 3.6, -0.6, -0.6, 0.0, 4.8, -0.1, 2.8, 1.5, 1.3, 1.8, ~
## $ WS.48  <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ OBPM   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ DBPM   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ BPM    <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ VORP   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
```

```
player_stat %>%
  glimpse()
```

```
## Rows: 4,979
## Columns: 8
## $ Player     <chr> "Alaa Abdelnaby", "Zaid Abdul-Aziz", "Kareem Abdul-Jabbar*"~
## $ From       <int> 1991, 1969, 1970, 1991, 1998, 1997, 1977, 1957, 1947, 2017,~
## $ To         <int> 1995, 1978, 1989, 2001, 2003, 2008, 1981, 1957, 1948, 2019,~
## $ Pos        <chr> "F-C", "C-F", "C", "G", "F", "F", "F", "G", "F", "G-F", "F"~
## $ Ht         <chr> "6-10", "6-9", "7-2", "6-1", "6-6", "6-9", "6-7", "6-3", "6~
## $ Wt         <int> 240, 235, 225, 162, 223, 225, 220, 180, 195, 200, 225, 185,~
## $ Birth.Date <chr> "June 24 1968", "April 7 1946", "April 16 1947", "March 9 1~
## $ Colleges   <chr> "Duke", "Iowa State", "UCLA", "LSU", "Michigan San Jose Sta~
```

When analyzing the null values present in *player_stat*, note how there are only 5 records corresponding to the weight column, which are removed because their presence historically has been marginal.

```
player_stat %>%
  select(everything()) %>%
  summarise_all(list(~sum(is.na(.))))
```

```
##   Player From To Pos Ht Wt Birth.Date Colleges
## 1      0    0  0   0  0  5          0        0
```

```
player_stat %>%
  filter(is.na(Wt) == T)
```

```
##            Player From   To Pos   Ht Wt       Birth.Date
## 1        Dick Lee 1968 1968   F  6-6 NA
## 2 Murray Mitchell 1950 1950   C  6-6 NA    March 19 1923
## 3      Paul Nolen 1954 1954   C 6-10 NA September 3 1929
## 4      Ray Wertis 1947 1948   G 5-11 NA     July 30 1923
## 5        Bob Wood 1950 1950   G 5-10 NA   October 7 1921
##                        Colleges
## 1                    Washington
## 2 Sam Houston State University
## 3                    Texas Tech
## 4                    St. John's
## 5             Northern Illinois
```

```
player_stat = player_stat %>%
  drop_na(Wt)
```

Any duplicates are removed.

```
player_stat = player_stat %>%
  distinct(Player, .keep_all = T)
```

The columns for weight and height are measured in pounds and feet, respectively; for convenience of use we convert the units to kilograms and centimeters.

```
convertHt <- function(x) {
  heights <- as.character(x)
  heights_split <- strsplit(heights, "-")
  feet <- as.numeric(sapply(heights_split, `[`, 1))
  inches <- as.numeric(sapply(heights_split, `[`, 2))
  heights_cm <- round(conv_unit(feet, "ft", "cm") + conv_unit(inches, "inch", "cm"), 0)

  return(heights_cm)
}

player_stat <- player_stat %>%
  rowwise() %>%
  mutate(Ht = convertHt(Ht), Wt = round(conv_unit(Wt, "lbs", "kg")))

player_stat %>%
  select(c(Wt, Ht)) %>%
head(10)
```

```
## # A tibble: 10 x 2
## # Rowwise:
##       Wt    Ht
##    <dbl> <dbl>
##  1   109   208
##  2   107   206
##  3   102   218
```

```
##  4    73   185
##  5   101   198
##  6   102   206
##  7   100   201
##  8    82   190
##  9    88   190
## 10    91   198
```

Regarding the *nba* dataset, we remove the rows for the year 2022 because it is the same as the year 2021.

```
nba = nba %>%
  filter(Year<2022)
```

Some players' names are flanked by an asterisk to symbolize their presence in the Hall of Fame. For convenience of use, we remove the asterisks and add an appropriate column.

```
nba %>%
  filter(str_detect(nba$Player, ".*\\*$")) %>%
  select(Player) %>%
  head(10)
```

```
##              Player
## 1         Al Cervi*
## 2       Bob Davies*
## 3        Joe Fulks*
## 4   Harry Gallatin*
## 5      Alex Hannum*
## 6      Red Holzman*
## 7  Buddy Jeannette*
## 8      Ed Macauley*
## 9    Slater Martin*
## 10    Dick McGuire*
```

```
nba = nba %>%
  mutate(HallOfFame = if_else(str_detect(Player, ".*\\*$"), "Yes", "No"))

nba$Player = gsub("\\*$", "", nba$Player)
player_stat$Player <- gsub("\\*$", "", player_stat$Player)
```

Per-game statistics and FTr are added.

```
nba = nba %>%
  mutate(MpG = round(MP/G,3), PpG = round(PTS/G,3), ApG = round(AST/G,3),
         RpG = round(TRB/G,3), TOpG = round(TOV/G,3), BpG = round(BLK/G,3),
         SpG = round(STL/G,3),FpG = round(PF/G, 3), .before = 9)

nba = nba %>%
  mutate(FTr = round(FT/FGA,3), .before = 30)
```

Positions are redefined.

```r
nba <- nba %>%
  mutate(Pos = case_when(
    Pos == "PF-C" ~ "PF",
    Pos == "C-F" ~ "C",
    Pos == "SF-SG" ~ "SF",
    Pos == "C-PF" ~ "C",
    Pos == "SG-SF" ~ "SG",
    Pos == "PF-SF" ~ "PF",
    Pos == "SF-PF" ~ "SF",
    Pos == "SG-PG" ~ "SG",
    Pos == "SF-PG" ~ "SF",
    Pos == "C-SF" ~ "C",
    Pos == "PG-SG" ~ "PG",
    Pos == "PG-SF" ~ "PG",
    Pos == "SG-PF" ~ "SG",
    Pos == "SF-C" ~ "SF",
    Pos == "F-C" ~ "PF",
    Pos == "F-G" ~ "SF",
    Pos == "G-F" ~ "SF",
    Pos == "F" ~ "PF",
    Pos == "G" ~ "SG",
    TRUE ~ Pos
  ))

table(nba$Pos)
```

```
##
##    C   PF   PG   SF   SG
## 5351 5786 5194 5372 5649
```

At this point the two datasets can be merged to get a comprehensive overview of the information. In addition to that, we also extract two datasets namely *nba_withTOT* and *nba_performance*. The former excludes partial statistics for those players who changed teams in the middle of the season; while the latter is filtered by minutes played.

```r
player_stat = player_stat %>%
  select(c("Player","Ht", "Wt"))
player_stat = as.data.frame(player_stat)

nba = left_join(nba, player_stat, by = "Player", relationship = "many-to-many")

nba = nba %>%
  select(Year:Tm, HallOfFame:Wt, everything())

nba = nba %>%
  distinct(Player,Year, Tm, Age, .keep_all = T)

nba_withTOT = nba %>%
  group_by(Year, Player) %>%
  mutate(count_tot = sum(Tm == "TOT")) %>%
  filter(count_tot == 0 | (count_tot > 0 & Tm == "TOT")) %>%
  select(-count_tot)
```

```
nba_withTOT = as.data.frame(nba_withTOT)

nba_performance = nba_withTOT %>%
  filter(MpG > mean(MpG, na.rm = T) & Year > 1999)

nba_performance = as.data.frame(nba_performance)
```

The full dataset contains several null values since not all statistics date back to the same historical period, so from time to time we are going to handle them without removing them. Also in the column for colleges, there are also blank spaces, these may be due to lack of information or because non-American players actually did not belong to any college. The NA values related to the Ht, Wt columns are due to the fact that those players are not present in the player_stat dataset.

```
nba_performance %>%
  summarise_all(list(~sum(is.na(.))))
```

```
##   Year Player Pos Age Tm HallOfFame  Ht  Wt X G GS MpG PpG ApG RpG TOpG BpG SpG
## 1    0      0   0   0  0          0 416 416 0 0  0   0   0   0   0    0   0   0
##   FpG MP FG FGA FG. X3P X3PA X3P. X2P X2PA X2P. eFG. FT FTA FT. ORB DRB TRB AST
## 1   0  0  0   0   0   0    0  252   0    0    0    0  0   0   4   0   0   0   0
##   STL BLK TOV PF PTS PER TS. X3PAr FTr ORB. DRB. TRB. AST. STL. BLK. TOV. USG.
## 1   0   0   0  0   0   0   0     0   0    0    0    0    0    0    0    0    0
##   OWS DWS WS WS.48 OBPM DBPM BPM VORP
## 1   0   0  0     0    0    0   0    0
```

```
nba_performance$X3P. = replace(nba_performance$X3P., is.na(nba_performance$X3P.), 0)
nba_performance$FT. = replace(nba_performance$FT., is.na(nba_performance$FT.), 0)

nba_performance = nba_performance %>%
  group_by(Pos) %>%
  mutate(MeanWt = mean(Wt, na.rm = TRUE),
         MeanHt = mean(Ht, na.rm = TRUE)) %>%
  ungroup()

nba_performance$Wt = ifelse(is.na(nba_performance$Wt), nba_performance$MeanWt, nba_performance$Wt)
nba_performance$Ht = ifelse(is.na(nba_performance$Ht), nba_performance$MeanHt, nba_performance$Ht)

nba_performance <- nba_performance %>% select(-c(MeanWt, MeanHt))

nba_performance = as.data.frame(nba_performance)
```

Physique evolution

In recent decades, the evolution of NBA players' physiques has reflected a significant transformation in the game. Different positions on the court have shown distinctive variations in players' physiques over time. This evolution of physique reflects not only changes in the game itself, but also the training strategies and demands of the modern NBA style of play.

```
PosColorCode <- c("C"="red", "PF"="orange",
                  "SF"="yellow" ,"SG"="blue", "PG"="green")
```

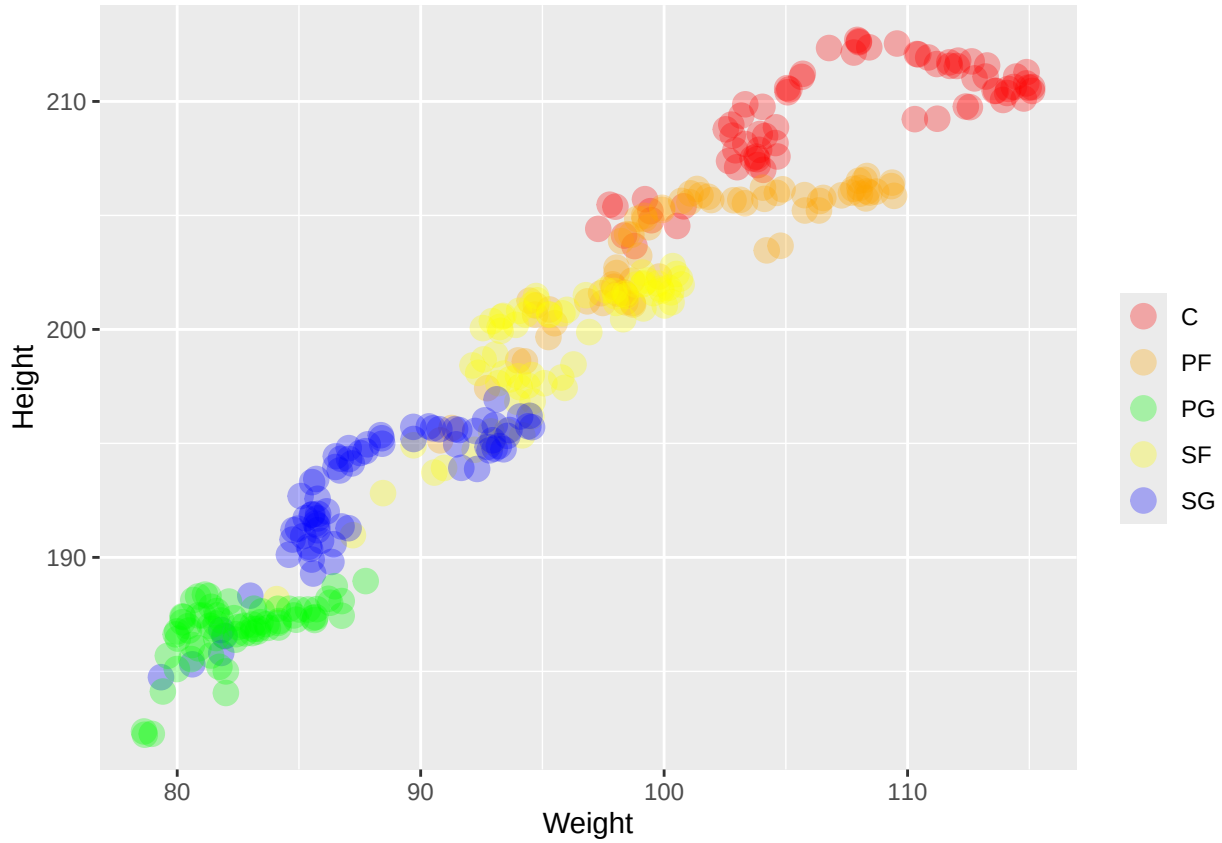
Note how centers and power forward are the strongest athletes physically, while point guards and small forwards are the lightest. Shooting guards have a very wide distribution, so they can be defined, in terms of physical stature, as the ideal NBA player prototype.

```
physique <- nba_withTOT %>%
  group_by(Year, Pos) %>%
  summarise("Height" = mean(Ht, na.rm = T), "Weight"= mean(Wt, na.rm = T))
```

```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

```
#physique

ggplot(physique, aes(x=Weight, y=Height, color=Pos)) +
  geom_point(size=4, alpha = 0.3)+
  scale_color_manual(values = PosColorCode, name = "")
```



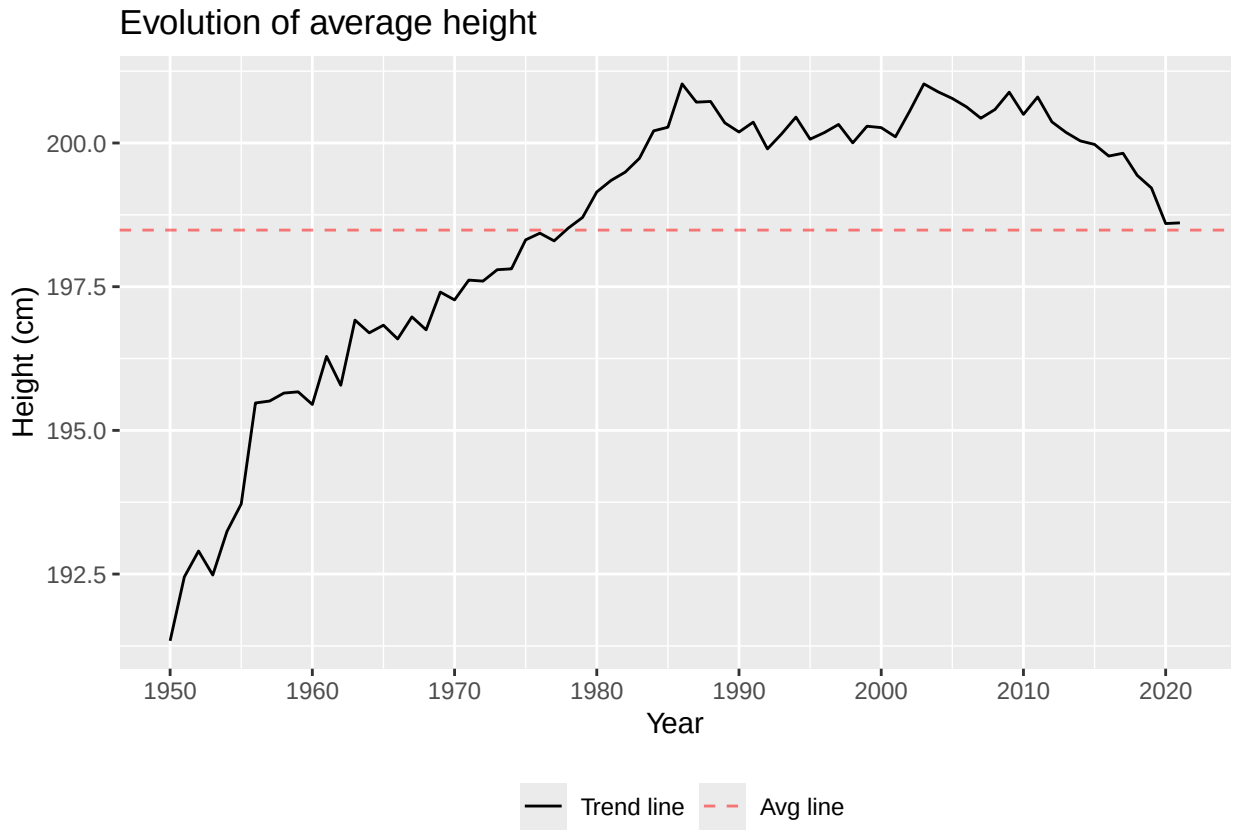
The game of basketball has inevitably changed a great deal since its inception, therefore, even the same conventional positions have adapted to the modern NBA. The most significant change came with the introduction of the three-point shot in the early 1980s, which radically changed the way the game was played. The new rule increased the importance of shooting skills from long distance, giving more responsibility for scoring points to players who, as opposed to pivots, played away from the basket. Traditionally, having a dominant center was fundamental to the structure of a team, but, however, in recent years, many teams have embraced a playing philosophy that emphasizes speed, mobility, and versatility, preferring lineups without a traditional center. This significant evolution in the position concept is called **"small ball"**. This has

resulted in taller players also moving to the perimeter, with shorter players filling more interior roles. These new centers, referred to as **"big stretch"**, in addition to dominating in the painted area, are also able to shoot long range, pushing opponents out of their defensive comfort zone. This has opened up spaces for teammates and created new offensive opportunities.

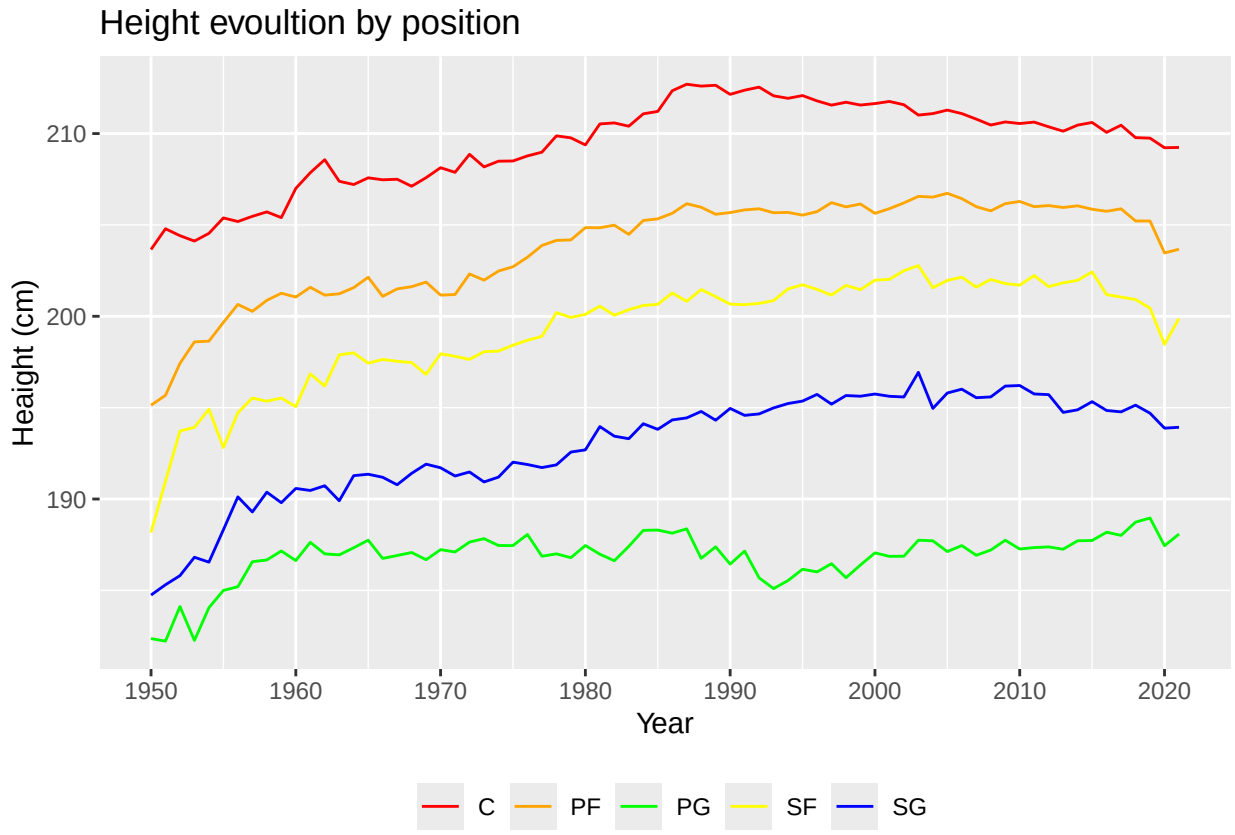
The evolution of the game led to a consequent transformation of the players, both in terms of individual technique and from the point of view of physical structure. It can be seen that, until the early 1980s, growth in terms of stature was stable. From there on there is no clear positive or negative trend, but it is evident how the last decade was the first in history in which NBA players became shorter than the previous decade. This decline is also tangible for individual positions, except for point guards who have reached their maximum height in recent seasons, the other positions have become shorter and shorter.

```
avg <- nba_withTOT %>%
  group_by(Year) %>%
  summarise("Height"=mean(Ht, na.rm = T), "Weight"=mean(Wt, na.rm = T))

ggplot(avg, aes(x=Year, y=Height, linetype = "Trend line")) +
  geom_line()+
  labs(x="Year", y="Height (cm)", title = "Evolution of average height")+
  geom_hline(aes(yintercept = mean(Height), linetype = "Avg line"), col = "red", alpha = 0.5) +
  scale_x_continuous(breaks = seq(1950, 2021, 10)) +
  scale_linetype_manual(name = "", values = c(2, 1), guide = guide_legend(reverse = TRUE))+
  theme(legend.position = "bottom")
```

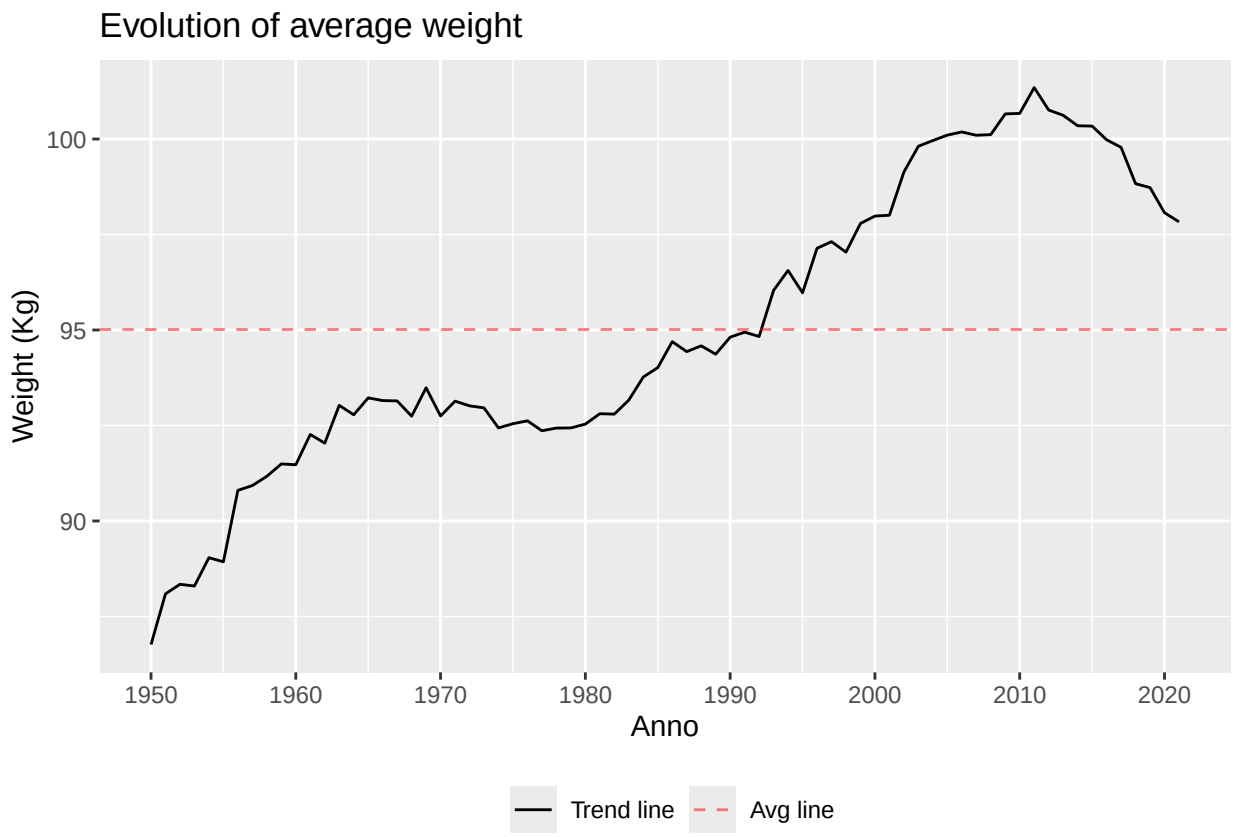


```
physique %>%
  ggplot( aes(x=Year, y=Height, group=Pos, color=Pos)) +
  geom_line()+
  labs(x="Year", y = "Heaight (cm)", title = "Height evoultion by position")+
  theme(legend.position = "bottom")+
  scale_color_manual(values = PosColorCode , name = "")+
  scale_x_continuous(breaks = seq(1950, 2021, 10))
```

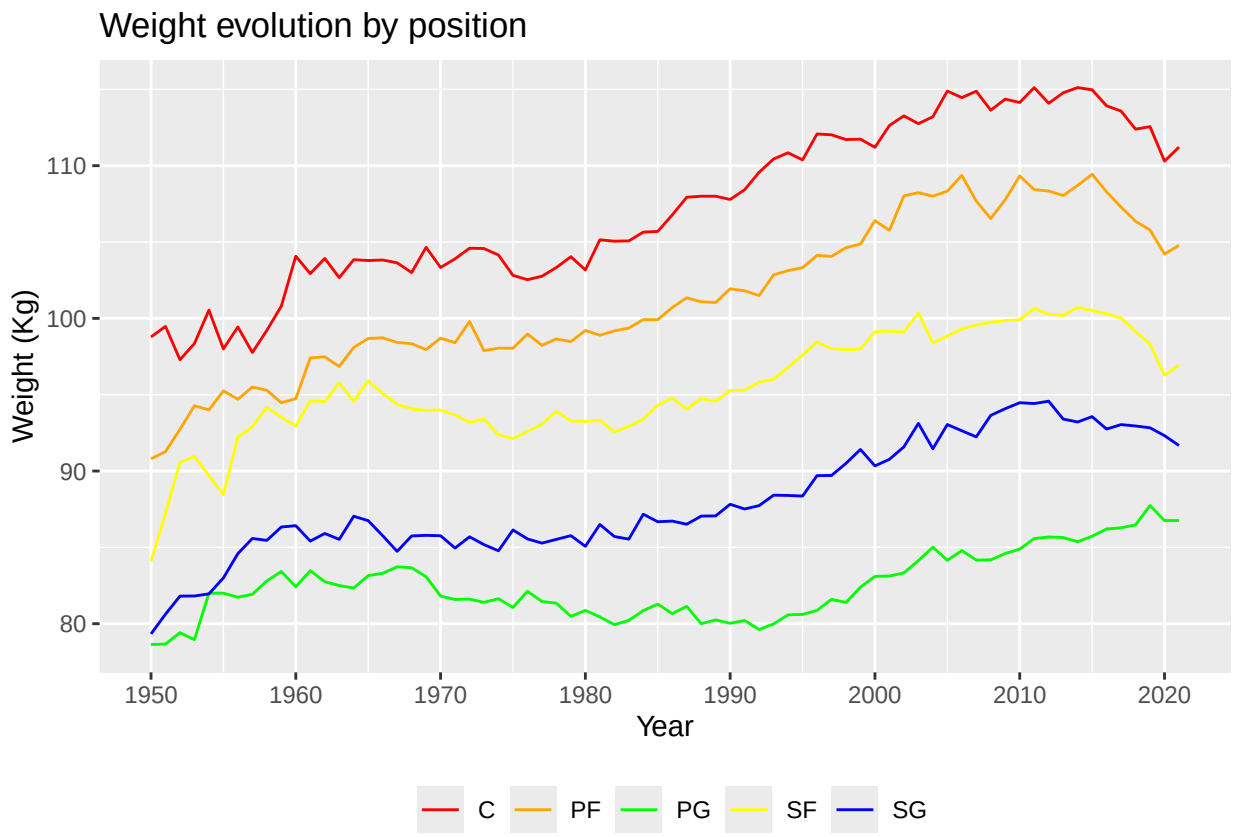


In terms of weight, its average grew steadily until the 1970s, peaked in the 2010/11 season, and has been declining ever since, in fact the players are found to be among the lightest in the 21st century. It can be seen that since 2010 height and weight have undertaken a decreasing trend. This phenomenon can be attributed to the fact that NBA athletes, particularly the **"big men"**, have had to become faster and leaner to adapt to the perimeter-oriented game. This is why NBA centers and forwards are facing the greatest decline in their physical stature. Similar to the evolution of height, point guards are among the heaviest they have ever been, while all others have become lighter.

```
ggplot(avg, aes(x=Year, y=Weight, linetype = "Trend line")) +
  geom_line()+
  labs(x="Anno", y="Weight (Kg)", title = "Evolution of average weight")+
  geom_hline(aes(yintercept = mean(Weight), linetype = "Avg line"), col = "red", alpha = 0.5) +
  scale_x_continuous(breaks = seq(1950, 2021, 10)) +
  scale_linetype_manual(name = "", values = c(2, 1), guide = guide_legend(reverse = TRUE))+
  theme(legend.position = "bottom")
```




```
physique %>%
  ggplot( aes(x=Year, y= Weight, group=Pos, color=Pos)) +
  geom_line()+
  labs(x="Year", y = "Weight (Kg)", title = "Weight evolution by position")+
  theme(legend.position = "bottom")+
  scale_color_manual(values = PosColorCode , name = "")+
  scale_x_continuous(breaks = seq(1950, 2021, 10))
```

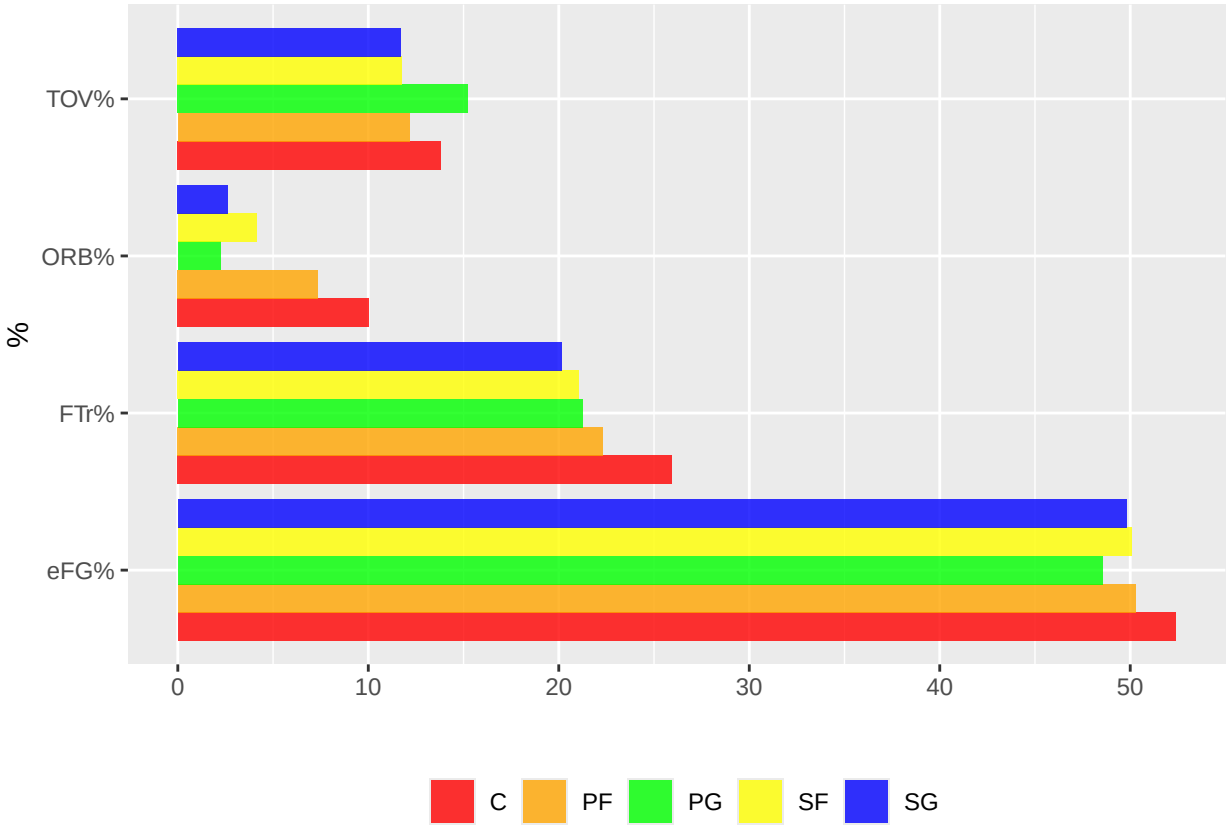


These technical and physical differences between positions are made explicit by some advanced statistics of the game represented by Oliver's four factors. It can be seen that centers and power forwards, taking advantage of their physicality, record high values for the percentage of rebounds collected per game and the rate of free throws obtained, the latter quantifying the player's aggressiveness to attack the basket and thus to obtain fouls; the point guards, being the players from whom each team's offensive actions start, are those who have the greatest ability to distribute smarcating passes to the forwards and thus simplify their finalization, this implies that by having possession of the ball several times, the turnover rate increases; shooting guards, on the other hand, whose job is to score points, especially from behind the arc, turn out to have mediocre values for these indicators.

```
x = nba_performance %>%
  group_by(Pos) %>%
  summarise("ORB%"= mean(ORB., na.rm = T), "eFG%"=mean(eFG., na.rm = T)*100,
            "TOV%"=mean(TOV., na.rm = T), "FTr%" = mean(FTr, na.rm = T)*100)

df <- data.frame(stats = rep(c("ORB%", "eFG%", "TOV%", "FTr%"), each = 5),
                 position = rep(x$Pos, times = 4),
                 value = c(x$`ORB%`,x$`eFG%`, x$`TOV%`, x$`FTr%`))

ggplot(df, aes(fill = position, y = value, x = stats))+
  geom_bar(position = "dodge", stat = "identity", alpha = 0.8)+
  labs(x="%", y="")+
  coord_flip()+
  theme(legend.position = "bottom")+
  scale_fill_manual(values = PosColorCode, name = "")
```



Cluster analysis

In the following section, the non-hierarchical k-means technique is applied to a group of NBA players related to the 2020/2021 season.

For the purpose of the analysis, *Oliver's four factors* were used:

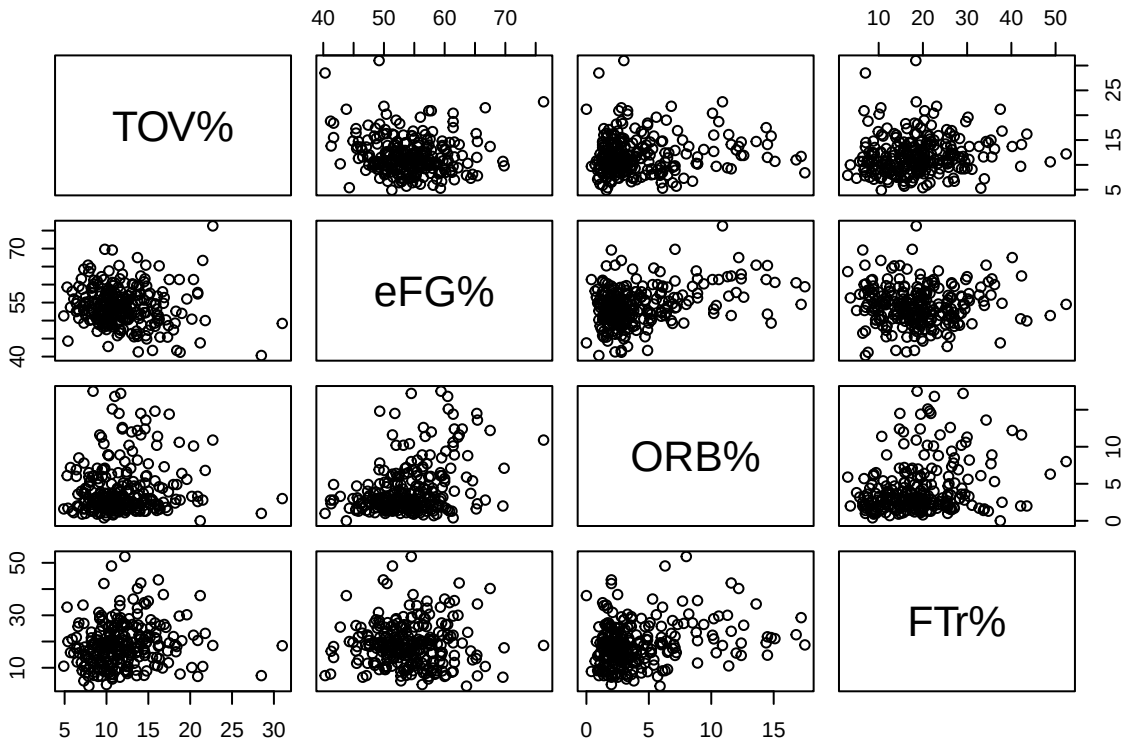
1. **Effective Field Goal Percentage (eFG%)**: A statistic that adjusts the traditional field goal percentage to account for the added value of three-point shots. It is calculated as (Field Goals Made + 0.5 * Three-Point Field Goals Made) / Field Goals Attempted.
2. **Turnover Percentage (TOV%)**: The percentage of possessions that end with a turnover by the team. It is calculated as Turnovers / (Field Goals Attempted + 0.44 * Free Throws Attempted + Turnovers).
3. **Offensive Rebounding Percentage (ORB%)**: The percentage of available offensive rebounds grabbed by the team during a game. It is calculated as Offensive Rebounds / (Offensive Rebounds + Opponent's Defensive Rebounds).
4. **Free Throw Rate (FTr%)**: The ratio of free throws attempted to field goals attempted by the team. It is calculated as Free Throws Attempted / Field Goals Attempted.

```
players = nba_performance %>%
  filter(Year == 2021) %>%
  mutate("TOV%"= TOV., "eFG%"=(eFG.*100),
```

```
         "ORB%"= ORB., "FTr%" = (FTr*100)) %>%
  select(Player, Pos, "TOV%", "eFG%", "ORB%", "FTr%")
table(players$Pos)
```

```
##
##  C PF PG SF SG
## 41 48 51 44 67
```

```
players_data <- players[3:6]
plot(players_data)
```

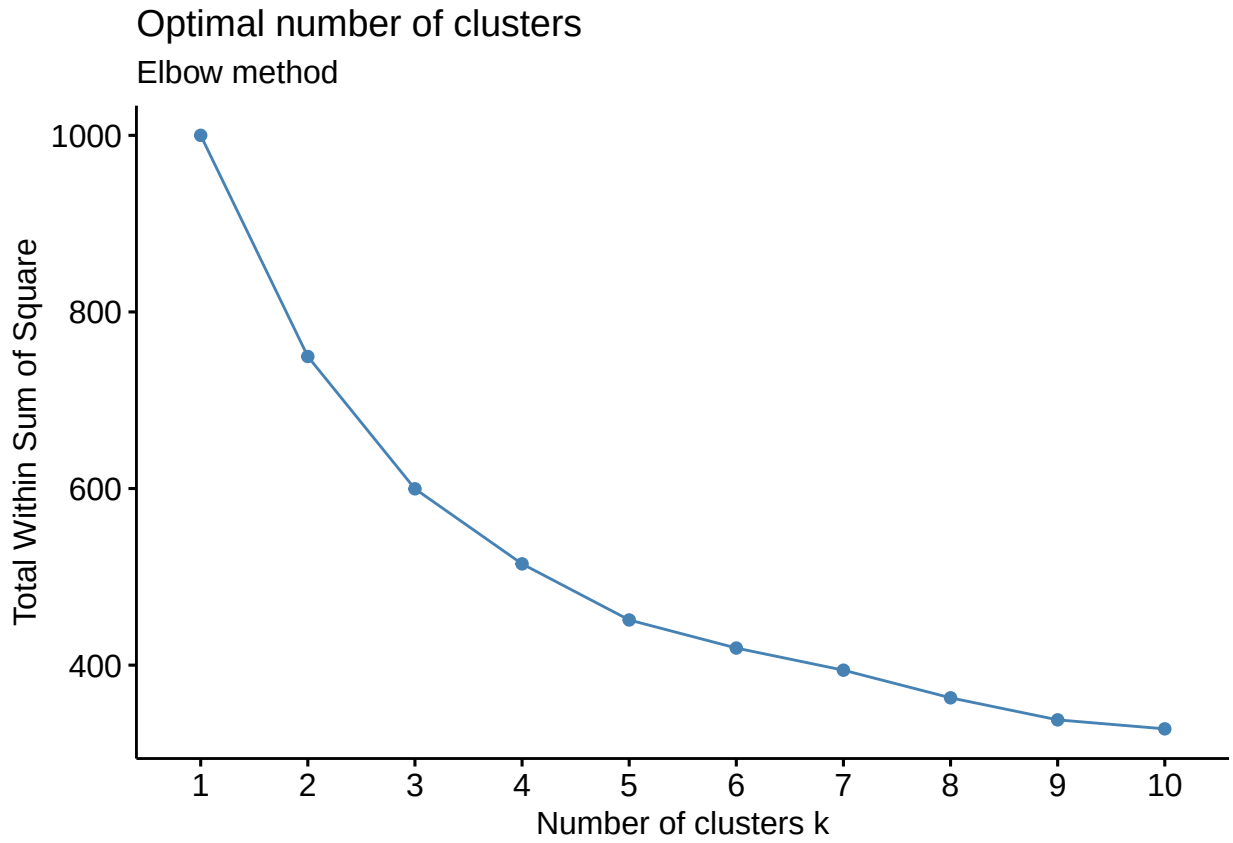


Before beginning the analysis, it should be pointed out how the use of k-means on raw data could provide results that are not very useful because they are highly variable. Therefore, as with hierarchical clustering procedures, we chose to normalize each variable, that is, transform them so that they have a mean of zero and a standard deviation of one, through the scale() function.

```
players_data_scale <- scale(players_data)
```

The methodology involves a preliminary hierarchical analysis to determine the appropriate number of partitions. The technique is that of the elbow method, which is based on analyzing the variation of the within-cluster sum-of-squares index, also known as WSS (Within-Cluster Sum of Squares). The main idea of the elbow method is to identify the point at which the WSS index curve exhibits an "elbow" or an abrupt reversal of its trend. This point indicates the optimal number of clusters to be used.

```
fviz_nbclust(players_data_scale, kmeans, method = "wss")+
  labs(subtitle = "Elbow method")
```

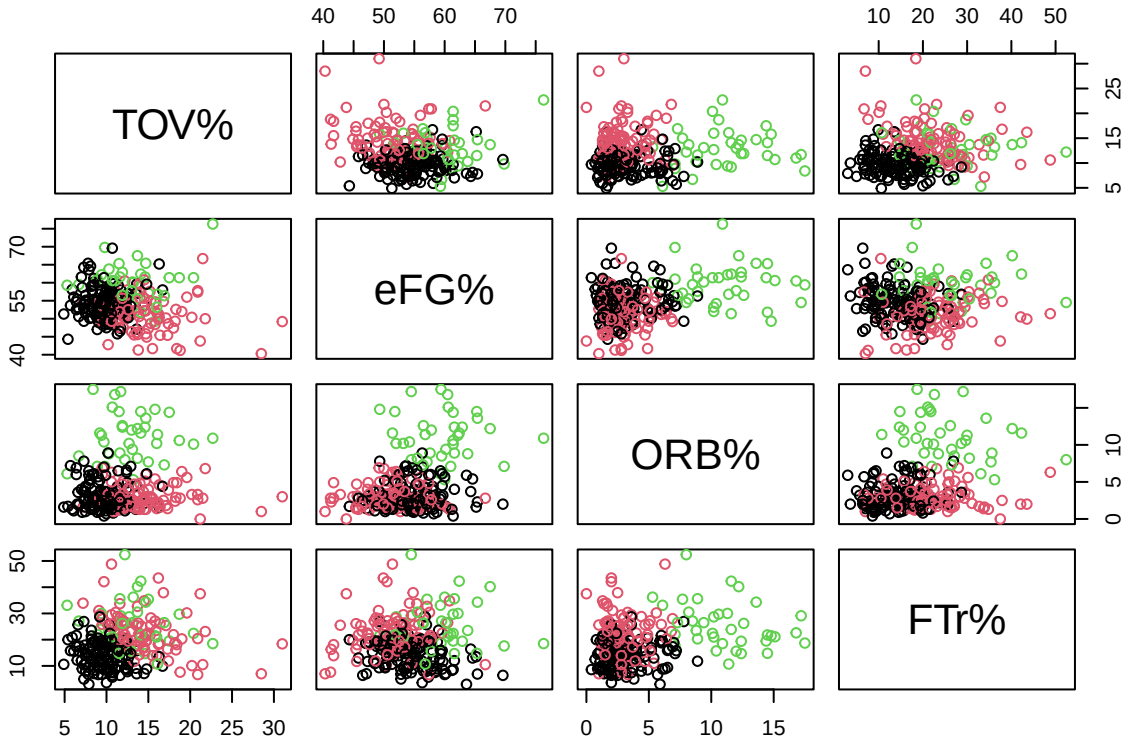



```
set.seed(123)
km.out <- kmeans(players_data_scale, centers = 3)
print(km.out)
```

```
## K-means clustering with 3 clusters of sizes 128, 86, 37
##
## Cluster means:
##         TOV%        eFG%       ORB%       FTr%
## 1 -0.5478098  0.07348199 -0.3315934 -0.5554997
## 2  0.7025981 -0.53854971 -0.3601627  0.4762851
## 3  0.2620600  0.99755623  1.9842689  0.8146876
##
## Clustering vector:
##   [1] 3 3 1 1 1 3 1 3 1 2 1 1 1 3 1 3 2 1 1 2 1 2 1 2 2 1 1 1 1 3 2 1 2 2 3 1 1
##  [38] 1 3 1 1 1 3 1 3 1 1 2 3 1 1 2 1 2 3 3 2 2 1 1 3 1 1 1 1 1 2 2 1 2 2 2 1 2
##  [75] 1 1 2 3 2 1 1 3 2 1 1 2 3 2 2 2 1 2 2 3 2 2 1 2 1 2 1 2 1 1 1 2 3 1 1 1 1
## [112] 2 2 1 1 1 1 1 1 3 1 1 1 2 1 2 1 2 1 2 2 1 2 1 1 1 3 1 3 2 1 1 1 2 1 1 2 1
## [149] 1 2 2 1 2 1 1 2 1 1 2 1 2 2 1 2 1 1 1 1 2 1 3 1 3 1 1 1 2 2 1 2 1 1 2 2 3
## [186] 3 2 2 1 1 1 1 2 2 2 1 1 1 3 2 1 1 1 2 2 3 2 2 2 2 1 2 2 2 1 1 3 1 1 1 1 1
## [223] 3 2 3 1 2 2 3 1 1 1 1 2 1 2 2 2 1 1 1 2 2 3 1 1 2 3 2 3 3
##
## Within cluster sum of squares by cluster:
```

```
## [1] 208.4834 252.4876 138.6934
##  (between_SS / total_SS =  40.0 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
plot(players_data, col = km.out$cluster)
```



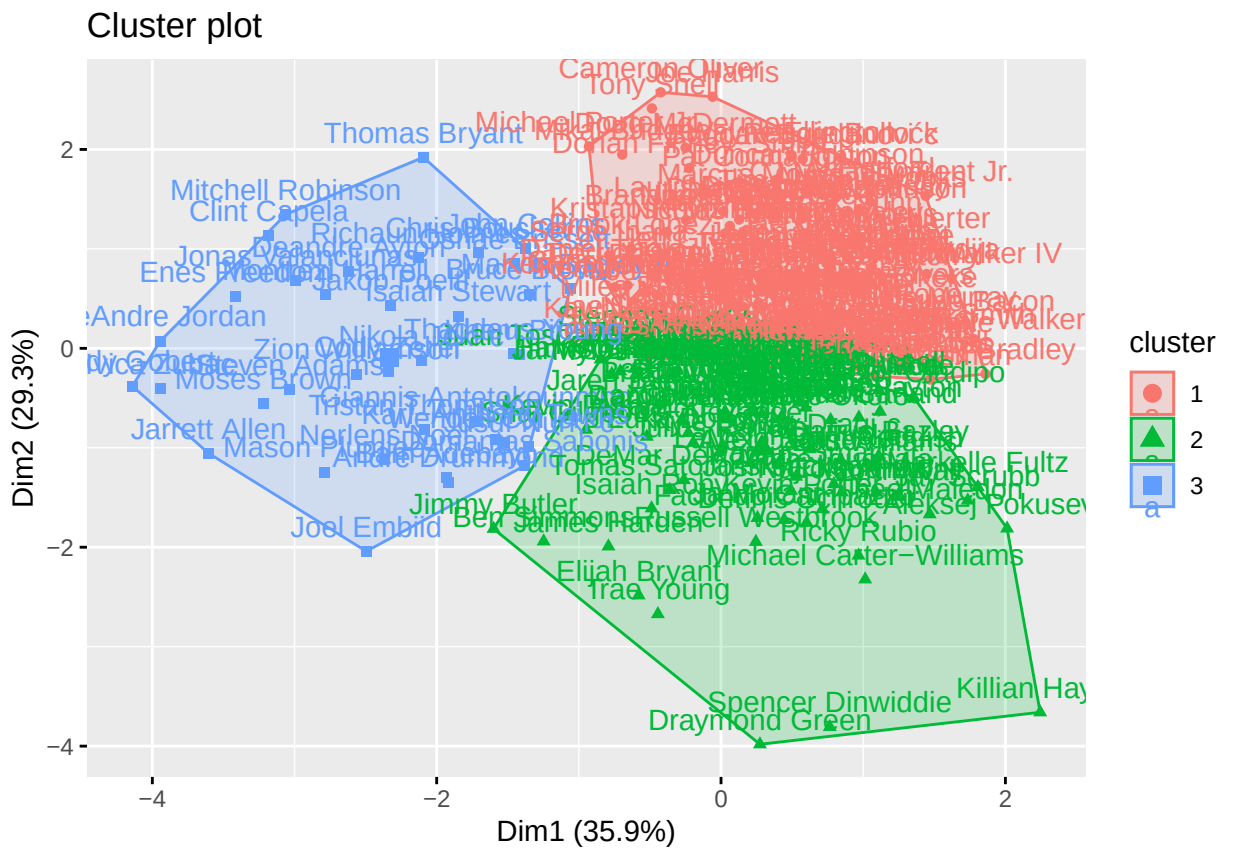
Results

The main output produced by the algorithm is as follows:

```
km.clusters <- km.out$cluster

rownames(players_data_scale) <- players$Player

fviz_cluster(list(data = players_data_scale, cluster= km.clusters))
```




```
table(km.clusters, players$Pos)
```

```
##
## km.clusters  C PF PG SF SG
##           1 10 23 19 30 46
##           2  3 18 31 13 21
##           3 28  7  1  1  0
```

```
players %>%
  mutate(Cluster = km.clusters) %>%
  group_by(Cluster) %>%
  select(-Player, -Pos) %>%
  summarise_all(mean)
```

```
## # A tibble: 3 x 5
##   Cluster `TOV%` `eFG%` `ORB%` `FTr%`
##     <int>  <dbl>  <dbl>  <dbl>  <dbl>
## 1       1   9.80   54.5   3.09   14.3
## 2       2  14.6    51.2   3.00   22.8
## 3       3  12.9    59.5  11.1    25.7
```

1. ***Smooth Operators***. This cluster embodies a harmonious blend of efficiency across multiple facets of the game. With a steady hand in ball control, these players navigate the court with poise, reflected in their low turnover rate. Their shooting prowess shines through an above-average effective field goal

percentage, showcasing their ability to convert opportunities into points. While they may not dominate the offensive boards, their balanced approach ensures they capitalize on scoring chances without overly relying on drawing fouls.

2. ***Attack Mode Squad.*** Characterized by a more aggressive offensive mindset, this cluster shows a tendency to push the envelope in search of scoring opportunities. While their shooting efficiency remains average, turnovers present a recurring challenge, hinting at occasional lapses in ball control. Their eagerness to attack the basket is evident in their high free throw rate, often compensating for shortcomings in other areas. However, their limited presence on the offensive glass suggests a potential area for improvement.

3. ***Dynamic Dominators.*** Players in this cluster embody a dynamic and tenacious approach to the game, combining efficiency with grit. Despite occasional turnovers, their high shooting efficiency underscores their ability to make the most of scoring opportunities. What sets them apart is their strong presence on the offensive boards, relentlessly pursuing second chances and adding an extra dimension to their team's scoring arsenal. Their knack for drawing fouls further amplifies their impact, making them formidable forces on the offensive end.