```
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://
cran.us.r-project.org")
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-
project.org")
if(!require(data.table)) install.packages("data.table", repos = "http://
cran.us.r-project.org")
if(!require(readr)) install.packages("readr", repos = "http://cran.us.r-
project.org")
if(!require(dplyr)) install.packages("dplyr", repos = "http://cran.us.r-
project.org")
if(!require(measurements)) install.packages("dplyr", repos = "http://
cran.us.r-project.org")
if(!require(janitor)) install.packages("janitor", repos = "http://
cran.us.r-project.org")
if(!require(rvest)) install.packages("rvest", repos = "http://cran.us.r-
project.org")
if(!require(corrplot)) install.packages("corrplot", repos = "http://
cran.us.r-project.org")
library(data.table)
library(lubridate)
library(tidyverse)
library(readr)
library(dplyr)
library(measurements)
library(janitor)  # for data cleaning)
library(rvest)      # for web scraping
library(corrplot)   # correlation plots
library(caret)



set.seed(1)


setwd("C:\\Users\\Utilisateur\\Documents\\projects\\Capstone2-NBa")
#
#Download and extracting the files.
#Dataset from: https://www.kaggle.com/drgilermo/nba-players-stats/version/2
#Hide
#fileURL <- "https://www.kaggle.com/drgilermo/nba-players-stats/downloads/
nba-players-stats.zip/2"
#filename <- "NBASeason1950-2017.zip"
#
# Checking if archieve already exists.
#if (!file.exists(filename)){
#  download.file(fileURL, filename, method="curl")
#}
#
# Checking if folder exists
#if (!file.exists("NBA Season Dataset")) {
#  unzip(filename)
#}
#




#players <- read.csv("players.csv", header=TRUE)
```

```
#player_data <- read.csv("player_data.csv", header=TRUE)
#season <- read.csv("Seasons_Stats.csv", header=TRUE)
#player_of_week<-read.csv("NBA_player_of_the_week.csv", header=TRUE)
#Coach<-read.csv("NBA_head_coaches.csv", header=TRUE)

#
-------------------------------------------------------------------------------
#
# The below function will extract players data from basketball
reference.com by season
#
-------------------------------------------------------------------------------
#
scrape_stats <- function(season){
  #total stats
  #scrape
  url <- paste0("https://www.basketball-reference.com/leagues/
NBA_",season,"_totals.html")
  stats_tot <- url %>%
    read_html() %>%
    html_table() %>%
    .[[1]]
  #
  #stats_tot<-stats_tot%>%mutate(Y=season)
  #
  #clean
  player_stats_tot <- stats_tot %>%
    remove_empty("cols") %>%
    clean_names() %>%
    dplyr::filter(!player=="Player") %>%
    mutate_at(vars(-c(player,tm,pos)),as.numeric) %>%
    mutate_at(vars(-c(player,tm,pos)), funs(replace(., is.na(.), 0))) %>%
    as_tibble() %>%
    group_by(player) %>%
    slice(1) %>%
    ungroup() %>%
    select(-rk)

  #per minute
  url <- paste0("https://www.basketball-reference.com/leagues/
NBA_",season,"_per_minute.html")
  stats_pm <- url %>%
    read_html() %>%
    html_table() %>%
    .[[1]]


  #stats_pm<-stats_pm%>%mutate(Y=season)

  player_stats_pm <- stats_pm %>%
    remove_empty("cols") %>%
    clean_names() %>%
    dplyr::filter(!player=="Player") %>%
    mutate_at(vars(-c(player,tm,pos)),as.numeric) %>%
    mutate_at(vars(-c(player,tm,pos)), funs(replace(., is.na(.), 0))) %>%
    as_tibble() %>%
    group_by(player) %>%
    slice(1) %>%
```

```
    ungroup() %>%
    rename_at(vars(9:29),funs(paste0(.,"_pm"))) %>%
    select(-rk)

  #advanced
  url <- paste0("https://www.basketball-reference.com/leagues/
NBA_",season,"_advanced.html")
  stats_adv <- url %>%
    read_html() %>%
    html_table() %>%
    .[[1]]

  #stats_adv<-stats_adv%>%mutate(Y=season)

  player_stats_adv <- stats_adv %>%
    remove_empty("cols") %>%
    clean_names() %>%
    dplyr::filter(!player=="Player") %>%
    mutate_at(vars(-c(player,tm,pos)),as.numeric) %>%
    mutate_at(vars(-c(player,tm,pos)), funs(replace(., is.na(.), 0))) %>%
    as_tibble() %>%
    mutate(year=season)%>%
    group_by(player) %>%
    slice(1) %>%
    ungroup() %>%
    select(-rk)

  player_stats <- full_join(player_stats_tot,player_stats_pm,
                            by = c("player", "pos", "age", "tm", "g", "gs",
"mp")) %>%
    full_join(player_stats_adv,
              by = c("player", "pos", "age", "tm", "g", "mp"))



  return(player_stats)
}

#
# url <- paste0("https://www.basketball-reference.com/players/a/
#players::none")
#
#
------------------------------------------------------------------------------------
scrape_player <- function(alpha){
  # player stats
  # scrape
  lien <- paste0("https://www.basketball-reference.com/players/" ,alpha, "/
#players::none",sep="",collapse=NULL)
  url<-lien
  #
  p_stats <- url %>%
    read_html() %>%
    html_table() %>%
    .[[1]]

  #clean
  p_stats <- p_stats %>%
```

```
    remove_empty_cols() %>%
    clean_names() %>%
    dplyr::filter(!player=="Player") %>%
    as_tibble() %>%
    group_by(player) %>%
    slice(1) %>%
    ungroup()

  return(p_stats)

}
#
#
#size conversion
#
c_height<-function(x){
  x<-as.character(x)
  split<-strsplit(x,"-")
  feet<-as.numeric(split[[1]][1])
  inch<-as.numeric(split[[1]][2])
  x<-round(conv_unit(feet,"ft","cm")+conv_unit(inch,"inch","cm"),0)
}


c_weight<-function(x){
  x<-as.numeric(as.character(x))
  round(conv_unit(x,"lbs","kg"),0)
}
#




theme_set(theme_minimal()+
            theme(legend.position = "bottom",
                  text=element_text(size = 12)))


#
# Baseball player , if csv exist use them, if not download
#
if (file.exists("player_stats_1980.csv")){
  player_stats_1980 <- read.csv("player_stats_1980.csv", header=TRUE)
}
if (file.exists("player_stats_1990.csv")){
  player_stats_1990 <- read.csv("player_stats_1990.csv", header=TRUE)
}
if (file.exists("player_stats_2000.csv")){
  player_stats_2000 <- read.csv("player_stats_2000.csv", header=TRUE)
}
  if (file.exists("player_stats_last.csv")){
    player_stats_last <- read.csv("player_stats_last.csv", header=TRUE)
  }


#
#  If no csv saved, reload data from website
#
```

```
setwd('C:\\Users\\Utilisateur\\Documents\\projects\\Capstone2-NBa')
getwd()
if (!file.exists("player_stats_last.csv")){
player_stats_last <- map_dfr(2018:2019,scrape_stats)
player_stats_1980 <-map_dfr(1980:1989,scrape_stats)
player_stats_1990 <- map_dfr(1990:1999,scrape_stats)
player_stats_2000 <- map_dfr(2000:2017,scrape_stats)
#
write.csv(player_stats_last,"player_stats_last.csv")
write.csv(player_stats_1980,"player_stats_1980.csv")
write.csv(player_stats_1990,"player_stats_1990.csv")
write.csv(player_stats_2000,"player_stats_2000.csv")
}
#
#  merge data
#
player_stats<-player_stats_last
player_stats<-merge(player_stats,player_stats_1990,all=TRUE)
player_stats<-merge(player_stats,player_stats_2000,all=TRUE)
player_stats<-merge(player_stats,player_stats_1980,all=TRUE)

#filter by player with more the  mn played
player_stats <- player_stats %>%
  dplyr::filter(mp>=500)

write.csv(player_stats,"player_stats.csv", row.names = FALSE)



#
# research all players in alphabetical; no x in databse
#
#
alphabet<-letters[seq( from = 1, to = 26 )]
alphabet<-alphabet[-24]
alphabet


if (file.exists("player_data.csv")){
  player_data <- read.csv("player_data.csv", header=TRUE)
}
#
if (!file.exists("player_data.csv")){
player_data <-map_dfr(alphabet,scrape_player)
#
write.csv(player_data,"player_data.csv")
}

#
#
#  i keep the rw data in player_stats and create a working set NBA
#
#
player_data$Player <- gsub("\\*$", "", player_data$player)
player_data<-player_data%>%filter(!is.na(wt) &!is.na(ht))%>%rowwise()
%>%mutate(p_cm=(c_height(ht)),p_kg=(c_weight(wt)))
player_d<-player_data%>%select(player,p_cm,p_kg)
#
```

```
NBA <- left_join(player_stats,player_d, by=c("player"))


# --------------------             DATA Cleaning
-----------------------------------------------
# Remove NA rows
NBA <- NBA %>% filter(!is.na(year), !is.na(player))
# Remove Team = TOT (which indicates total, when player played in more than
1 team in a season)
NBA <- NBA[NBA$tm != "TOT",]
# Remove of "*" which indicates a player is a member of NBA Hall of Fame
#NBA$Player <- gsub("\\*$", "", NBA$Player)
# Fix player data
#PlayerData[2143, 4] = as.factor("6-2")
#PlayerData[2143, 5] = 190
#NBA[21304, 3] = "SG"
#
str(NBA)
head(NBA)
dim(NBA)
#
str(player_data)
head(player_data)
dim(player_data)
#
#str(season)
#head(season)
#dim(season)
#season$Player
#season$year
#
# removing na rows
#
#season <- season %>% filter(!is.na(Player), !is.na(Year))

#
NBA %>%
       ggplot(aes(year)) +
       geom_histogram(binwidth=0.2, color="darkblue", fill="lightblue") +
       ggtitle("Nb Players by season")

Team_stat<-NBA %>% group_by(year) %>%
summarise(nb_player=n_distinct(player),
nb_teams=n_distinct(tm),nb_game=max(g), play_by_team=round(nb_player/
nb_teams))


Team_stat %>%
  ggplot()+geom_line(aes(year,nb_player)) +
  ggtitle("Number of Players by Year")

Team_stat %>%
  ggplot()+geom_line(aes(year,nb_teams)) +
  ggtitle("Number of Teams by year")

Team_stat %>%
  ggplot()+geom_line(aes(year,nb_game)) +
  ggtitle("Number of game by year")
```

```
-------------------------------------------------------------------------------
#
# Player statitics - convert height and weight
#
#-------------------------------------------------------------------------------
player_data<-player_data%>%rowwise()
%>%mutate(height=c_height(ht),weight=c_weight(wt))
print(player_data)
p_stat<-player_data%>%filter(!is.na(wt) &!is.na(ht))%>%rowwise()
%>%mutate(p_cm=(c_height(ht)),p_kg=(c_weight(wt)))
p_stat%>%group_by(from) %>% summarise(avg_h=mean(p_cm), avg_p=mean(p_kg))

# Players Height distribution for all players
p_stat %>%
  ggplot(aes(p_cm,fill=TRUE),color="BLUE") +
  geom_density() +
  ggtitle("Height Distribution of all Player")
#
# Evolution of players heights by year
#
j<-p_stat  %>% group_by(from)
%>%summarize(avg_h=mean(p_cm),avg_p=mean(p_kg))
j %>%
     ggplot(aes(x=as.numeric(from),y=as.numeric(avg_h))) +
     geom_line()+geom_smooth()+ggtitle("Players Heights by Starting Years")

# Evolution of players weight by year

j %>%
  ggplot(aes(x=as.numeric(from),y=as.numeric(avg_p))) +
  geom_line()+geom_smooth()+ggtitle("Players Weights by Starting Years")

# compare weight and height

j %>%
  ggplot(aes(x=as.numeric(from)))+geom_line(aes(y=as.numeric(avg_p))) +
  geom_line(aes(y=as.numeric(avg_h)))
#
#
#
#
#
# now analyse players by position
#
D_by_pos<-p_stat%>%group_by(from,pos)%>%summarise
(Nb=n(),start=mean(from),avg_w=mean(p_kg),avg_h=mean(p_cm))
#
#
#
print(D_by_pos)
#
D_by_pos %>%  group_by(pos)
%>%summarise(pos_h=mean(avg_h),pos_w=mean(avg_w))
print(D_by_pos)
#
D_by_pos%>%ggplot(aes(x=(pos)))+geom_line(aes(y=as.numeric(avg_w)))
D_by_pos%>%ggplot(aes(x=(pos)))+geom_line(aes(y=as.numeric(avg_h)))
```

```
#
#  Big one by position short and tall
#
player_data %>%
     group_by(height, player) %>%
     summarise(pos,YearActive = paste(mean(from), "-", mean(to)))%>%
     head()

player_data %>%
  group_by(p_cm, player) %>%
  summarise(pos,YearActive = paste(mean(from), "-", mean(to)))%>%
  tail()
#
# Stats by position
#

p_stat %>%
  group_by(pos) %>%
  summarise(MinHeight = min(p_cm),
            MaxHeight = max(p_cm),
            MedianHeight = median(p_cm),
            MeanHeight = round(mean(`p_cm`), 2))

p_stat %>%
  ggplot(aes(pos, p_cm, color=pos)) +
  geom_violin() +
  ggtitle("Height distribution by position") +
  stat_summary(fun.y=mean, geom="point", shape=8, size=6) +
  geom_point() +
  geom_hline(aes(yintercept = mean(p_stat$p_cm, na.rm=T), linetype =
"Average NBA players"),
             col = "red",
             alpha = 0.5) +
  geom_hline(aes(yintercept = 179, linetype = "Average American male"),
             col = "blue",
             alpha = 0.5) +
  theme(legend.position="bottom")


p_stat %>%
  ggplot(aes(pos, p_kg, color=pos)) +
  geom_violin() +
  ggtitle("Weight distribution by position") +
  stat_summary(fun.y=mean, geom="point", shape=8, size=6) +
  geom_point() +
  geom_hline(aes(yintercept = mean(p_stat$p_kg, na.rm=T), linetype =
"Average NBA players"),
             col = "red",
             alpha = 0.5) +
  geom_hline(aes(yintercept = 80, linetype = "Average American male"),
             col = "blue",
             alpha = 0.5) +
  theme(legend.position="bottom")
# Players last year
#  Points by position
#
NBA20<-NBA%>% filter(year>2017)
```

```
NBA20%>%group_by(pos) %>%
  summarise(Games=mean(g),FieldGoal=mean(fg),Attemps=mean(fga))
#
NBA20%>%group_by(player,pos) %>%
  summarise(Games=mean(g),FieldGoal=mean(fg),Attemps=mean(fga))
%>%arrange(desc(FieldGoal))%>%head()
#
# filtering players with more than 50 games a year
#
NBA20 %>% filter(g>50)%>%
  ggplot(aes(pos, fg, color=pos)) +
  geom_violin() +
  ggtitle("Field Goal by Position") +
  stat_summary(fun.y=mean, geom="point", shape=8, size=6) +
  geom_point() +
  geom_hline(aes(yintercept = mean(NBA20$fg, na.rm=T), linetype = "Average
FG by players"),
             col = "red",
             alpha = 0.5)  +
  theme(legend.position="bottom")
#
# 3points by position
#
NBA20 %>% filter(g>50)%>%
  ggplot(aes(pos, x3p, color=pos)) +
  geom_violin() +
  ggtitle("3 Points by Position") +
  stat_summary(fun.y=mean, geom="point", shape=8, size=6) +
  geom_point() +
  geom_hline(aes(yintercept = mean(NBA20$x3p, na.rm=T), linetype = "Average
3Pts by players"),
             col = "red",
             alpha = 0.5)  +
  theme(legend.position="bottom")
#
# 2 points by positions
#
NBA20 %>% filter(g>50)%>%
  ggplot(aes(pos, x2p, color=pos)) +
  geom_violin() +
  ggtitle("2 Points by Position (50 games played)") +
  stat_summary(fun.y=mean, geom="point", shape=8, size=6) +
  geom_point() +
  geom_hline(aes(yintercept = mean(NBA20$x2p, na.rm=T), linetype = "Average
3Pts by players"),
             col = "red",
             alpha = 0.5)  +
  theme(legend.position="bottom")
#
# shotters
#
NBA20 %>% filter(g>50)%>%
  ggplot(aes(pos, x2p+x3p, color=pos)) +
  geom_violin() +
  ggtitle("Points by Position(>50 gmas played)") +
  stat_summary(fun.y=mean, geom="point", shape=8, size=6) +
  geom_point() +
```

```
  geom_hline(aes(yintercept = mean(NBA20$x2p+NBA20$x3p, na.rm=T), linetype
= "Average Pts by players"),
             col = "red",
             alpha = 0.5)  +
  theme(legend.position="bottom")

#
# REbound by player
#
NBA20 %>% filter(g>50 )%>%
  ggplot(aes(pos, orb+drb, color=pos)) +
  geom_violin() +
  ggtitle("Rebounds by Position (>50 games played") +
  stat_summary(fun.y=mean, geom="point", shape=8, size=6) +
  geom_point() +
  geom_hline(aes(yintercept = mean(NBA20$orb+NBA20$drb, na.rm=T), linetype
= "Average Rebound by players"),
             col = "red",
             alpha = 0.5)  +
  theme(legend.position="bottom")
#
#  DEfense
#
NBA20 %>% filter(g>50)%>%
  ggplot(aes(pos, stl+blk, color=pos)) +
  geom_violin() +
  ggtitle("Steal/Block by Position (50 Games played)") +
  stat_summary(fun.y=mean, geom="point", shape=8, size=6) +
  geom_point() +
  geom_hline(aes(yintercept = mean(NBA20$stl+NBA20$blk, na.rm=T), linetype
= "AverageBlock+Steal by players"),
             col = "red",
             alpha = 0.5)  +
  theme(legend.position="bottom")
#
#  assist + turnover
#
NBA20 %>% filter(g>50)%>%
  ggplot(aes(pos, ast+tov, color=pos)) +
  geom_violin() +
  ggtitle("Assist/Turnover by Position (>50 games played)") +
  stat_summary(fun.y=mean, geom="point", shape=8, size=6) +
  geom_point() +
  geom_hline(aes(yintercept = mean(NBA20$ast+NBA20$tov, na.rm=T), linetype
= "Average Assist+Turnover by players"),
             col = "red",
             alpha = 0.5)  +
  theme(legend.position="bottom")
#
#  compare with stats by minutes players
#
NBA20 %>% filter(g>50 & mp>1500)%>%
  ggplot(aes(pos, x3p_pm+x2p_pm, color=pos)) +
  geom_violin() +
  ggtitle("Points per Minutes by Position") +
  stat_summary(fun.y=mean, geom="point", shape=8, size=6) +
  geom_point() +
```

```
  geom_hline(aes(yintercept = mean(NBA20$x3p_pm+NBA20$x2p_pm, na.rm=T),
linetype = "Average Pts per Min by players"),
             col = "red",
             alpha = 0.5)  +
  theme(legend.position="bottom")

NBA20 %>% filter(g>50 & mp>1500)%>%
  ggplot(aes(pos, ast_pm+tov_pm, color=pos)) +
  geom_violin() +
  ggtitle("Assist/Turnover played >1500 mn") +
  stat_summary(fun.y=mean, geom="point", shape=8, size=6) +
  geom_point() +
  geom_hline(aes(yintercept = mean(NBA20$ast_pm+NBA20$tov_pm, na.rm=T),
linetype = "Average Assist+Turnover by players"),
             col = "red",
             alpha = 0.5)  +
  theme(legend.position="bottom")
#
#Player info dataset
#
#ORpG: (Offensive Rebounds per Game): Average offensive rebounds a player
made in a game. (ORB/G)
#DRpG: (Defensive Rebounds per Game): Average defensive rebounds a player
made in a game. (DRB/G)
#RpG (Rebounds per Game): Average rebounds a player made in a game. (TRB/G)
#ApG (Assists per Game): Average assists a player made in a game. (AST/G)
#SPG (Steals per Game): Average rebounds a player made in a game. (STL/G)
#BPG (Blocks Per Game): Average rebounds a player made in a game. (BLK/G)
#TPG (Turnovers Per Game): Average turnovers a player made in a game. (TOV/
G)
#PpG (Points per Game): Average points a player made in a game. (PTS/G)
#Pos Position

P_info<-NBA20%>%filter(g>50)%>%
  rowwise() %>%
  mutate(ORpG = orb / g,
         DRpG = drb / g,
         RpG = trb / g,
         ApG = ast / g,
         SpG = stl / g,
         BpG = blk / g,
         TpG = tov / g,
         PpG = pts / g)%>%
select(player,pos,ORpG,DRpG,RpG,ApG,SpG,BpG,TpG,PpG,p_cm,p_kg)

#
# Computing the Principal Components (PC)
# I will use NBA dataset with 8 components for the demonstration. The data
contain 8 continuous variables
# which corresponds to ability and a categorical variable describing the
player position.
#  I will not use age, height and weight, as they impact the 8 datas

# data transform for the selected dataset on P_info
log.player <-(P_info[, 3:12])
pl.pos <- P_info[, 2]
pl.name<-P_info[,1]
print(log.player)
```

```
#
sum(is.na (log.player))
log.player<-na.omit(log.player)
#
# search for correlation
#
c<-cor(log.player)
print(c)
corrplot.mixed(cor(log.player), order="hclust", tl.col="black")
#
# apply PCA - scale. = TRUE is highly
# advisable, but default is FALSE.
#
player.pca <- prcomp(log.player,
                 center = TRUE,
                 scale. = TRUE)
print(player.pca)
#
plot(player.pca)

player_var <- get_pca_var(player.pca)
pc_s <- player_var$contrib[,1:8]
colnames(pc_s) <- paste0("PC",1:8)

as_tibble(pc_s   ,rownames = "stat") %>%
  gather(pc,contrib,PC1:PC8) %>%
  mutate(pc=factor(pc,levels=paste0("PC",1:10))) %>%
  group_by(pc) %>%
  top_n(5,contrib) %>%
  ggplot(aes(x=stat,y=contrib))+
  geom_col()+
  coord_flip()+
  facet_wrap(~pc,scales = "free",ncol=5)+
  labs(x="",y="")
#
# PC1: (Turnovers Per Game)-RpG(Rebounds per Game)-(Defensive Rebounds per
Game)-(Points per Game)
# Overall player with good Rebounds capabilities
#
# PC2: (Offensive Rebounds per Game)-(Assists per Game)
# Offensive player with good assist and offensive rebound capabilities
#
# PC3: (Steals per Game)
# Defensive player able to steal the ball
#
# PC4: (Blocks Per Game)
# Big defensive player focus on blocking
#
# PC5: (Points per Game)-(Assists per Game)
# The offensive top scorer, focus on Points and assist
#
# PC6: (Offensive Rebounds per Game)-(Defensive Rebounds per Game)
# The overall rebounder, focus on offensive and defensive rebound
#
# PC7: ((Turnovers Per Game)-(Assists per Game))
# Offensive Support player
#
#PC8 : (Rebounds per Game) (Defensive Rebounds per Game)
```

```
# Defensive support player
#
#ORpG: (Offensive Rebounds per Game): Average offensive rebounds a player
made in a game. (ORB/G)
#DRpG: (Defensive Rebounds per Game): Average defensive rebounds a player
made in a game. (DRB/G)
#RpG (Rebounds per Game): Average rebounds a player made in a game. (TRB/G)
#ApG (Assists per Game): Average assists a player made in a game. (AST/G)
#SPG (Steals per Game): Average steal a player made in a game. (STL/G)
#BPG (Blocks Per Game): Average rebounds a player made in a game. (BLK/G)
#TPG (Turnovers Per Game): Average turnovers a player made in a game. (TOV/
G)
#PpG (Points per Game): Average points a player made in a game. (PTS/G)
#

#
# The summary method describe the importance of the PCs.
# The first row describe again the standard deviation associated with each
PC.
# The second row shows the proportion of the variance in the data explained
by each component
# while the third row describe the cumulative proportion of explained
variance.
# We can see there that the first five PCs accounts for more than 95% of
the variance of the data.
#
summary(player.pca)
#
# Predict Principal Components PCs
#
predict(player.pca,
        newdata=tail(log.player, 2))

player_stats_ld <- player.pca$x[,1:8]

P_info<-na.omit(P_info)


player_clus <- kmeans(player_stats_ld,centers = 5,iter.max = 150)
summary(player_clus)

aggregate(player_stats_ld,by=list(player_clus$cluster),mean)

#The following code is used to visualizes the cluster centers
as_tibble(player_clus$centers) %>%
  gather(component,value,PC1:PC8) %>%
  mutate(clust = rep(1:5,8)) %>%
  ggplot(aes(x=factor(component,levels = paste0("PC",10:1)),y=value))+
  geom_col()+
  coord_flip()+
  facet_wrap(~clust)+
  labs(x="",y="")

print(pc_s)



X <- sapply(1:7, function(i){
```

```
  kmeans(pc_s,i,nstart=50,iter.max=15)$tot.withinss

})

print(X)
plot(1:7, X,type="b",pch=19,frame=FALSE,xlab="NB Clustr", ylab="Total
within cluster")
#
# decide to choose Nb cluster = 5 to tke winthin >4000
#
print(player.pca)
player_stats_ld <- player.pca$x[,1:10]

aggregate(player_stats_ld,by=list(player_clus$cluster),mean)


player_clus <- kmeans(player_stats_ld,centers = 5,iter.max = 150)
summary(player_clus)

#The following code is used to visualizes the cluster centers
as_tibble(player_clus$centers) %>%
  gather(component,value,PC1:PC8) %>%
  mutate(clust = rep(1:5,8)) %>%
  ggplot(aes(x=factor(component,levels = paste0("PC",10:1)),y=value))+
  geom_col()+
  coord_flip()+
  facet_wrap(~clust)+
  labs(x="",y="")

#
#  Pc1 to PC6 define the clusters
# what do they represent ?
#Traditionally, basketball has 5 specific positions on the court.
#Two guards, two forwards, and a center.
#1. Point guard
#2. Shooting guard
#3. Small forward
#4. Power forward
#5. Center
#


class.players = cbind(P_info, player.pca$x)

class.players$km.cluster = player.pca$cluster

summary(class.players)







set.seed(5)

num.clusters = 12
```

```
cluster_data = P_info %>%
  select(-player,-pos) %>%
  scale()
km.mod = kmeans(cluster_data, centers=num.clusters, iter.max=50)
km.mod



# ------------------------------------------------------------------------
# plot results by PCs
#-------------------------------------------------------------------------
class.players = cbind(P_info, player.pca$x)
class.players$km.cluster = km.mod$cluster


class.players %>% filter(km.cluster == 8)
summary(class.players)


my.cols = c('black', 'blue', 'yellow', 'lightgreen', 'cadetblue2',
            'darkorange', 'forestgreen', 'darkorchid', 'goldenrod', 'red',
'green2', 'lightpink3')



palette(my.cols)



plot(PC2 ~ PC1, data=class.players, col=km.cluster, pch=16, main = "Player
clusters by first two principal components")
pt.labels = ifelse(class.players$g > 70, class.players$player, "")
text(class.players$PC1, class.players$PC2, pt.labels, pos=2, cex=0.5)
cluster.labels = c("1: banger",
                   "2: exterior distributor",
                   "3: defensive stopper",
                   "4: offensive hub",
                   "5: size and distance",
```

```
                    "6: under the basket",

                    "7: stretch big",

                    "8: attacking shooter",

                    "9: inside / outside",

                    "10: 3-point specialist",

                    "11: attacking distributor",

                    "12: exterior shooter")

legend(5.4, -1.2, legend=cluster.labels, col=my.cols, pch=16, cex=0.6)
legend("topright", legend=cluster.labels, col=1:num.clusters, pch=16,
cex=0.45)
palette("default")

#
#
#  PC1 size / physical
#  PC2 Quicknes/Ball handling
#  PC3 Ball Catcher
# ---------------------------------- GLOBAL VIEW
-------------------------------------------------------

plot(PC2 ~ PC1, data=class.players, col=km.cluster, pch=16, main =
"Position clusters by first two principal components",
     xlab="PC1: 'size/physicality'", ylab="PC2: 'quickness/ballhandling'")

pt.labels = class.players$pos

text(class.players$PC1, class.players$PC2, pt.labels, pos=2, cex=0.5)

cluster.labels = class.players$pos

legend(5.4, -1.2, legend=cluster.labels, col=my.cols, pch=16, cex=0.6)
palette("default")



plot(PC3 ~ PC2, data=class.players, col=km.cluster, pch=16, main =
"Position clusters by PC2 vs PC3 components",
     xlab="PC1: 'Quick/BallHandling'", ylab="PC3: 'PickPocket'")

pt.labels = class.players$pos

text(class.players$PC1, class.players$PC2, pt.labels, pos=2, cex=0.5)

cluster.labels = class.players$pos

legend(5.4, -1.2, legend=cluster.labels, col=my.cols, pch=16, cex=0.6)
palette("default")

# -------------------------------------1 PF Only
-------------------------------------------------
```

```
d<-class.players%>%filter(class.players$pos=="PF")


plot(PC2 ~ PC1, data=d, col=km.cluster, pch=16, main = " PF by first two
principal components",
     xlab="PC1: 'size/physicality'", ylab="PC2: 'quickness/ballhandling'")

pt.labels = d$player
cluster.labels = d$player

text(d$PC1, d$PC2, pt.labels, pos=2, cex=0.5)
palette("default")

# ---------------------------------------2 CEnter only
-----------------------------------------

d<-class.players%>%filter(class.players$pos=="C")


plot(PC2 ~ PC1, data=d, col=km.cluster, pch=16, main = " C by first two
principal components",
     xlab="PC1: 'size/physicality'", ylab="PC2: 'quickness/ballhandling'")

pt.labels = d$player
cluster.labels = d$player
text(d$PC1, d$PC2, pt.labels, pos=2, cex=0.5)
palette("default")

# ------------------------------------ 3 PG Only
-------------------------------------------------

d<-class.players%>%filter(class.players$pos=="PG")


plot(PC2 ~ PC1, data=d, col=km.cluster, pch=16, main = " PG by first two
principal components",
     xlab="PC1: 'size/physicality'", ylab="PC2: 'quickness/ballhandling'")

pt.labels = d$player
cluster.labels = d$player
text(d$PC1, d$PC2, pt.labels, pos=2, cex=0.5)
palette("default")



# ---------------------------------------- 4 SG only
-----------------------------------------

d<-class.players%>%filter(class.players$pos=="SG")


plot(PC2 ~ PC1, data=d, col=km.cluster, pch=16, main = " SG by first two
principal components",
     xlab="PC1: 'size/physicality'", ylab="PC2: 'quickness/ballhandling'")

pt.labels = d$player
cluster.labels = d$player
```

```
text(d$PC1, d$PC2, pt.labels, pos=2, cex=0.5)
palette("default")

# ------------------------------- 5 SF only
-------------------------------------------------

d<-class.players%>%filter(class.players$pos=="SF")


plot(PC2 ~ PC1, data=d, col=km.cluster, pch=16, main = " SF by first two
principal components",
     xlab="PC1: 'size/physicality'", ylab="PC2: 'quickness/ballhandling'")

pt.labels = d$player
cluster.labels = d$player
text(d$PC1, d$PC2, pt.labels, pos=2, cex=0.5)
palette("default")


#
-----------------------------------------------------------------------------------------


plot(PC3 ~ PC1, data=class.players, col=km.cluster, pch=16, main = "Player
clusters by first two principal components",
     xlab="PC1: 'size/physicality'", ylab="PC3: 'pickpockets'")

pt.labels = ifelse(class.players$g > 70, class.players$player, "")

text(class.players$PC1, class.players$PC2, pt.labels, pos=2, cex=0.5)

cluster.labels = c("1: banger",

                   "2: exterior distributor",

                   "3: defensive stopper",

                   "4: offensive hub",

                   "5: size and distance",

                   "6: under the basket",

                   "7: stretch big",

                   "8: attacking shooter",

                   "9: inside / outside",

                   "10: 3-point specialist",

                   "11: attacking distributor",

                   "12: exterior shooter")

legend(5.4, -1.2, legend=cluster.labels, col=my.cols, pch=16, cex=0.6)
legend("topright", legend=cluster.labels, col=1:num.clusters, pch=16,
cex=0.45)
```

```
palette("default")
```

```
#
-------------------------------------------------------------------------------------
#
#                  using more players  data
#
#
-------------------------------------------------------------------------------------

# ----------------------------------FULL set
17-----------------------------------------------

#
-------------------------------------------------------------------------------
#
#  same with full set of data
#
P_info_all<-NBA %>% filter(mp>500) %>% group_by(player)%>%
  select(player,pos,p_cm,p_kg,fg,fga,x3p,x3pa,x2p,x2pa,ft,fta,orb,drb,trb,ast,stl,blk,t
sum(is.na (nba.pca))
P_info_all<-na.omit(P_info_all)
print(P_info_all)


log.player <-(P_info_all[, 5:21])
pl.pos <- P_info[, 2]
pl.name<-P_info[,1]
print(log.player)
#
sum(is.na (log.player))
log.player<-na.omit(log.player)
#
# search for correlation
#
c<-cor(log.player)
print(c)
corrplot.mixed(cor(log.player), order="hclust", tl.col="black")
```

```
P_info_all<-aggregate(P_info_all,by=list(P_info_all$player),mean)
```

```
print(P_info_all)
#

nba.pca <- P_info_all %>% select(fg:pts)%>%
  as.matrix() %>%
  prcomp(center = TRUE,scale = TRUE,retx = TRUE)

print (nba.pca)
```

```
nb<-17

player_stats_ld <- nba.pca$x[,1:nb]

print(nba.pca)
#
plot(nba.pca)
summary(nba.pca)
#
# let' s try with 17 vars
#
library("factoextra")
nba_var <- get_pca_var(nba.pca)
pcs <- nba_var$contrib[,1:17]
colnames(pcs) <- paste0("PC",1:17)

print(pcs)
summary(pcs)

as_tibble(pcs,rownames = "stat") %>%
  gather(pc,contrib,PC1:PC17) %>%
  mutate(pc=factor(pc,levels=paste0("PC",1:17))) %>%
  group_by(pc) %>%
  top_n(5,contrib) %>%
  ggplot(aes(x=stat,y=contrib))+
  geom_col()+
  coord_flip()+
  facet_wrap(~pc,scales = "free",ncol=5)+
  labs(x="",y="")
#
-------------------------------------------------------------------------------
# data definition
#
-------------------------------------------------------------------------------
#Season -- If listed as single number, the year the season ended.
#★ - Indicates All-Star for league.
#Only on regular season tables.
#Age -- Player's age on February 1 of the season
#Tm -- Team
#Lg -- League
#Pos -- Position
#G -- Games
#GS -- Games Started
```

```
#MP -- Minutes Played Per Game
#FG -- Field Goals Per Game
#FGA -- Field Goal Attempts Per Game
#FG% -- Field Goal Percentage
#3P -- 3-Point Field Goals Per Game
#3PA -- 3-Point Field Goal Attempts Per Game
#3P% -- 3-Point Field Goal Percentage
#2P -- 2-Point Field Goals Per Game
#2PA -- 2-Point Field Goal Attempts Per Game
#2P% -- 2-Point Field Goal Percentage
#eFG% -- Effective Field Goal Percentage
#
#This statistic adjusts for the fact that a 3-point field goal is worth one
more point than a 2-point field goal.
#
#FT -- Free Throws Per Game
#FTA -- Free Throw Attempts Per Game
#FT% -- Free Throw Percentage
#ORB -- Offensive Rebounds Per Game
#DRB -- Defensive Rebounds Per Game
#TRB -- Total Rebounds Per Game
#AST -- Assists Per Game
#STL -- Steals Per Game
#BLK -- Blocks Per Game
#TOV -- Turnovers Per Game
#PF -- Personal Fouls Per Game
#PTS -- Points Per Game
# ------------------------------------------------------------
# PC1,PC11: 2pts Shooter,fied goal attemps
# PC2: 3pts shooter,offensive rebonds, block
# PC3,PC13:  3 pts shooter
# PC4,PC8: steal and assist (mixt)
# PC5,6: pure blocker (def)
# PC7: aggresive faulty player
# PC9,PC14: Rebound expert
# PC10: support player
# PC12: passing
#

#corrplot(p_info_all ,type="upper",method="number")
#
X <- sapply(1:15, function(i){
  kmeans(pcs,i,nstart=50,iter.max=15)$tot.withinss

})

print(X)
plot(1:15, X,type="b",pch=19,frame=FALSE,xlab="NB Clustr", ylab="Total
within cluster")
#
# decide to choose Nb cluster = 5 to tke winthin >4000
#
player_stats_ld <- nba.pca$x[,1:10]
fit<-kmeans(player_stats_ld,centers=5)
print(fit)
str(fit)
summary(fit)
```

```
aggregate(player_stats_ld,by=list(fit$cluster),mean)


player_clus <- kmeans(player_stats_ld,centers = 5,iter.max = 150)
summary(player_clus)

#The following code is used to visualizes the cluster centers
as_tibble(player_clus$centers) %>%
  gather(component,value,PC1:PC10) %>%
  mutate(clust = rep(1:5,10)) %>%
  ggplot(aes(x=factor(component,levels = paste0("PC",10:1)),y=value))+
  geom_col()+
  coord_flip()+
  facet_wrap(~clust)+
  labs(x="",y="")

#
#  Pc1 to PC17 define the clusters
# what do they represent ?
#Traditionally, basketball has 5 specific positions on the court.
#Two guards, two forwards, and a center.
#1. Point guard
#2. Shooting guard
#3. Small forward
#4. Power forward
#5. Center
#


class.players = cbind(P_info_all, nba.pca$x)

class.players$km.cluster = nba.pca$cluster

head(class.players)

class.players %>%

  select(-player,-pos) %>%

  group_by(km.cluster) %>%

  summarise_all(mean)







set.seed(5)

num.clusters = 12

cluster_data = P_info_all %>%

  select(-player,-pos) %>%
```

```
  scale()

km.mod = kmeans(cluster_data, centers=num.clusters, iter.max=50)

km.mod




################

### results ####

################

class.players = cbind(P_info_all, nba.pca$x)

class.players$km.cluster = km.mod$cluster


class.players %>% filter(km.cluster == 8)

summary(class.players)


my.cols = c('black', 'blue', 'yellow', 'lightgreen', 'cadetblue2',
            'darkorange', 'forestgreen', 'darkorchid', 'goldenrod', 'red',
'green2', 'lightpink3')



palette(my.cols)



plot(PC2 ~ PC1, data=class.players, col=km.cluster, pch=16, main = "Player
clusters by first two principal components")

pt.labels = ifelse(class.players$g > 70, class.players$player, "")

text(class.players$PC1, class.players$PC2, pt.labels, pos=2, cex=0.5)

cluster.labels = c("1: Shooter",

                   "2: exterior distributor",

                   "3: defensive stopper",

                   "4: offensive hub",

                   "5: size and distance",

                   "6: under the basket",

                   "7: stretch big",
```

```
                    "8: attacking shooter",

                    "9: inside / outside",

                    "10: 3-point specialist",

                    "11: attacking distributor",

                    "12: exterior shooter")

legend(5.4, -1.2, legend=cluster.labels, col=my.cols, pch=16, cex=0.6)
legend("topright", legend=cluster.labels, col=1:num.clusters, pch=16,
cex=0.45)
palette("default")

#
#
#  PC1 size / physical
#  PC2 Quicknes/Ball handling
#  PC3 Ball Catcher
# ---------------------------------- GLOBAL VIEW
-------------------------------------------------------

plot(PC2 ~ PC1, data=class.players, col=km.cluster, pch=16, main =
"Position clusters by first two principal components",
     xlab="PC1: 'size/physicality'", ylab="PC2: 'quickness/ballhandling'")

pt.labels = class.players$pos

text(class.players$PC1, class.players$PC2, pt.labels, pos=2, cex=0.5)

cluster.labels = class.players$pos

legend(5.4, -1.2, legend=cluster.labels, col=my.cols, pch=16, cex=0.6)
palette("default")



plot(PC3 ~ PC2, data=class.players, col=km.cluster, pch=16, main =
"Position clusters by PC2 vs PC3 components",
     xlab="PC1: 'Quick/BallHandling'", ylab="PC3: 'PickPocket'")

pt.labels = class.players$pos

text(class.players$PC1, class.players$PC2, pt.labels, pos=2, cex=0.5)

cluster.labels = class.players$pos

legend(5.4, -1.2, legend=cluster.labels, col=my.cols, pch=16, cex=0.6)
palette("default")

# -------------------------------------1 PF Only
-------------------------------------------------


d<-class.players%>%filter(class.players$pos=="PF")
```

```
plot(PC2 ~ PC1, data=d, col=km.cluster, pch=16, main = " PF by first two
principal components",
     xlab="PC1: 'size/physicality'", ylab="PC2: 'quickness/ballhandling'")

pt.labels = d$player
cluster.labels = d$player

text(d$PC1, d$PC2, pt.labels, pos=2, cex=0.5)
palette("default")

# ---------------------------------------2 CEnter only
-----------------------------------------

d<-class.players%>%filter(class.players$pos=="C")


plot(PC2 ~ PC1, data=d, col=km.cluster, pch=16, main = " C by first two
principal components",
     xlab="PC1: 'size/physicality'", ylab="PC2: 'quickness/ballhandling'")

pt.labels = d$player
cluster.labels = d$player
text(d$PC1, d$PC2, pt.labels, pos=2, cex=0.5)
palette("default")

# ------------------------------------ 3 PG Only
-------------------------------------------------

d<-class.players%>%filter(class.players$pos=="PG")


plot(PC2 ~ PC1, data=d, col=km.cluster, pch=16, main = " PG by first two
principal components",
     xlab="PC1: 'size/physicality'", ylab="PC2: 'quickness/ballhandling'")

pt.labels = d$player
cluster.labels = d$player
text(d$PC1, d$PC2, pt.labels, pos=2, cex=0.5)
palette("default")



# ---------------------------------------- 4 SG only
-------------------------------------------

d<-class.players%>%filter(class.players$pos=="SG")


plot(PC2 ~ PC1, data=d, col=km.cluster, pch=16, main = " SG by first two
principal components",
     xlab="PC1: 'size/physicality'", ylab="PC2: 'quickness/ballhandling'")

pt.labels = d$player
cluster.labels = d$player
text(d$PC1, d$PC2, pt.labels, pos=2, cex=0.5)
palette("default")
```

```
# ------------------------------- 5 SF only
-------------------------------------------------

d<-class.players%>%filter(class.players$pos=="SF")


plot(PC2 ~ PC1, data=d, col=km.cluster, pch=16, main = " SF by first two
principal components",
     xlab="PC1: 'size/physicality'", ylab="PC2: 'quickness/ballhandling'")

pt.labels = d$player
cluster.labels = d$player
text(d$PC1, d$PC2, pt.labels, pos=2, cex=0.5)
palette("default")


#
-------------------------------------------------------------------------------------------


plot(PC3 ~ PC1, data=class.players, col=km.cluster, pch=16, main = "Player
clusters by first two principal components",
     xlab="PC1: 'size/physicality'", ylab="PC3: 'pickpockets'")

pt.labels = ifelse(class.players$g > 70, class.players$player, "")

text(class.players$PC1, class.players$PC2, pt.labels, pos=2, cex=0.5)

cluster.labels = c("1: Shooter",

                   "2: exterior distributor",

                   "3: defensive stopper",

                   "4: offensive hub",

                   "5: size and distance",

                   "6: under the basket",

                   "7: stretch big",

                   "8: attacking shooter",

                   "9: inside / outside",

                   "10: 3-point specialist",

                   "11: attacking distributor",

                   "12: exterior shooter")

legend(5.4, -1.2, legend=cluster.labels, col=my.cols, pch=16, cex=0.6)
legend("topright", legend=cluster.labels, col=1:num.clusters, pch=16,
cex=0.45)
palette("default")
```

#
