NBA analysis

Asim Mohammed Thayyil

## 1. Introduction

Basketball is a sport that has gained significant popularity around the world, and the National Basketball Association (NBA) is one of the most prominent professional basketball leagues in the world. In basketball, each team is made up of five players who are assigned to different positions, namely point guard, shooting guard, small forward, power forward, and center.

The Chicago Bulls is one of the NBA teams that have been struggling in recent seasons, with the team placing 27th out of 30 in the most recent season (2018-19). The team’s budget for player contracts in the upcoming season (2019-20) is also limited, ranked 26th out of 30 teams. As a result, there is a need to optimize player selection to improve the team’s performance while also managing costs.

The aim of this project is to identify the best five starting players for the Chicago Bulls for the upcoming 2019-20 NBA season while staying within the team’s budget of $118 million. To achieve this aim, we will perform a data analysis project that involves identifying the key performance metrics for each basketball position, evaluating the players on the Chicago Bulls team, and analyzing potential external candidates that the team can afford.

This project is important because it can help the Chicago Bulls improve their performance in the upcoming season while also managing costs. By identifying the best starting players for the team, we can maximize the team’s chances of winning games and potentially making it to the playoffs. Additionally, this project can serve as a template for other NBA teams looking to optimize their player selection while managing costs.

## 2. Reading and cleaning the raw data

We first load the the necessary libraries.

*# loading necessary libraries*  
suppressPackageStartupMessages(library(tidyverse))   
suppressPackageStartupMessages(library(ggplot2))   
suppressPackageStartupMessages(library(ggpubr))   
suppressPackageStartupMessages(library(reshape2))

Read the data:

*# load data*  
player\_salary <- read.csv('2018-19\_nba\_player-salaries.csv') *# player salary data*  
player\_stats <- read.csv('2018-19\_nba\_player-statistics.csv') *# player statistics data*  
team\_stats\_1 <- read.csv('2018-19\_nba\_team-statistics\_1.csv') *# team statistics data 1*  
team\_stats\_2 <- read.csv('2018-19\_nba\_team-statistics\_2.csv') *# team statistics data 2*  
team\_payroll <- read.csv('2019-20\_nba\_team-payroll.csv') *# team payroll data*

A brief description of each data set and the variables in it, with a link to the full variable descriptions:

1. **2018-19\_nba\_player-salaries** : This data set contains information about the salaries of NBa players for the session 2018-19 in millions (USD). There are three variables in the data namely, player\_id, player\_name and salary. The terms are self explanatory.

This data is sourced from: hoopshype.com/salaries (<https://hoopshype.com/salaries/>)

1. **2018-19\_nba\_player-statistics.csv** : This data set contains total statistics for individual NBA players during the 2018-19 season. The dataset includes various variables such as player name, position, age, team, games played, games started, minutes played, field goal attempts and percentage, 3-point field goal attempts and percentage, 2-point field goal attempts and percentage, effective field goal percentage, free throw attempts and percentage, offensive and defensive rebounds, total rebounds, assists, steals, blocks, turnovers, personal fouls, and points. The dataset has a total of 708 entries, with one row for each player in the 2018-19 NBA season.

This data is sourced from: basketball-reference.com (<https://www.basketball-reference.com/leagues/NBA_2019_totals.html>)

1. **2018-19\_nba\_team-statistics\_1.csv** : This data set provides miscellaneous statistics for NBA teams during the 2018-19 season. The dataset includes various variables such as team rank, mean age of players, total number of wins and losses, Pythagorean wins and losses, margin of victory, strength of schedule, simple rating system, offensive rating, defensive rating, net rating, pace factor, free throw attempt rate, 3-point attempt rate, true shooting percentage, effective field goal percentage, turnover percentage, offensive rebound percentage, free throws per field goal attempt, and defensive rebound percentage.

This data is sourced from: basketball-reference.com ([https://www.basketball-](about:blank) reference.com/leagues/NBA\_2019.html)

1. **2018-19\_nba\_team-statistics\_2.csv** : This data set is a complementary to the above data set which mentions addition statistics for NBA teams during the session 2018-19. The dataset includes data on team performance in both offensive and defensive categories, such as points scored, field goal percentage, rebounds, blocks, steals, and turnovers. Additionally, it includes more advanced statistics like true shooting percentage, player efficiency rating, and offensive and defensive ratings

This data is sourced from: basketball-reference.com ([https://www.basketball-](about:blank) reference.com/leagues/NBA\_2019.html)

1. **2019-20\_nba\_team-payroll.csv**: This data set contains the team payroll budget for each team in the NBA during the 2018-19 season. The variables in the data set include:

* team\_id: a unique identification number for each team
* team: the name of the NBA team
* salary: the team payroll budget for the 2018-19 season, measured in US dollars

This data is sourced from: hoopshype.com/salaries (<https://hoopshype.com/salaries/>)

**NOTE:** To look at the data dictionary for the above data sets please refer to this link: (<https://uclearn.canberra.edu.au/courses/13262/pages/data-description-reproducible-data-analysis-project>)

## 3. Exploratory analysis

In this section we will explore the data set closely and will perform necessary pre-processing along the way.

### Handling missing values

We will now check if there are any missing values in the data sets.

*# checking for missing values:*  
  
print(paste0('player\_salary : ', sum(is.na(player\_salary))))

## [1] "player\_salary : 0"

print(paste0('team\_payroll : ', sum(is.na(team\_payroll))))

## [1] "team\_payroll : 0"

print(paste0('player\_stats : ', sum(is.na(player\_stats))))

## [1] "player\_stats : 117"

print(paste0('team\_stats\_1 : ', sum(is.na(team\_stats\_1))))

## [1] "team\_stats\_1 : 90"

print(paste0('team\_stats\_2 : ', sum(is.na(team\_stats\_2))))

## [1] "team\_stats\_2 : 0"

We see that the data sets player\_stats have 117 missing values and team\_stats\_1 have 90 missing values. Let's find out in which columns these missing values are present.

*# finding columns with missing values*  
colSums(is.na(player\_stats))

## player\_name Pos Age Tm G GS   
## 0 0 0 0 0 0   
## MP FG FGA FG. X3P X3PA   
## 0 0 0 6 0 0   
## X3P. X2P X2PA X2P. eFG. FT   
## 47 0 0 15 6 0   
## FTA FT. ORB DRB TRB AST   
## 0 43 0 0 0 0   
## STL BLK TOV PF PTS   
## 0 0 0 0 0

we observe that there are missing values in the following columns of player\_stats : FG., X3P., X2P., eFG., FT.

colSums((is.na(team\_stats\_1)))

## ï..Rk Team Age W L PW PL MOV SOS SRS ORtg   
## 0 0 0 0 0 0 0 0 0 0 0   
## DRtg NRtg Pace FTr X3PAr TS. eFG. TOV. ORB. FT.FGA DRB.   
## 0 0 0 0 0 0 0 0 0 0 0   
## X X.1 X.2   
## 30 30 30

we observe that there are missing values in following columns of team\_stats\_1: X, X.1, X.2. All these variables are completely empty.

We will now impute the missing values in the data set with the mean of respective columns

*# removing missing values with the mean of the respective column in player\_stats*  
player\_stats$FG.[is.na(player\_stats$FG.)] <- mean(player\_stats$FG., na.rm = TRUE)  
player\_stats$eFG.[is.na(player\_stats$eFG.)] <- mean(player\_stats$eFG., na.rm = TRUE)  
player\_stats$FT.[is.na(player\_stats$FT.)] <- mean(player\_stats$FT., na.rm = TRUE)  
player\_stats$X3P.[is.na(player\_stats$X3P.)] <- mean(player\_stats$X3P., na.rm = TRUE)  
player\_stats$X2P.[is.na(player\_stats$X2P.)] <- mean(player\_stats$X2P., na.rm = TRUE)  
  
  
*# dropping the missing values in team\_stats\_1 dataframe*  
team\_stats\_1[,c('X', 'X.1', 'X.2')] <- NULL  
  
print(paste0('player\_stats : ', sum(is.na(player\_stats))))

## [1] "player\_stats : 0"

print(paste0('team\_stats\_1 : ', sum(is.na(team\_stats\_1))))

## [1] "team\_stats\_1 : 0"

### Distribution of variables

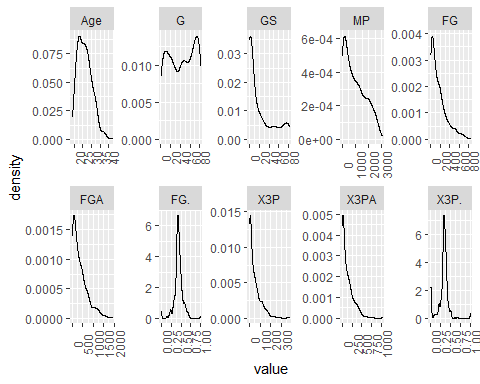
We will now look at the distributions of the features presented in dataframes: player\_stats, team\_stats\_1 and team\_stats\_2.

We will define a function named ‘distribution’ to help in the process of plotting the graphs and reduce the line of codes.

*# function to create density plot for provided columns in dataframe*  
distribution <- **function**(df, cols, ncol, nrow){  
   
 df\_stats <- melt(df[,cols], id.vars = 'index', variable.name = 'columns')  
 p3 <- ggplot(aes(x = value, group = columns), data=df\_stats)  
 p3 + geom\_density() + facet\_wrap(~columns, ncol = ncol, nrow = nrow, scales = 'free') + theme(axis.text.x = element\_text(angle = 90))  
   
}

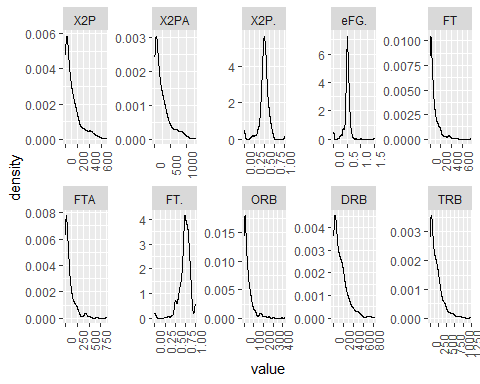
*distribution of features in player statistics:*

*# add an index column to the dataframe*  
player\_stats['index'] <- rownames(player\_stats)  
  
*# plot the distribution of columns: 'Age', 'G', 'GS', 'MP', 'FG', 'FGA', 'FG.','X3P', 'X3PA', 'X3P.'*  
distribution(player\_stats,c('index','Age', 'G', 'GS', 'MP', 'FG', 'FGA', 'FG.','X3P', 'X3PA', 'X3P.'), 5, 2)

 From the plot of distributions above we observe that:

* Most players in the data set are fairly young ranging from 22 to 28 years of age approximately. There are very few players above the age of 30-35.
* There are three groups of players:
  + players who played more than 60 games : These players have a high frequency in the data.
  + players who played less than 30 games : These players have the second most high frequency.
  + players who played 30-60 games.
* Most of the players had a field goals rate of approx 40% .
* Most of the players had a field goal rate on 3-Pt field goal attempts (X3P.) of approx 40%.

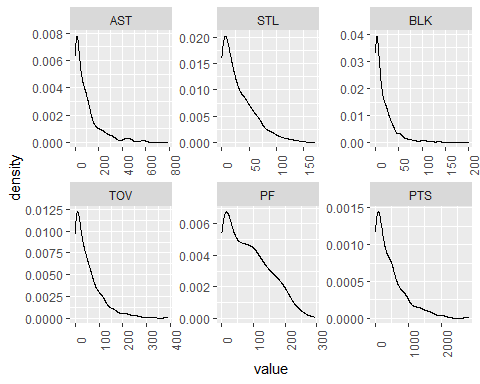
distribution(player\_stats, c('index','X2P', 'X2PA', 'X2P.', 'eFG.', 'FT', 'FTA', 'FT.','ORB', 'DRB', 'TRB'), 5, 2)



From the plot of distributions above we observe that:

* Most of the players had an effective field goals rate of approx 50% .
* Most of the players had field goal rate on 2-Pt field goal attempts (X2P.) of approx 50%.
* Most of the players had a free throw rate of approx 60-80%.
* Most of the players had Total rebounds of less than 250.

distribution(player\_stats,c('index','AST', 'STL', 'BLK', 'TOV', 'PF', 'PTS'), 3, 2)

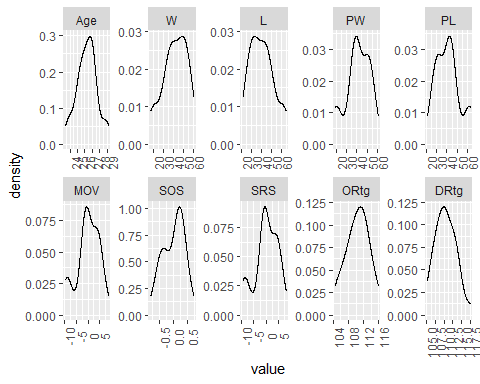


From the plot of distribution above we observe that:

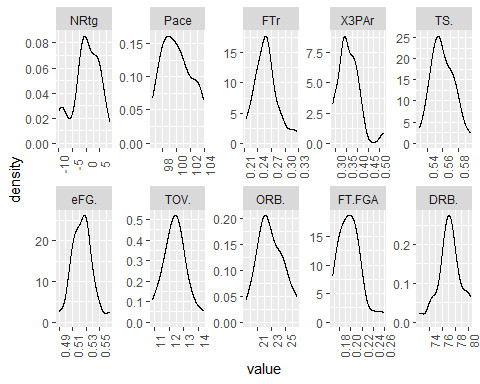
* Most of the players had an assist of less than 150.
* Most of the players had steals of less than 50.
* Most of the players had blocks of less than 25.
* Most of the players had turnovers of less than 75.
* Most of the players had up to 150 personal fouls.

*distribution of features in Team statistics 1*:

team\_stats\_1['index'] <- rownames(team\_stats\_1)  
distribution(team\_stats\_1, c('index', 'Age', 'W', 'L', 'PW', 'PL', 'MOV', 'SOS', 'SRS', 'ORtg', 'DRtg'), 5, 2)

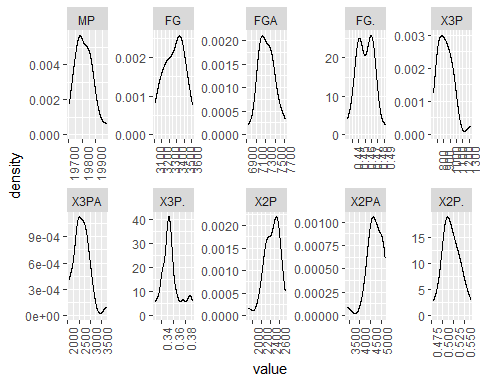


distribution(team\_stats\_1, c('index', 'NRtg', 'Pace', 'FTr', 'X3PAr', 'TS.', 'eFG.', 'TOV.', 'ORB.', 'FT.FGA', 'DRB.'), 5, 2)

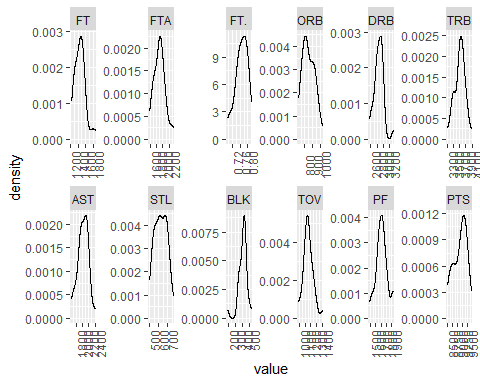


distribution of features in team stats 2:

team\_stats\_2['index'] <- rownames(team\_stats\_2)  
  
distribution(team\_stats\_2, c('index', 'MP', 'FG', 'FGA', 'FG.', 'X3P', 'X3PA', 'X3P.', 'X2P', 'X2PA', 'X2P.'), 5, 2)



distribution(team\_stats\_2, c('index', 'FT', 'FTA', 'FT.', 'ORB', 'DRB', 'TRB', 'AST', 'STL', 'BLK', 'TOV','PF', 'PTS'), 6, 2)



## Player wise analysis:

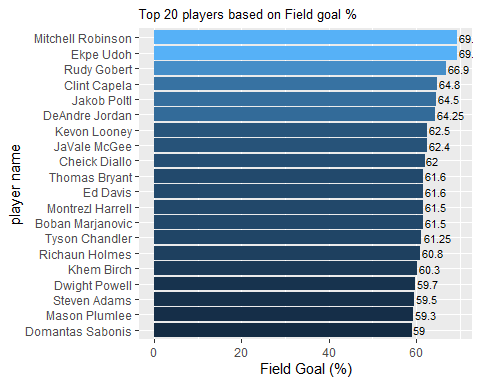
In this section we will look at top and bottom 20 players based on various features in the player\_stats dataframe.

While performing this analysis we keep in mind that there are multiple rows for one player in the data set, hence we choose to group the data by player\_name and summarize it by taking means of the feature of interest.

For the analysis to be backed by enough data we will only consider those players who have played at least 40 games in the season.

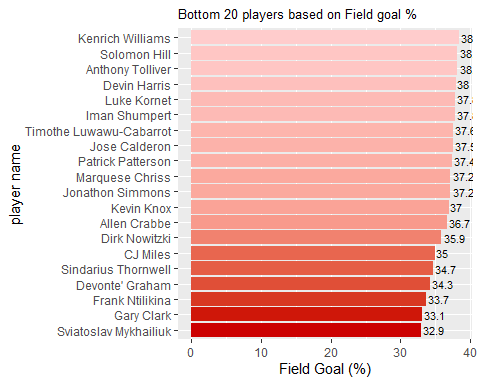
### Top and bottom 20 players based on Field Goal percentage:

*# multiplying by 100 to convert into more readable percentage form in the graphs*  
player\_stats['FG.'] <- player\_stats['FG.'] \* 100  
  
n = 20  
  
*# top n*  
  
p1.1 <- player\_stats%>%filter( G >= 40)%>%group\_by(player\_name)%>%summarise(FG. = mean(FG.))%>%arrange(-FG.)%>%head(n)%>%ggplot(aes(reorder(player\_name, FG.), FG., fill = FG.)) +   
 geom\_bar(stat="identity") +   
 labs(x = 'player name', y = 'Field Goal (%)', title = 'Top 20 players based on Field goal %') +   
 theme(plot.title = element\_text(size = 10),legend.position = "none") +  
 geom\_text(aes(player\_name, FG.,label = round(FG.,2) ), size = 3, color = 'black', hjust = -0.1) +   
 coord\_flip()  
  
p1.1



The above graph represents the analysis of the top 20 players (who have played more than 40 games) based on their Field Goal percentages. Upon examining the graph, it can be observed that Mitchell Robinson and Ekpe Udoh have the highest field goal percentages of 69%, making them the top two players.

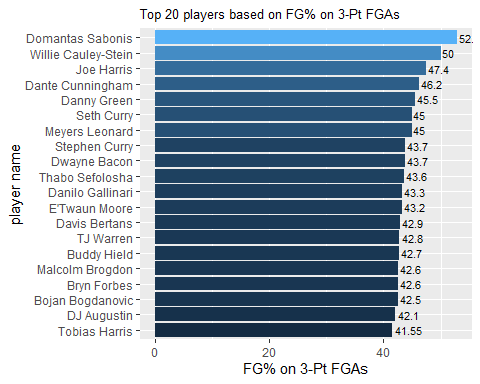
*# bottom n*  
  
p1.2 <- player\_stats%>%filter( G >= 40)%>%group\_by(player\_name)%>%summarise(FG. = mean(FG.))%>% arrange(-FG.)%>% tail(n)%>%ggplot(aes(reorder(player\_name, FG.), FG., fill = FG.)) + geom\_bar(stat="identity") +  
 labs(x = 'player name', y = 'Field Goal (%)', title = 'Bottom 20 players based on Field goal %') +  
 theme(plot.title = element\_text(size = 10),legend.position = "none")+  
 scale\_fill\_gradient(low = '#CC0000', high= '#FFCCCC') +  
 geom\_text(aes(player\_name, FG.,label = round(FG.,2) ), size = 3, color = 'black', hjust = -0.1) +  
 coord\_flip()  
  
p1.2



The above graph represents the analysis of the bottom 20 players (who have played more than 40 games) based on their Field Goal percentages. Upon examining the graph, it can be observed that Sviatoslav Mykhailiuk, Gary Clark and Frank Ntilikina are the lowest performing players.

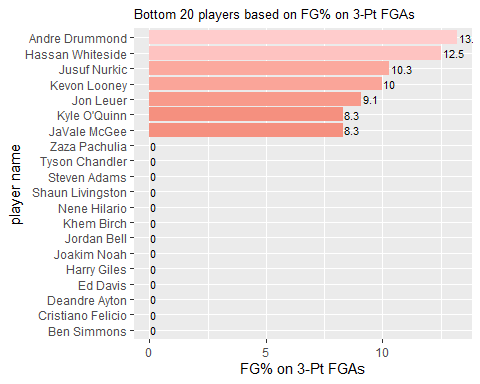
### Top and bottom 20 players based on FG% on 3-Pt FGAs

*# multiplying by 100 to convert into more readable percentage form in the graphs*  
player\_stats['X3P.'] <- player\_stats$X3P. \* 100  
  
  
*# top n*  
p2.1 <- player\_stats%>%filter(G >= 40)%>%group\_by(player\_name)%>% summarise(X3P. = mean(X3P.))%>%arrange(-X3P.)%>%head(n)%>%  
 ggplot(aes(reorder(player\_name, X3P.), X3P., fill = X3P.)) + geom\_bar(stat="identity") +  
 labs( x = 'player name',y = 'FG% on 3-Pt FGAs', title = 'Top 20 players based on FG% on 3-Pt FGAs') +  
 theme(plot.title = element\_text(size = 10),legend.position = "none") +  
 geom\_text(aes(player\_name, X3P.,label = round(X3P.,2) ), size = 3, color = 'black', hjust = -0.1) +  
 coord\_flip()  
  
p2.1



The above graph shows the top 20 players based on their Field Goal percentage on 3-Point Field Goal Attempts. The top player is Domantas Sabonis with a rate of 52.9% followed by Willie Cauley-Stein with a rate of 50%.

*# bottom n*  
  
p2.2 <- player\_stats%>%filter(G >= 40)%>%group\_by(player\_name)%>% summarise(X3P. = mean(X3P.))%>%arrange(-X3P.)%>%tail(n)%>%  
 ggplot(aes(reorder(player\_name, X3P.), X3P., fill = X3P.)) + geom\_bar(stat="identity") +  
 labs(x = 'player name', y = 'FG% on 3-Pt FGAs', title = 'Bottom 20 players based on FG% on 3-Pt FGAs') +  
 theme(plot.title = element\_text(size = 10),legend.position = "none") +  
 scale\_fill\_gradient(low = '#CC0000', high= '#FFCCCC') +  
 geom\_text(aes(player\_name, X3P.,label = round(X3P.,2) ), size = 3, color = 'black', hjust = -0.1) +  
 coord\_flip()  
  
p2.2

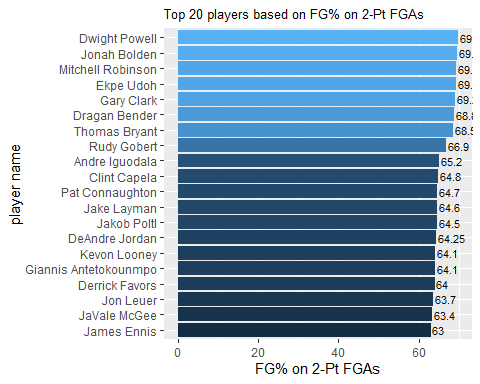


*# display*  
*#ggarrange(p1.1, p1.2, row = 2)*

The above plot shows that a no. of players have almost 0% X3P. We may say that Kylie O’Quinn and JaVale McGee are the lowest performers based on X3P.

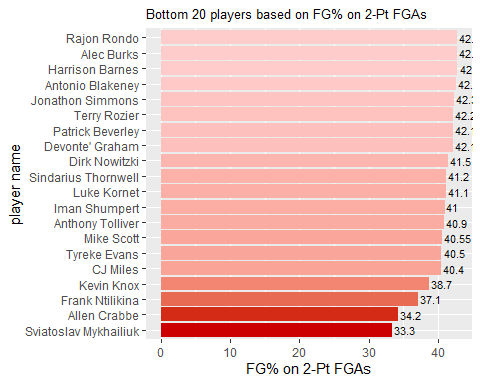
### Top and bottom 20 players based on FG% on 3-Pt FGAs

*# multiplying by 100 to convert into more readable percentage form in the graphs*  
player\_stats['X2P.'] <- player\_stats$X2P. \* 100  
  
  
*# top n*  
p3.1 <- player\_stats%>%filter( G >= 40 )%>%group\_by(player\_name)%>% summarise(X2P. = mean(X2P.))%>%arrange(-X2P.)%>%head(n)%>%  
 ggplot( aes(reorder(player\_name,X2P.), X2P., fill = X2P.)) + geom\_bar(stat="identity") +  
 labs(x = 'player name', y = 'FG% on 2-Pt FGAs', title = 'Top 20 players based on FG% on 2-Pt FGAs') +  
 theme(plot.title = element\_text(size = 10),legend.position = "none") +  
 geom\_text(aes(player\_name, X2P.,label = round(X2P.,2) ), size = 3, color = 'black', hjust = -0.1) +  
 coord\_flip()  
  
p3.1



The above graph shows the top 20 players based on their Field Goal percentage on 2-Point Field Goal Attempts. The top player is Dwight Powell with a rate of 69.9% followed by Jonah Bolden with a rate of 69.7%.

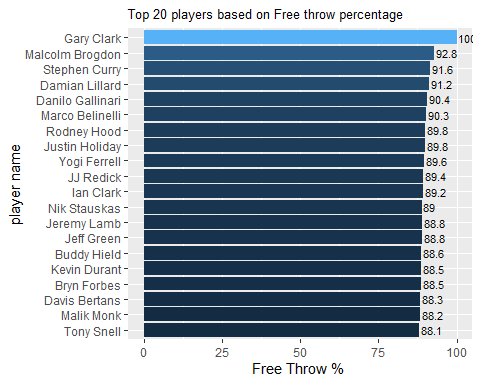
*# bottom n*  
  
p3.2 <- player\_stats%>%filter( G >= 40)%>%group\_by(player\_name)%>% summarise(X2P. = mean(X2P.))%>%arrange(-X2P.)%>%tail(n)%>%  
 ggplot(aes(reorder(player\_name,X2P.), X2P., fill = X2P.)) + geom\_bar(stat="identity") +  
 labs(x = 'player name', y = 'FG% on 2-Pt FGAs', title = 'Bottom 20 players based on FG% on 2-Pt FGAs') +  
 theme(plot.title = element\_text(size = 10),legend.position = "none") +  
 scale\_fill\_gradient(low = '#CC0000', high= '#FFCCCC') +  
 geom\_text(aes(player\_name, X2P.,label = round(X2P.,2) ), size = 3, color = 'black', hjust = -0.1) +  
 coord\_flip()  
  
p3.2



The above graph shows the bottom 20 players based on their Field Goal percentage on 2-Point Field Goal Attempts. The bottom player is Sviatoslav MykhailiuK with a rate of 33.3% followed by Allen Crabbe with a rate of 34.2%.

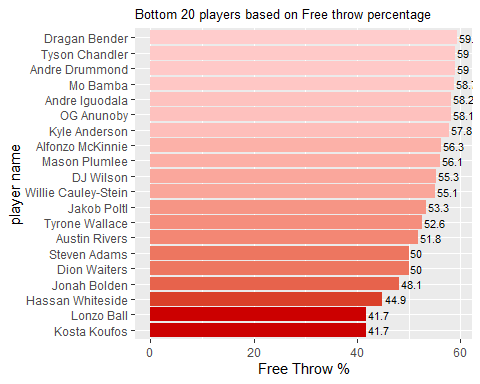
### Top and bottom 20 players based on Free Throw Percentage

*# multiplying by 100 to convert into more readable percentage form in the graphs*  
player\_stats['FT.'] <- player\_stats$FT. \* 100  
  
  
*# top n*  
p4.1 <- player\_stats%>%filter( G >= 40)%>%group\_by(player\_name)%>% summarise(FT. = mean(FT.))%>%arrange(-FT.)%>%head(n)%>%  
 ggplot(aes(reorder(player\_name, FT.), FT., fill = FT.)) + geom\_bar(stat="identity") +  
 labs(x = 'player name', y = 'Free Throw % ', title = 'Top 20 players based on Free throw percentage') +  
 theme(plot.title = element\_text(size = 10),legend.position = "none") +  
 geom\_text(aes(player\_name, FT.,label = round(FT.,2) ), size = 3, color = 'black', hjust = -0.1) +  
 coord\_flip()  
  
p4.1



The above graph shows the top 20 players based on their free throw percentage. The top player is Gary Clark with a rate of 100% followed by Malcolm Brogdon with a rate of 92.8%.

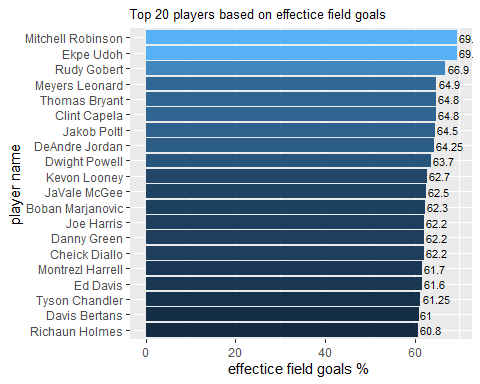
*# bottom n*  
  
p3.2 <- player\_stats%>%filter(G >= 40)%>%group\_by(player\_name)%>% summarise(FT. = mean(FT.))%>%arrange(-FT.)%>%tail(n)%>%  
 ggplot(aes(reorder(player\_name, FT.), FT., fill = FT.)) + geom\_bar(stat="identity") +  
 labs(x = 'player name', y = 'Free Throw %', title = 'Bottom 20 players based on Free throw percentage') +  
 theme(plot.title = element\_text(size = 10),legend.position = "none") +  
 scale\_fill\_gradient(low = '#CC0000', high= '#FFCCCC') +  
 geom\_text(aes(player\_name, FT.,label = round(FT.,2) ), size = 3, color = 'black', hjust = -0.1) +  
 coord\_flip()  
  
p3.2



The above graph shows the bottom 20 players based on their free throw percentage. The bottom most players are Kosta Koufos and Lonzo Ball with a rate of 41.7%.

### Top and bottom 20 players based on Effectice Field Goals

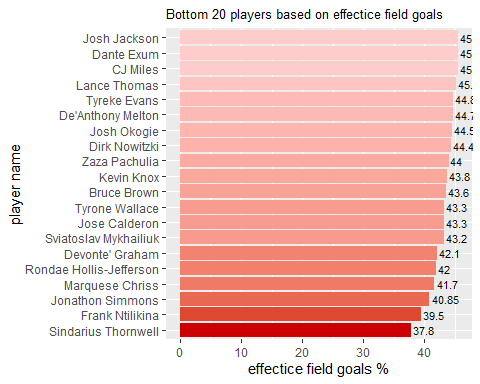
*# multiplying by 100 to convert into more readable percentage form in the graphs*  
player\_stats['eFG.'] <- player\_stats$eFG. \* 100  
  
*# top n*  
p5.1 <- player\_stats%>%filter( G >= 40)%>%group\_by(player\_name)%>% summarise(eFG. = mean(eFG.))%>%arrange(-eFG.)%>%head(n)%>%  
 ggplot(aes(reorder(player\_name,eFG.), eFG., fill = eFG.)) + geom\_bar(stat="identity") +  
 labs( x = 'player name', y = 'effectice field goals % ', title = 'Top 20 players based on effectice field goals') +  
 theme(plot.title = element\_text(size = 10),legend.position = "none") +  
 geom\_text(aes(player\_name, eFG.,label = round(eFG.,2) ), size = 3, color = 'black', hjust = -0.1) +  
 coord\_flip()  
  
p5.1



The above plot shows the top 20 players based on Effective Field Goals, the highest performing player comes out to be Mitchell Robinson and Ekpe Udoh with a rate of 69.4%. They were the top performers based on field goals percentage as well.

Effective Field Goals take into account the value of three-point shots, giving players who make more three-pointers a higher effective field goal percentage.

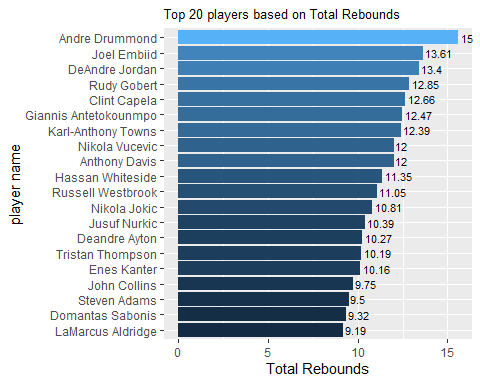
*# bottom n*  
  
p5.2 <- player\_stats%>%filter( G >= 40)%>%group\_by(player\_name)%>% summarise(eFG. = mean(eFG.))%>%arrange(-eFG.)%>%tail(n)%>%  
 ggplot(aes(reorder(player\_name,eFG.), eFG., fill = eFG.)) + geom\_bar(stat="identity") +  
 labs(x = 'player name', y = 'effectice field goals %', title = 'Bottom 20 players based on effectice field goals') +  
 theme(plot.title = element\_text(size = 10),legend.position = "none") +  
 scale\_fill\_gradient(low = '#CC0000', high= '#FFCCCC') +  
 geom\_text(aes(player\_name, eFG.,label = round(eFG.,2) ), size = 3, color = 'black', hjust = -0.1) +  
 coord\_flip()  
  
p5.2



The lowest performing player as per effective field goal is Sindarius Thornwell with a rate of 37.8% followed by Frank Ntilikina 39.5%.

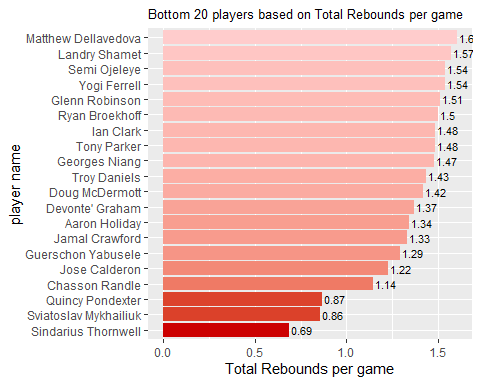
### Top and bottom 20 players based on Total Rebounds

*# top n*  
p6.1 <- player\_stats%>%filter( G >= 40)%>%group\_by(player\_name)%>% summarise(TRB = mean(TRB/G))%>%arrange(-TRB)%>%head(n)%>%  
 ggplot(aes(reorder(player\_name,TRB), TRB, fill = TRB)) + geom\_bar(stat="identity") +  
 labs(x = 'player name', y = 'Total Rebounds', title = 'Top 20 players based on Total Rebounds') +  
 theme(plot.title = element\_text(size = 10),legend.position = "none") +  
 geom\_text(aes(player\_name, TRB,label = round(TRB,2) ), size = 3, color = 'black', hjust = -0.1) +  
 coord\_flip()  
  
p6.1



Above graph shows the top 20 players based on total rebounds. Andre Drummond stands out with the highest total rebounds of 15.5 followed by Joel Embiid with TRB of 13.61.

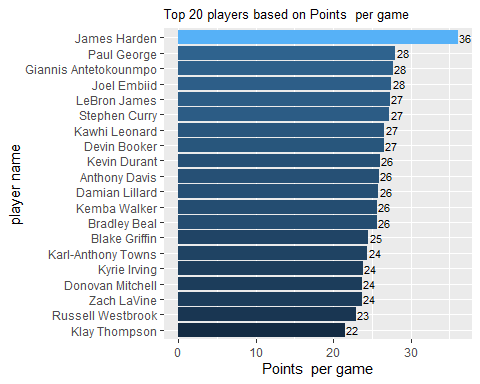
*# bottom n*  
  
p6.2 <- player\_stats%>%filter( G >= 40)%>%group\_by(player\_name)%>% summarise(TRB = mean(TRB/G))%>%arrange(-TRB)%>%tail(n)%>%  
 ggplot(aes(reorder(player\_name,TRB), TRB, fill = TRB)) + geom\_bar(stat="identity") +  
 labs(x = 'player name', y = 'Total Rebounds per game', title = 'Bottom 20 players based on Total Rebounds per game') + theme(plot.title = element\_text(size = 10),legend.position = "none") +  
 scale\_fill\_gradient(low = '#CC0000', high= '#FFCCCC') +  
 geom\_text(aes(player\_name, TRB,label = round(TRB,2) ), size = 3, color = 'black', hjust = -0.1) +  
 coord\_flip()  
  
p6.2



Based on the above graph we can say that Sindarius Thornwell had the least TRB of 0.69 followed by Sviatoslav Mykhailiuk with a TRB of 0.86.

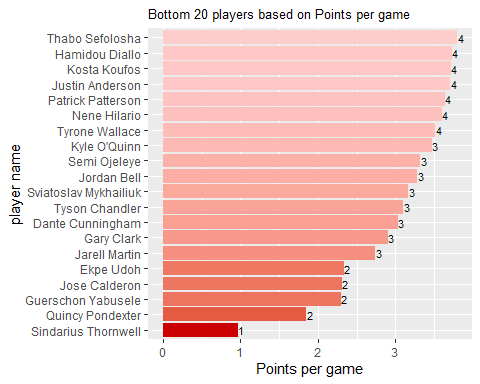
### Top and bottom 20 players based on Points per game

n = 20  
  
*# top n*  
p7.1 <- player\_stats%>%filter( G >= 40)%>%group\_by(player\_name)%>% summarise(PTS = mean(PTS/G))%>%arrange(-PTS)%>%head(n)%>%  
 ggplot(aes(reorder(player\_name,PTS), PTS, fill = PTS)) + geom\_bar(stat="identity") +  
 labs(x = 'player name', y = 'Points per game', title = 'Top 20 players based on Points per game') +  
 theme(plot.title = element\_text(size = 10),legend.position = "none") +  
 geom\_text(aes(player\_name, PTS,label = round(PTS) ), size = 3, color = 'black', hjust = -0.1) +  
 coord\_flip()  
  
p7.1



According to the plotted graph, the top 20 players based on points per game have been identified. James Harden has the highest points per game with a score of 36.

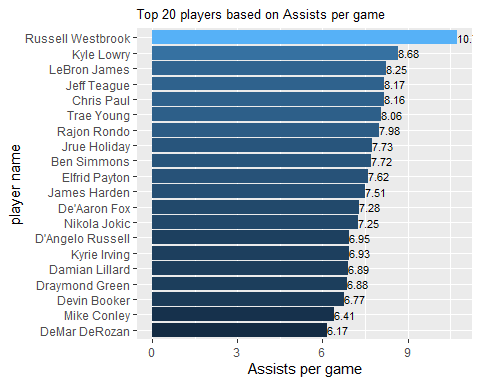
*# bottom n*  
  
p7.2 <- player\_stats%>%filter( G >= 40)%>%group\_by(player\_name)%>% summarise(PTS = mean(PTS/G))%>%arrange(-PTS)%>%tail(n)%>%  
 ggplot(aes(reorder(player\_name, PTS), PTS, fill = PTS)) + geom\_bar(stat="identity") +  
 labs(x = 'player name', y = 'Points per game', title = 'Bottom 20 players based on Points per game') +  
 theme(plot.title = element\_text(size = 10),legend.position = "none") +  
 scale\_fill\_gradient(low = '#CC0000', high= '#FFCCCC') +  
 geom\_text(aes(player\_name, PTS,label = round(PTS) ), size = 3, color = 'black', hjust = -0.1) +  
 coord\_flip()  
  
p7.2



Based on the above graph Sindarius Thornwell performed the lowest based on points per game.

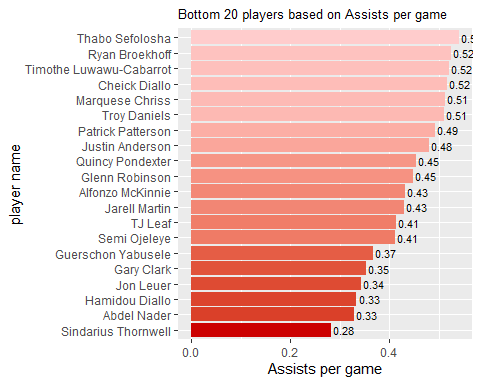
### Top and bottom 20 players based on Assists per game

n = 20  
  
*# top n*  
p8.1 <- player\_stats%>%filter( G >= 40)%>%group\_by(player\_name)%>% summarise(AST = mean(AST/G))%>%arrange(-AST)%>%head(n)%>%  
 ggplot(aes(reorder(player\_name, AST), AST, fill = AST)) + geom\_bar(stat="identity") +  
 labs(x = 'player name', y = 'Assists per game', title = 'Top 20 players based on Assists per game') +  
 theme(plot.title = element\_text(size = 10),legend.position = "none") +  
 geom\_text(aes(player\_name, AST,label = round(AST,2) ), size = 3, color = 'black', hjust = 0) +  
 coord\_flip()  
  
p8.1



According to the graph, the top 20 players based on assists per game are led by Russell Westbrook, who has the highest number of assists per game.

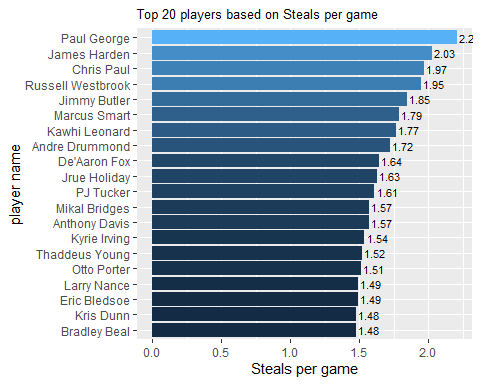
*# bottom n*  
  
p8.2 <- player\_stats%>%filter( G >= 40)%>%group\_by(player\_name)%>% summarise(AST = mean(AST/G))%>%arrange(-AST)%>%tail(n)%>%  
 ggplot(aes(reorder(player\_name, AST), AST, fill = AST)) + geom\_bar(stat="identity") +  
 labs(x = 'player name', y = 'Assists per game', title = 'Bottom 20 players based on Assists per game') +  
 theme(plot.title = element\_text(size = 10),legend.position = "none") +  
 scale\_fill\_gradient(low = '#CC0000', high= '#FFCCCC') +  
 geom\_text(aes(player\_name, AST,label = round(AST,2) ), size = 3, color = 'black', hjust = -0.1) +  
 coord\_flip()  
  
p8.2



Based on the above graph Sindarius Thornwell performed the lowest based on assists per game.

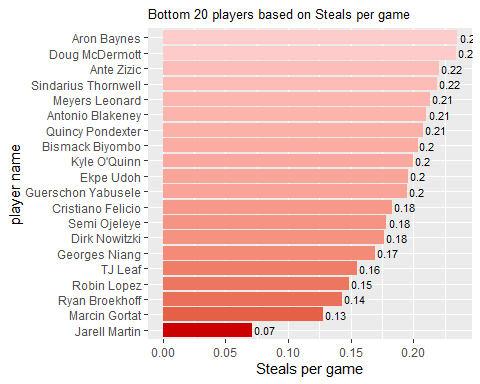
### Top and bottom 20 players based on Steals per game

n = 20  
  
*# top n*  
p9.1 <- player\_stats%>%filter( G >= 40 )%>%group\_by(player\_name)%>% summarise(STL = mean(STL/G))%>%arrange(-STL)%>%head(n)%>%  
 ggplot(aes(reorder(player\_name, STL), STL, fill = STL)) + geom\_bar(stat="identity") +  
 labs(x = 'player name', y = 'Steals per game', title = 'Top 20 players based on Steals per game') +  
 theme(plot.title = element\_text(size = 10),legend.position = "none") +  
 geom\_text(aes(player\_name, STL,label = round(STL,2) ), size = 3, color = 'black', hjust = -0.1) +  
 coord\_flip()  
  
p9.1



According to the below graph, Paul George has the highest number of steals per game among all players.

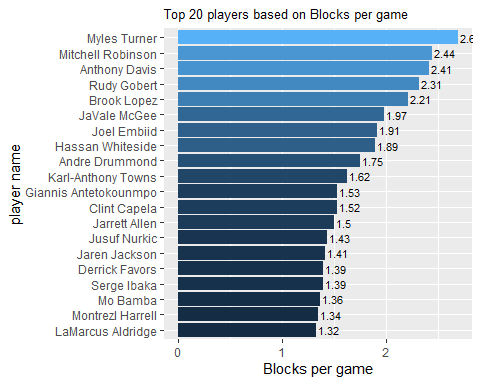
*# bottom n*  
  
p9.2 <- player\_stats%>%filter( G >= 40)%>%group\_by(player\_name)%>% summarise(STL = mean(STL/G))%>%arrange(-STL)%>%tail(n)%>%  
 ggplot(aes(reorder(player\_name,STL), STL, fill = STL)) + geom\_bar(stat="identity") +  
 labs(x = 'player name', y = 'Steals per game', title = 'Bottom 20 players based on Steals per game') +  
 theme(plot.title = element\_text(size = 10),legend.position = "none") +  
 scale\_fill\_gradient(low = '#CC0000', high= '#FFCCCC') +  
 geom\_text(aes(player\_name, STL,label = round(STL,2) ), size = 3, color = 'black', hjust = -0.1) +  
 coord\_flip()  
  
p9.2



Based on the above graph Jarell Martin performed the lowest based on steals per game.

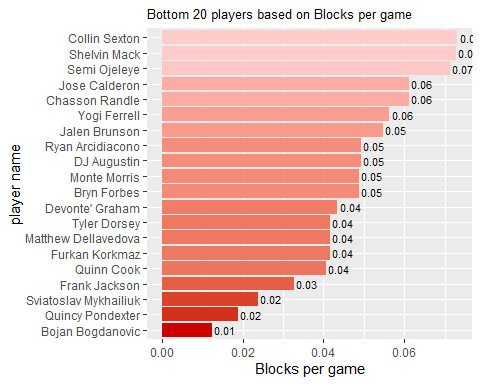
### Top and bottom 20 players based on Blocks per game

n = 20  
  
*# top n*  
p10.1 <- player\_stats%>%filter( G >= 40)%>%group\_by(player\_name)%>% summarise(BLK = mean(BLK/G))%>%arrange(-BLK)%>%head(n)%>%  
 ggplot(aes(reorder(player\_name,BLK), BLK, fill = BLK)) + geom\_bar(stat="identity") +  
 labs(x = 'player name', y = 'Blocks per game', title = 'Top 20 players based on Blocks per game') +  
 theme(plot.title = element\_text(size = 10),legend.position = "none") +  
 geom\_text(aes(player\_name, BLK,label = round(BLK,2) ), size = 3, color = 'black', hjust = -0.1) +  
 coord\_flip()  
  
p10.1



Above graph shows the top 20 players based on Blocks per game and they are led by Myles Turner, who has the highest number of Blocks per game among all players.

*# bottom n*  
  
p10.2 <- player\_stats%>%filter( G >= 40)%>%group\_by(player\_name)%>% summarise(BLK = mean(BLK/G))%>%arrange(-BLK)%>%tail(n)%>%  
 ggplot(aes(reorder(player\_name, BLK), BLK, fill = BLK)) + geom\_bar(stat="identity") +  
 labs( x = 'player name',y = 'Blocks per game', title = 'Bottom 20 players based on Blocks per game') +  
 theme(plot.title = element\_text(size = 10),legend.position = "none") +  
 scale\_fill\_gradient(low = '#CC0000', high= '#FFCCCC') +  
 geom\_text(aes(player\_name, BLK,label = round(BLK,2) ), size = 3, color = 'black', hjust = -0.1) +  
 coord\_flip()  
  
p10.2

 Based on the above graph Bojan Bogdanovic performed the lowest based on steals per game.

## Player analysis based on positions:

### PG ( Point Guard)

The point guard (PG) is typically the team’s primary ball-handler, responsible for bringing the ball up the court and initiating the team’s offensive plays. They are usually the shortest players on the court, but also one of the fastest and most agile.

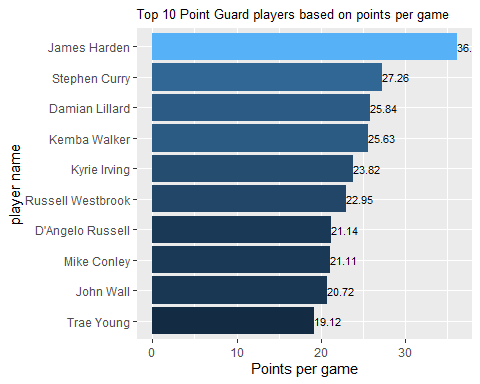
Top 10 point guard players based on points:

We first filter the Point guards players in the data and then make the plot.

PG\_df <- player\_stats%>%filter(Pos == 'PG')%>%group\_by(player\_name)%>%summarise(PTS = mean(PTS/G))%>% arrange(-PTS)%>%head(10)  
  
PG\_df

## # A tibble: 10 x 2  
## player\_name PTS  
## <chr> <dbl>  
## 1 James Harden 36.1  
## 2 Stephen Curry 27.3  
## 3 Damian Lillard 25.8  
## 4 Kemba Walker 25.6  
## 5 Kyrie Irving 23.8  
## 6 Russell Westbrook 22.9  
## 7 D'Angelo Russell 21.1  
## 8 Mike Conley 21.1  
## 9 John Wall 20.7  
## 10 Trae Young 19.1

PG\_df%>%ggplot(aes(reorder(player\_name, PTS), PTS, fill = PTS)) + geom\_bar(stat="identity") +  
 labs( x = 'player name',y = 'Points per game', title = 'Top 10 Point Guard players based on points per game') + theme(plot.title = element\_text(size = 10),legend.position = "none") +  
 geom\_text(aes(player\_name, PTS,label = round(PTS,2) ), size = 3, color = 'black', hjust = 0) +  
 coord\_flip()



Top 10 point guard players based on assist to turnover ratio:

We first filter out those players who play at point guard positions with an Assist of more than 7 and who have played more than 40 games. We then calculate the Assist to turnover ratio for them in order to rank them.

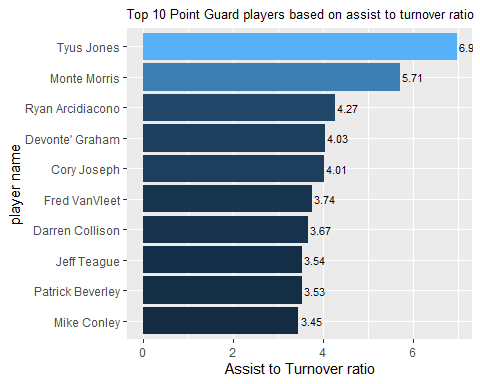
*# top 10 PG players with highest assist to turnover ratio*  
PG\_df2 <- player\_stats %>%filter(Pos == 'PG' & AST > 7.1 & G >= 40)%>%group\_by(player\_name)%>%summarise(AST\_TOV = AST / TOV)%>% arrange(-AST\_TOV)%>%head(10)

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

PG\_df2

## # A tibble: 10 x 2  
## # Groups: player\_name [10]  
## player\_name AST\_TOV  
## <chr> <dbl>  
## 1 Tyus Jones 6.96  
## 2 Monte Morris 5.71  
## 3 Ryan Arcidiacono 4.27  
## 4 Devonte' Graham 4.03  
## 5 Cory Joseph 4.01  
## 6 Fred VanVleet 3.74  
## 7 Darren Collison 3.67  
## 8 Jeff Teague 3.54  
## 9 Patrick Beverley 3.53  
## 10 Mike Conley 3.45

PG\_df2%>% ggplot(aes(reorder(player\_name,AST\_TOV), AST\_TOV, fill = AST\_TOV)) + geom\_bar(stat="identity") +  
 labs( x = 'player name',y = 'Assist to Turnover ratio', title = 'Top 10 Point Guard players based on assist to turnover ratio') + theme(plot.title = element\_text(size = 10),legend.position = "none") +  
 geom\_text(aes(player\_name, AST\_TOV,label = round(AST\_TOV,2) ), size = 3, color = 'black', hjust = -0.1) +  
 coord\_flip()



### SG (Shooting Guard)

The shooting guard (SG) is responsible for scoring points and is often the team’s best outside shooter. They are typically taller than the point guard but not as tall as the small forward.

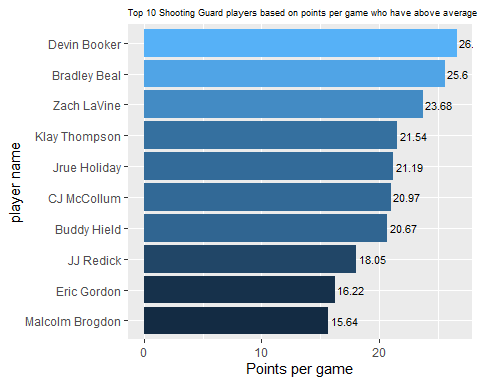
Top 10 Shooting Guard players based on points:

We first filter the Shooting guards players in the data and then make the plot.

SG\_df <- player\_stats%>%filter(Pos == 'SG')%>%group\_by(player\_name)%>%summarise(X3P. = mean(X3P.), X2P. = mean(X2P.), FT. = mean(FT.), G = sum(G), PTS = mean(PTS/G))  
   
SG\_df2 <- SG\_df%>%filter(X3P. > mean(SG\_df$X3P.), X2P. > mean(SG\_df$X2P.), FT. > mean(SG\_df$FT.), G >= 40 )%>% arrange(-PTS)%>%head(10)  
  
SG\_df2

## # A tibble: 10 x 6  
## player\_name X3P. X2P. FT. G PTS  
## <chr> <dbl> <dbl> <dbl> <int> <dbl>  
## 1 Devin Booker 32.6 53.6 86.6 64 26.6  
## 2 Bradley Beal 35.1 54.8 80.8 82 25.6  
## 3 Zach LaVine 37.4 50.4 83.2 63 23.7  
## 4 Klay Thompson 40.2 51.6 81.6 78 21.5  
## 5 Jrue Holiday 32.5 53.9 76.8 67 21.2  
## 6 CJ McCollum 37.5 50.6 82.8 70 21.0  
## 7 Buddy Hield 42.7 48.7 88.6 82 20.7  
## 8 JJ Redick 39.7 50.2 89.4 76 18.1  
## 9 Eric Gordon 36 49.7 78.3 68 16.2  
## 10 Malcolm Brogdon 42.6 54.4 92.8 64 15.6

SG\_df2%>%ggplot(aes(reorder(player\_name, PTS),PTS, fill = PTS)) + geom\_bar(stat="identity") +  
 labs( x = 'player name',y = 'Points per game', title = 'Top 10 Shooting Guard players based on points per game who have above average 3 and 2 point goal attempts and free throws') +  
 geom\_text(aes(player\_name, PTS,label = round(PTS,2) ), size = 3, color = 'black', hjust = -0.1) + theme(plot.title = element\_text(size = 7.1), legend.position = "none") +  
 coord\_flip()



### PF (Power Forward)

The power forward (PF) is usually the team’s best rebounder and inside scorer. They are typically taller and stronger than the small forward, but not as tall as the center.

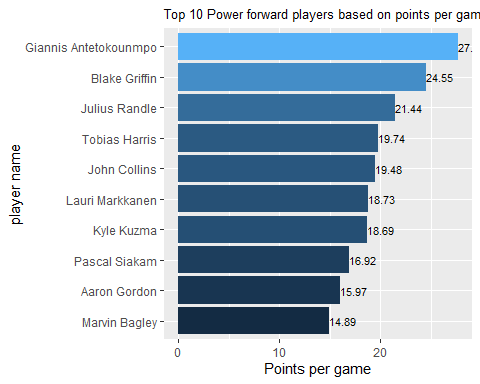
Top 10 power forward players based on points:

We first filter the power forward players in the data and then make the plot.

PF\_df <- player\_stats%>%filter(Pos == 'PF')%>%group\_by(player\_name)%>%summarise(PTS = mean(PTS/G), TRB = mean(TRB), BLK = mean(BLK), G = sum(G))  
  
PF\_df2 <- PF\_df%>%filter(TRB > mean(PF\_df$TRB), BLK > mean(PF\_df$BLK), PTS > mean(PF\_df$PTS) , G >= 40)%>% arrange(-PTS)%>% head(10)  
  
PF\_df2

## # A tibble: 10 x 5  
## player\_name PTS TRB BLK G  
## <chr> <dbl> <dbl> <dbl> <int>  
## 1 Giannis Antetokounmpo 27.7 898 110 72  
## 2 Blake Griffin 24.5 565 28 75  
## 3 Julius Randle 21.4 634 45 73  
## 4 Tobias Harris 19.7 430 24.7 164  
## 5 John Collins 19.5 595 39 61  
## 6 Lauri Markkanen 18.7 470 33 52  
## 7 Kyle Kuzma 18.7 382 26 70  
## 8 Pascal Siakam 16.9 549 52 80  
## 9 Aaron Gordon 16.0 574 56 78  
## 10 Marvin Bagley 14.9 471 59 62

PF\_df2%>% ggplot(aes(reorder(player\_name,PTS), PTS, fill = PTS)) + geom\_bar(stat="identity")+  
 labs( x = 'player name', y = 'Points per game', title = 'Top 10 Power forward players based on points per game') + geom\_text(aes(player\_name, PTS,label = round(PTS,2) ), size = 3, color = 'black', hjust = 0) +   
 theme(plot.title = element\_text(size = 10),legend.position = "none") +  
 coord\_flip()



### C (Center)

Finally, the center (C) is usually the tallest player on the team and is responsible for playing close to the basket on both offense and defense. Their main responsibilities include rebounding, blocking shots, and scoring close to the basket. The center is often considered the anchor of the team’s defense, and their size and strength make them a dominant presence on the court.

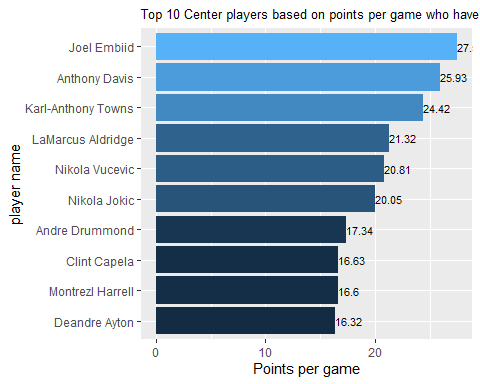
Top 10 center players based on points:

We first filter the Center forward players in the data and then make the plots.

C\_df <- player\_stats%>%filter(Pos == 'C')%>%group\_by(player\_name)%>%summarise(PTS = mean(PTS/G), TRB = mean(TRB), BLK = mean(BLK), G = sum(G))  
  
C\_df2 <- C\_df%>%filter(TRB > mean(PF\_df$TRB), BLK > mean(PF\_df$BLK), PTS > mean(PF\_df$PTS) , G >= 40)%>%arrange(-PTS)%>% head(10)  
  
C\_df2

## # A tibble: 10 x 5  
## player\_name PTS TRB BLK G  
## <chr> <dbl> <dbl> <dbl> <int>  
## 1 Joel Embiid 27.5 871 122 64  
## 2 Anthony Davis 25.9 672 135 56  
## 3 Karl-Anthony Towns 24.4 954 125 77  
## 4 LaMarcus Aldridge 21.3 744 107 81  
## 5 Nikola Vucevic 20.8 960 89 80  
## 6 Nikola Jokic 20.0 865 55 80  
## 7 Andre Drummond 17.3 1232 138 79  
## 8 Clint Capela 16.6 848 102 67  
## 9 Montrezl Harrell 16.6 535 110 82  
## 10 Deandre Ayton 16.3 729 67 71

C\_df2%>% ggplot(aes(reorder(player\_name,PTS), PTS, fill = PTS)) + geom\_bar(stat="identity") +labs(x = 'player name', y = 'Points per game', title = 'Top 10 Center players based on points per game who have more than average TRB and BLK') +geom\_text(aes(player\_name, PTS,label = round(PTS,2) ), size = 3, color = 'black', hjust = 0) +  
 theme(plot.title = element\_text(size = 10),legend.position = "none") +  
 coord\_flip()



### SF (Small Forward)

The small forward (SF) is a versatile player who can play both inside and outside. They are typically taller than the shooting guard and often have a combination of speed and strength.

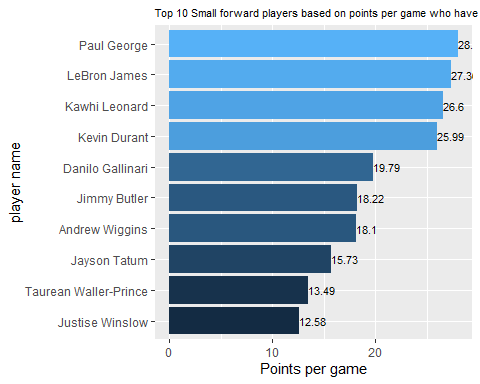
Top 10 small forward players based on points:

We first filter the small forward players in the data and then make the plots.

SF\_df <- player\_stats%>%filter(Pos == 'SF')%>%group\_by(player\_name)%>%summarise(PTS = mean(PTS/G), G = sum(G), TRB = mean(TRB), AST = mean(AST), BLK = mean(BLK), STL = mean(STL))  
  
  
SF\_df2 <- SF\_df%>%  
 filter(TRB > mean(SF\_df$TRB), AST > mean(SF\_df$AST), STL > mean(SF\_df$STL), BLK > mean(SF\_df$BLK), PTS > mean(SF\_df$PTS) , G > 40)%>%arrange(-PTS)%>% head(10)  
  
SF\_df2

## # A tibble: 10 x 7  
## player\_name PTS G TRB AST BLK STL  
## <chr> <dbl> <int> <dbl> <dbl> <dbl> <dbl>  
## 1 Paul George 28.0 77 628 318 34 170  
## 2 LeBron James 27.4 55 465 454 33 72  
## 3 Kawhi Leonard 26.6 60 439 199 24 106  
## 4 Kevin Durant 26.0 78 497 457 84 58  
## 5 Danilo Gallinari 19.8 68 417 178 23 49  
## 6 Jimmy Butler 18.2 55 290 220 29 99  
## 7 Andrew Wiggins 18.1 73 352 184 48 70  
## 8 Jayson Tatum 15.7 79 477 168 57 84  
## 9 Taurean Waller-Prince 13.5 55 199 118 19 53  
## 10 Justise Winslow 12.6 66 355 282 19 72

SF\_df2%>% ggplot(aes(reorder(player\_name,PTS), PTS, fill = PTS)) + geom\_bar(stat="identity") +  
 labs( x = 'player name', y = 'Points per game', title = 'Top 10 Small forward players based on points per game who have above average AST, TRB, STL, BLK') +  
 geom\_text(aes(player\_name, PTS,label = round(PTS,2) ), size = 3, color = 'black', hjust = 0) +  
 theme(plot.title = element\_text(size = 8.6),legend.position = "none") +  
 coord\_flip()

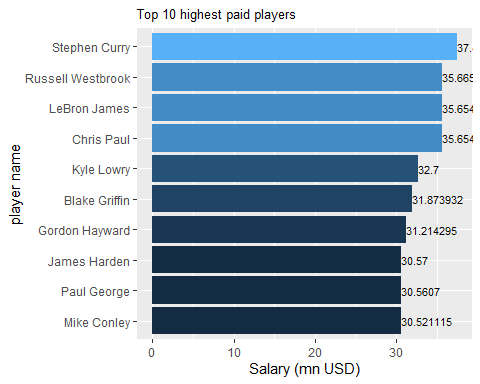


### Top 10 highest paid players

We will find the top 10 highest paid players among all the players.

For this we will join the player\_salary dataframe with the player\_stats dataframe and then group the resulting data frame by player name and take the average of all numerical columns/features.

*# reformat the salary column*  
player\_salary['salary'] <- player\_salary$salary / 10^6  
  
  
*# Join the two data frames ( player\_stats and player\_salary) by the "player\_name" column*  
merged\_df <- merge(player\_stats, player\_salary, by = "player\_name", all = TRUE)  
  
  
merged\_df%>%arrange(-salary)%>%head(10)%>%ggplot(aes(reorder(player\_name, salary), salary, fill = salary)) + geom\_bar(stat="identity") +labs(x = 'player name', y = 'Salary (mn USD)', title = 'Top 10 highest paid players') + geom\_text(aes(player\_name, salary,label = salary ), size = 3, color = 'black', hjust = 0) + theme(plot.title = element\_text(size = 10),legend.position = "none") + coord\_flip()



### Position wise player salaries

## Point guards

We will look at the salaries of the top 10 point guards based on assist to turnover ratio from the analysis we performed above.

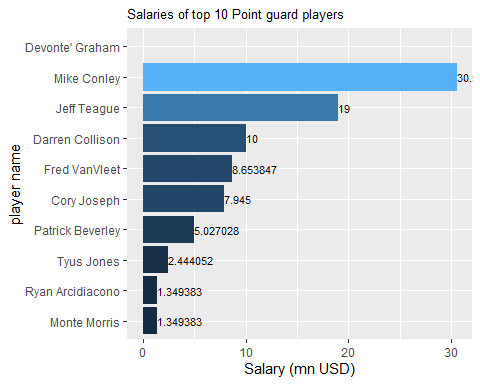
df1 <- merged\_df%>% filter(player\_name %in% PG\_df2$player\_name)%>%select(c('player\_name', 'Age', 'salary' ))  
df2 <- merge(PG\_df2, df1, by = 'player\_name', all = TRUE)  
df2

## player\_name AST\_TOV Age salary  
## 1 Cory Joseph 4.012500 27 7.945000  
## 2 Darren Collison 3.672000 31 10.000000  
## 3 Devonte' Graham 4.033333 23 NA  
## 4 Fred VanVleet 3.743902 24 8.653847  
## 5 Jeff Teague 3.536082 30 19.000000  
## 6 Mike Conley 3.453846 31 30.521115  
## 7 Monte Morris 5.711538 23 1.349383  
## 8 Patrick Beverley 3.529412 30 5.027028  
## 9 Ryan Arcidiacono 4.269841 24 1.349383  
## 10 Tyus Jones 6.957447 22 2.444052

merged\_df%>% filter(player\_name %in% PG\_df2$player\_name)%>%ggplot(aes(reorder(player\_name, salary), salary, fill = salary)) + geom\_bar(stat="identity") +labs(x = 'player name', y = 'Salary (mn USD)', title = 'Salaries of top 10 Point guard players') + geom\_text(aes(player\_name, salary,label = salary ), size = 3, color = 'black', hjust = 0) + theme(plot.title = element\_text(size = 10),legend.position = "none") + coord\_flip()

## Warning: Removed 1 rows containing missing values (position\_stack).

## Warning: Removed 1 rows containing missing values (geom\_text).



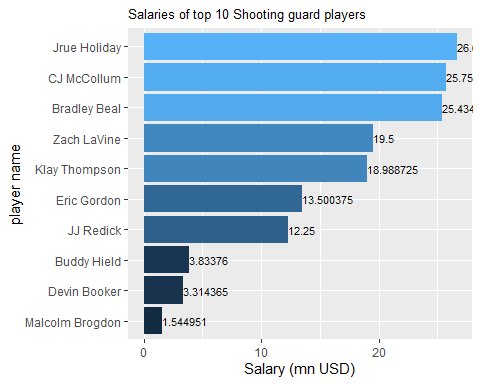
## Shooting guards

We will look at the salaries of the top 10 shooting guards based on above avg performance from the analysis we did above.

df1 <- merged\_df%>% filter(player\_name %in% SG\_df2$player\_name)%>%select(c('player\_name', 'Age', 'salary' ))  
df2 <- merge(SG\_df2, df1, by = 'player\_name', all = TRUE)  
df2

## player\_name X3P. X2P. FT. G PTS Age salary  
## 1 Bradley Beal 35.1 54.8 80.8 82 25.59756 25 25.434262  
## 2 Buddy Hield 42.7 48.7 88.6 82 20.67073 26 3.833760  
## 3 CJ McCollum 37.5 50.6 82.8 70 20.97143 27 25.759766  
## 4 Devin Booker 32.6 53.6 86.6 64 26.56250 22 3.314365  
## 5 Eric Gordon 36.0 49.7 78.3 68 16.22059 30 13.500375  
## 6 JJ Redick 39.7 50.2 89.4 76 18.05263 34 12.250000  
## 7 Jrue Holiday 32.5 53.9 76.8 67 21.19403 28 26.641111  
## 8 Klay Thompson 40.2 51.6 81.6 78 21.53846 28 18.988725  
## 9 Malcolm Brogdon 42.6 54.4 92.8 64 15.64062 26 1.544951  
## 10 Zach LaVine 37.4 50.4 83.2 63 23.68254 23 19.500000

merged\_df%>% filter(player\_name %in% SG\_df2$player\_name)%>%ggplot(aes(reorder(player\_name, salary), salary, fill = salary)) + geom\_bar(stat="identity") +labs(x = 'player name', y = 'Salary (mn USD)', title = 'Salaries of top 10 Shooting guard players') + geom\_text(aes(player\_name, salary,label = salary ), size = 3, color = 'black', hjust = 0) + theme(plot.title = element\_text(size = 10),legend.position = "none") + coord\_flip()

 ## Power Forward

We will look at the salaries of the top 10 power forward players based on above average performance from the analysis we did above.

df1 <- merged\_df%>% filter(player\_name %in% PF\_df2$player\_name)%>%select(c('player\_name', 'Age', 'salary' ))%>%group\_by\_at(vars(player\_name))%>%summarise(across(everything(), mean))  
df2 <- merge(PF\_df2, df1, by = 'player\_name', all = TRUE)  
df2

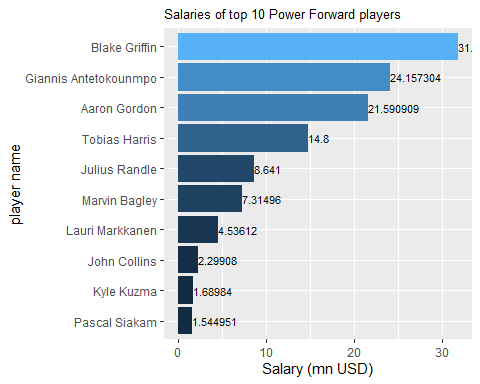
## player\_name PTS TRB BLK G Age salary  
## 1 Aaron Gordon 15.97436 574 56.00000 78 23 21.590909  
## 2 Blake Griffin 24.54667 565 28.00000 75 29 31.873932  
## 3 Giannis Antetokounmpo 27.69444 898 110.00000 72 24 24.157304  
## 4 John Collins 19.47541 595 39.00000 61 21 2.299080  
## 5 Julius Randle 21.43836 634 45.00000 73 24 8.641000  
## 6 Kyle Kuzma 18.68571 382 26.00000 70 23 1.689840  
## 7 Lauri Markkanen 18.73077 470 33.00000 52 21 4.536120  
## 8 Marvin Bagley 14.88710 471 59.00000 62 19 7.314960  
## 9 Pascal Siakam 16.92500 549 52.00000 80 24 1.544951  
## 10 Tobias Harris 19.73882 430 24.66667 164 26 14.800000

merged\_df%>% filter(player\_name %in% PF\_df2$player\_name)%>% group\_by\_at(vars(player\_name))%>% summarise(across(everything(), mean))%>%ggplot(aes(reorder(player\_name, salary), salary, fill = salary)) + geom\_bar(stat="identity") +labs( x = 'player name', y = 'Salary (mn USD)', title = 'Salaries of top 10 Power Forward players') + geom\_text(aes(player\_name, salary,label = salary ), size = 3, color = 'black', hjust = 0) + theme(plot.title = element\_text(size = 10),legend.position = "none") + coord\_flip()

## Warning in mean.default(Pos): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Pos): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Pos): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Pos): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Pos): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Pos): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Pos): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Pos): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Pos): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Pos): argument is not numeric or logical: returning NA

## Warning in mean.default(Tm): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Tm): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Tm): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Tm): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Tm): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Tm): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Tm): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Tm): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Tm): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Tm): argument is not numeric or logical: returning NA

## Warning in mean.default(index): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(index): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(index): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(index): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(index): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(index): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(index): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(index): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(index): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(index): argument is not numeric or logical: returning NA



## Small Forward

We will look at the salaries of the top 10 small forward players based on above average performance from the analysis we did above.

df1 <- merged\_df%>% filter(player\_name %in% SF\_df2$player\_name)%>%select(c('player\_name', 'Age', 'salary' ))%>%group\_by\_at(vars(player\_name))%>%summarise(across(everything(), mean))  
df2 <- merge(SF\_df2, df1, by = 'player\_name', all = TRUE)  
df2

## player\_name PTS G TRB AST BLK STL Age salary  
## 1 Andrew Wiggins 18.09589 73 352 184 48 70 23 25.467250  
## 2 Danilo Gallinari 19.79412 68 417 178 23 49 30 21.587579  
## 3 Jayson Tatum 15.73418 79 477 168 57 84 20 6.700800  
## 4 Jimmy Butler 18.21818 55 290 220 29 99 29 19.841627  
## 5 Justise Winslow 12.57576 66 355 282 19 72 22 3.448926  
## 6 Kawhi Leonard 26.60000 60 439 199 24 106 27 23.114066  
## 7 Kevin Durant 25.98718 78 497 457 84 58 30 30.000000  
## 8 LeBron James 27.36364 55 465 454 33 72 34 35.654150  
## 9 Paul George 28.03896 77 628 318 34 170 28 30.560700  
## 10 Taurean Waller-Prince 13.49091 55 199 118 19 53 24 NA

merged\_df%>% filter(player\_name %in% SF\_df2$player\_name)%>% group\_by\_at(vars(player\_name))%>% summarise(across(everything(), mean))%>%ggplot(aes(reorder(player\_name, salary), salary, fill = salary)) + geom\_bar(stat="identity") +labs( x = 'player name',y = 'Salary (mn USD)', title = 'Salaries of top 10 Small Forward players') + geom\_text(aes(player\_name, salary,label = salary ), size = 3, color = 'black', hjust = 0) + theme(plot.title = element\_text(size = 10),legend.position = "none") + coord\_flip()

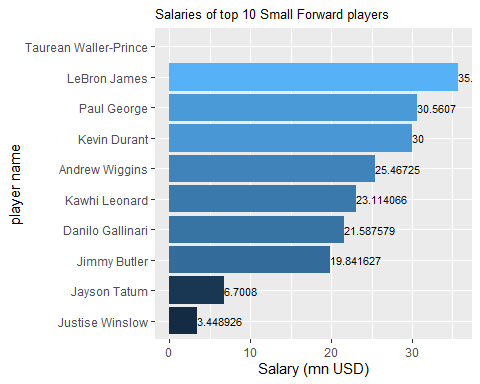
## Warning in mean.default(Pos): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Pos): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Pos): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Pos): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Pos): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Pos): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Pos): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Pos): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Pos): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Pos): argument is not numeric or logical: returning NA

## Warning in mean.default(Tm): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Tm): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Tm): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Tm): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Tm): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Tm): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Tm): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Tm): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Tm): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(Tm): argument is not numeric or logical: returning NA

## Warning in mean.default(index): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(index): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(index): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(index): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(index): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(index): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(index): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(index): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(index): argument is not numeric or logical: returning NA  
  
## Warning in mean.default(index): argument is not numeric or logical: returning NA

## Warning: Removed 1 rows containing missing values (position\_stack).

## Warning: Removed 1 rows containing missing values (geom\_text).



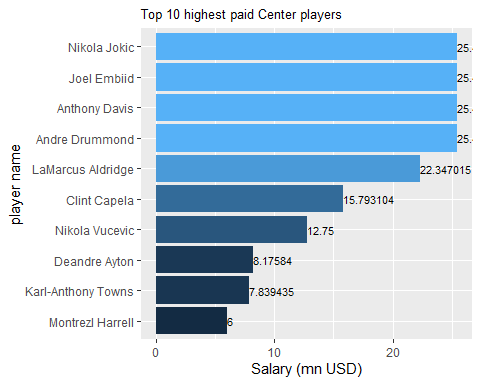
## Center

We will look at the salaries of the top 10 center players based on above average performance from the analysis we did above.

df1 <- merged\_df%>% filter(player\_name %in% C\_df2$player\_name)%>%select(c('player\_name', 'Age', 'salary' ))  
df2 <- merge(C\_df2, df1, by = 'player\_name', all = TRUE)  
df2

## player\_name PTS TRB BLK G Age salary  
## 1 Andre Drummond 17.34177 1232 138 79 25 25.434262  
## 2 Anthony Davis 25.92857 672 135 56 25 25.434263  
## 3 Clint Capela 16.62687 848 102 67 24 15.793104  
## 4 Deandre Ayton 16.32394 729 67 71 20 8.175840  
## 5 Joel Embiid 27.51562 871 122 64 24 25.467250  
## 6 Karl-Anthony Towns 24.41558 954 125 77 23 7.839435  
## 7 LaMarcus Aldridge 21.32099 744 107 81 33 22.347015  
## 8 Montrezl Harrell 16.59756 535 110 82 25 6.000000  
## 9 Nikola Jokic 20.05000 865 55 80 23 25.467250  
## 10 Nikola Vucevic 20.81250 960 89 80 28 12.750000

merged\_df%>% filter(player\_name %in% C\_df2$player\_name)%>%ggplot(aes(reorder(player\_name, salary), salary, fill = salary)) + geom\_bar(stat="identity") +labs(x = 'player name', y = 'Salary (mn USD)', title = 'Top 10 highest paid Center players') + geom\_text(aes(player\_name, salary,label = salary ), size = 3, color = 'black', hjust = 0) + theme(plot.title = element\_text(size = 10),legend.position = "none") + coord\_flip()



# Players in Chicago Bulls

We will now find all the players in the Chicago bulls team.

*# Join the two data frames ( player\_stats and player\_salary) by the "player\_name" column*  
CHI\_df <- merged\_df%>%filter(Tm == 'CHI')%>%select(player\_name, Age, Pos, salary)  
CHI\_df

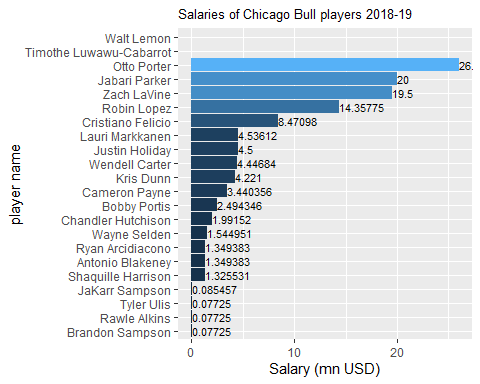
## player\_name Age Pos salary  
## 1 Antonio Blakeney 22 SG 1.349383  
## 2 Bobby Portis 23 PF 2.494346  
## 3 Brandon Sampson 21 SG 0.077250  
## 4 Cameron Payne 24 PG 3.440356  
## 5 Chandler Hutchison 22 SF 1.991520  
## 6 Cristiano Felicio 26 C 8.470980  
## 7 Jabari Parker 23 PF 20.000000  
## 8 JaKarr Sampson 25 SF 0.085457  
## 9 Justin Holiday 29 SG 4.500000  
## 10 Kris Dunn 24 PG 4.221000  
## 11 Lauri Markkanen 21 PF 4.536120  
## 12 Otto Porter 25 SF 26.011913  
## 13 Rawle Alkins 21 SG 0.077250  
## 14 Robin Lopez 30 C 14.357750  
## 15 Ryan Arcidiacono 24 PG 1.349383  
## 16 Shaquille Harrison 25 PG 1.325531  
## 17 Timothe Luwawu-Cabarrot 23 SF NA  
## 18 Tyler Ulis 23 PG 0.077250  
## 19 Walt Lemon 26 PG NA  
## 20 Wayne Selden 24 SG 1.544951  
## 21 Wendell Carter 19 C 4.446840  
## 22 Zach LaVine 23 SG 19.500000

And now we will plot the salaries of all these players in the year 2018-19.

CHI\_df%>%arrange(-salary)%>%ggplot(aes(reorder(player\_name, salary), salary, fill = salary)) + geom\_bar(stat="identity") +labs( x = 'player name', y = 'Salary (mn USD)', title = 'Salaries of Chicago Bull players 2018-19') + geom\_text(aes(player\_name, salary,label = salary ), size = 3, color = 'black', hjust = 0) + theme(plot.title = element\_text(size = 10),legend.position = "none") + coord\_flip()

## Warning: Removed 2 rows containing missing values (position\_stack).

## Warning: Removed 2 rows containing missing values (geom\_text).



cat("Chicago Bulls paid :", sum(CHI\_df$salary, na.rm = TRUE), 'mn USD for the team in 2018-19')

## Chicago Bulls paid : 119.8573 mn USD for the team in 2018-19

team\_payroll%>%filter(ï..team\_id == 24)

## ï..team\_id team salary  
## 1 24 Chicago $112,598,201

We observe that the team payroll of Chicago bulls for the year 2018-19 was approx $112 mn it seems that they overspent on players based on above findings.

Chicago Bulls budget for player contracts next season is $118 million, which is approx 2 millions less than before.

In the National Basketball Association (NBA), each team can have up to 15 players on its roster, which is known as the “active roster.” However, teams are also allowed to have up to two additional players on their “inactive roster,” bringing the total number of players on a team’s full roster to 17. The inactive players are typically reserves who are not currently needed to play, but can be activated in case of injury or other circumstances.

Based on this info on average we can spend approx 7 mn USD on each player. But its not necessary to keep all the players within this budget, we can make adjustments to rope in some better players in each position.

Five best starting players that the team can afford, one from each position:

* PG : Either Tyus JOnes (age: 22, salary: $2.4 mn) or Monte Morris (age: 23, salary: $1.3 mn)
* SG : Malcolm Brogdon (age 26, salary: $1.5 mn)
* PF : Julius Randle (age: 24, salary: $8.6 mn)
* SF : Jayson Tatum (age: 20, salary: $6.7 mn)
* C : Karl-Anthony Towns (age: 23, salary: $7.8 mn)

# 4. Data Modelling:

In this section we will create a linear regression model that would predict points per game scored by a player based on his stats.

We will group the player\_stats dataframe by player\_name and use the average values of the following columns as features: ‘Age’,‘G’,‘MP’,‘X3P.’,‘X2P.’,‘eFG.’,‘FT.’,‘TRB’,‘AST’,‘STL’,‘BLK’,‘TOV’,‘PF’,‘PTS’

We have not selected some of the columns as they are inherent in the columns present above for example: X3P. <- X3P/X3PA X2P. <- X2P/X2PA and so on.

It is done in order to avoid multi-colinearity in the features.

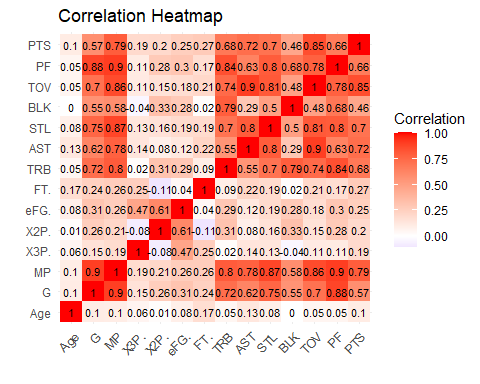
*# we extract the relevant columsn and average the dataset by player name and convert the points to points per game.*  
data = player\_stats[,c('player\_name','Age','G','MP','X3P.','X2P.','eFG.','FT.','TRB','AST','STL','BLK','TOV','PF','PTS')]%>%group\_by\_at(vars(player\_name))%>%summarise(across(everything(), mean), PTS = round((PTS/G)))  
  
*# dropping player names*  
data[,c(1)] <- NULL

We will now perform feature selection. We will try to methods and compare them.

## Feature selection approach 1: Correlation

We will make use of Correlation plot for feature selection. We will keep only one of the two highly correlated features, if any.

*# Compute the correlation matrix*  
cor\_mat <- cor(data)  
  
*# Melt the correlation matrix into long format*  
cor\_melted <- melt(cor\_mat)  
  
*# Create the heatmap using ggplot2*  
ggplot(cor\_melted, aes(x=Var1, y=Var2, fill=value)) +  
 geom\_tile() +  
 scale\_fill\_gradient2(low="blue", mid="white", high="red", midpoint=0) +  
 geom\_text(aes(label=round(value,2)), color="black", size=3) +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 labs(x="", y="", fill="Correlation") +  
 ggtitle("Correlation Heatmap")



From the above heatmap for correlation we observe that there is a : - high positive correlation between G and MP (0.9) - high positive correlation of G and MP with TRB (0.72, 0.8), AST(0.62, 0.78), STL(0.75, 0.87), TOV(0.7, 0.86), PF (0.88, 0.9), PTS(0.75, 0.92) - high positive correlation of TRB with STL(0.7), BLK(0.79), TOV(0.74), PF (0.84), PTS(0.77) - high positive correlation of AST with STL(0.8), TOV(0.9), PTS(0.8) - high positive correlation of TOV with PF(0.78) and PTS(0.92)

So we will drop both G, MP, STL, BLK, TOV and PF from the data and train the model.

data2 = data  
  
data2[,c('G','MP','STL', 'BLK', 'TOV', 'PF' )] <- NULL  
  
***## scale the features, not scaling the target (PTS)***  
data\_scaled <- as.data.frame(scale(data2[,c(1:7)]))  
data\_scaled['PTS'] <- data2$PTS  
  
  
***## train test split***  
set.seed(123)  
  
*# Split the data into a training set (80%) and a testing set (20%)*  
split = sample(nrow(data\_scaled), floor(0.8\*nrow(data\_scaled)), replace = FALSE)  
  
*# Create the training set*  
train = data\_scaled[split, ]  
  
*# Create the testing set*  
test = data\_scaled[-split, ]  
  
*# train model on train data*  
model <- lm(PTS ~ ., data = train)  
  
*# prediction*  
y\_pred <- predict(model, newdata = test)  
  
*# Calculate the R-squared value*  
r2 <- summary(model)$r.squared  
  
*# Calculate the Root Mean Squared Error*  
rmse <- sqrt(mean((y\_pred - test$PTS)^2))  
  
cat("R-squared:", r2, "\n")

## R-squared: 0.6483023

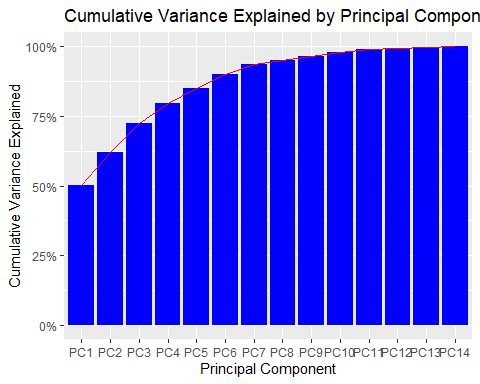
cat("Root Mean Squared Error:", rmse, "\n")

## Root Mean Squared Error: 3.410153

## Feature selection approach 2: PCA

We will perform PCA to reduce the dimensions of the data.

*# Perform PCA on the dataset*  
pca\_result <- prcomp(data, scale. = TRUE)  
  
*# Extract the variance explained by each principal component*  
var\_explained <- pca\_result$sdev^2 / sum(pca\_result$sdev^2)  
  
*# Calculate the cumulative variance explained by each component*  
cumulative\_var <- cumsum(var\_explained)  
  
*# Create a data frame with the variance and cumulative variance explained*  
var\_df <- data.frame(PCA\_Component = paste0("PC", 1:length(var\_explained)),   
 Variance\_Explained = var\_explained,   
 Cumulative\_Variance = cumulative\_var)  
  
*# Reorder PCA\_Component factor according to Cumulative\_Variance*  
var\_df <- var\_df %>%   
 mutate(PCA\_Component = factor(PCA\_Component, levels = PCA\_Component[order(Cumulative\_Variance)]))  
  
*# Plot the cumulative variance explained by the principal components*  
ggplot(var\_df, aes(x = PCA\_Component, y = Cumulative\_Variance)) +   
 geom\_col(fill = "blue") +   
 geom\_line(aes(y = Cumulative\_Variance, group = 1), color = "red") +   
 scale\_y\_continuous(labels = scales::percent) +   
 labs(x = "Principal Component", y = "Cumulative Variance Explained",   
 title = "Cumulative Variance Explained by Principal Components")



*# Select only those components that explain up to 95% of the variance*  
n\_components <- sum(cumulative\_var <= 0.95)  
  
*# Create a new data frame with the selected components*  
pca\_data <- data.frame(pca\_result$x[,1:n\_components])  
  
pca\_data['PTS'] <- data$PTS  
  
***## train test split***  
set.seed(123)  
  
*# Split the data into a training set (80%) and a testing set (20%)*  
split = sample(nrow(pca\_data), floor(0.8\*nrow(pca\_data)), replace = FALSE)  
  
*# Create the training set*  
train = pca\_data[split, ]  
  
*# Create the testing set*  
test = pca\_data[-split, ]  
  
*# train model on train data*  
model2 <- lm(PTS ~ ., data = train)  
  
*# prediction*  
y\_pred2 <- predict(model2, newdata = test)  
  
*# Calculate the R-squared value*  
r2\_2 <- summary(model2)$r.squared  
  
*# Calculate the Root Mean Squared Error*  
rmse2 <- sqrt(mean((y\_pred2 - test$PTS)^2))  
  
cat("R-squared:", r2\_2, "\n")

## R-squared: 0.9054921

cat("Root Mean Squared Error:", rmse2, "\n")

## Root Mean Squared Error: 1.899778

Based on the r-squared and rmse values, PCA model performed better at predicting points per game of players given their statistics.

#################### END ###################################