Data cleaning project

Ricard Trinchet Arnejo

11th of June 2019

Índice

0.1. packages

These are the packages that will be needed in this project:

```
library(plyr)
library(tidyverse)

library(knitr)
library(ggthemes)
library(stringr)
library(VIM)
library(car)
library(gridExtra)
```

0.2. Dataset load into R

We upload the datasets into R.

```
season_stats <- read.csv('nba-players-stats/Seasons_Stats.csv')

salaries <- read.csv('salaries.csv')
```

1. Dataset description

Two datasets are used:

- `season_stats`: Dataset containing advanced stats since the year 1950. It was obtained through *this link*. A glossary with an explanation of the attributes of this dataset can be cheked here.
- `salaries`: Dataset containing the salaries of the NBA players from 1990 until 2018. Accessed via *this link*.

Now let us describe briefly both datasets.

The first dataset has 24691 observations each one with 53 variables. The second one, contains 11837 observations of 7 variables.

That amount of data is too big for the purpose of this analysis, so we will reduce the dimensions of both datasets. The idea is to join both datasets to get a full dataset with player stats and salaries. After that, we will select only the data from the 2016-17 season. We will also use the salaries of the 2017-18 season.

The remaining data will not be considered in the main analyses, though in the last section, it will be used to make some visualizations.

Also, some variable selection will be made. We will choose 23 of the variables from the first dataset and only one from the second.

The selected variables are the following. From the first dataset:

- `Player`: Name of the player.
- `Pos`: Position of the player. It can be one of the following 5: PG, SG, SF, PF, C, or a combination of those.
- `Age`: Age of the player.
- `Tm`: Team of the player.
- `G`: Games played by the player on that season.
- `GS`: Games started by the player on that season, i.e., games where the player was on the starting line-up.
- `MPG`: Minutes Played per game.
- `PPG`: Points scored per game.
- `APG`: Assists made per game.
- `RPG`: Rebounds per game.
- `SPG`: Steals per game.
- `BPG`: Blocks per game.
- `TOPG`: Turnovers per game.
- `PFPG`: Personal fouls per game.
- `PER`: Player Efficiency Rating (available since the 1951-52 season); PER is a rating developed by ESPN.com columnist John Hollinger. In John's words, "The PER sums up all a player's positive accomplishments, subtracts the negative accomplishments, and returns a per-minute rating of a player's performance."
- `TS.`: TS %, True Shooting Percentage, a measure of shooting efficiency that takes into account field goals, 3-point field goals, and free throws.
- `USG.`: Usage Percentage (available since the 1977-78 season in the NBA). Usage percentage is an estimate of the percentage of team plays used by a player while he was on the floor.
- `WS`: an estimate of the number of wins contributed by a player.
- `VORP`: Value Over Replacement Player (available since the 1973-74 season in the NBA); a box score estimate of the points per 100 TEAM possessions that a player contributed above a replacement-level (-2.0) player, translated to an average team and prorated to an 82-game season
- `FG.`: Field Goal Percentage.
- `X2P.`: 2-Point Field Goal Percentage.
- `X3P.`: 3-Point Field Goal Percentage (available since the 1979-80 season in the NBA).

And from the second dataset:

- `Salary.in..`: The amount of money that the players gets for the season, in dollars.

1.1. Importance of the dataset and questions that will be studied

There are three main questions that we would like to answer:

- Are the Win Shares the factor that is more correlated to the salary of a player?
- Is there any significant difference in any of the main statistics for the different five positions of the game?
- Can the players' salaries be predicted?

To answer these questions, we will work with the data from the 2016-17 season, as it is the last for which there is data on both salaries and in-game stats.

Furthermore, we try to answer some small questions with the help of some plots. The questions that we pose are the following:

- How did the number of points scored evolved through the years?
- Which teams are the ones which expent the most on salaries? Were they successfull teams?
- Was there any change in the number of three points attempts in the history of the league?

2. Data cleaning

We start with some basic cleaning of the datasets. First, let us remove two variables that are empty and will not be used.

```
# remove blanl & blank2: empty variables

season_stats <- season_stats %>% select(-c(blanl, blank2))
```

Next, we will clean the salary variable, by removing the dollar sign and changing the variable type to numeric. For that, we have to take care of the decimal separator and the commas

```
# remove $ sign
salaries$Salary.in.. <- str_replace_all(salaries$Salary.in.., fixed("$"), "")

# remove point as it confuses the conversion to numeric
salaries$Salary.in.. <- str_replace_all(salaries$Salary.in.., fixed("."), "")

# replace the decimal separator from comma to dot
salaries$Salary.in.. <- str_replace_all(salaries$Salary.in.., fixed(","), ".")

# convert the salaries to numeric
salaries$Salary.in.. <- as.numeric(salaries$Salary.in..)
```

We now restrict the datasets to the 2016-17 season, as it is the one which has more interest to our analysis (we want to study the statistics for the las available season, that is, 2016-17).

```r
# year 2017: season 16_17

salaries_2017 <- salaries %>% filter(Season.End==2017)

season_stats_2017 <- season_stats %>% filter(Year==2017)

# update the levels in the subset 2017

#stats
season_stats_2017$Pos <- factor(season_stats_2017$Pos)
season_stats_2017$Player <- factor(season_stats_2017$Player)
season_stats_2017$Tm <- factor(season_stats_2017$Tm)

#salaries
salaries_2017$Player.Name <- factor(salaries_2017$Player.Name)
salaries_2017$Team <- factor(salaries_2017$Team)


# update and match the levels on both teams
salaries_2017$Team <- revalue(salaries_2017$Team, c("CHA"="CHO", "NJN"="BRK", "NOH" = "NOP"))
```

2.1. Zeroes and NAs handling

We start with the NA handling for both datasets. The salaries_2017 dataset doesn't contain any null value, as it is shown below.

```r
salaries_2017 %>%
  select(everything()) %>%
  summarise_all(funs(sum(is.na(.))))
```

```
##   Register.Value Player.Name Salary.in.. Season.Start Season.End Team
## 1              0           0           0            0          0    0
##   Full.Team.Name
## 1              0
```

The other dataset does have some null values that should be analyzed.

```r
# number of NAs
season_stats_2017 %>%
  select(everything()) %>%
  summarise_all(funs(sum(is.na(.))))
```

```
##   X Year Player Pos Age Tm G GS MP PER TS. X3PAr FTr ORB. DRB. TRB. AST.
## 1 0    0      0   0   0  0 0  0  0   0   2     2   2    0    0    0    0
##   STL. BLK. TOV. USG. OWS DWS WS WS.48 OBPM DBPM BPM VORP FG FGA FG. X3P
## 1    0    0    2    0   0   0  0     0    0    0   0    0  0   0   2   0
##   X3PA X3P. X2P X2PA X2P. eFG. FT FTA FT. ORB DRB TRB AST STL BLK TOV PF
## 1    0   46   0    0    5    2  0   0  24   0   0   0   0   0   0   0  0
##   PTS
## 1   0
```

Now we should think if those values should be imputed or else if we can drop them from the dataset. Let us examine those samples which have some missing values and then decide what is more suitable for each case.

```
### lists--NA

# residual: players who didn't play much
season_stats_2017 %>%
  filter(is.na(FT.) == TRUE) %>%
  select(c("Player", "Tm", "Pos", "G", "MP")) %>%
  arrange(desc(G)) %>%
  head(n=10)
```

```
##              Player  Tm Pos  G  MP
## 1      Damjan Rudez ORL  SF 45 314
## 2     Anthony Brown TOT  SF 11 159
## 3     Anthony Brown NOP  SF  9 143
## 4     Bruno Caboclo TOR  SF  9  40
## 5       Steve Novak MIL  PF  8  22
## 6     Arinze Onuaku ORL   C  8  28
## 7       Roy Hibbert DEN   C  6  11
## 8  Patricio Garino ORL  SG  5  43
## 9        John Lucas MIN  PG  5  11
## 10     Axel Toupane TOT  SF  4  47
```

```
# important: some big stars included
season_stats_2017 %>%
  filter(is.na(X3P.) == TRUE) %>%
  select(c("Player", "Tm", "Pos", "G", "MP")) %>%
  arrange(desc(G)) %>%
  head(n=10)
```

```
##                Player  Tm Pos  G   MP
## 1     Bismack Biyombo ORL   C 81 1793
## 2           David Lee SAS  PF 79 1477
## 3    Hassan Whiteside MIA   C 77 2513
## 4      Dewayne Dedmon SAS   C 76 1330
## 5         Aron Baynes DET   C 75 1163
## 6  Cristiano Felicio CHI   C 66 1040
## 7        Clint Capela HOU   C 65 1551
## 8        Cole Aldrich MIN   C 62  531
## 9        Jakob Poeltl TOR   C 54  626
## 10      Jahlil Okafor PHI   C 50 1134
```

While the first case is residual, as only players with few minutes played through the season appear, the second case is more important, as it contains some good overall players, like Capela or Whiteside, and others who logged a big number of minutes.

For the first case, we will drop the values and for the second we will impute them.

```
#drop values with NA values in FT.
season_stats_2017<- season_stats_2017 %>% drop_na(FT.)
```

Let us see how many NAs are still present on the dataset:

```r
season_stats_2017 %>%
  select(everything()) %>%
  summarise_all(funs(sum(is.na(.))))
```

```
##   X Year Player Pos Age Tm G GS MP PER TS. X3PAr FTr ORB. DRB. TRB. AST.
## 1 0    0      0   0   0  0 0  0  0   0   0     0   0    0    0    0    0
##   STL. BLK. TOV. USG. OWS DWS WS WS.48 OBPM DBPM BPM VORP FG FGA FG. X3P
## 1    0    0    0    0   0   0  0     0    0    0   0    0  0   0   0   0
##   X3PA X3P. X2P X2PA X2P. eFG. FT FTA FT. ORB DRB TRB AST STL BLK TOV PF
## 1    0   39   0    0    1    0  0   0   0   0   0   0   0   0   0   0  0
##   PTS
## 1   0
```

We will see which are the samples with NAs in `X2p.`

```r
# check X2P.
season_stats_2017 %>%
  filter(is.na(X2P.) == TRUE) %>%
  select(c("Player", "Tm", "Pos", "G", "MP")) %>%
  arrange(desc(G))
```

```
##             Player  Tm Pos G MP
## 1 Chris McCullough WAS  PF 2  8
```

It is just one player who didn't play many minutes, so we will drop this sample from the dataset.

```r
# drop the sample
season_stats_2017<- season_stats_2017 %>%
  drop_na(X2P.)
```

Next, we deal with the imputation of the NAs values of the variable `X3P.`.

```r
season_stats_2017 %>%
  filter(is.na(X3P.) == TRUE) %>%
  select(c("Player", "Tm", "Pos", "G", "MP", "X3P", "X3PA")) %>%
  arrange(desc(X3P)) %>%
  head(n=10)
```

```
##              Player  Tm Pos  G   MP X3P X3PA
## 1      Cole Aldrich MIN   C 62  531   0    0
## 2      Joel Anthony SAS   C 19  122   0    0
## 3         Omer Asik NOP   C 31  482   0    0
## 4       Aron Baynes DET   C 75 1163   0    0
## 5   Bismack Biyombo ORL   C 81 1793   0    0
## 6      Clint Capela HOU   C 65 1551   0    0
## 7    Tyson Chandler PHO   C 47 1298   0    0
## 8  Rakeem Christmas IND  PF 29  219   0    0
## 9     Deyonta Davis MEM   C 36  238   0    0
## 10         Ed Davis POR  PF 46  789   0    0
```

We see that those values are empties because those players did not shoot any 3-pointer through the entire season. Thus, it would make sense not to consider the variable for them. To simplify the analysis, we can imput the value 0. If they didn't shoot any 3, they probably would have a bad percentage anyway.

```
#replace NAs by 0
season_stats_2017 <- season_stats_2017 %>%
  mutate(X3P. = replace_na(X3P., 0) )

##check the results on two of the players
season_stats_2017 %>%
  filter(Player %in% c("Clint Capela", "Hassan Whiteside")) %>%
  select(c(Player, X3P. ))
```

```
##             Player X3P.
## 1     Clint Capela    0
## 2 Hassan Whiteside    0
```

2.2. Computing Per Game statistics

We will compute some basic per game statistics that are not included in the `season_stats` dataset, namely, points per game (PPG), assiste per game (APG), rebounds per game (RPG), blocks per game (BPG), steals per game (SPG), turnovers per game (TOPG), personal fouls per game (PFPG) and minutes per game (MPG). Those variables will be used in the following analyses and are more convenient to understand them and to compare the values of the different players.

```
season_stats_2017 <- season_stats_2017 %>%
  mutate(PPG = PTS/G, APG = AST/G, RPG = TRB/G,
         BPG = BLK/G, SPG = STL/G, MPG = MP/G,
         TOPG = TOV/G, PFPG = PF/G )
```

2.3. Join the two tables

Finally, we can create the table that resulsts from the union of the stats of 2017 table and the 2017 salaries.

```
# join stats with salaries: join by player and team to avoid duplicities of players

salaries_2017_join <- salaries_2017 %>%
  select(c(Player.Name, Team, Salary.in..))

stats_with_salaries <- inner_join(season_stats_2017, salaries_2017_join,
                                  by = c('Player' = 'Player.Name', "Tm"="Team"))


head(stats_with_salaries[,c(1:10,53)])
```

```
##       X Year         Player Pos Age  Tm  G GS   MP  PER       APG
## 1 24096 2017  Alex Abrines  SG  23 OKC 68  6 1055 10.1 0.5882353
## 2 24098 2017    Quincy Acy  PF  26 DAL  6  0   48 -1.4 0.0000000
## 3 24099 2017    Quincy Acy  PF  26 BRK 32  1  510 13.1 0.5625000
## 4 24100 2017  Steven Adams   C  23 OKC 80 80 2389 16.5 1.0750000
## 5 24101 2017 Arron Afflalo  SG  31 SAC 61 45 1580  9.0 1.2786885
## 6 24102 2017 Alexis Ajinca   C  28 NOP 39 15  584 12.9 0.3076923
```

Now let us analyze if there were any NAs created with this process.

```
stats_with_salaries %>%
  select(everything()) %>%
  summarise_all(funs(sum(is.na(.))))
```

```
##   X Year Player Pos Age Tm G GS MP PER TS. X3PAr FTr ORB. DRB. TRB. AST.
## 1 0    0      0   0   0  0 0  0  0   0   0     0   0    0    0    0    0
##   STL. BLK. TOV. USG. OWS DWS WS WS.48 OBPM DBPM BPM VORP FG FGA FG. X3P
## 1    0    0    0    0   0   0  0     0    0    0   0    0  0   0   0   0
##   X3PA X3P. X2P X2PA X2P. eFG. FT FTA FT. ORB DRB TRB AST STL BLK TOV PF
## 1    0    0   0    0    0    0  0   0   0   0   0   0   0   0   0   0  0
##   PTS PPG APG RPG BPG SPG MPG TOPG PFPG Salary.in..
## 1   0   0   0   0   0   0   0    0    0           0
```

We see that no null values were introduced.

2.4. Subset selection

After cleaning the data, we will select a subset of all the variables, with which we will continue the analysis. Nevertheless, the data cleaning made in the previous section can be useful for further analyses on the dataset.

```
subset <- stats_with_salaries %>%
  select(c(Player,Tm, Pos, Age,  G, GS, MPG, PPG, APG, RPG, BPG,
           SPG, TOPG, PFPG, WS, PER, VORP, X2P., X3P., FG., TS.,
           USG., Salary.in.. ) )

# variables that we will keep
names(subset)
```

```
##  [1] "Player"      "Tm"          "Pos"         "Age"         "G"
##  [6] "GS"          "MPG"         "PPG"         "APG"         "RPG"
## [11] "BPG"         "SPG"         "TOPG"        "PFPG"        "WS"
## [16] "PER"         "VORP"        "X2P."        "X3P."        "FG."
## [21] "TS."         "USG."        "Salary.in.."
```

2.5. Outliers

Now we can check for outliers on every variable of the previously selected ones.

```
boxplot.stats(subset$Age)$out
```

```
## [1] 40 39 39 39
```

```
boxplot.stats(subset$G)$out
```

```
## integer(0)
```

```r
boxplot.stats(subset$MPG)$out
```

```
## numeric(0)
```

```r
boxplot.stats(subset$PPG)$out
```

```
##  [1] 22.90000 22.41892 23.10390 22.12821 23.89474 24.35294 25.30380
##  [8] 27.98667 27.29730 25.08065 23.66667 21.57377 29.08642 21.93151
## [15] 25.22222 26.40541 25.51351 26.98667 22.40000 22.96250 28.93421
## [22] 22.33333 25.13415 23.16456 23.14103 31.58025 23.57317
```

```r
boxplot.stats(subset$APG)$out
```

```
##  [1]  5.425000  5.514286  5.909091  6.333333  5.486842  6.275362  6.620253
##  [8]  5.794521  5.153846  7.013158 11.185185  7.283582  4.955882  5.805556
## [15]  5.192308  8.729730  4.948276  5.853333  5.111111  6.950000  6.592593
## [22]  5.133333  9.229508  6.451220  5.058824  6.681159  9.093333  6.316456
## [29]  5.160494  7.792683  5.907895  5.506329 10.653846 10.370370  6.850000
```

```r
boxplot.stats(subset$RPG)$out
```

```
##  [1] 11.468085 12.470588 11.813333 13.777778 12.777778 10.353659 12.702703
##  [8]  9.835616 13.753086 11.100000 10.350000 10.000000  9.179487 12.280488
## [15]  9.487500 10.386667 10.666667 14.129870
```

```r
boxplot.stats(subset$BPG)$out
```

```
##  [1] 1.236111 1.887500 1.123457 1.230769 1.117647 2.226667 1.158537
##  [8] 1.098765 1.596774 2.451613 1.337838 1.093750 2.641975 1.394737
## [15] 1.344828 1.279412 1.243243 1.500000 1.434783 1.131579 1.666667
## [22] 1.272727 1.653333 1.444444 1.136364 1.578947 1.900000 1.074074
## [29] 1.969697 6.000000 1.076923 1.256098 2.135802 2.090909
```

```r
boxplot.stats(subset$SPG)$out
```

```
##  [1] 1.837500 1.881579 1.895522 1.810127 2.026316 1.783784 2.000000
##  [8] 1.934426 1.706667 2.012821
```

```r
boxplot.stats(subset$TOPG)$out
```

```
##  [1] 3.378788 3.089744 3.647059 3.025316 3.774194 5.728395 4.094595
##  [8] 3.100000 3.265823 4.141026 5.407407
```

```r
boxplot.stats(subset$PFPG)$out
```

```
## [1] 4.411765
```

```r
boxplot.stats(subset$WS)$out
```

```
##  [1] 12.4 13.8 10.0 12.6 11.0  9.0 12.0 14.3 15.0 10.4  8.9 12.9  9.7 11.8
## [15] 13.6 10.3 10.1 10.6  9.4 12.6 12.7 13.1  9.5
```

```r
boxplot.stats(subset$PER)$out
```

```
##  [1] -1.4 26.1 27.5 27.6 -2.1 27.3 -2.2 -2.2 26.1 30.8 27.0 26.4 27.5 29.6
## [15] -3.4 -4.3  0.1 -1.9 26.2 31.5 -1.2  0.3 26.5 26.0 30.6
```

```r
boxplot.stats(subset$VORP)$out
```

```
##  [1]  6.9  2.6  2.8  2.9  6.3  4.5  6.2  3.9  2.5  2.9  5.2  4.0  3.2  5.4
## [15]  4.6  3.4  9.0  4.0  2.8  2.5  2.9  7.3  2.5  5.3  3.9  6.2  4.3  4.9
## [29]  2.7  5.2  3.9  2.6  4.8  5.4  2.8  3.9  4.3 12.4
```

```r
boxplot.stats(subset$X2P.)$out
```

```
##  [1] 0.167 0.273 0.200 0.231 0.000 1.000 0.717 0.222 0.200 0.000 0.694
## [12] 0.714 0.222 0.143 0.250 1.000 1.000 0.231 1.000 0.750 0.200 0.294
## [23] 0.200 0.306 0.250
```

```r
boxplot.stats(subset$X3P.)$out
```

```
##  [1] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
## [12] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.000
## [23] 0.000 0.000 0.600 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.000
## [34] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
## [45] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
## [56] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.625
## [67] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
## [78] 0.000 0.000 0.000 0.000 0.000 0.000
```

```r
boxplot.stats(subset$FG.)$out
```

```
##  [1] 0.625 0.643 0.671 0.622 0.100 0.214 0.662 0.652 0.200 0.000 0.617
## [12] 0.633 0.667 0.750 0.714 0.652 0.642 0.143 0.000 0.250 0.660 0.714
## [23] 0.200 0.167 0.250 0.625 1.000 0.231 1.000 0.750 0.143 0.268 0.250
## [34] 0.258
```

```r
boxplot.stats(subset$TS.)$out
```

```
##  [1] 0.355 0.703 0.387 0.381 0.184 0.306 0.798 0.239 0.102 0.773 0.753
## [12] 0.357 0.190 0.228 0.285 0.799 0.276 0.272 0.820 0.798 0.339 0.190
## [23] 0.378 0.376 0.392 0.329 0.346
```

```r
boxplot.stats(subset$USG.)$out
```

```
## [1] 33.1 32.6 34.3 36.0 34.2 35.4 39.9 34.0 41.7
```

```r
boxplot.stats(subset$Salary.in..)$out
```

```
##  [1] 20575005 24559380 22116750 20869566 22116750 26540100 22116750
##  [8] 26540100 22116750 26540100 21165675 20140838 26540100 26540100
## [15] 23180275 30963450 21165675 24328425 21165675 21165675 20072033
## [22] 25000000 22116750 22868827 21323250 23200000 26540100 22116750
```

We see that all of them correspond to different players, so those are possible values that we can leave as they are.

3. Data analysis

In this section, we will state three questions that we want to answer with the dataset that we constructed in the previous section. Then, analyses will be made in order to answer the questions.

First, we start by some preliminary exploratory analysis.

3.1. Exploratory analysis

Let us explore the players with the highest values of some of the statistics: MPG, PPG, APG, RPG, TOPG, PFPG and Salary. We will output the 10 players with the higher values for the statistics.

```r
# MPG

subset %>%
  select(c(Player, Pos, MPG )) %>%
  arrange(desc(MPG)) %>%
  head (n=10)
```

```
##                Player Pos      MPG
## 1        LeBron James  SF 37.75676
## 2          Kyle Lowry  PG 37.40000
## 3         Zach LaVine  SG 37.21277
## 4      Andrew Wiggins  SF 37.17073
## 5        Jimmy Butler  SF 36.96053
## 6  Karl-Anthony Towns   C 36.95122
## 7        James Harden  PG 36.38272
## 8           John Wall  PG 36.35897
## 9       Anthony Davis   C 36.10667
## 10     Damian Lillard  PG 35.92000
```

```r
# PPG

subset %>%
  select(c(Player, Pos, PPG )) %>%
  arrange(desc(PPG)) %>%
  head (n=10)
```

```
##                Player Pos      PPG
## 1  Russell Westbrook  PG 31.58025
## 2       James Harden  PG 29.08642
## 3      Isaiah Thomas  PG 28.93421
## 4      Anthony Davis   C 27.98667
## 5      DeMar DeRozan  SG 27.29730
## 6     Damian Lillard  PG 26.98667
## 7       LeBron James  SF 26.40541
## 8      Kawhi Leonard  SF 25.51351
## 9      Stephen Curry  PG 25.30380
## 10      Kyrie Irving  PG 25.22222
```

```r
# APG

subset %>%
  select(c(Player, Pos, APG )) %>%
  arrange(desc(APG)) %>%
  head (n=10)
```

```
##                Player Pos       APG
## 1       James Harden  PG 11.185185
## 2          John Wall  PG 10.653846
## 3  Russell Westbrook  PG 10.370370
## 4         Chris Paul  PG  9.229508
## 5        Ricky Rubio  PG  9.093333
## 6       LeBron James  SF  8.729730
## 7        Jeff Teague  PG  7.792683
## 8       Jrue Holiday  PG  7.283582
## 9     Draymond Green  PF  7.013158
## 10        Kyle Lowry  PG  6.950000
```

```r
# RPG

subset %>%
  select(c(Player, Pos, RPG )) %>%
  arrange(desc(RPG)) %>%
  head (n=10)
```

```
##                 Player Pos      RPG
## 1    Hassan Whiteside   C 14.12987
## 2      Andre Drummond   C 13.77778
## 3      DeAndre Jordan   C 13.75309
## 4         Rudy Gobert   C 12.77778
## 5       Dwight Howard   C 12.70270
## 6    DeMarcus Cousins   C 12.47059
## 7  Karl-Anthony Towns   C 12.28049
## 8       Anthony Davis   C 11.81333
## 9      Tyson Chandler   C 11.46809
## 10         Kevin Love  PF 11.10000
```

```r
# TOPG
subset %>%
  select(c(Player, Pos, TOPG )) %>%
```

```r
  arrange(desc(TOPG)) %>%
  head (n=10)
```

```
##                Player Pos     TOPG
## 1        James Harden  PG 5.728395
## 2   Russell Westbrook  PG 5.407407
## 3           John Wall  PG 4.141026
## 4        LeBron James  SF 4.094595
## 5         Joel Embiid   C 3.774194
## 6     DeMarcus Cousins   C 3.647059
## 7        Eric Bledsoe  PG 3.378788
## 8      Dennis Schroder  PG 3.265823
## 9         Jusuf Nurkic   C 3.100000
## 10       Devin Booker  SG 3.089744
```

```r
# PFPG

subset %>%
  select(c(Player, Pos, PFPG)) %>%
  arrange(desc(PFPG)) %>%
  head (n=10)
```

```
##                 Player Pos     PFPG
## 1     DeMarcus Cousins   C 4.411765
## 2   Kristaps Porzingis  PF 3.696970
## 3          Jusuf Nurkic   C 3.650000
## 4           Joel Embiid   C 3.612903
## 5        Julius Randle  PF 3.351351
## 6       Markieff Morris  PF 3.342105
## 7     Patrick Beverley  SG 3.313433
## 8           Serge Ibaka  PF 3.304348
## 9          Myles Turner   C 3.234568
## 10       JaMychal Green  PF 3.220779
```

```r
# Salary

subset %>%
  select(c(Player, Pos, Salary.in.. )) %>%
  arrange(desc(Salary.in..)) %>%

  head (n=10)
```

```
##                Player Pos Salary.in..
## 1        LeBron James  SF    30963450
## 2         Mike Conley  PG    26540100
## 3       DeMar DeRozan  SG    26540100
## 4        Kevin Durant  SF    26540100
## 5        James Harden  PG    26540100
## 6          Al Horford   C    26540100
## 7   Russell Westbrook  PG    26540100
## 8       Dirk Nowitzki  PF    25000000
## 9     Carmelo Anthony  SF    24559380
## 10     Damian Lillard  PG    24328425
```

3.2. Are the Win Shares the factor that is more correlated to the salary of a player?

We start with the hypothesis that whe Win Shares (WS) is the factor that contributes the most to a player's salary. Let's see if that hypothesis is true.

If not, we will examine some other factors to determine which is the one more related to the salary of a player.

First, let's check who are the 30 players with the higher WS and see how much they are getting paid

```r
# 30 players with higher WS
subset %>%
  select(c(Player, Tm, Pos,  WS,  Salary.in.. )) %>%
  arrange(desc(WS)) %>%
  head(n=30)
```

```
##                   Player  Tm Pos   WS Salary.in..
## 1           James Harden HOU  PG 15.0    26540100
## 2            Rudy Gobert UTA   C 14.3     2121288
## 3           Jimmy Butler CHI  SF 13.8    17552209
## 4          Kawhi Leonard SAS  SF 13.6    17638063
## 5      Russell Westbrook OKC  PG 13.1    26540100
## 6           LeBron James CLE  SF 12.9    30963450
## 7     Karl-Anthony Towns MIN   C 12.7     5960160
## 8          Stephen Curry GSW  PG 12.6    12112359
## 9          Isaiah Thomas BOS  PG 12.6     6587132
## 10 Giannis Antetokounmpo MIL  SF 12.4     2995421
## 11          Kevin Durant GSW  SF 12.0    26540100
## 12        DeAndre Jordan LAC   C 11.8    21165675
## 13         Anthony Davis NOP   C 11.0    22116750
## 14            Chris Paul LAC  PG 10.6    22868827
## 15        Gordon Hayward UTA  SF 10.4    16073140
## 16        Damian Lillard POR  PG 10.3    24328425
## 17            Kyle Lowry TOR  PG 10.1    12000000
## 18           Mike Conley MEM  PG 10.0    26540100
## 19          Nikola Jokic DEN   C  9.7     1358500
## 20       Hassan Whiteside MIA   C  9.5    22116750
## 21           Otto Porter WAS  SF  9.4     5893981
## 22         DeMar DeRozan TOR  SG  9.0    26540100
## 23          Kyrie Irving CLE  PG  8.9    17638063
## 24             John Wall WAS  PG  8.8    16957900
## 25          Bradley Beal WAS  SG  8.5    22116750
## 26         Dwight Howard ATL   C  8.3    23180275
## 27        Draymond Green GSW  PF  8.2    15330435
## 28           Jeff Teague IND  PG  8.1     8800000
## 29          Kemba Walker CHO  PG  8.1    12000000
## 30          Myles Turner IND   C  8.0     2463840
```

In general, those players have high salaries, except for some exceptions like Antetokounmpo or Jokic, who probably were on their rookie deals.

As a curiosity, let's calculate the Win shares for the teams:

```r
#teams with more WS
stats_with_salaries %>%
  group_by(Tm) %>%
  summarise( sum_WS = sum(WS), sum_salary = sum(Salary.in..)) %>%
  select(c(Tm, sum_WS, sum_salary ))  %>%
  arrange(desc(sum_WS))
```

```
## # A tibble: 30 x 3
##    Tm    sum_WS sum_salary
##    <chr>  <dbl>      <dbl>
##  1 GSW     66.8   99365032
##  2 SAS     60.4  105395531
##  3 HOU     54.3   87392332
##  4 UTA     52.7   80323193
##  5 TOR     51.2  106868829
##  6 LAC     48.7  111299700
##  7 CLE     48.6  127012355
##  8 BOS     48.1   91915088
##  9 WAS     46.5   99594431
## 10 MIA     44.6   77542376
## # ... with 20 more rows
```

We see that it is almost identical to the number of victories of the team that season.

3.2.1. Normality and homogeneity check for the variance

We want to check the correlation between the WS and the salaries. We will check if the variables are normally distributed and homocedastic

```r
#normality test shapiro-wilk WS and salaries

shapiro.test(stats_with_salaries$WS)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  stats_with_salaries$WS
## W = 0.81405, p-value < 2.2e-16
```

```r
shapiro.test(stats_with_salaries$Salary.in..)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  stats_with_salaries$Salary.in..
## W = 0.81072, p-value < 2.2e-16
```

```r
#homocedasticity

fligner.test(WS ~ Salary.in.., data = subset)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  WS by Salary.in..
## Fligner-Killeen:med chi-squared = 383.55, df = 340, p-value =
## 0.05164
```

3.2.2. Computing the correlation

We saw that the variables are normal but there isn't homocedasticity, so the Pearson correlation cannot be applied. Instead, we shall apply the Spearman correlation for these two variables. Let us see the results:

```
# correlation: WS and Salary
cor(stats_with_salaries$WS, stats_with_salaries$Salary.in.., method = 'spearman')
```

```
## [1] 0.5858834
```

The result is a correlation of 0,58. There is some positive correlation, but is not a great one. Let's check some other variables, to see if there is another one which is more positively correlated to the salary:

```
subset1 <- stats_with_salaries %>%
  select(c(Player, Tm, Pos, Age, TS., WS, PER, G,
           PPG, APG, RPG, BPG, SPG, TOPG, MPG, Salary.in.. ))

cor(subset1$Salary.in.., subset1[4:15], method="spearman")
```

```
##            Age       TS.        WS       PER         G       PPG       APG
## [1,] 0.3863301 0.1910448 0.5858834 0.3985885 0.4902147 0.6183281 0.4207545
##            RPG       BPG       SPG      TOPG       MPG
## [1,] 0.5430776 0.3554451 0.4625887 0.5144228 0.6280021
```

We see that the variable that is more positively correlated to the salary is MPG, closely followed by PPG.

It can be concluded that, usually, the players that earn the most are also the ones that play more minutes per game and the ones that score more points per game. This makes sense, as if you have to spend more money on a certain player, you would expect him to ve valuable for the team and thus make him play more minutes and allow him to shoot more often.

The teams value more the points and the minutes, rather than some advanced metrics like the PER, or the True Shooting %. Nevertheless, the metric of Win Shares follows closely the PPG and MPG in correlation with salary.

Let's compute the correlation between salaries and FG.(field goal %) to check if the players who are getting paid the most are also the ones who shoot more efficiently.

```
cor.test(subset$FG., subset$Salary.in.., method = 'spearman',  exact=F)
```

```
##
##  Spearman's rank correlation rho
##
## data:  subset$FG. and subset$Salary.in..
## S = 14232000, p-value = 0.0001784
```

```
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.1722231
```

The correlation is not so high, so being an efficient scorer seems not to be highly correlated with a higher salary.

3.3. Is there any significant difference in any of the main statistics for the different five positions of the game?

Let's check if there are any differences in some of the different stats for the five player positions. If any differences are found, We will perform different statistical tests to check if those differences are significative or not.

3.3.1. Analysis planification

```
subset2 <- stats_with_salaries %>%
  select(c(Player, Tm, Pos, Age, WS, PPG, APG, MPG, Salary.in.. ))
  #select(c(Player, Tm, Pos, Age, TS., WS, PER, G, PPG, APG, RPG, BPG, SPG, MPG, Salary.in.. ))
```

We select the five analysis groups, i.e., the five types of players in the game. Let us see a sample of these groups.

```
group.PG <- subset2 %>%  filter(Pos == 'PG')

head(group.PG)
```

```
##           Player  Tm Pos Age   WS       PPG      APG      MPG Salary.in..
## 1  D.J. Augustin ORL  PG  29  1.2  7.897436 2.679487 19.71795     7250000
## 2   Wade Baldwin MEM  PG  20 -0.4  3.212121 1.848485 12.27273     1793760
## 3     J.J. Barea DAL  PG  32  1.3 10.885714 5.514286 22.02857     4096950
## 4 Jerryd Bayless PHI  PG  28 -0.1 11.000000 4.333333 23.66667     9424084
## 5   Eric Bledsoe PHO  PG  27  5.4 21.060606 6.333333 32.96970    14000000
## 6   Aaron Brooks IND  PG  32  0.3  4.953846 1.923077 13.75385     2700000
```

```
group.SG <- subset2 %>%  filter(Pos == 'SG')

head(group.SG)
```

```
##            Player  Tm Pos Age   WS      PPG       APG      MPG Salary.in..
## 1    Alex Abrines OKC  SG  23  2.1 5.970588 0.5882353 15.51471     5994764
## 2   Arron Afflalo SAC  SG  31  1.4 8.442623 1.2786885 25.90164    12500000
## 3      Tony Allen MEM  SG  35  3.1 9.056338 1.3802817 26.95775     5505618
## 4   Kyle Anderson SAS  SG  23  2.7 3.416667 1.2638889 14.16667     1192080
## 5       Ron Baker NYK  SG  23 -0.1 4.134615 2.0576923 16.48077      543471
## 6 Leandro Barbosa PHO  SG  34  0.6 6.253731 1.2089552 14.37313     4000000
```

```r
group.SF <- subset2 %>%  filter(Pos == 'SF')

head(group.SF)
```

```
##                  Player  Tm Pos Age   WS       PPG       APG      MPG
## 1       Al-Farouq Aminu POR  SF  26  1.9  8.721311 1.6229508 29.06557
## 2         Alan Anderson LAC  SF  34  0.1  2.866667 0.3666667 10.26667
## 3       Justin Anderson PHI  SF  23  0.9  8.458333 1.4166667 21.58333
## 4 Giannis Antetokounmpo MIL  SF  22 12.4 22.900000 5.4250000 35.56250
## 5       Carmelo Anthony NYK  SF  32  4.7 22.418919 2.8783784 34.29730
## 6          Trevor Ariza HOU  SF  31  6.0 11.700000 2.1875000 34.66250
##   Salary.in..
## 1     7680965
## 2     1315448
## 3     1514160
## 4     2995421
## 5    24559380
## 6     7806971
```

```r
group.PF <- subset2 %>%  filter(Pos == 'PF')

head(group.PF)
```

```
##              Player  Tm Pos Age   WS       PPG       APG      MPG
## 1        Quincy Acy DAL  PF  26 -0.1  2.166667 0.0000000  8.00000
## 2        Quincy Acy BRK  PF  26  1.1  6.531250 0.5625000 15.93750
## 3 LaMarcus Aldridge SAS  PF  31  7.3 17.263889 1.9305556 32.43056
## 4       Lavoy Allen IND  PF  27  1.7  2.901639 0.9344262 14.27869
## 5     Ryan Anderson HOU  PF  28  5.2 13.597222 0.9444444 29.38889
## 6    Darrell Arthur DEN  PF  28  1.1  6.390244 1.0243902 15.58537
##   Salary.in..
## 1     1050961
## 2     1914544
## 3    20575005
## 4     4000000
## 5    18735364
## 6     8070175
```

```r
group.C <- subset2 %>%  filter(Pos == 'C')

head(group.C)
```

```
##          Player  Tm Pos Age  WS       PPG       APG       MPG Salary.in..
## 1  Steven Adams OKC   C  23 6.4 11.312500 1.0750000 29.862500     3140517
## 2 Alexis Ajinca NOP   C  28 1.0  5.307692 0.3076923 14.974359     4600000
## 3  Cole Aldrich MIN   C  28 1.3  1.693548 0.4032258  8.564516     7643979
## 4  Joel Anthony SAS   C  34 0.4  1.315789 0.1578947  6.421053      663810
## 5     Omer Asik NOP   C  30 1.0  2.741935 0.4838710 15.548387     9904494
## 6   Aron Baynes DET   C  30 3.0  4.866667 0.4266667 15.506667     6500000
```

We are going to study if there are any significative differences in the following variables, depending on the position of the player:

- Salary
- Points per Game
- Assists per Game
- Minutes per Game
- Win shares

First of all, we must check if each of those variables are normally distributed and if there is homocedasticity. If the answer is affirmative, then an ANOVA test can be performed. If the answer is negative, we should perform a non-parametric test like the Kruskal-Wallis.

3.3.1.1. Salary

```
# normality test

shapiro.test(subset2$Salary.in..)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  subset2$Salary.in..
## W = 0.81072, p-value < 2.2e-16
```

```
# homocedasticity test
leveneTest(Salary.in..  ~ Pos, data = subset2)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group   4  0.6833 0.6038
##       464
```

The variable is normally distributed, but there is no homocedasticity. Thus, we will apply the Kruskal-Wallis test to see if there are any differences with respect to the Salary by position.

```
test_salary <- kruskal.test(Salary.in..~ Pos, data= subset2)
test_salary
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  Salary.in.. by Pos
## Kruskal-Wallis chi-squared = 7.7001, df = 4, p-value = 0.1032
```

There is no evidence that there exist a difference on the salary by the different position that a player has.

3.3.1.2. PPG

```r
# normality test

shapiro.test(subset2$PPG)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  subset2$PPG
## W = 0.89668, p-value < 2.2e-16
```

```r
# homocedasticity test

leveneTest(PPG  ~ Pos, data = subset2)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group   4  1.0646 0.3735
##       464
```

The variable is normally distributed, but there is no homocedasticity. Thus, we will apply the Kruskal-Wallis test to see if there are any differences with respect to the PPG by position.

```r
test_PPG <-  kruskal.test(PPG~ Pos, data= subset2)

test_PPG
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  PPG by Pos
## Kruskal-Wallis chi-squared = 4.4811, df = 4, p-value = 0.3448
```

After running the test, no evidence was found that the PPG change significatively according to the player's position on the field.

3.3.1.3. APG

```r
# diferencias en APG

shapiro.test(subset2$APG)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  subset2$APG
## W = 0.79879, p-value < 2.2e-16
```

```r
# homocedasticity test

leveneTest(APG  ~ Pos, data = subset2)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   4  18.472 4.253e-14 ***
##       464
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The variable is normally distributed and there is homocedasticity, so we can apply the ANOVA test to see if there are any differences with respect to the APG by position.

```
# normal--- we can do the anova test
test_APG <- aov(APG~ Pos, data= subset2)
summary(test_APG)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## Pos           4  397.8   99.45   42.44 <2e-16 ***
## Residuals   464 1087.3    2.34
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(test_APG)[[1]]$'Pr(>F)'[1]
```

```
## [1] 2.435007e-30
```

In this case, the Assists per Game depend on the position that the player has on the field. The p-value is really small (less than 10^{-16}). Later, we will explore which of the possitions has a higher number of APG.

3.3.1.4. MPG

```
# normality test

shapiro.test(subset2$MPG)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  subset2$MPG
## W = 0.97304, p-value = 1.324e-07
```

```
# homocedasticity test
leveneTest(MPG  ~ Pos, data = subset2)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group   4  0.4066 0.8039
##       464
```

The variable is normally distributed, but there is no homocedasticity. Thus, we will apply the Kruskal-Wallis test to see if there are any differences with respect to the MPG by position.

```r
test_MPG <- kruskal.test(MPG~ Pos, data= subset2)
test_MPG
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  MPG by Pos
## Kruskal-Wallis chi-squared = 13.891, df = 4, p-value = 0.007651
```

In the case of the MPG, we found that there is evidence to support the claim that the minutes played depend on the position. In this case, the p-value is still small (in the order of 10^{-3}).

3.3.1.5. WS

```r
# normality test

shapiro.test(subset2$WS)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  subset2$WS
## W = 0.81405, p-value < 2.2e-16
```

```r
leveneTest(WS  ~ Pos, data = subset2)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group   4  3.3091 0.0109 *
##       464
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The variable is normally distributed and there is homocedasticity, so we can apply the ANOVA test to see if there are any differences with respect to the WS by position.

```r
# normal--- we can do the anova test
test_WS <- aov(WS~ Pos, data= subset2)
summary(test_WS)
```

```
##              Df Sum Sq Mean Sq F value  Pr(>F)
## Pos           4    114  28.465   3.408 0.00921 **
## Residuals   464   3875   8.352
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In this case, the WS also have some dependence on the position played, but the p-value was closer to the significance level (0,009) than in the other two variables, meaning that further investigations should be made in order to determine whether there is a true effect or just a result of chance.

Let us see a summary of the results of the analyses.

```
DT = data.frame(
  Variable = c("Salary","PPG","APG","MPG","WS"),
  Result = c("No significative difference observed",
             "No significative difference observed",
             "Significative difference",
             "Significative difference",
             "Significative difference"),
  p.value = c(sprintf("%0.3f", test_salary$p.value),
            sprintf("%0.3f", test_PPG$p.value),
            sprintf("%0.2e", summary(test_APG)[[1]]$'Pr(>F)'[1]),
            sprintf("%0.2e", test_MPG$p.value),
            sprintf("%0.3f", summary(test_WS)[[1]]$'Pr(>F)'[1]))
)

DT
```

```
##   Variable                               Result  p.value
## 1   Salary No significative difference observed    0.103
## 2      PPG No significative difference observed    0.345
## 3      APG             Significative difference 2.44e-30
## 4      MPG             Significative difference 7.65e-03
## 5       WS             Significative difference    0.009
```

3.3.2. Further analysis with WS, APG and MPG.

Now, let us check the source of the differences in the variables WS, APG and MPG are coming from, i.e., which possitions have a higher value for those variables.

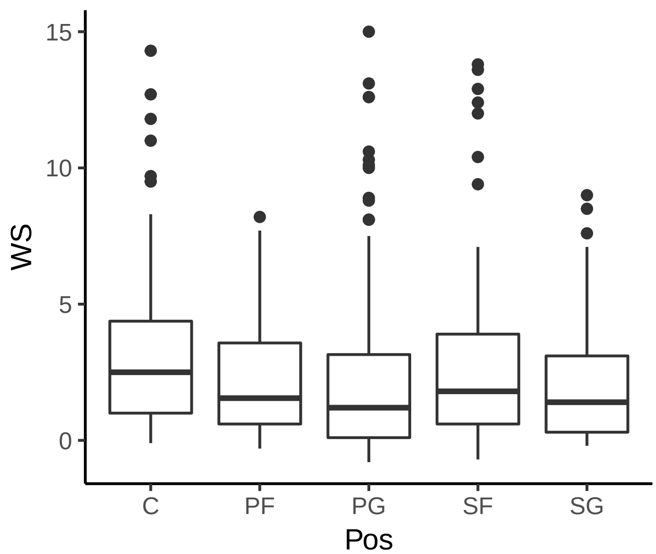
3.3.2.1. WS

```
# WS
summary(lm(WS ~ Pos, data = subset2))
```

```
##
## Call:
## lm(formula = WS ~ Pos, data = subset2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5576 -1.9752 -0.9273  1.1424 12.4677
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.2793     0.3013  10.884  < 2e-16 ***
## PosPF        -1.0521     0.4309  -2.442 0.014998 *
## PosPG        -0.7470     0.4185  -1.785 0.074909 .
## PosSF        -0.4217     0.4348  -0.970 0.332589
## PosSG        -1.4041     0.4127  -3.402 0.000726 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 2.89 on 464 degrees of freedom
## Multiple R-squared:  0.02854,    Adjusted R-squared:  0.02017
## F-statistic: 3.408 on 4 and 464 DF,  p-value: 0.009211
```

```
subset2 %>%
  ggplot(aes(x = Pos, y = WS)) +
  geom_boxplot()+
  theme_classic()
```



It seems that the positions with the higher WS are the Centers, followed by the Small Forwards.

3.3.2.2. APG

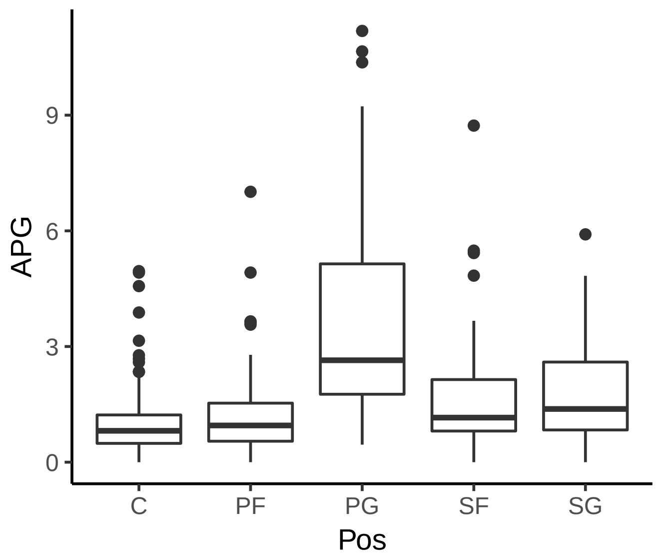
```
# APG
summary(lm(APG ~ Pos, data = subset2))
```

```
##
## Call:
## lm(formula = APG ~ Pos, data = subset2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1482 -0.9170 -0.3549  0.6379  7.5825
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.09569    0.15959   6.865 2.13e-11 ***
## PosPF        0.09255    0.22825   0.405  0.68531
## PosPG        2.50704    0.22167  11.310  < 2e-16 ***
## PosSF        0.56473    0.23030   2.452  0.01457 *
## PosSG        0.67383    0.21860   3.082  0.00218 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 1.531 on 464 degrees of freedom
## Multiple R-squared:  0.2679, Adjusted R-squared:  0.2615
## F-statistic: 42.44 on 4 and 464 DF,  p-value: < 2.2e-16
```

We can plot those differences in a boxplot:

```r
subset2 %>%
  ggplot(aes(x = Pos , y = APG)) +
  geom_boxplot()+
  theme_classic()
```




```r
pairwise.t.test(subset2$APG, subset2$Pos)
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  subset2$APG and subset2$Pos
##
##    C       PF      PG      SF
## PF 1.000   -       -       -
## PG < 2e-16 < 2e-16 -       -
## SF 0.058   0.129   1.1e-15 -
## SG 0.013   0.044   1.3e-15 1.000
##
## P value adjustment method: holm
```

The results of the pairwise t-test mean that the PG has significant differences in APG with every other position in the game, while the other positions (except for SG with PF and C) do not have significant differences betweent each other.

It is quite clear that the PG are the ones with the higher APG.

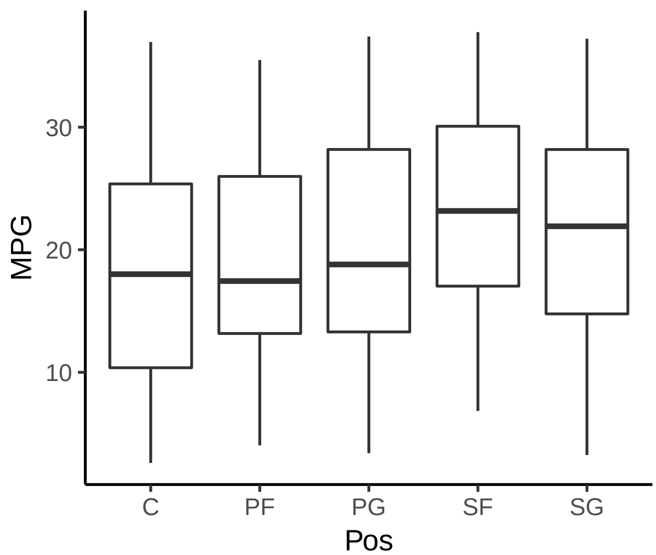
3.3.2.3. MPG

Let us repeat the analysis for the MPG attribute.

```r
# MPG
summary(lm(MPG ~ Pos, data = subset2))
```

```
##
## Call:
## lm(formula = MPG ~ Pos, data = subset2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.8959  -6.7727  -0.4421   6.9765  18.5709
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  18.3803     0.9067  20.271  < 2e-16 ***
## PosPF         0.9271     1.2968   0.715 0.475035
## PosPG         2.2422     1.2595   1.780 0.075681 .
## PosSF         4.6387     1.3085   3.545 0.000432 ***
## PosSG         2.7656     1.2420   2.227 0.026448 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.697 on 464 degrees of freedom
## Multiple R-squared:  0.03101,    Adjusted R-squared:  0.02265
## F-statistic: 3.712 on 4 and 464 DF,  p-value: 0.005493
```

```r
subset2 %>%
  ggplot(aes(x = Pos, y = MPG)) +
  geom_boxplot()+
  theme_classic()
```



It seems that the Small Forwards are getting more minutes per game. Let us see how many players are in the league for every position in the 2016-17 season.

```r
summary(subset$Pos)
```

```
##    C   PF PF-C   PG   SF   SG
##   92   88    0   99   85  105
```

We see that the SF is the position with less players, which can be the reason that is the position with the higher MPG.

3.4. Salary prediction model

We are going to use the following subset of variables to predict the salary:

```r
subset3 <- subset %>%
  select(-c(Tm, Player))

names(subset3)
```

```
##  [1] "Pos"          "Age"          "G"            "GS"           "MPG"
##  [6] "PPG"          "APG"          "RPG"          "BPG"          "SPG"
## [11] "TOPG"         "PFPG"         "WS"           "PER"          "VORP"
## [16] "X2P."         "X3P."         "FG."          "TS."          "USG."
## [21] "Salary.in.."
```

3.4.1. Models creation

Let us create different models and compare their adjusted R-squared as an indicator of which model will make a better adjustment to the data.

Let us first use a model which uses all the variables in the subset. Let us display the summary and make some comments about it.

```r
# all variables
model1 <- lm(Salary.in.. ~ .,
             data=subset3)

summary(model1)
```

```
##
## Call:
## lm(formula = Salary.in.. ~ ., data = subset3)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -11522379  -2537283   -287102   2510987  16497540
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3700675    3181663  -1.163 0.245402
```

```
## PosPF        -881931     765393  -1.152 0.249833
## PosPG       -3005998    1037316  -2.898 0.003943 **
## PosSF       -2011648     924789  -2.175 0.030137 *
## PosSG       -2635173     961622  -2.740 0.006384 **
## Age           403168      48388   8.332 9.95e-16 ***
## G              10859      13049   0.832 0.405734
## GS             40041      13226   3.027 0.002610 **
## MPG          -142218     114103  -1.246 0.213275
## PPG           807431     201991   3.997 7.50e-05 ***
## APG           705696     378200   1.866 0.062708 .
## RPG           719306     228368   3.150 0.001744 **
## BPG          -414566     646726  -0.641 0.521838
## SPG          1147922     943379   1.217 0.224318
## TOPG         -361199     972248  -0.372 0.710435
## PFPG        -1876593     492161  -3.813 0.000157 ***
## WS             75524     292422   0.258 0.796318
## PER          -165753     129563  -1.279 0.201448
## VORP         -582967     444824  -1.311 0.190684
## X2P.        -5850339    4627825  -1.264 0.206832
## X3P.         -144102    1826245  -0.079 0.937143
## FG.          6099856    6574182   0.928 0.353988
## TS.         -2616527    6659917  -0.393 0.694599
## USG.         -101227      90498  -1.119 0.263931
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4248000 on 445 degrees of freedom
## Multiple R-squared:  0.5937, Adjusted R-squared:  0.5727
## F-statistic: 28.27 on 23 and 445 DF,  p-value: < 2.2e-16
```

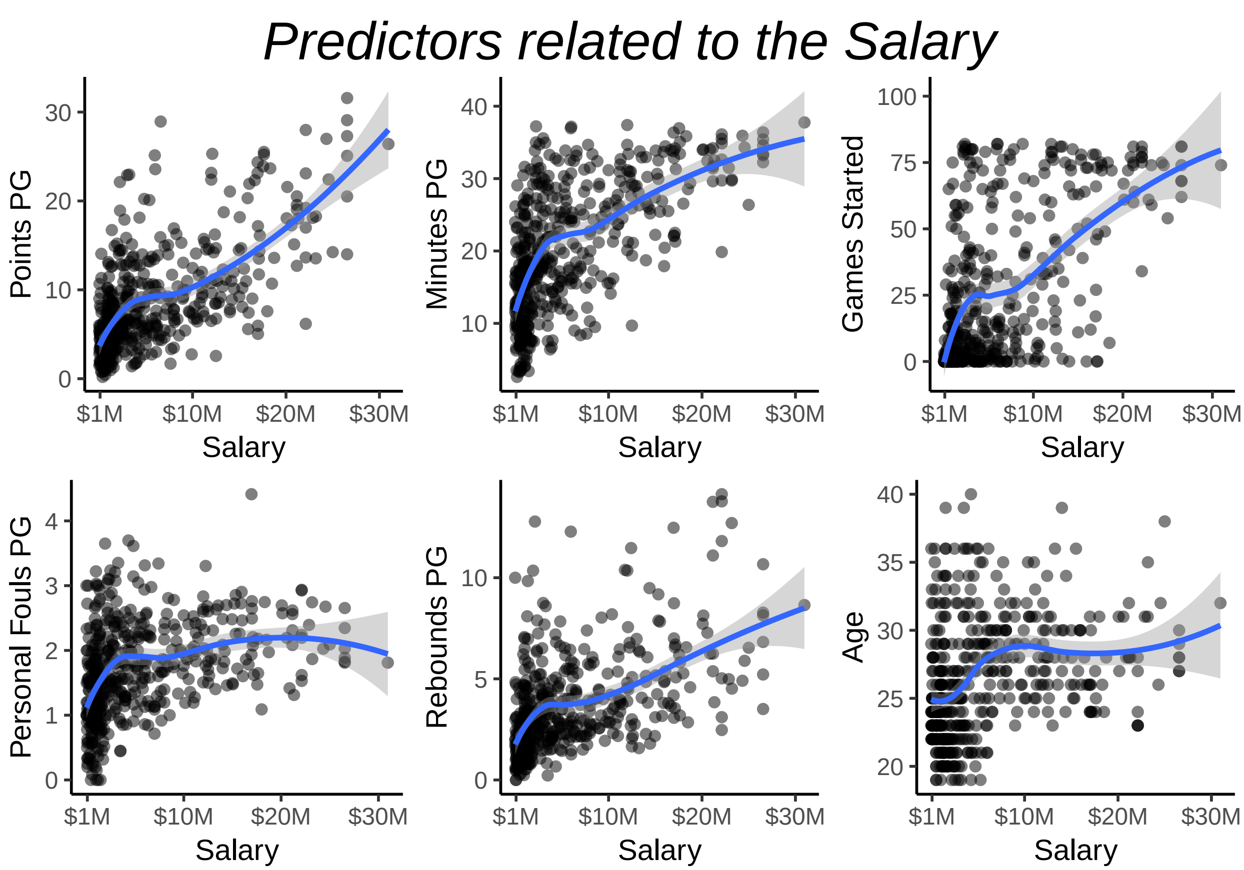
We see that some of the variables that seem to explain better the salary are the Position, the Games Started, PPG, RPG and PFPG.

Let us create a plot to see how the variables with the most significant p-value from the previous output (those are: PPG, MPG, RPG, GS, PFPG and AGE) are related to the Salary. The following code gives the desired plot.

```r
# Salary vs PPG
p1 <- ggplot(subset, aes(x=Salary.in.., y=PPG)) +
  geom_point(alpha=0.5) + geom_smooth() +
  labs(x="Salary", y="Points PG") +
  theme(legend.position="none") +
  scale_x_continuous(breaks = c(100000, 10000000,
                                20000000, 30000000),
                     labels = c("$1M", "$10M",
                                "$20M", "$30M"))+
  theme_classic() +
  theme(
    plot.margin = margin(3, 7, 3, 1.5)
  )


# Salary vs MPG
p2 <- ggplot(subset, aes(x=Salary.in.., y=MPG)) +
```

```r
  geom_point(alpha=0.5) + geom_smooth() +
  labs(x="Salary", y="Minutes PG") +
  theme(legend.position="none")+
  scale_x_continuous(breaks = c(100000, 10000000,
                                20000000, 30000000),
                     labels = c("$1M", "$10M",
                                "$20M", "$30M"))+
  theme_classic() +
  theme(
    plot.margin = margin(3, 7, 3, 1.5)
  )

# Salary vs GS
p3 <- ggplot(subset, aes(x=Salary.in.., y=GS)) +
  geom_point(alpha=0.5) + geom_smooth() +
  labs(x="Salary", y="Games Started") +
  theme(legend.position="none")+
  scale_x_continuous(breaks = c(100000, 10000000,
                                20000000, 30000000),
                     labels = c("$1M", "$10M",
                                "$20M", "$30M"))+
  theme_classic() +
  theme(
    plot.margin = margin(3, 7, 3, 1.5)
  )


# Salary vs PFPG
p4 <- ggplot(subset, aes(x=Salary.in.., y=PFPG)) +
  geom_point(alpha=0.5) + geom_smooth() +
  labs(x="Salary", y="Personal Fouls PG") +
  theme(legend.position="none")+
  scale_x_continuous(breaks = c(100000, 10000000,
                                20000000, 30000000),
                     labels = c("$1M", "$10M",
                                "$20M", "$30M"))+
  theme_classic() +
  theme(
    plot.margin = margin(3, 7, 3, 1.5)
  )

# Salary vs RPG
p5 <- ggplot(subset, aes(x=Salary.in.., y=RPG)) +
  geom_point(alpha=0.5) + geom_smooth() +
  labs(x="Salary", y= "Rebounds PG") +
  theme(legend.position="none")+
  scale_x_continuous(breaks = c(100000, 10000000,
                                20000000, 30000000),
                     labels = c("$1M", "$10M",
                                "$20M", "$30M"))+
  theme_classic() +
  theme(
    plot.margin = margin(3, 7, 3, 1.5)
```

```
  )

# Salary vs Age
p6 <- ggplot(subset, aes(x=Salary.in.., y=Age)) +
  geom_point(alpha=0.5) + geom_smooth() +
  labs(x="Salary", y="Age") +
  scale_x_continuous(breaks = c(100000, 10000000,
                                20000000, 30000000),
                     labels = c("$1M", "$10M",
                                "$20M", "$30M"))+
  theme_classic() +
  theme(
    plot.margin = margin(3, 7, 3, 1.5)
  )

# Represent all the subplots
grid.arrange(p1, p2, p3, p4, p5, p6,
             layout_matrix=cbind(c(1,4), c(2,5), c(3,6)),
             top = textGrob("Predictors related to the Salary",
                            gp=gpar(fontsize=20,font=3)))
```



It seems that the variables whose correlation is more clear with the salary are MPG, PPG and GS.

Let us build some more linear regression models. Then, we will select the better performing one.

```r
model2 <- lm(Salary.in.. ~ APG + RPG + PPG + Age + BPG + MPG*PPG
             + Age*MPG,
             data=subset3)
```

```r
model3 <- lm(Salary.in.. ~ MPG +  RPG + Age + BPG + MPG*PPG + Age*MPG,
             data=subset3)
```

```r
model4 <- lm(Salary.in.. ~  Age*MPG + WS + TOPG + PFPG + GS,
             data=subset3)
```

```r
model5 <- lm(Salary.in.. ~  I(Age)^2 + I(PPG)^3+ Age*MPG +
               Pos + RPG + TS. + GS + APG*TOPG+ USG.,
             data=subset3)
```

```r
model6 <- lm(Salary.in.. ~  I(Age)^2 + I(PPG)^3+  I(PFPG)^2+
               Age*MPG + Pos + I(RPG)^2 + TS. + I(GS)^2 + APG*TOPG+
               USG.*TOPG + VORP + WS, data=subset3)
```

We print the results in the following table. We include the F-statistic as an indicator of wheter there is a relationship between the Response and the Predictors (the larger the F-statistic, the more relationship there is).

```r
models.table <-
  data.frame(c(1, 2, 3 ,4 ,5, 6),
                         c(summary(model1)$adj.r.squared,
                           summary(model2)$adj.r.squared,
                           summary(model3)$adj.r.squared,
                           summary(model4)$adj.r.squared,
                           summary(model5)$adj.r.squared,
                           summary(model6)$adj.r.squared),
                         c(summary(model1)$fstatistic[1],
                           summary(model2)$fstatistic[1],
                           summary(model3)$fstatistic[1],
                           summary(model4)$fstatistic[1],
                           summary(model5)$fstatistic[1],
                           summary(model6)$fstatistic[1]))

names(models.table) <- c("Model", "Adj. R-squared", "F-statistic")

models.table
```

```
##   Model Adj. R-squared F-statistic
## 1     1      0.5726652    28.26783
## 2     2      0.5957975    87.22942
## 3     3      0.5966328    99.89043
## 4     4      0.5566585    84.94565
## 5     5      0.6081355    49.41935
## 6     6      0.6220001    41.53135
```

It seems that the model number 6 has the highest adjusted R-squared, so that is the one that we will use. But we will also keep an eye on the model number 3, the one with the higher F-statistic.

3.4.2. What salary would the model predict for Free Agent players?

Now, let us now predict the salaries that some of the free agents of the 2016-2017 season would make in the season 2017-2018, according to the best performing model from the previous section. We can check the list of free agents in this link.

To make this test, we will choose the following players:

- Blake Griffin.
- Steph Curry.
- Kyle Lowry.
- George Hill.
- Serge Ibaka.
- Pau Gasol.
- J.J. Redick.

To make the predictions, we will use the model named `model6`, as it was the one with the higher adjusted R squared of all the models built in the previous section.

```r
griffin <- subset %>% filter(Player == 'Blake Griffin')
# predicted salary
pred_griffin <- predict(model6, griffin)

curry <- subset %>% filter(Player == 'Stephen Curry')
# predicted salary
pred_curry <- predict(model6, curry)

lowry <- subset %>% filter(Player == 'Kyle Lowry')
# predicted salary
pred_lowry <- predict(model6, lowry)

hill <- subset %>% filter(Player == 'George Hill')
# predicted salary
pred_hill <- predict(model6, hill)

ibaka <- subset %>% filter(Player == 'Serge Ibaka')
# predicted salary
pred_ibaka <- predict(model6, ibaka)

p.gasol <- subset %>% filter(Player == 'Pau Gasol')
# predicted salary
pred_gasol <- predict(model6, p.gasol)

redick <- subset %>% filter(Player == 'J.J. Redick')
# predicted salary
pred_redick <- predict(model6, redick)

preds_model6 <- c(pred_griffin, pred_hill, pred_redick,
                  pred_lowry, pred_gasol, pred_ibaka, pred_curry)
```

Now we can compare those predictions with the actual salaries those players got in that season. These values are included in the `salaries` dataset. We extract that information and compare the predictions with the true values in the following lines of code.

```
player_names <- c('Blake Griffin', 'Stephen Curry', 'Kyle Lowry',
                  'George Hill', 'Serge Ibaka', 'Pau Gasol', 'J.J. Redick' )

# see the filtered players
subset %>% filter(Player %in% player_names)
```

```
##          Player  Tm Pos Age  G GS      MPG      PPG       APG      RPG
## 1 Stephen Curry GSW  PG  28 79 79 33.39241 25.30380 6.6202532 4.468354
## 2     Pau Gasol SAS   C  36 64 39 25.42188 12.37500 2.3437500 7.828125
## 3 Blake Griffin LAC  PF  27 61 61 34.03279 21.57377 4.9180328 8.131148
## 4   George Hill UTA  PG  30 49 49 31.51020 16.91837 4.1428571 3.408163
## 5   Serge Ibaka TOR  PF  27 23 23 30.95652 14.21739 0.6521739 6.782609
## 6    Kyle Lowry TOR  PG  30 60 60 37.40000 22.40000 6.9500000 4.800000
## 7   J.J. Redick LAC  SG  32 78 78 28.17949 15.03846 1.4102564 2.192308
##         BPG       SPG     TOPG     PFPG   WS  PER VORP  X2P.  X3P.   FG.
## 1 0.2151899 1.8101266 3.025316 2.316456 12.6 24.6  6.2 0.537 0.411 0.468
## 2 1.0937500 0.3750000 1.265625 1.718750  6.4 20.2  2.4 0.494 0.538 0.502
## 3 0.3770492 0.9508197 2.327869 2.573770  7.7 22.7  3.4 0.514 0.336 0.493
## 4 0.2244898 1.0204082 1.734694 2.326531  5.9 19.3  2.2 0.523 0.403 0.477
## 5 1.4347826 0.3043478 1.695652 3.304348  1.3 13.8  0.0 0.494 0.398 0.459
## 6 0.3166667 1.4666667 2.883333 2.833333 10.1 22.9  4.9 0.518 0.412 0.464
## 7 0.1666667 0.7051282 1.256410 1.602564  4.8 14.8  1.1 0.462 0.429 0.445
##     TS. USG. Salary.in..
## 1 0.624 30.1    12112359
## 2 0.578 21.3    15500000
## 3 0.569 28.0    20140838
## 4 0.599 23.5     8000000
## 5 0.556 20.9    12250000
## 6 0.623 24.9    12000000
## 7 0.599 21.9     7377500
```

```
# salaries in 2016-17
true_salaries <-
  salaries_2017 %>%
  filter(Player.Name %in% player_names) %>%
  select(Player.Name, Salary.in..)
names(true_salaries) <- c("Player", "salary_2017")

#predicted sal. for 2017-18 for these players
true_salaries$predicted_2018 <- preds_model6

# salaries they obtained in the 2017-18 season
true_salaries_2018 <-
  salaries %>%
  filter(Season.End==2018, Player.Name %in% player_names) %>%
  select(Player.Name, Salary.in..)
names(true_salaries_2018) <- c("Player", "true_salary_2018")

# see the differences
```

```r
salaries_comparison <- full_join(true_salaries, true_salaries_2018,
                                 by =c('Player' = 'Player' ) )
salaries_comparison$pred_error <- salaries_comparison$true_salary_2018 -
  salaries_comparison$predicted_2018

salaries_comparison
```

```
##          Player salary_2017 predicted_2018 true_salary_2018 pred_error
## 1 Blake Griffin    20140838       16232826         29512900   13280074
## 2   George Hill     8000000       12099458         20000000    7900542
## 3   J.J. Redick     7377500       14079457         23000000    8920543
## 4    Kyle Lowry    12000000       17190876         28703704   11512828
## 5     Pau Gasol    15500000       17251344         16000000   -1251344
## 6   Serge Ibaka    12250000        8964770         20061729   11096959
## 7 Stephen Curry    12112359       16708167         34682550   17974383
```

We have quite different results. We can see that for Pau Gasol, the prediction error is less than 1.5 M$, but for the other players, the differences are unacceptable. It is intereseting, though, to see how the model detected a salary raise in the J.J. Redick and George Hill salaries for the 2017-18 season, though the raise was shorter than in the real life.

3.4.3. Which predictions would other models make?

We can show the predictions made by the different models, to see if the model with the highest adjusted R-squared was actually the one that made the best predictions. Let us check `model1`, which used all variables and `model3`, the one with the higher F-statistic.

```r
# model 1
pred_griffin_model1 <- predict(model1, griffin)

pred_curry_model1 <- predict(model1, curry)

pred_lowry_model1 <- predict(model1, lowry)

pred_hill_model1 <- predict(model1, hill)

pred_ibaka_model1 <- predict(model1, ibaka)

pred_gasol_model1 <- predict(model1, p.gasol)

pred_redick_model1 <- predict(model1, redick)

preds_model1 <-  c(pred_griffin_model1, pred_hill_model1, pred_redick_model1,
                  pred_lowry_model1, pred_gasol_model1, pred_ibaka_model1,
                  pred_curry_model1)

# model 3
pred_griffin_model3 <- predict(model3, griffin)

pred_curry_model3 <- predict(model3, curry)

pred_lowry_model3 <- predict(model3, lowry)
```

```r
pred_hill_model3 <- predict(model3, hill)

pred_ibaka_model3 <- predict(model3, ibaka)

pred_gasol_model3 <- predict(model3, p.gasol)

pred_redick_model3 <- predict(model3, redick)

preds_model3 <- c(pred_griffin_model3, pred_hill_model3, pred_redick_model3,
                  pred_lowry_model3, pred_gasol_model3, pred_ibaka_model3,
                  pred_curry_model3)
```

```r
true_salaries$predicted_model1 <-  preds_model1

true_salaries$predicted_model3 <-  preds_model3

# see the differences
salaries_comparison <- full_join(true_salaries, true_salaries_2018, by =c('Player' = 'Player' ) )
```

```
## Warning: Column `Player` joining factors with different levels, coercing to
## character vector
```

```r
salaries_comparison
```

```
##          Player salary_2017 predicted_2018 predicted_model1
## 1 Blake Griffin    20140838       16232826         17036348
## 2   George Hill     8000000       12099458         10332166
## 3   J.J. Redick     7377500       14079457         11955640
## 4    Kyle Lowry    12000000       17190876         14461343
## 5     Pau Gasol    15500000       17251344         15187828
## 6   Serge Ibaka    12250000        8964770          6923571
## 7 Stephen Curry    12112359       16708167         16954000
##   predicted_model3 true_salary_2018
## 1         17506777         29512900
## 2         13877478         20000000
## 3         12390258         23000000
## 4         20688659         28703704
## 5         15972730         16000000
## 6         11045350         20061729
## 7         17793142         34682550
```

It seems that `model3` is the one making the most accurate predictions overall. Some further tests would be necessary to determine which of the models has more accurate predictions.

3.4.4. Conclusion

In general, the models that were built in the last sections are too simple to reflect the complexity of the NBA salary system. Some improvements are necessary in order to achieve a higher performing model, that is able to predicto more accurately the salaries.

Some ideas to improve the model would be the following:

- Change the model from regression to other ones that are able to detect the nature of the salaries. Let us notice that the best adjusted R-squared obtained was around 0.6, which is not a great value.
- There are other considerations in the salary predictions that were not considered. For example, **the salary cap available for a season** (that is, the limit of money that every team can spend on salaries) or the **player's eligibility for a super-max contract**. In the case of Steph Curry, he was eligible for a super-max extension, thus, he earned a lot more money than predicted.
- If a player enters free-agency from a rookie contract, he is expected to earn more money the next season. This wasn't considered in this model.

4. Data visualizations

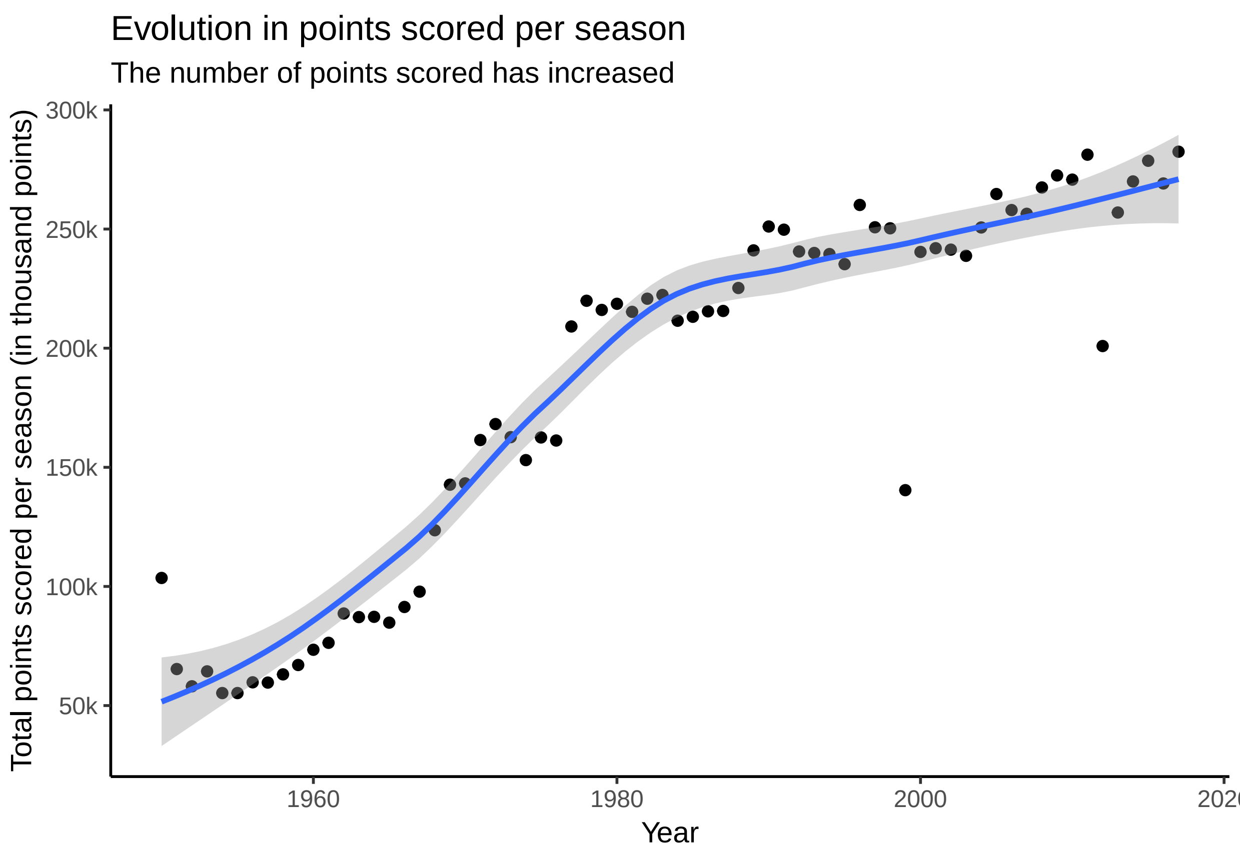
In this section, some data visualizations are included, aiming to answer some small questions about the evolution of the league in the last years.

4.1. How many points were scored each season through the NBA history?

```
## evolution of points scored though the years------------------------

# group points by year

points_by_year <- season_stats %>%
  group_by(Year) %>%
  summarise(total = sum(PTS))

str(points_by_year)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    69 obs. of  2 variables:
##  $ Year : int  1950 1951 1952 1953 1954 1955 1956 1957 1958 1959 ...
##  $ total: int  103562 65338 58096 64356 55252 55253 59768 59654 63093 67031 ...
```

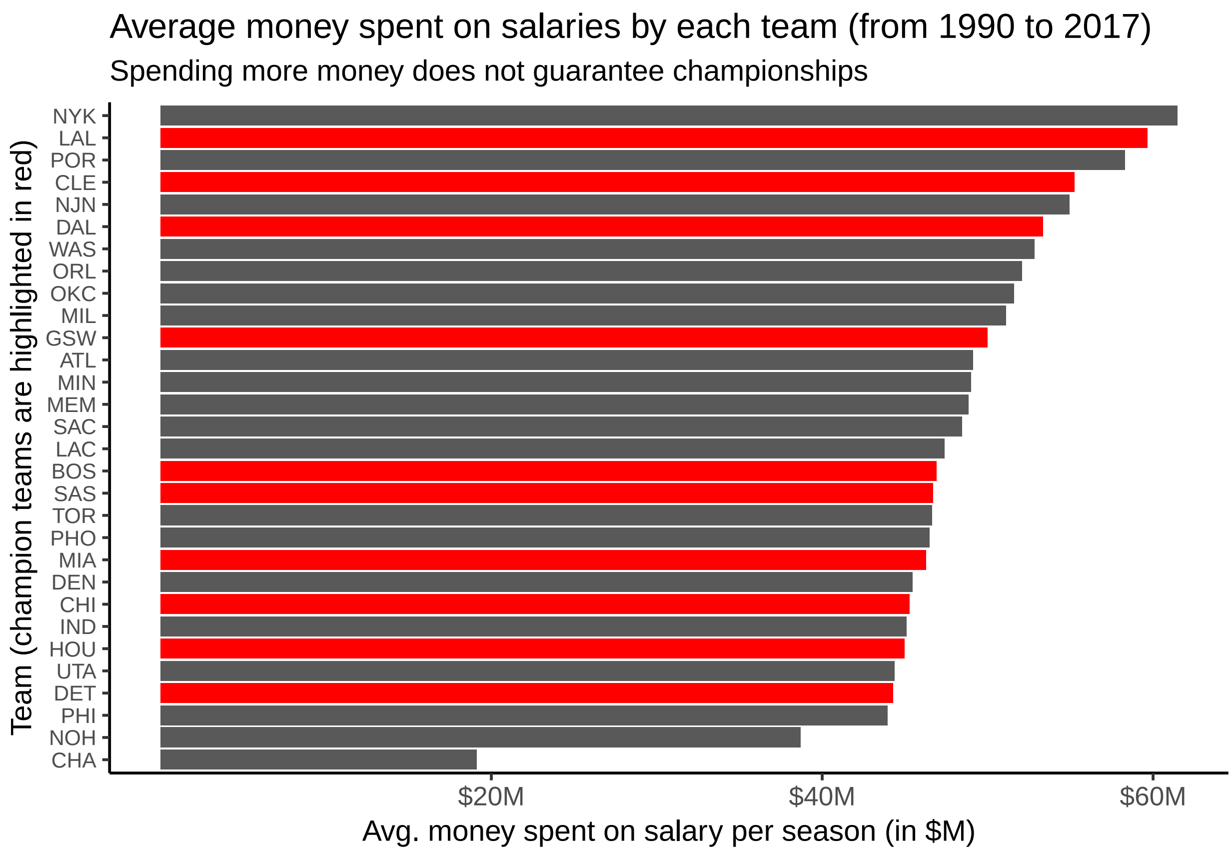
```
# plot; geom_smooth options: method = "lm", se = FALSE

points_by_year %>%
  ggplot(aes(Year, total)) +
  geom_point()+
  geom_smooth()+
  theme_classic()+
  scale_y_continuous(breaks = c(50000, 100000,
                                150000, 200000,
                                250000, 300000),
                     labels = c("50k", "100k", "150k",
                                "200k", "250k", "300k"))+
  ggtitle('Evolution in points scored per season') +
  labs(subtitle = "The number of points scored has increased")+
  ylab("Total points scored per season (in thousand points)")
```



4.2. Which are the teams that spent the most money on players' salaries?

```
### averaged salaries grouped by teams -----------
library(tidyverse)

champions_list <- c("LAL", "CLE", "DET", "GSW", "BOS",
                    "MIA", "CHI", "SAS", "HOU", "DAL")

n_seasons_salaries <- n_distinct(salaries$Season.Start)
salaries_by_team <- salaries %>% group_by(Team) %>%
  summarise(total = sum(Salary.in..), avg = sum(Salary.in..)/n_seasons_salaries) %>%
  mutate(highlight_flag = ifelse(Team %in% champions_list, T, F))



# plot: champion teams are displayed in red colour
salaries_by_team %>%
  ggplot(aes(x = reorder(Team, avg), avg)) +
  geom_bar(stat = "identity",
           aes(fill = highlight_flag)) +
  scale_fill_manual(values = c('#595959', 'red')) +
  theme_classic() +
  ggtitle('Average money spent on salaries by each team (from 1990 to 2017)') +
  labs(subtitle = "Spending more money does not guarantee championships") +
```

```r
  theme(axis.text.y = element_text(size=8),
        axis.text.x = element_text(size=10),
        legend.position = 'none') +
  scale_y_continuous(breaks = c(20000000, 40000000, 60000000),
                     labels = c("$20M", "$40M", "$60M"))+
  ylab("Avg. money spent on salary per season (in $M)") +
  xlab("Team (champion teams are highlighted in red)")+
  coord_flip()
```



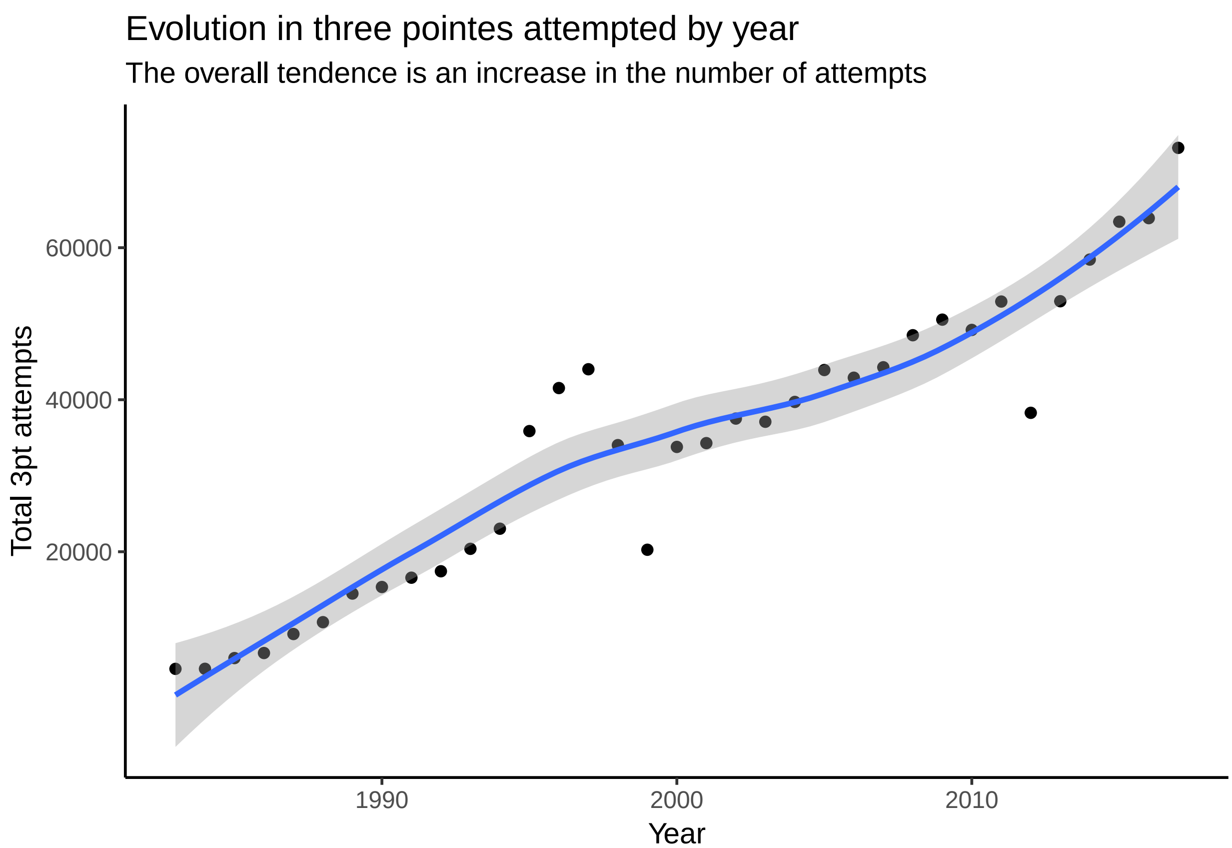
4.3. Was there an evolution in the number of three point attempts through the NBA history?

```r
## evolution of 3pts attemps though the years------------------------

# group 3 pts by year :after 1982, when first collected data

threes_att_by_year <- season_stats %>% filter(Year>1982) %>%
  group_by(Year) %>%
  summarise(total = sum(X3PA))

str(threes_att_by_year)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    35 obs. of  2 variables:
```

```
##  $ Year : int  1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 ...
##  $ total: int  4592 4599 6008 6677 9177 10735 14496 15361 16587 17433 ...
```

```
# plot; geom_smooth options: method = "lm", se = FALSE

threes_att_by_year %>% ggplot(aes(Year, total)) +
  geom_point()+ geom_smooth()+
  #geom_bar(stat = "identity") +
  scale_y_continuous(breaks = c(20000, 40000,
                                60000, 80000))+
  ggtitle('Evolution in three pointes attempted by year') +
  theme_classic()+
  labs(subtitle = "The overall tendence is an increase in the number of attempts")+
  ylab('Total 3pt attempts')
```



5. Conclusions

In this document we explored three hypothesis about the NBA stats and salaries datasets. First, we tried to determine if the Win Shares are the factor that is more correlated to the salary of a player. It was concluded that the variable that is more positively correlated to the salary is MPG, closely followed by PPG.

The reason why this happens is that, usually, the players that earn the most are also the ones that play more minutes per game and the ones that score more points per game. This makes sense, as if you have to spend more money on a certain player, you would expect him to ve valuable for the team and thus make him play more minutes and allow him to shoot more often.

Second, we wondered if there there is a significant difference in any of the main statistics for the different five positions of the game. The statistics we have checked are the following: Salary, Points per Game, Assists per Game, Minutes per Game and Win shares. We found differences by positions in the Assists per game (the PG has the higher average), in Minutes per game (the Small Forwards are getting more minutes per game) and Win Shares (the positions with the higher WS are the Centers, followed by the Small Forwards).

Third, we created a regression model that would predict the salary of a player, given the statistics of the previous season. The conclusion is that some improvements need to be made with the model, in order to be able to obtain more precise predictions.

6. Save the clean dataset to .csv

```r
write.csv(stats_with_salaries, file = "clean_data.csv")
```