NBA: Behind the Scenes

Nicolas Fairweather

7/22/2021

The NBA Collective Bargaining Agreement (CBA) is a contract between the National Basketball Association (the commissioner and the 30 team owners) and the National Basketball Players Association, the players' union, that dictates the rules of player contracts, trades, revenue distribution, the NBA Draft, and the salary cap, among other things. In June 2005, the NBA's 1999 CBA expired, meaning the League and the players' union had to negotiate a new agreement; in light of the 2004–05 NHL lockout, the two sides quickly came to an agreement, and ratified a new CBA in July 2005. This agreement expired following the 2010–11 season, leading to the 2011 NBA lockout. A new CBA was ratified in December 2011, ending the lockout.Little changed in terms of the salary cap between the 1999 and 2005 versions of the CBA. In exchange for agreeing to the controversial player age minimum, the players received a slightly higher percentage of the League's revenues over the course of the new agreement. Additionally, the League's maximum salary decreased slightly in comparison to the 1999 CBA. Under the 2011 CBA, the players received a lower percentage of league revenues.In 2005, players received 57% of the income, and as of the 2016 CBA, they are receiving about 49–51% of revenue. At that time, the next CBA discussion was set for ten years. or if necessary, in 2017.In 2016, the NBA and NBA Players Association met to work on a new CBA, which both sides approved in December of that year. This most recent agreement started with the 2016–17 season and runs through 2023–24, with a mutual opt-out after 2022–23. Players average their highest salaries at the age of 33, while teams spent the most funds on players aged between 26 and 31. The average salary in the NBA has increased more than 7x since the 1990-91 season. Median salary has slower growth than the average; this suggests that the financial gap between the top talents and the rest is growing larger. The top talents in the NBA all share high win shares, which is a player's statistic that distributes credit from team success to the individuals on the team. This is all worked out through the general manager whose job is to put together a winning team by evaluating performance and productivity to determine salaries, which is then balanced to fit under the NBA salary cap.

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)
library(ggplot2)
library(plotly)
library(caret)
```

We upload the datasets into R.

```
library(readr)
seasons_stats <- read_csv("Seasons_Stats.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
##   Player = col_character(),
##   Pos = col_character(),
##   Tm = col_character(),
##   GS = col_logical(),
##   `3PAr` = col_logical(),
##   `ORB%` = col_logical(),
##   `DRB%` = col_logical(),
##   `TRB%` = col_logical(),
##   `AST%` = col_logical(),
##   `STL%` = col_logical(),
##   `BLK%` = col_logical(),
##   `TOV%` = col_logical(),
##   `USG%` = col_logical(),
##   blanl = col_logical(),
##   blank2 = col_logical(),
##   OBPM = col_logical(),
##   DBPM = col_logical(),
##   BPM = col_logical(),
##   VORP = col_logical(),
##   `3P` = col_logical()
##   # ... with 7 more columns
## )
## i Use `spec()` for the full column specifications.
```

```
library("readxl")
salaries <- read_excel("Player - Salaries per Year (1990 - 2017).xlsx")
```

Dataset description

Two datasets are used:

- `season_stats` : Dataset containing advanced stats since the year 1950. It was obtained through Kaggle.
- `salaries` : Dataset containing the salaries of the NBA players from 1990 until 2018. Accessed via Kaggle.

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 and use the salaries of the 2017-18 season, because that is when the last collective bargaining agreements between the NBA (the commissioner and the 30 team owners) and the National Basketball Players Association, the players' union.

The remaining data will not be considered in the main analyses, though 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: `X` : Number of column in the dataset `Year` : Year that the season occurred. Since the NBA season is split over two calendar years, the year given is the last year for that season. For example, the year for the 1999-00 season would be 2000.

`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 where the players were in 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

`TS.` : TS%, True Shooting Percentage, a measure of shooting efficiency that takes into account field goals, 3-point field goals, and free throws.

`3PAr` : 3-Point Field Goal Attempts

`USG.` : Usage Percentage. 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; 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.

`2P%` : 2-Point Field Goal Percentage.

`3P%` : 3-Point Field Goal Percentage

`3PA` : 3-Point Field Goal Attempts

And from the second dataset:

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

Research Questions

Are the Win Shares the factor that is more correlated to the salary of a player? How did the number of points scored evolved through the years? Which teams are the ones which spent the most on salaries? Were they successful teams? What's the NBA's common age and how do a player's age affect their time on the court?

- Alternative Hypothesis

The Win Shares are the factor that is more correlated to the salary of a player.

- Null Hypothesis

Another statistic is the factor that is more correlated to the salary of a player.

Data Wrangling

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 <- select(seasons_stats, -c(blanl, blank2))
```

We now restrict the datasets to the 2016-17 season, as it is the one which has more interest to our analysis

```
# year 2017: season 16_17
names(salaries)<-str_replace_all(names(salaries), c(" " = "." , "," = "" ))
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"))
```

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

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

Register.Value	Player.Name	Salary.in.$	Season.Start	Season.End	Team	Full.Team.Name
0	0	0	0	0	0	0

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

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

X1	Year	Player	Pos	Age	Tm	G	GS	MP	PER	TS%	3PAr	FTr	ORB%	DRB%	TRB%	AST%	STL%	BLK%	TOV%	USG%	OWS	DWS	WS	WS
0	0	0	0	0	0	0	389	0	0	2	548	2	561	590	590	574	532	516	575	593	0	0	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
kable(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
Damjan Rudez	ORL	SF	45	314
Anthony Brown	TOT	SF	11	159
Anthony Brown	NOP	SF	9	143
Bruno Caboclo	TOR	SF	9	40
Steve Novak	MIL	PF	8	22
Arinze Onuaku	ORL	C	8	28
Roy Hibbert	DEN	C	6	11
Patricio Garino	ORL	SG	5	43
John Lucas	MIN	PG	5	11
Axel Toupane	TOT	SF	4	47

```
# important: some big stars included
kable(season_stats_2017 %>%
  filter(is.na(`3P%`) == TRUE) %>%
  select(c("Player", "Tm", "Pos", "G", "MP")) %>%
  arrange(desc(G)) %>%
  head(n=10))
```

Player	Tm	Pos	G	MP
Corey Brewer	TOT	SF	82	1281
Marquese Chriss	PHO	PF	82	1743
Jordan Clarkson	LAL	SG	82	2397
Jamal Crawford	LAC	SG	82	2157
Gorgui Dieng	MIN	PF	82	2653
Tobias Harris	DET	PF	82	2567
Buddy Hield	TOT	SG	82	1888
Justin Holiday	NYK	SG	82	1639
Ersan Ilyasova	TOT	PF	82	2142
Joe Ingles	UTA	SF	82	1972

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 replace 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:

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

X1	Year	Player	Pos	Age	Tm	G	GS	MP	PER	TS%	3PAr	FTr	ORB%	DRB%	TRB%	AST%	STL%	BLK%	TOV%	USG%	OWS	DWS	WS	WS
0	0	0	0	0	0	0	389	0	0	0	531	0	547	569	569	561	522	505	558	571	0	0	0	

We will see which are the samples with NAs in `2P%` 2 point percentage.

```
# check X2P.
kable(season_stats_2017 %>%
  filter(is.na(`2P%`) == TRUE) %>%
  select(c("Player", "Tm", "Pos", "G", "MP")) %>%
  arrange(desc(G)))
```

Player	Tm	Pos	G	MP
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.

```
# drop the sample
season_stats_2017<- season_stats_2017 %>%
  drop_na(`2P%`)
```

Next, we deal with the imputation of the NAs values of the variable `3P%` .

```
kable(season_stats_2017 %>%
  filter(is.na(`3P%`) == TRUE) %>%
  select(c("Player", "Tm", "Pos", "G", "MP", "3P%", "3PA")) %>%
  arrange(desc(`3P%`)) %>%
  head(n=10))
```

Player	Tm	Pos	G	MP	3P%	3PA
Alex Abrines	OKC	SG	68	1055	NA	NA
Quincy Acy	TOT	PF	38	558	NA	NA

Player	Tm	Pos	G	MP	3P%	3PA
Quincy Acy	DAL	PF	6	48	NA	NA
Quincy Acy	BRK	PF	32	510	NA	NA
Arron Afflalo	SAC	SG	61	1580	NA	NA
Cole Aldrich	MIN	C	62	531	NA	FALSE
LaMarcus Aldridge	SAS	PF	72	2335	NA	NA
Tony Allen	MEM	SG	71	1914	NA	NA
Al-Farouq Aminu	POR	SF	61	1773	NA	NA
Alan Anderson	LAC	SF	30	308	NA	NA

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 these specific players. To simplify the analysis, we can input the value 0. If they didn't shoot any 3 pointers, they probably would have a bad percentage anyways.

```
#replace NAs by 0
season_stats_2017 <- season_stats_2017 %>%
  mutate(`3P%` = replace_na(`3P%`, 0) )
##check the results on two of the players
kable(season_stats_2017 %>%
  filter(Player %in% c("Clint Capela", "Hassan Whiteside")) %>%
  select(c(Player, `3P%` )))
```

Player	3P%
Clint Capela	0
Hassan Whiteside	0

FOllowing we will compute some basic per game statistics that are not included in the `season_stats` dataset, namely, points per game (PPG), assist 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 )
```

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

```
# join stats with salaries: join by player and team to avoid duplicates 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"))
kable(head(stats_with_salaries))
```

X1	Year	Player	Pos	Age	Tm	G	GS	MP	PER	TS%	3PAr	FTr	ORB%	DRB%	TRB%	AST%	STL%	BLK%	TOV%	USG%	OWS
24096	2017	Alex Abrines	SG	23	OKC	68	NA	1055	10.1	0.560	NA	0.144	NA	NA	NA	NA	NA	NA	NA	NA	1.2
24098	2017	Quincy Acy	PF	26	DAL	6	FALSE	48	-1.4	0.355	NA	0.176	NA	NA	NA	FALSE	FALSE	FALSE	NA	NA	-0.2
24099	2017	Quincy Acy	PF	26	BRK	32	TRUE	510	13.1	0.587	NA	0.373	NA	NA	NA	NA	NA	NA	NA	NA	0.6
24100	2017	Steven Adams	C	23	OKC	80	NA	2389	16.5	0.589	NA	0.392	NA	NA	NA	NA	NA	NA	NA	NA	3.3
24101	2017	Arron Afflalo	SG	31	SAC	61	NA	1580	9.0	0.559	NA	0.221	NA	NA	NA	NA	NA	NA	NA	NA	1.2
24102	2017	Alexis Ajinca	C	28	NOP	39	NA	584	12.9	0.529	NA	0.225	NA	NA	NA	NA	NA	NA	NA	NA	0.0

After wrangling the data, we will select a subset of all the variables, with which we will continue the analysis.

```
subset <- stats_with_salaries %>%
  select(c(Player,Tm, Pos, Age,  G, GS, MPG, PPG, APG, RPG, BPG,
           SPG, TOPG, PFPG, WS, PER, VORP, `2P%`, `3P%`, `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"        "2P%"         "3P%"         "FG%"
## [21] "TS%"         "USG%"        "Salary.in.$"
```

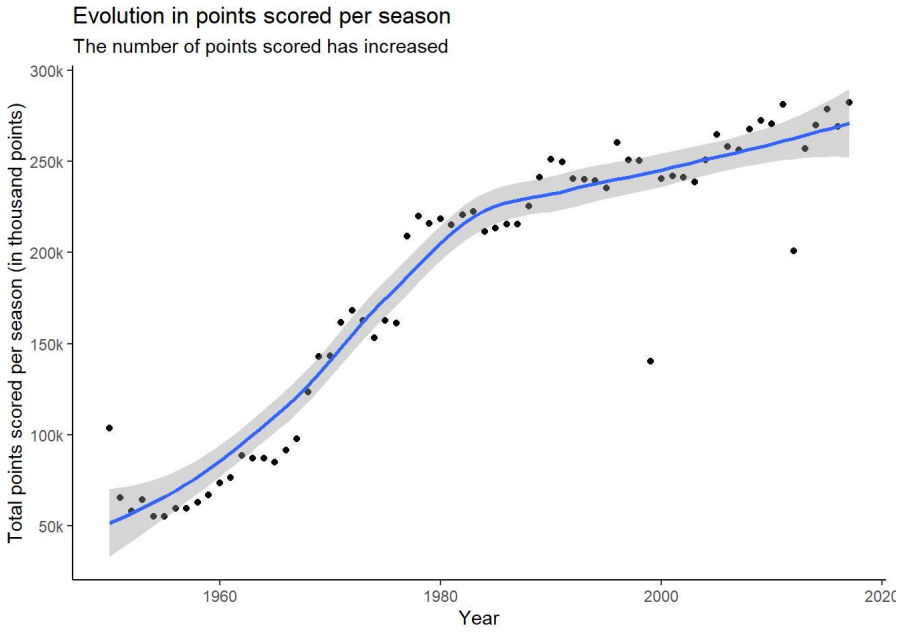
Data visualizations

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

```
## 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)
```

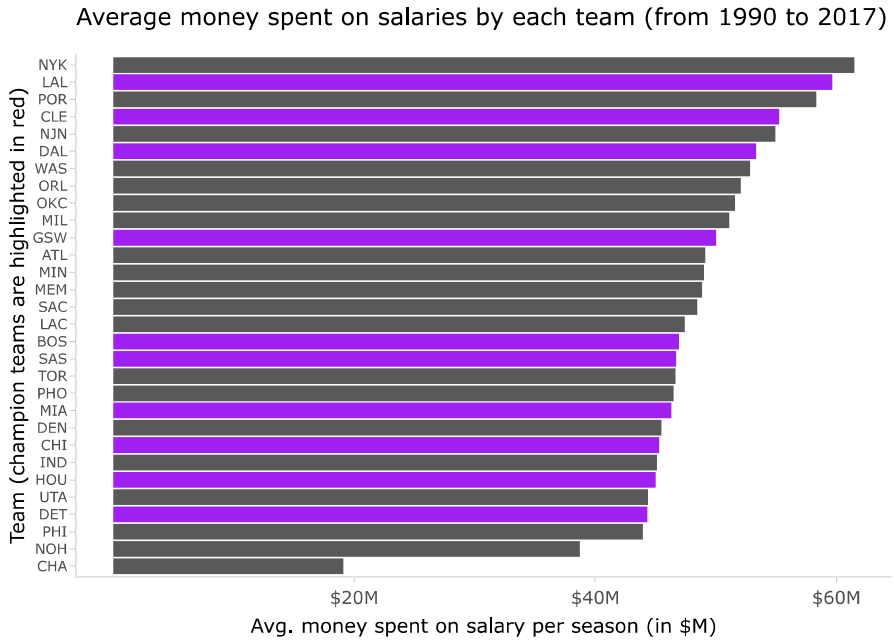
```
## tibble [69 x 2] (S3: tbl_df/tbl/data.frame)
##  $ Year : num [1:69] 1950 1951 1952 1953 1954 ...
##  $ total: num [1:69] 103562 65338 58096 64356 55252 ...
```

```
# 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)")
```



The graph shows the points scored in a NBA year increasing over the years. This is actually a depiction of what analysts argue that the NBA is playing a more offense less defense brand of basketball, which is more favorable for the fans.

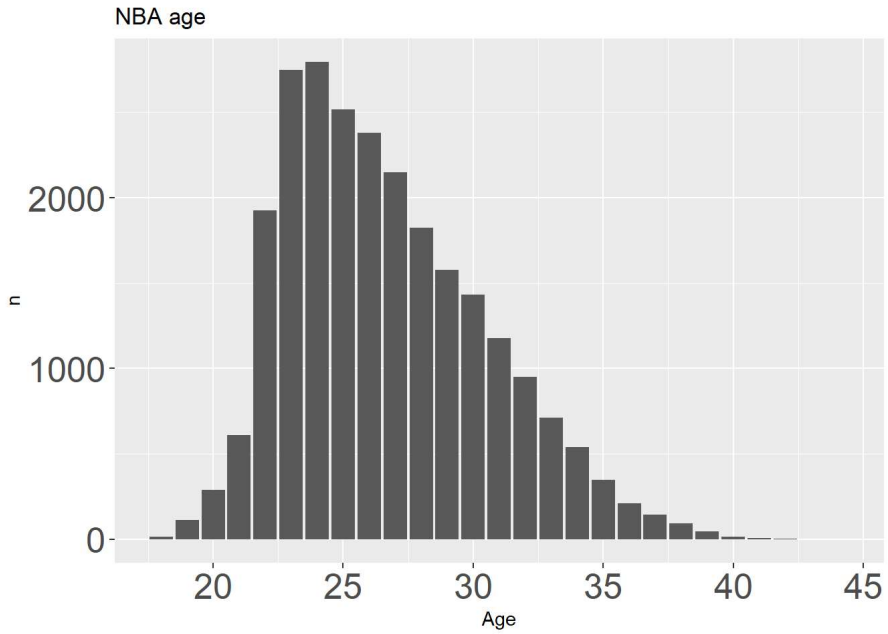
```
### averaged salaries grouped by teams -----------
library(tidyverse)
champions_list <- c("LAL", "CLE", "DET", "GSW", "BOS",
                    "MIA", "CHI", "SAS", "HOU", "DAL")
n_seasons_salaries <- n_distinct(salaries$Season.Start)
salaries_by_team <- salaries %>% group_by(Team) %>%
  summarise(total = sum(`Salary.in.$`), avg = sum(`Salary.in.$`)/n_seasons_salaries) %>%
  mutate(highlight_flag = ifelse(Team %in% champions_list, T, F))
# plot: champion teams are displayed in purple color
salaries_barplot <- salaries_by_team %>%
  ggplot(aes(x = reorder(Team, avg), avg)) +
  geom_bar(stat = "identity",
           aes(fill = highlight_flag)) +
  scale_fill_manual(values = c('#595959', 'purple')) +
  theme_classic() +
  ggtitle('Average money spent on salaries by each team (from 1990 to 2017)') +
  labs(subtitle = "Spending more money does not guarantee championships") +
  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()
  ggplotly(salaries_barplot)
```



This graph shows how successful a team was in respect to the average amount they spend on salaries from 1990 to 2017. The highlighted teams are teams that have won the championships in the those years. This graph also shows that not all money produce championships, you have to invest wisely.

```
season_stats2 <- season_stats %>%
  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 )
```

```
age <- season_stats2 %>% group_by(Age) %>% summarise(n=n()) %>% ggplot(aes(x=Age,y=n))+
geom_bar(position='dodge',stat = 'identity')+
theme(axis.text=element_text(size=20))+
ggtitle('NBA age')
age
```

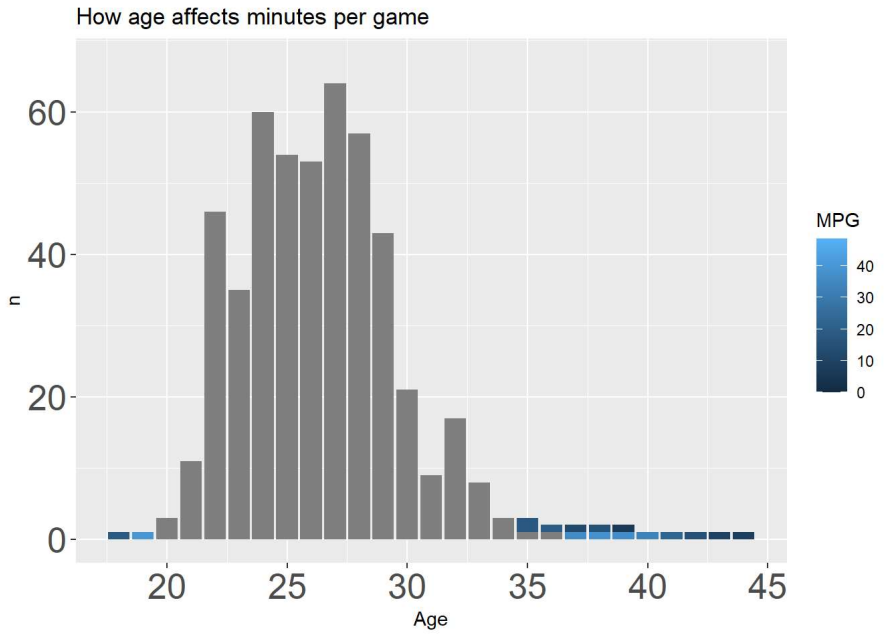


This graph gives us a spread of the age is of the NBA. The most common age is 24, with the least common being 42. The graph starts decreasing after age 24 because that is typically around the time where rookies don't get a second contract and are replaced with younger player like 18-20 because they have not shown good progress from their time in the league.

```
age_2 <- season_stats2 %>% group_by(Age, MPG) %>% summarise(n=n()) %>% ggplot(aes(x=Age,y=n, fill = MPG,))+
geom_bar(position='dodge',stat = 'identity')+
theme(axis.text=element_text(size=20))+
ggtitle('How age affects minutes per game')
```

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

```
age_2
```



This graph shows how age affects playing time. It also just emphasizes the norm that older you get the less you can perform with anything. But there are players that defy the test of time, this is also contributed more now than before because of the more favorable style the NBA has now.

Statistical Data Analysis

We start with the hypothesis that 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.

Let's compute the correlation of Win Shares. We use the spearman correlation test because we have two ranked variables, and we want to see whether the two variables covary

```
# 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 it 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  NA  NA   NA 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 be 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 MPG and PPG 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.

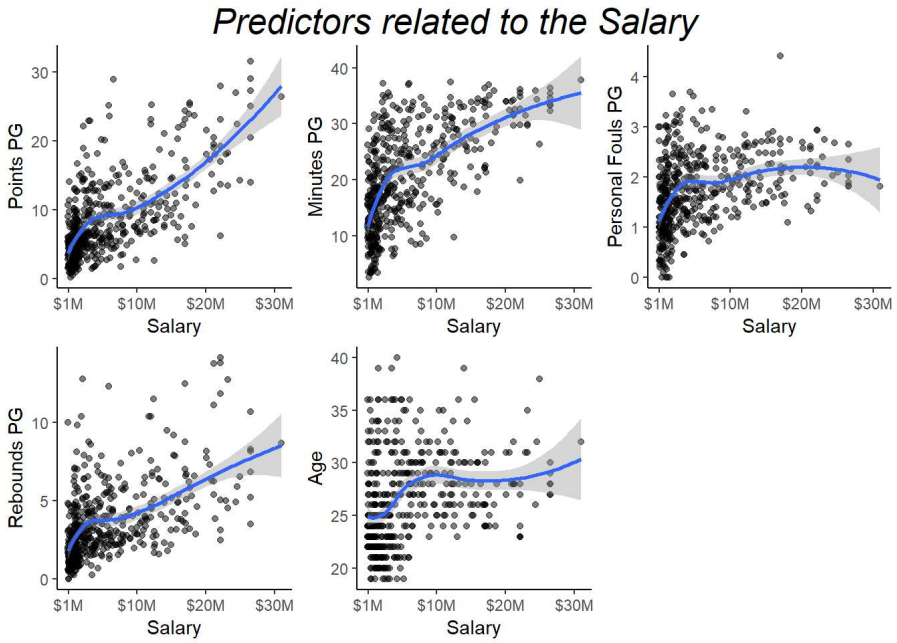
```
cor.test(subset$`FG%`, subset$`Salary.in.$`, method = 'spearman')
```

```
##
##  Spearman's rank correlation rho
##
## data:  subset$`FG%` and subset$`Salary.in.$`
## S = 14232416, p-value = 0.0001784
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.1722231
```

The correlation of 0.17 is not high, so being an efficient scorer seems not to be highly correlated with a higher salary.

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, PFPG and AGE) are related to the Salary. The following code gives the desired plot.

```
# 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", "$30M"))+
  theme_classic() +
  theme(plot.margin = margin(3, 7, 3, 1.5))
p2 <- ggplot(subset, aes(x=`Salary.in.$`, y=MPG)) +
  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", "$30M"))+
  theme_classic() +
  theme(plot.margin = margin(3, 7, 3, 1.5))
p3 <- ggplot(subset, aes(x=`Salary.in.$`, 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", "$30M"))+
  theme_classic() +
  theme(plot.margin = margin(3, 7, 3, 1.5))
p4 <- ggplot(subset, aes(x=`Salary.in.$`, 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", "$30M"))+
  theme_classic() +
  theme(plot.margin = margin(3, 7, 3, 1.5))
p5 <- ggplot(subset, aes(x=`Salary.in.$`, 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))
grid.arrange(p1, p2, p3, p4, p5,
             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)))
```



Machine Learning

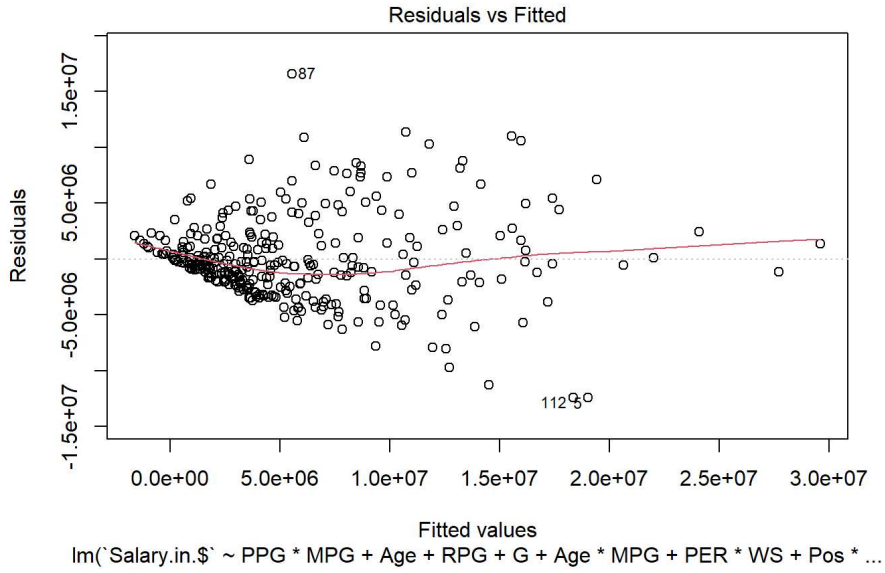
A machine learning technique for linear regression will be used to help answer to find the statistics that work best in cooperation when predicting salaries. The code below is used to only split the data once into the training and test sets so that the same output is received when running the linear models and predictions multiple times.I chose the model with the highest adjusted R-squared 0.62 and the p-value of the F-statistic is < 2.2e-16, which is highly significant.

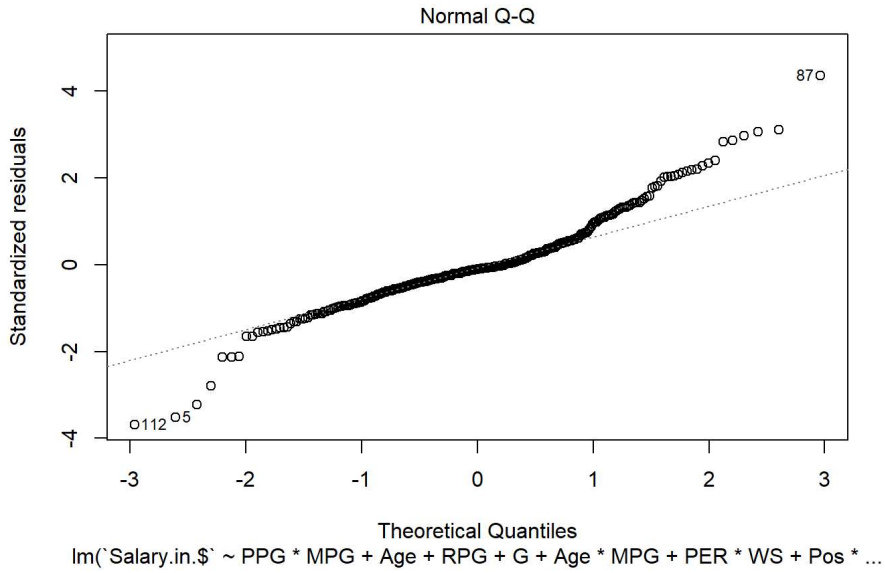
```
set.seed(100)
smp_size <- floor(0.7 * nrow(subset))
train_ind <- sample(seq_len(nrow(subset)), size = smp_size)
train <- subset[train_ind,]
test <- subset[-train_ind,]
```

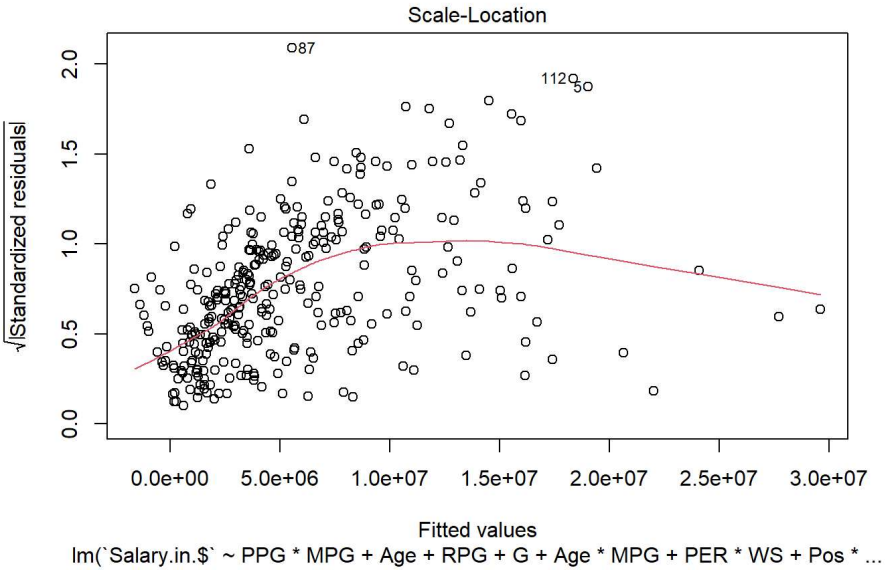
```
model1 <- lm(`Salary.in.$` ~ PPG * MPG + Age + RPG + G + Age * MPG + PER * WS + Pos * WS, data = train)
summary(model1)
```

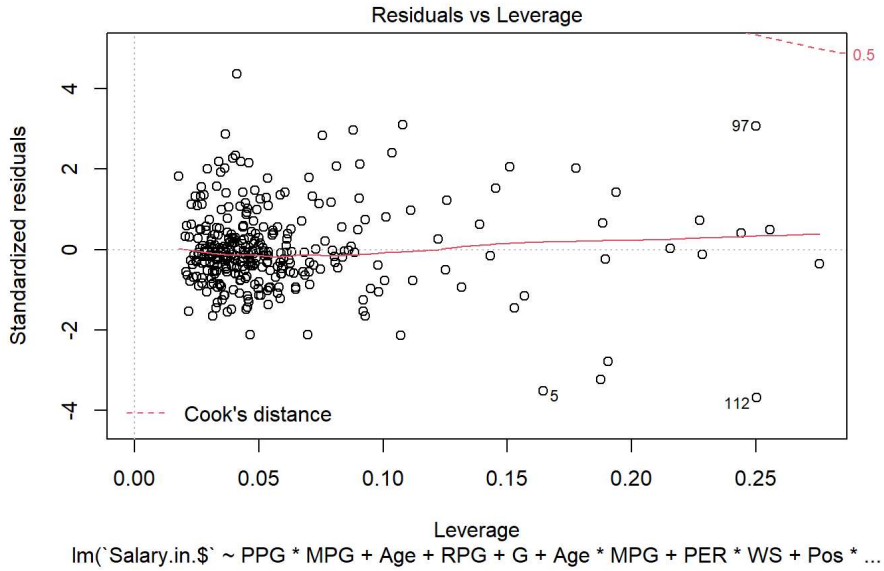
```
##
## Call:
## lm(formula = `Salary.in.$` ~ PPG * MPG + Age + RPG + G + Age *
##     MPG + PER * WS + Pos * WS, data = train)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -12443791  -2061056   -370750   1550408  16575839
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 12338065    3629264   3.400 0.000763 ***
## PPG          -559362     275303  -2.032 0.043029 *
## MPG         -1100802     199648  -5.514 7.43e-08 ***
## Age          -410290     131449  -3.121 0.001971 **
## RPG           256416     206892   1.239 0.216149
## G              47108      13662   3.448 0.000643 ***
## PER           -48560      69958  -0.694 0.488127
## WS          -1594895     513715  -3.105 0.002082 **
## PosPF       -1006266    1002074  -1.004 0.316076
## PosPG       -1988455    1006553  -1.976 0.049100 *
## PosSF       -1052729    1063527  -0.990 0.323023
## PosSG       -2530001    1045218  -2.421 0.016073 *
## PPG:MPG        27360       7041   3.886 0.000125 ***
## MPG:Age        46953       6745   6.962 2.02e-11 ***
## PER:WS         66702      19987   3.337 0.000949 ***
## WS:PosPF      240652     327137   0.736 0.462513
## WS:PosPG     -128489     236824  -0.543 0.587830
## WS:PosSF     -254408     237173  -1.073 0.284257
## WS:PosSG      208138     346485   0.601 0.548472
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3878000 on 309 degrees of freedom
## Multiple R-squared:  0.6434, Adjusted R-squared:  0.6227
## F-statistic: 30.98 on 18 and 309 DF,  p-value: < 2.2e-16
```

```
plot(model1)
```


Residuals vs Fitted
Residuals
Fitted values
lm(`Salary.in.$` ~ PPG * MPG + Age + RPG + G + Age * MPG + PER * WS + Pos * ...
87
112
5
1.5e+07
5.0e+06
-5.0e+06
-1.5e+07
0.0e+00
5.0e+06
1.0e+07
1.5e+07
2.0e+07
2.5e+07
3.0e+07


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(`Salary.in.$` ~ PPG * MPG + Age + RPG + G + Age * MPG + PER * WS + Pos * ...
87
112
5
4
2
0
-2
-4
-3
-2
-1
0
1
2
3

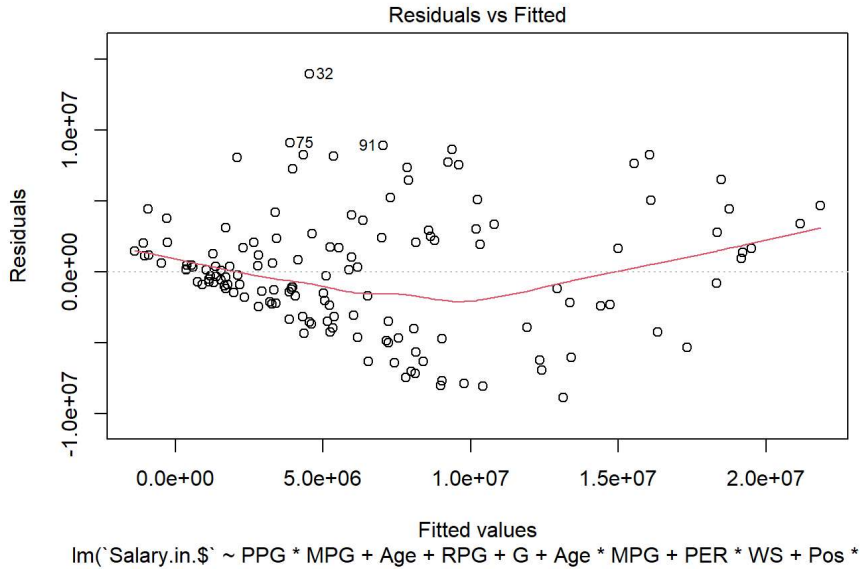

Scale-Location
√|Standardized residuals|
Fitted values
lm(`Salary.in.$` ~ PPG * MPG + Age + RPG + G + Age * MPG + PER * WS + Pos * ...
87
112
5
2.0
1.5
1.0
0.5
0.0
0.0e+00
5.0e+06
1.0e+07
1.5e+07
2.0e+07
2.5e+07
3.0e+07

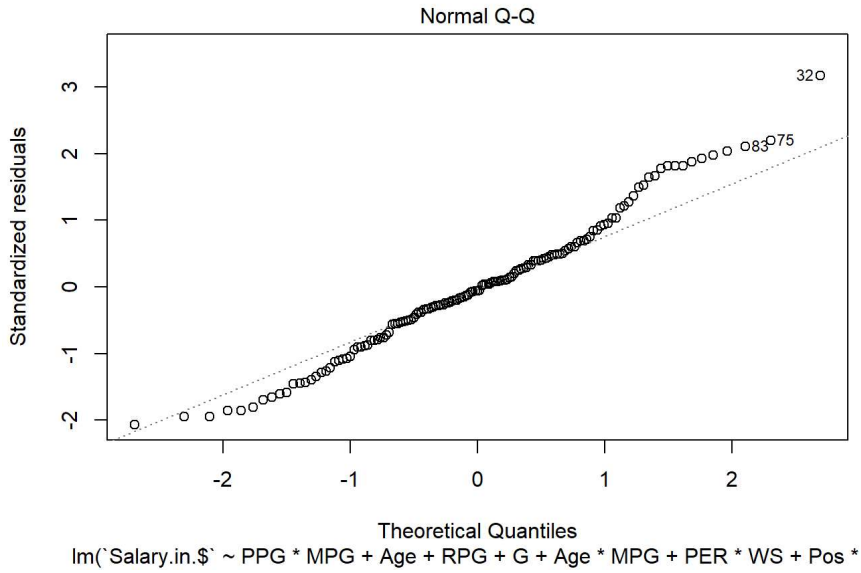


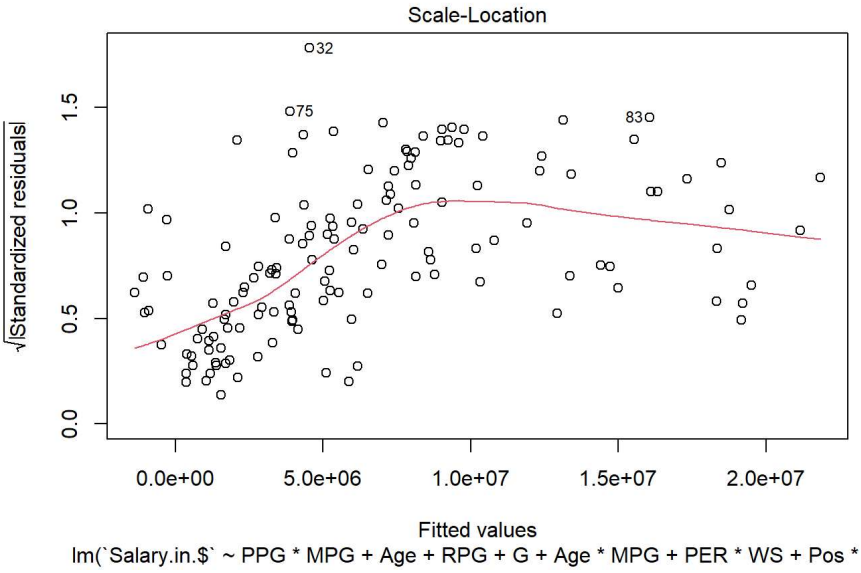

```
model2 <- lm(`Salary.in.$` ~ PPG * MPG + Age + RPG + G + Age * MPG + PER * WS + Pos * WS, data = test)
summary(model2)
```

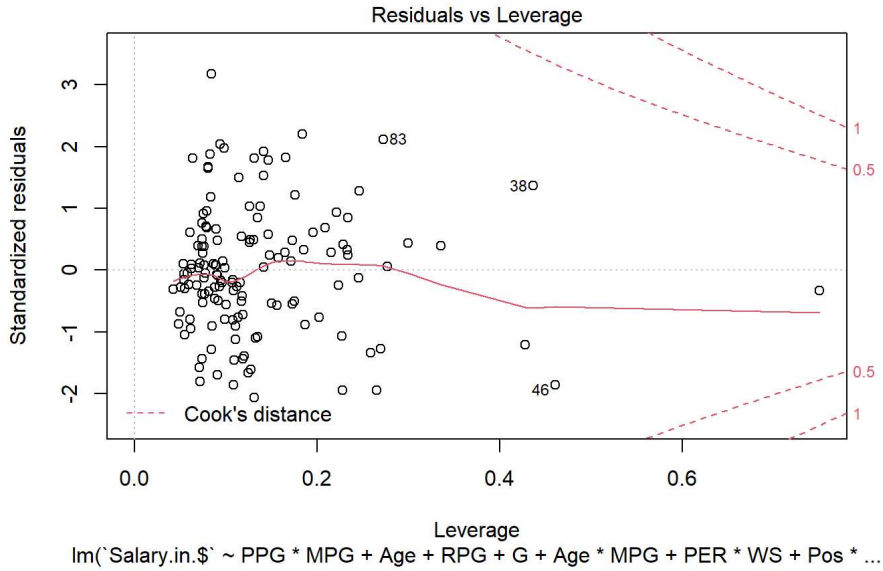
```
##
## Call:
## lm(formula = `Salary.in.$` ~ PPG * MPG + Age + RPG + G + Age *
##     MPG + PER * WS + Pos * WS, data = test)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8868560 -2420052  -251357  2211526 13955336
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13550506    6578073   2.060 0.041530 *
## PPG           391195     493919   0.792 0.429884
## MPG         -1136052     309068  -3.676 0.000354 ***
## Age          -434859     234081  -1.858 0.065619 .
## RPG           343763     380282   0.904 0.367793
## G              27965      25862   1.081 0.281689
## PER           -49723     176379  -0.282 0.778488
## WS          -1143369    1067196  -1.071 0.286115
## PosPF       -3203110    2166547  -1.478 0.141867
## PosPG       -3149362    2247235  -1.401 0.163622
## PosSF       -2728121    2146511  -1.271 0.206162
## PosSG       -4590437    2289185  -2.005 0.047147 *
## PPG:MPG         6555      13436   0.488 0.626534
## MPG:Age        43089      10694   4.029 9.77e-05 ***
## PER:WS         32961      40834   0.807 0.421133
## WS:PosPF      784958     489651   1.603 0.111499
## WS:PosPG      157485     444322   0.354 0.723621
## WS:PosSF      322866     456199   0.708 0.480462
## WS:PosSG      793439     566938   1.400 0.164195
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4595000 on 122 degrees of freedom
## Multiple R-squared:  0.6133, Adjusted R-squared:  0.5563
## F-statistic: 10.75 on 18 and 122 DF,  p-value: < 2.2e-16
```

```
plot(model2)
```

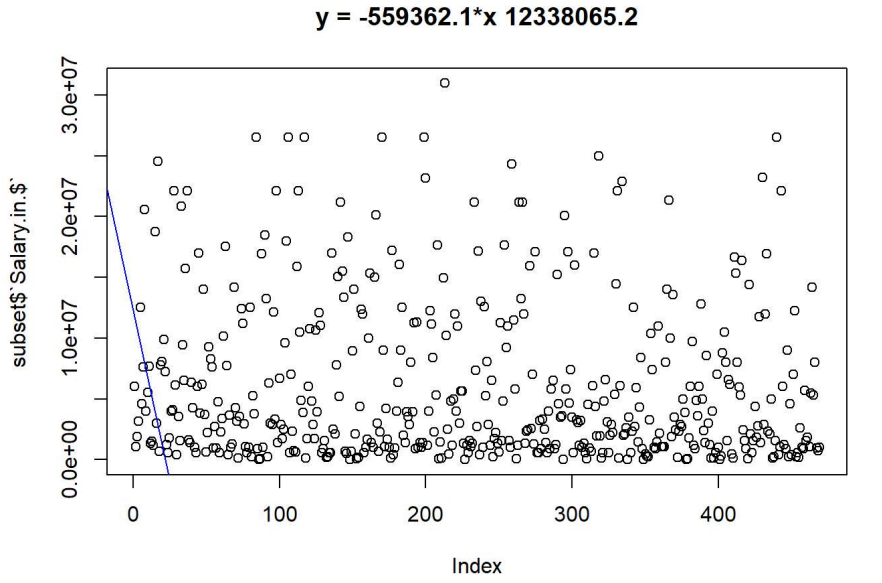

Residuals vs Fitted
Residuals
Fitted values
lm(`Salary.in.$` ~ PPG * MPG + Age + RPG + G + Age * MPG + PER * WS + Pos * ...
32
75
91
1.0e+07
0.0e+00
-1.0e+07
0.0e+00
5.0e+06
1.0e+07
1.5e+07
2.0e+07


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(`Salary.in.$` ~ PPG * MPG + Age + RPG + G + Age * MPG + PER * WS + Pos * ...
320
83
75
3
2
1
0
-1
-2
-2
-1
0
1
2

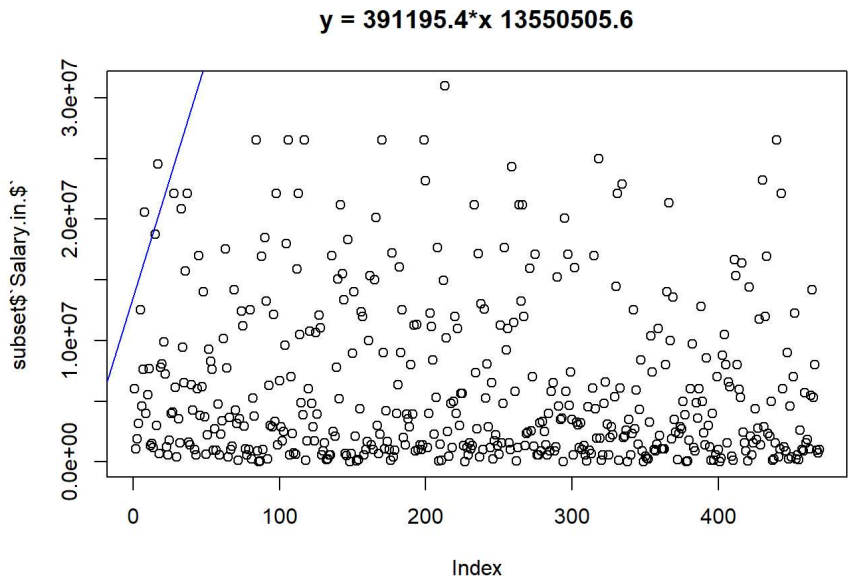

Scale-Location
√|Standardized residuals|
Fitted values
lm(`Salary.in.$` ~ PPG * MPG + Age + RPG + G + Age * MPG + PER * WS + Pos * ...
32
75
83
1.5
1.0
0.5
0.0
0.0e+00
5.0e+06
1.0e+07
1.5e+07
2.0e+07




```
model1 <- lm(`Salary.in.$` ~ PPG * MPG + Age + RPG + G + Age * MPG + PER * WS + Pos * WS, data = train)
coeff <- coefficients(model1)
eq <- paste0("y = ", round(coeff[2],1), "*x ", round(coeff[1],1))
plot(subset$`Salary.in.$`, main = eq)
abline(model1, col="blue")
```




```
model2 <- lm(`Salary.in.$` ~ PPG * MPG + Age + RPG + G + Age * MPG + PER * WS + Pos * WS, data = test)
coeff <- coefficients(model2)
eq <- paste0("y = ", round(coeff[2],1), "*x ", round(coeff[1],1))
plot(subset$`Salary.in.$`, main=eq)
abline(model2, col="blue")
```



For example, with Steph Curry 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.

```
model_1 <- lm(`Salary.in.$` ~ PPG * MPG + Age + RPG + G + Age * MPG + PER * WS + Pos * WS, data = subset)
curry <- subset %>% filter(Player == 'Stephen Curry')
# predicted salary
pred_curry <- predict(model_1, curry)
preds_model1 <- c(pred_curry)
```

```
player_names <- c('Stephen Curry')
# see the filtered players
kable(subset %>% filter(Player %in% player_names))
```

Player	Tm	Pos	Age	G	GS	MPG	PPG	APG	RPG	BPG	SPG	TOPG	PFPG	WS	PER	VORP	2P%	3P%	FG%	TS%	U
Stephen Curry	GSW	PG	28	79	NA	33.39241	25.3038	6.620253	4.468354	NA	NA	NA	2.316456	12.6	24.6	NA	0.537	0	0.468	0.624	N

```
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_model1
# 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.$`)
names(true_salaries_2018) <- c("Player", "true_salary_2018")
# see the differences
salaries_comparison <- full_join(true_salaries, true_salaries_2018,
                                 by =c('Player' = 'Player' ) )
salaries_comparison$pred_error <- salaries_comparison$true_salary_2018 -
  salaries_comparison$predicted_2018
kable(salaries_comparison)
```

Player	salary_2017	predicted_2018	true_salary_2018	pred_error
Stephen Curry	12112359	17357089	34682550	17325461

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 predict 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 $17,325,461 more 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.But on the flip side there is a drop of players after the age of 24 because that is around the time that most players are finishing their rookie contract, and teams are most likely letting go of them because their production is not of their liking, so why pay more for a aging player that has shown no progress than paying less for another promising player that in the NBA can reach a higher ceiling than the other.