2016 Fall Final Project

Stat 133

Professor Gaston Sanchez

December 2, 2016

Member: Seung Soo Kim, Zimei Yuan, Jiyoon Jeong, Jin Kweon

Research Question

*In the 2015-2016 season, how do the skills of a player relate to his salary?*

Abstract & Motivation

This report analyzes what National Basketball Association (NBA) players’ salaries have to do with their performance (performance including many aspects, and we are going to talk about these as well). The motivation is to think of and predict how each team/the NBA sets the salary for each player, based on indicators (we say skills based on how I defined for our project. We defined skills later in our report): games played, age, positions, goals, rebounds, blocks, etc. Throughout the work of the project, our group tried to find out which variables are affecting players’ salaries the most. At the end, we also analyze players’ value and see if players are getting paid more than their value or less than their value. Our group is formed of four students: Seung Soo, Zimei, Jiyoon, and Jin. We have used R language the most, to come up with the answers. All the files and results are uploaded to Open Science Framework (OSF), with the link following: osf.io/3nqkp. The types of file extensions are “.R,” “.Rmd,” “.md,” “.txt,” “.csv,” and “.pdf.”

README File

Before we talk about our project, we want to introduce README File beforehand. We want to include a README file to show the general structure of our working directory. The README.md file includes the title (Stat133 Final Project Fall 2016), the authors’ names, and a description of this project and the directory-files structure. This README file will let readers know about other files in the directory and therefore see our work clearly.

Introduction

Our analysis are focusing on the following five questions: 1. How can we evaluate the performance of the players? 2. Which skills are more correlated with salary? 3. Are there any differences in skills and salary depending on the players’ position? 4. Are the players really worth the amounts of money clubs pay for? 5. Are there any undervalued or undervalued players?

From these five analytical / exploratory research questions above, our group came up with an answer for the main question of our project: “In the 2015 - 2016 season, how do the skills of a player relate to his salary?”

Our group divided our work based on our skills. Seung Soo worked on “efficiency” statistics (EFF), Zimei worked on Readme, data dictionaries for each data file, and slides (Jerome also contributed on Readme, data dictionaries, and slides, as well), Clover worked on data acquisition, cleaning, exploratory data analysis (EDA), and Shiny Apps, and Jin worked on Shiny Apps, ggplots, and the report (all other members contributing on the report, as well).

Data

First of all, NBA is the abbreviation of the National Basketball Association: men’s professional basketball league in North America. All the players’ performances and salaries are based on the 2015 – 2016 season. And, the main source is the NBA official website:<http://www.basketball-reference.com/>. In that season, there are thirty teams in the league. So, in total, we acquired 90 raw data files.

We developed all the raw data tables for the following three types: roster, totals, and salaries, for each of the thirty teams in the NBA. The names of the teams are as follows: “Atlanta Hawks (ATL),” “Boston Celtics (BOS),” “Brooklyn Nets (BRK),” “Charlotte Hornets (CHO),” “Chicago Bulls (CHI),” “Cleveland Cavaliers (CLE),” “Dallas Mavericks (DAL),” “Denver Nuggets (DEN),” “Detroit Pistons (DET),” “Golden State Warriors (GSW),” “Houston Rockets (HOU),” “Indiana Pacers (IND),” “Los Angeles Clippers (LAC),” “Los Angeles Lakers (LAL),” “Memphis Grizzlies (MEM),” “Miami Heat (MIA),” “Milwaukee Bucks (MIL),” “Minnesota Timberwolves (MIN),” “New Orleans Pelicans (NOP),” “New York Knicks (NYK),” “Oklahoma City Thunder (OKC),” “Orland Magic (ORL),” “Philadelphia 76ers (PHI),” “Phoenix Suns (PHO),” “Portland Trail Blazers (POR),” “Sacramento Kings (SAC),” “San Antonio Spurs (SAS),” “Toronto Raptors (TOR),” “Utah Jazz (UTA),” and “Washington Wizards (WAS).”

All the roster tables include information on roster numbers, roster players’ names, positions, heights, weights, birth dates, countries of birth, NBA years of experiences, and colleges attended. All the totals tables (or stats tables) include information about total ranks, roster players’ names, ages, games, games started, minutes played, field goals, field goal attempts, field goal percentages, 3-point field goals, 3-point field goal attempts, 3-point field goal percentages, 2-point field goals, 2-point field goal attempts, 2-point field goal percentages, effective field goal percentages, free throws, free throw attempts, free throw percentages, offensive rebounds, defensive rebounds, total rebounds, assists, steals, blocks, turnovers, personal fouls, and points. All the salary tables include information about salary ranks, roster players’ names, and salaries.

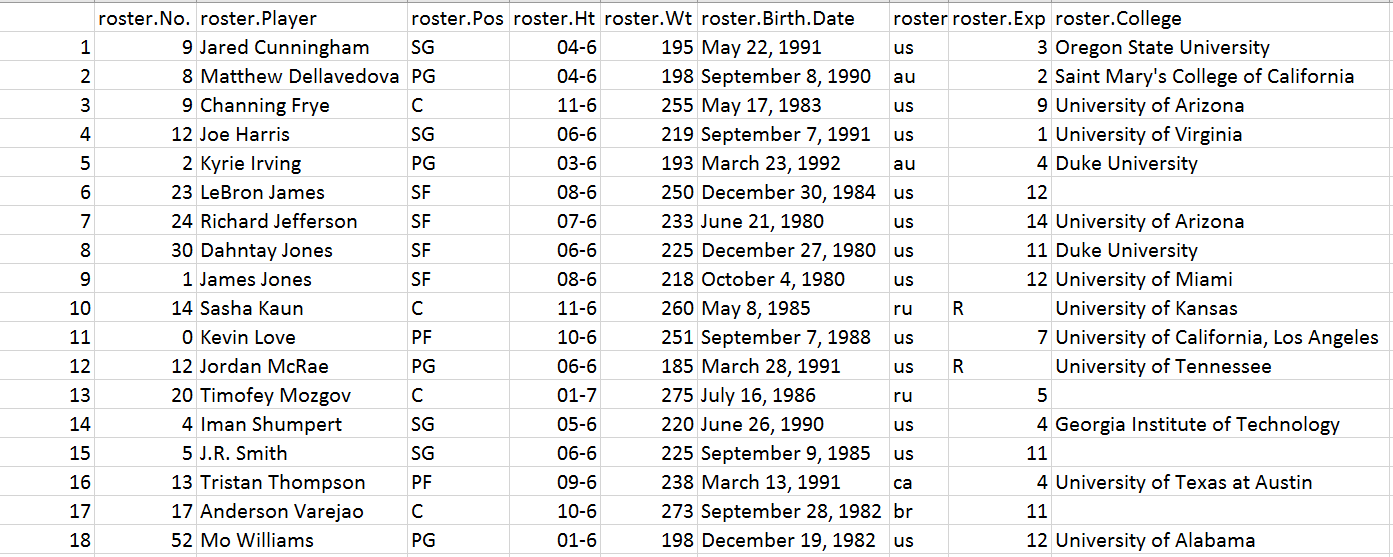


Figure 1. This is one of the roster table (csv) we acquired from the <http://www.basketball-reference.com/>.



Figure 2. This is one of the salary table (csv) we acquired from the <http://www.basketball-reference.com/>.

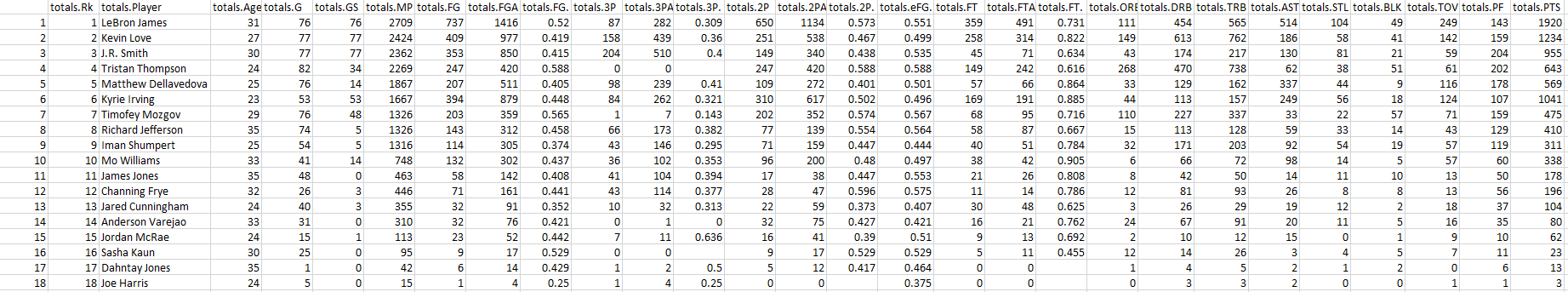


Figure 3. This is one of the totals (stats) table (csv) we acquired from the <http://www.basketball-reference.com/>.

After we obtained the raw data of each team’s records, we removed unnecessary data, edited the data structure and measurements of each variable, and added certain necessary data to the new data frame, to generate ‘roster-salary-stats.csv.’ The methods we used to acquire cleaned table are described under the section, Methodology.

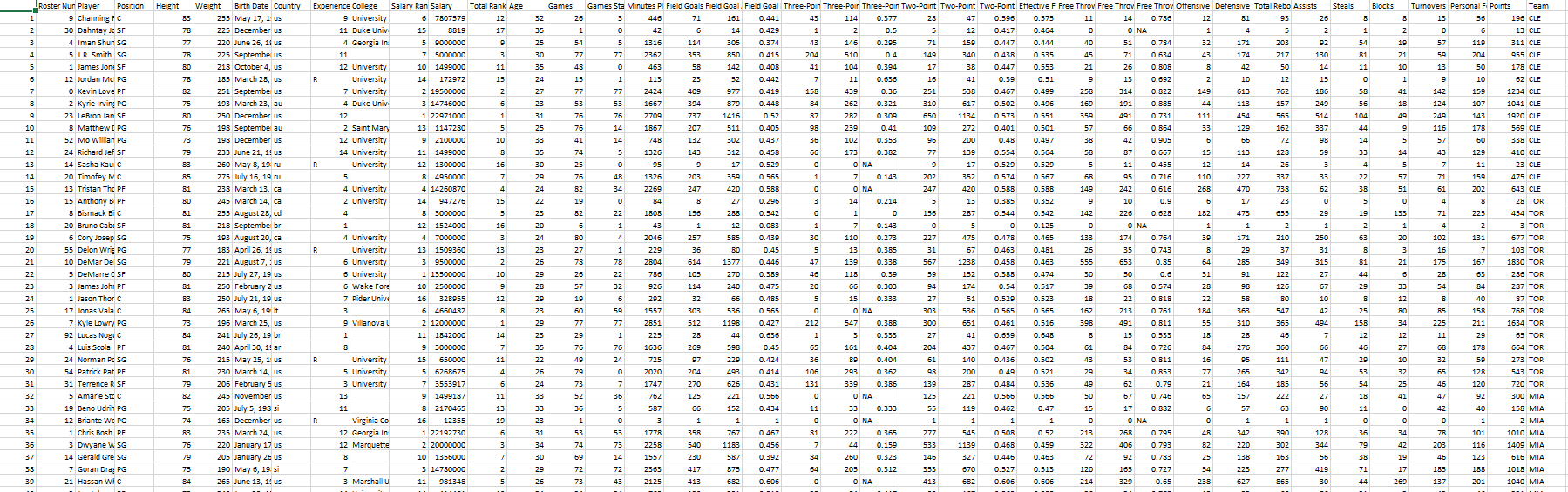


Figure 4. This is the part of a cleaned csv (roster-salary-stats.csv). This table includes 39 columns in total: roster numbers, roster players’ names, positions, heights, weights, birth dates, countries of birth, NBA years of experiences, colleges attended, salary ranks, salaries, total ranks, ages, games, games started, minutes played, field goals, field goal attempts, field goal percentages, 3-point field goals, 3-point field goal attempts, 3-point field goal percentages, 2-point field goals, 2-point field goal attempts, 2-point field goal percentages, effective field goal percentages, free throws, free throw attempts, free throw percentages, offensive rebounds, defensive rebounds, total rebounds, assists, steals, blocks, turnovers, personal fouls, points, and team names.

Methodology & Analysis & Results

We modified the provided code using loops and creating our own function to avoid repetitive types of codes, and to shorten the lengths of codes. The Data Scraping part is composed of two R scripts, which are ‘download-data-script.R’ and ‘download-data-functions.R’ inside of the functions and scripts files, respectively.

First, this is the methodology of the data scraping. We extracted each team’s information, such as href attributes, from the anchor tags, and teams’ abbreviations from the specified URL. ----(1) This part of the code is included in an R script named ‘download-data-functions.R’, since we need the data such as ‘team\_names’ in ‘clean-data-script.R’ to load each team’s raw data files by the team’s name. Merely Sourcing ‘download-data-functions.R’ enables us to get the NBA team’s name in ‘clean-data-script.R’ without writing the same code to get the team’s name.

=======================================================================

#extract\_team\_info

# base url

basketref <- 'http://www.basketball-reference.com'

# parse 'http://www.basketball-reference.com/leagues/NBA\_2016.html'

url <- paste0(basketref, '/leagues/NBA\_2016.html')

doc <- htmlParse(url)

# identify nodes with anchor tags for each team and

# extract the href attribute from the anchor tags

team\_rows <- getNodeSet(doc, "//th[@scope='row']/a") ----(1)

team\_hrefs <- xmlSApply(team\_rows, xmlAttrs) ----(1)

# just in case, here's the character vector with the team abbreviations

team\_names <- substr(team\_hrefs, 8, 10)

=======================================================================

Figure 5. This is part of the ‘download-data-functions.R.’ All explanations are above.

After we obtained each team’s name and URL for designated data from the previous code, we had to get the information on the roster, salary, and stat table in that URL by recognizing the initial line position of roster, salary, and stat html table. ----(1) The last line where those three html tables end was obtained by the variable ‘line\_counter’, where it counts the number of lines before </table>.---- (2)

After we obtain the first and last line position on the desired html table, we read the information on rosters, salaries, and stats html tables as data.frame. ----(3) Finally, we export the acquired data frame as a csv file, and name it in the form of ‘the type of data (roster, salaries, or totals)’-‘each team’s name’. ----(4) The purpose of this code is to obtain each team’s csv file, and it is more efficient run by functions, since the same logic (getting the initial and last line of the html table and exporting the table as a csv file) with different arguments (the different file path and name of the different type of tables) applies to all three html tables (roster, salary, and stats). Therefore, function ‘exportcsv’ is created, and this function is run by ‘download-data-script.R’ to generate each team’s csv file for three types of data tables.

========================================================================

exportcsv <- function(id, filepath, basketref, team\_hrefs, team\_names) {

for(i in 1:30) {

# Read html document (as a character vector) for a given team

# (first team is "CLE")

url\_team <- paste0(basketref, team\_hrefs[i])

html\_doc <- readLines(con = url\_team)

# initial line position of input file(such as roster) html table

begin\_roster <- grep(id, html\_doc) ----(1)

# find the line where the html ends

line\_counter <- begin\_roster ---- (2)

while (!grepl("</table>", html\_doc[line\_counter])) {

line\_counter <- line\_counter + 1

}

# read input argument file's(such as roster) table as data.frame

# and export it as csv

roster <- readHTMLTable(html\_doc[begin\_roster:line\_counter]) ----(3)

write.csv(roster, file = paste0('data/rawdata/', filepath,

team\_names[i], '.csv')) ----(4)

}

}

======================================================================

Figure 6. This is the last part of the ‘download-data-functions.R.’ All explanations are above.

Second, this is the methodology of the data cleaning.

**- Removed and edited data from raw data**

1. In salaries. Salary column in Salary csv, $(dollar sign) in front of each player’s salary is removed and the salary data is changed into numeric vectors. ----(1)

2. In roster csv, the original roster.Ht columns’ values is in inches and feet, but we removed dash(-) and changed the measurement to feet. We also changed this column vectors in numeric vectors to make the graph easier to understand, in the Analysis part. ----(2)

3. When we merged all the columns of each data table, we obtained duplicated players, belonging to different teams. Those players’ rows were removed, except for the first row of each player. For example, we have ‘Alex Stepheson’ who belongs to LAC, and ‘Alex Stepheson’ who belongs to MEM, and we just choose the ‘Alex Stepheson’ who belongs to LAC, since it is the first row that shows data for ‘Alex Stepheson.’ ----(3)

4. We removed the previous column names and added more descriptive column names, in order for us to be able to recognize the data easily. ----(4)

**- Added data**

1. We added a new column, called “Team”, since we need to arrange players by the names of their teams, to draw the graph in Shiny App (team-salaries) and obtain information in “eda-script.R.” ----(5)

**- Joined data**

1. First, we merged roster table data and salaries table data, which are ordered by players’ names. We saved this data into the variable called ‘a’ ----(6)

2. Secondly, we merged the data ‘a’ with the stats table data, which are also ordered by players’ names, and we got the merged data of the three tables. ----(7)

Here is the code we used to clean the data, below: (we numbered the instructions for the readers)

============================‘clean-data-script.R’==============================

...

a <- c(0)

b <- c(0)

cleaned <- c(0)

for(i in 1:30) {

# Save each team's roster, totals, and salaries data into each variables

roster\_raw = readfiles('data/rawdata/roster-data/roster-',team\_names[i])

totals\_raw = readfiles('data/rawdata/stat-data/stats-',team\_names[i])

salaries\_raw = readfiles('data/rawdata/salary-data/salaries-',team\_names[i])

# Order each team's data by name

order\_roster = orderfiles(roster\_raw,3)

order\_salaries = orderfiles(salaries\_raw,3)

order\_totals = orderfiles(totals\_raw,3)

# variable 'a' get the result of combined roster file's data with salary

# file's data by the name of the players. Only select and combine the

# player's information who are listed on both files.

a <- jointable(order\_roster,order\_salaries) -----(6)

# variable 'b' get the result of combined the data in 'a' with total file's

# data by the name of the players. Only select and combine the

# player's information who are listed on both files.

b <- jointable2(a,order\_totals) ----(7)

# Add team's name at the last columns

b$Team <- team\_names[i] ----(5)

...

# Change column's name into more descriptive name

colnames(cleaned)[1:38] <- c("Roster Number", "Player", "Position", "Height",

"Weight", "Birth Date", "Country", "Experience",

…

"Free Throw Attempts", "Free Throw Percentage",

"Offensive Rebounds", "Defensive Rebounds",

"Total Rebounds", "Assists", "Steals", "Blocks",

"Turnovers", "Personal Fouls", "Points") ----(4)

# Modify Salary column's data into numeric vectors and remove dollar sign

cleaned$Salary = as.numeric(gsub("\\$|,", "", cleaned$Salary)) ----(1)

# Modify Height column's data into numeric vectors and convert into inch

cleaned$Height = as.numeric(str\_split\_fixed(cleaned$Height,"-",n=2)[,1])\*12 +

as.numeric(str\_split\_fixed(cleaned$Height,"-",n=2)[,2]) ----(2)

# Remove dulicated players on the list

cleaned = cleaned[!duplicated(cleaned$Player),] // ----(3)

# Export cleaned dataframe as a csv file

write.csv(cleaned, "data/cleandata/roster-salary-stats.csv")

========================================================================

Figure 7. This is part of the “clean-data-script.R.’ All explanations are above.

For exploratory data analysis (in eda-script), we divide up quantitative and qualitative variables. The qualitative variables in our csv file are four in total: “player name,” “position,” “birth date,” “country origin,” “colleges players graduated,” and “teams.” The other 33 are all quantitative, such as roster number, height, weight, salary, etc. With quantitative variables, there are some special cases we have to modify for some columns. For example, the experience column has “R”, and the three-point field goal percentage has “NA,” and to make R program consider it numeric, we decided to make “R” and “NA” equal to zero (0) – there is either no information, or players have no experience in the columns.

Here is an example below:

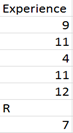


Figure 8. This picture shows one of the examples that some columns include a character while most of the elements are numeric.

In this R script, we used ‘sink’ (function in R) eda-output.txt to save minimum, median, tenth / twenty-five / seventy-five / ninety quartiles, inter-quartiles, maximum, minimum, median, standard deviation, range, difference (between the biggest and smallest numbers), and sums for quantitative variables in eda-output.txt. Most of the time, we used for-loop to get summaries for each column.

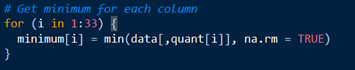


Figure 9. This picture is part of the codes how we used for-loop to get statistics for some variables.

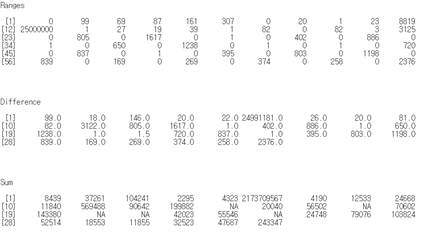


Figure 10. This is part of the ‘eda-output.txt,’ including all the statistics.

Also, we made some graphs, but there are some variables that we did not use in the graphs: roster number, player name, birth date, salary rank, and total rank. We skipped them because those variables have nothing to do with the skills of the players. Quantitative variables are interpreted with histograms and boxplots (and summary statistics in eda-output.txt), and qualitative variables are interpreted with bar-charts (and frequency tables in eda-output.txt). For every bar-chart, we displayed them horizontally, for visual purposes.

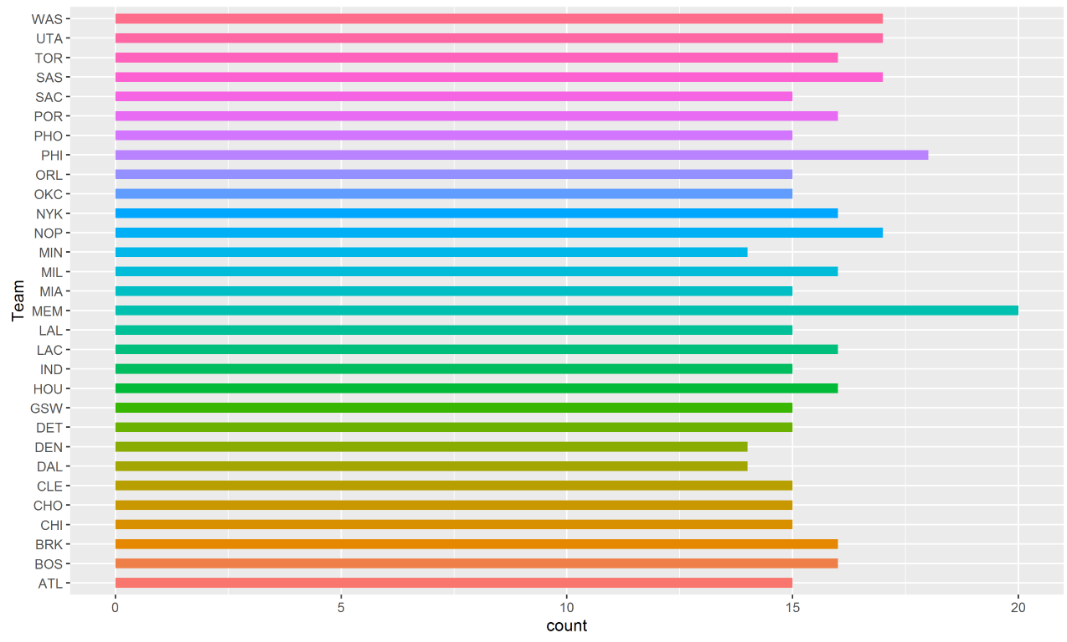


Figure 11. Here is one of the bar charts, displaying the counts of the teams.

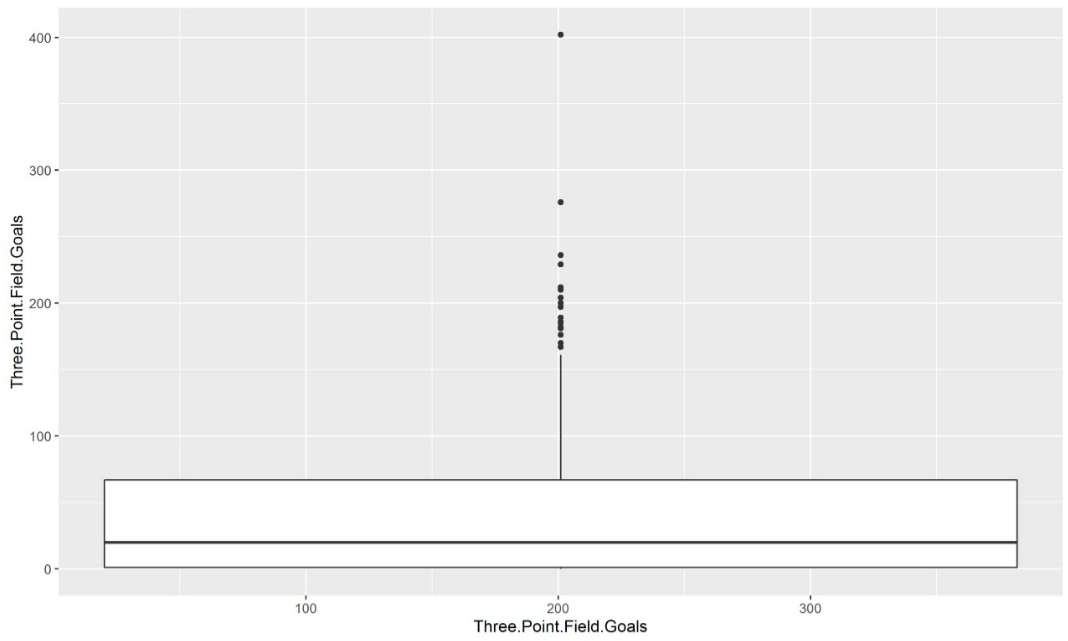


Figure 12. Here is one of the box plots, displaying the data (IQR, quartiles, media, etc) for three point field goals.

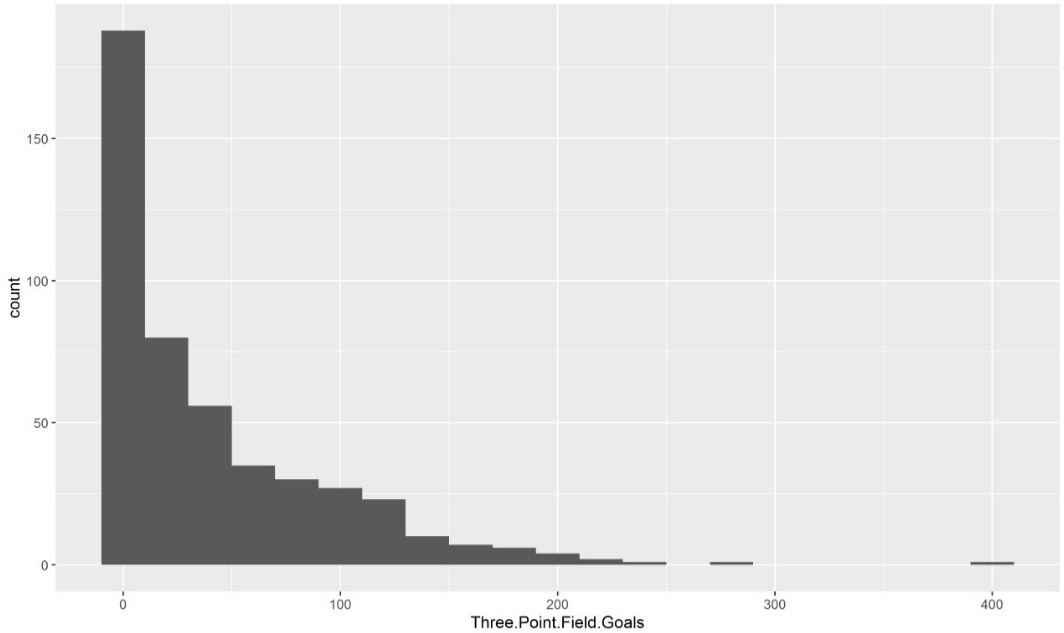


Figure 13. Here is one of the histograms, displaying the counts of three point field goals.

PCA - The original efficiency formula is offense-oriented, so defensive players might seem less efficient compared to those who have offensive positions. In order to balance the efficiency between offensive and defensive players, we applied Principal Component Analysis (PCA) to the original efficiency.

Modified EFF - In order to calculate the efficiency for each position, we classified the cleandata by players’ positions, and added missing records (missed free throws, missed field goals, and turnovers). Also, we divided records (points, assists, steals, etc.) by the number of played games, to see the efficiency for each game. The modified efficiency was obtained by multiplying each variable with a coefficient that is derived by PCA. To standardize the coefficient (weight), we divided the coefficient by a standard deviation of each position. So the modified EFF formula is the sum of coefficient \* variables (per game) / standard deviation.

“eff-stats-salary.csv” consists of 14 columns (refer the picture below). The efficiency index is the modified efficiency we obtained by using PCA.

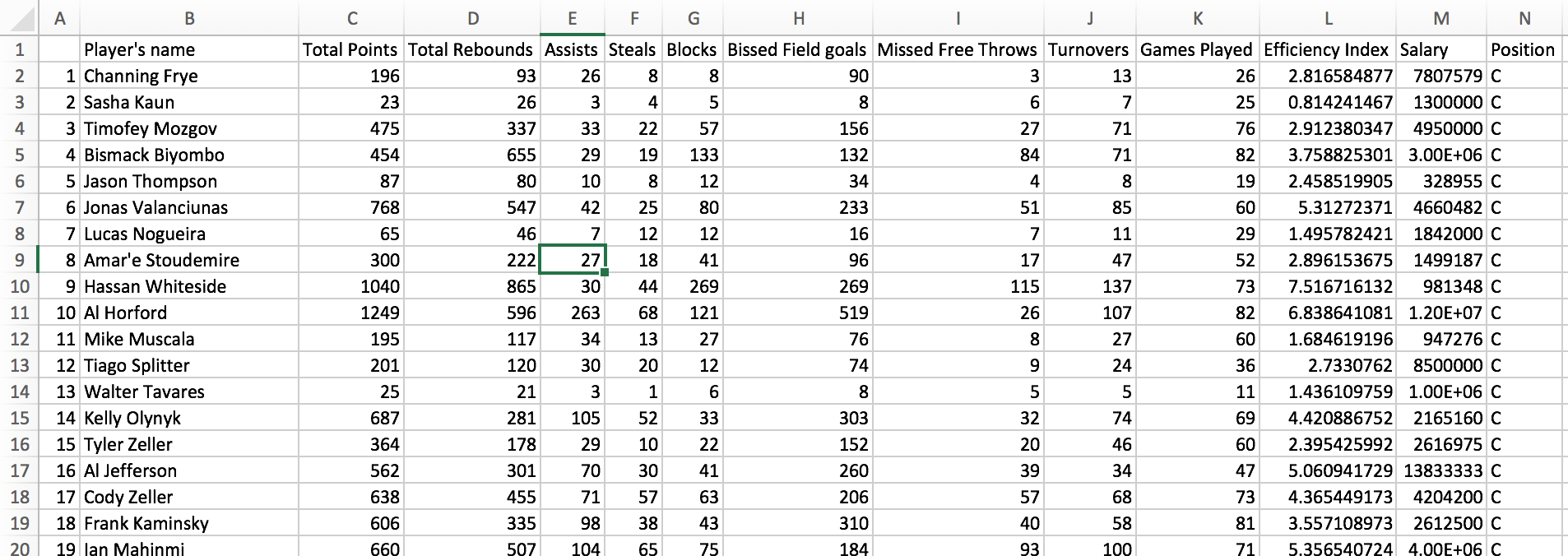


Figure 14. This is part of the “eff-stats-salary.csv.” We added an extra column, ‘position’ for Shiny app “stat-salaries.”

Shiny App shows the correlation between two variables easily. For example, in the Shiny App image below, we can see how total points and salary are positively correlated. Also, we see that most of the players are distributed between 0 and 300 points, regardless of their positions.

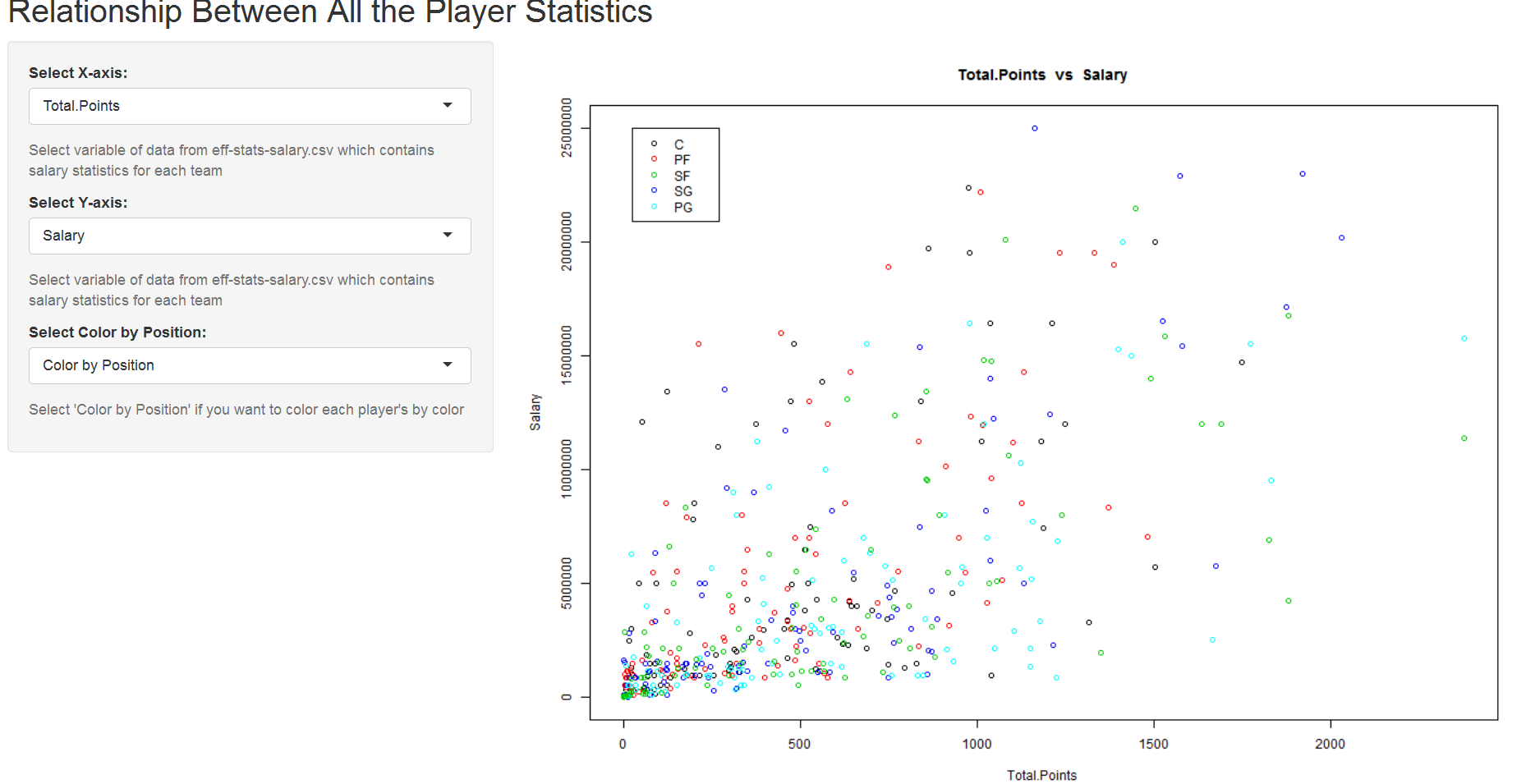
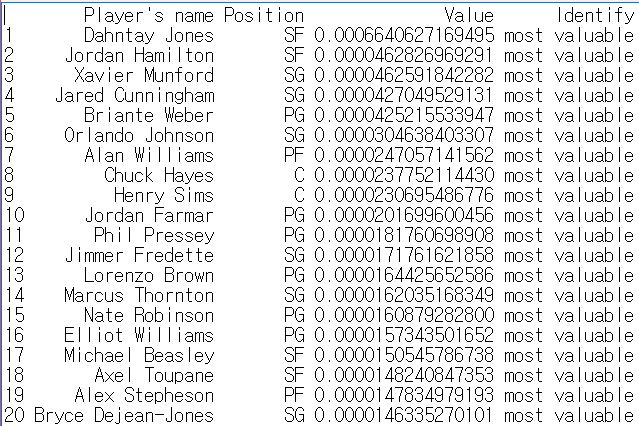


Figure 15. This is one of the Shiny pictures: showing the relationship between total points and salary.

“best-worst-value-players.txt” consists of 5 columns (refer the picture below). We picked the top 20 and bottom 20 players among all of them, ordering them by the value. The value of a player was obtained by a formula: efficiency divided by salary.



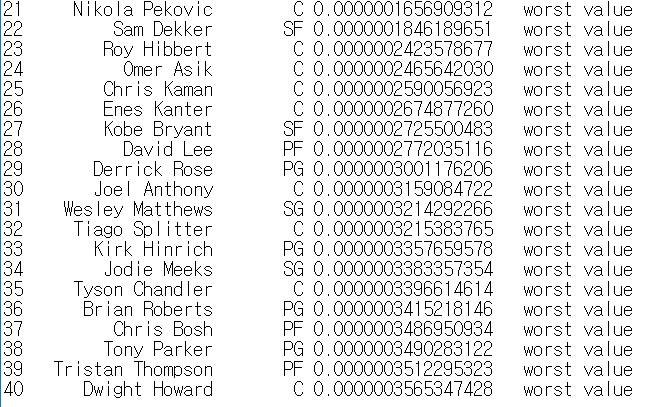


Figure 16, 17. This is picture of the “best-worst-value-players.txt,” showing the 20 most and worst value players.

From now on, we answered the five main questions below for analysis:

1. How can we evaluate the performance of the players?

We calculated applied modified efficiency by using PCA to standardize the indicator, because the original formula is offensive position oriented.

Formula = (w1\*PTS + w2\*REB + w3AST + , , + w8\*TO) / GP, (w1~w8 are weights)

**Efficiency of center**

Nazr Mohammed has the lowest efficiency (0.6529) among all center players. DeMarcus Cousins has the highest efficiency (12.294) among all center players. The median efficiency of centers is 3.6299, and the mean is 4.0088.

**Efficiency of power forward**

Branden Dawson has the lowest efficiency (0.3678) among all power forward players. Draymond Green has the highest efficiency (11.5831) among all power forward players. The median efficiency of power forwards is 3.6971, and the mean is 4.0242.

**Efficiency of small forward**

Sam Dekker has the lowest efficiency(0.304) among all small forward players. LeBron James has the highest efficiency (11.149) among all small forward players. The median efficiency of power forwards is 3.474, and the mean is 3.878.

**Efficiency of shooting guard**

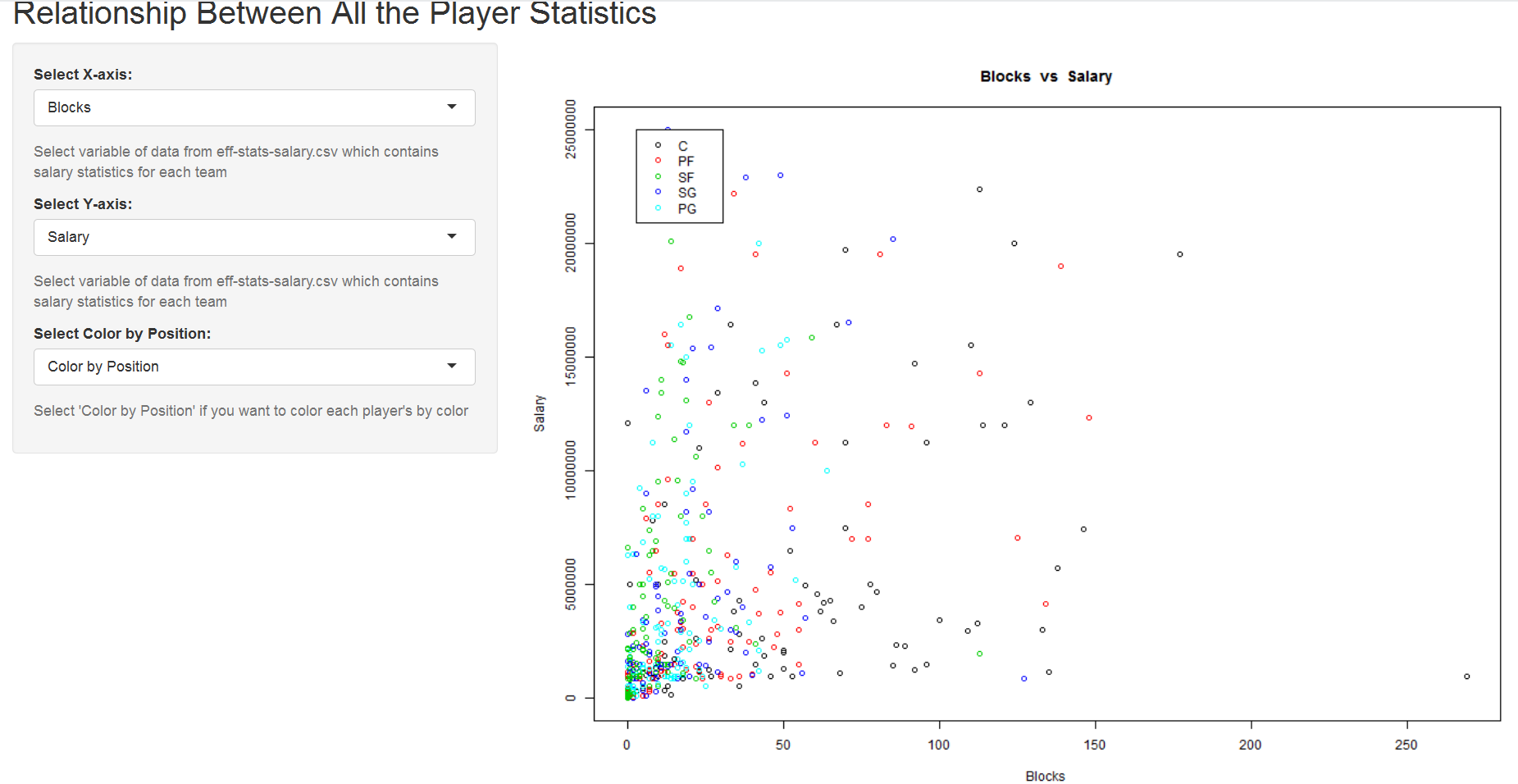
Luis Montero has the lowest efficiency (0.6448) among all shooting guard players. James Harden has the highest efficiency (13.7412) among all shooting guard players. The median efficiency of power forwards is 4.1755, and the mean is 4.5777.

**Efficiency of point guard**

Bryce Cotton has the lowest efficiency (0.0477) among all point guard players. Russell Westbrook has the highest efficiency (10.7583) among all point guard players. The median efficiency of power forwards is 3.67313, and the mean is 4.14635.

2. Which skills are more correlated with salary?

Our group defines skills as all the nine elements that consist of EFF. From the Shiny App “stat-salaries,” we can find some skills that have something to do with salaries, as follows: total points, total rebounds, assists, steals, missed field goals, and turnovers. However, some skills barely have relationships with salaries, as follows: blocks, missed free throws, and games played. These decisions are purely made through our observation (filtering out the skills that are likely to correlate with salary, before finding the correlations) of the graphs on Shiny App “stat-salaries.” For example, when you see the relationship between blocks and salaries below, most of the dots are concentrated near the origin, and many players have big gaps in salary, even though they have similar blocking numbers.

Figure 18. This is one of the Shiny app showing the relationship between blocks and salary. (show how each skill is correlated with salary)

After, our group tried to make relationships between skills (including the unlikely related skills, as well, to make sure our visual predictions were correct) and salaries.

We concluded that cor\_point most correlated with salary, with the value of 0.6400 rounded to. And, the least correlated skill with salary was cor\_blocks, with the value of 0.3589 rounded to.

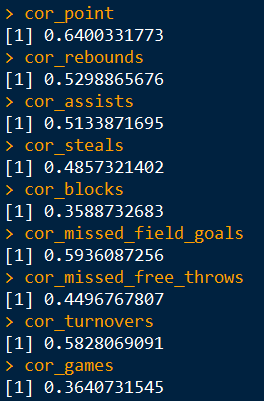
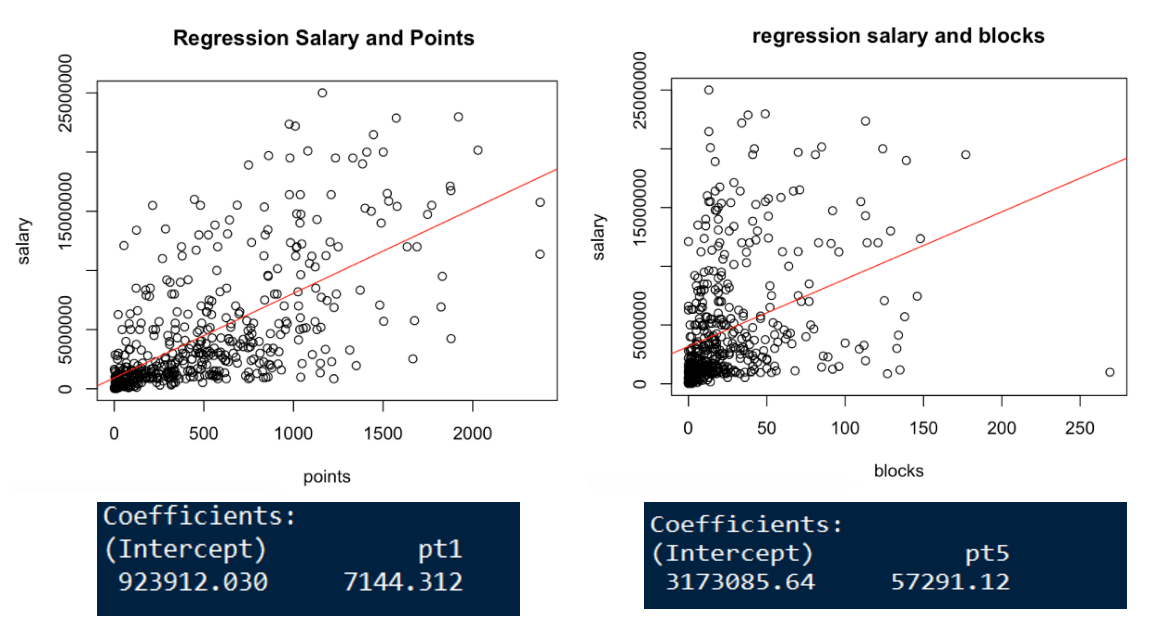


Figure 19. This is the table of the correlation for every skill.

Most correlated with salary, in ascending order, are the following:

1. cor\_blocks (blocks)
2. cor\_games (games played)
3. cor\_missed\_free\_throws (missed free throws)
4. cor\_steals (steals)
5. cor\_assists (assists)
6. cor\_rebounds (rebounds)
7. cor\_turnovers (turnovers)
8. cor\_missed\_field\_goals (missed field goals)
9. cor\_point(point)

Figure 20. These are some of the regressions between salary and other skills.

3. Are there any differences in skills and salary, depending on the players’ positions?

We came up with the most and least correlated skills with the salaries, for each position. By seeing how each position is related to skill, we could tell how each position requires different skills. First of all, the center’s salary has the most correlation with points, and the least correlation with steals. Second, the point forward’s salary has the most correlation with points, and the least correlation with games. Third, the small forward’s salary has the most correlation with assists, and the least correlation with games. Fourth, the shooting guard’s salary has the most correlation with assists, and the least correlation with games. Last, the point guard’s salary has the most correlation with assists, and the least correlation with blocks. From this analysis, definitely, points and assists are the important factors for every player in every position, for earning a higher salary, and the number of games they played is not that significant a factor.

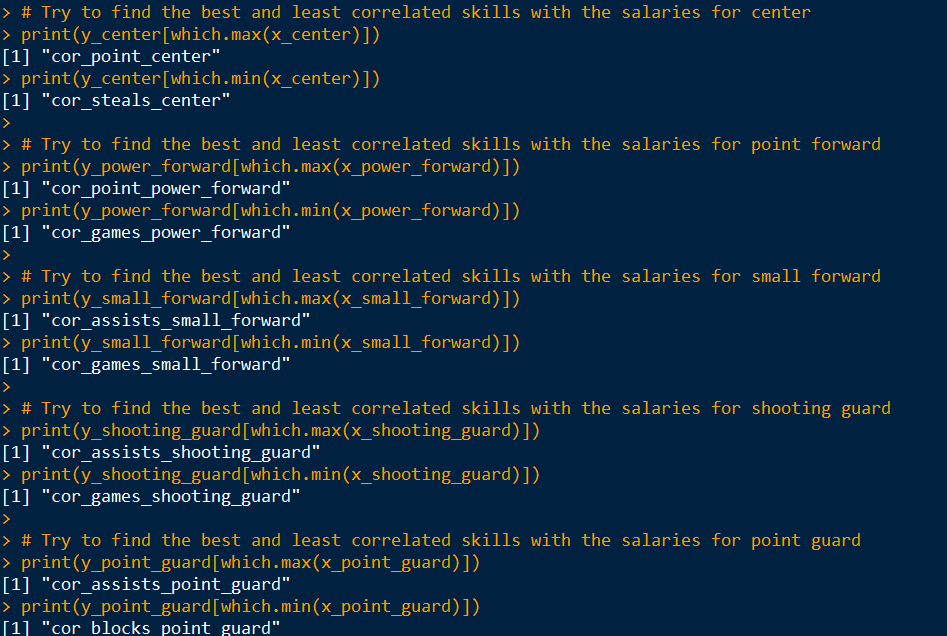
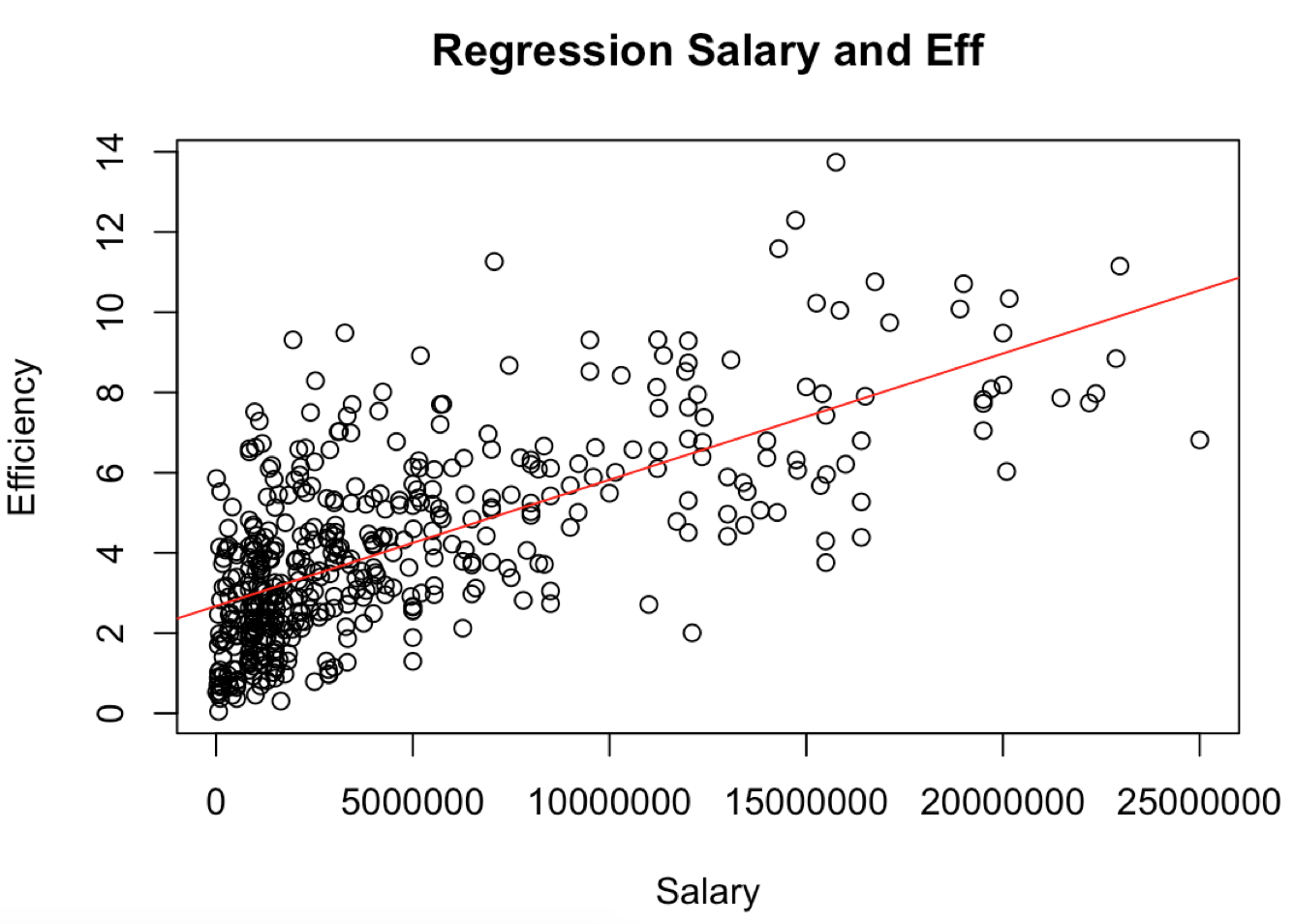


Figure 24. It shows the most and least correlated skill of each position.

4. Are the players really worth the amount of money clubs pay for?

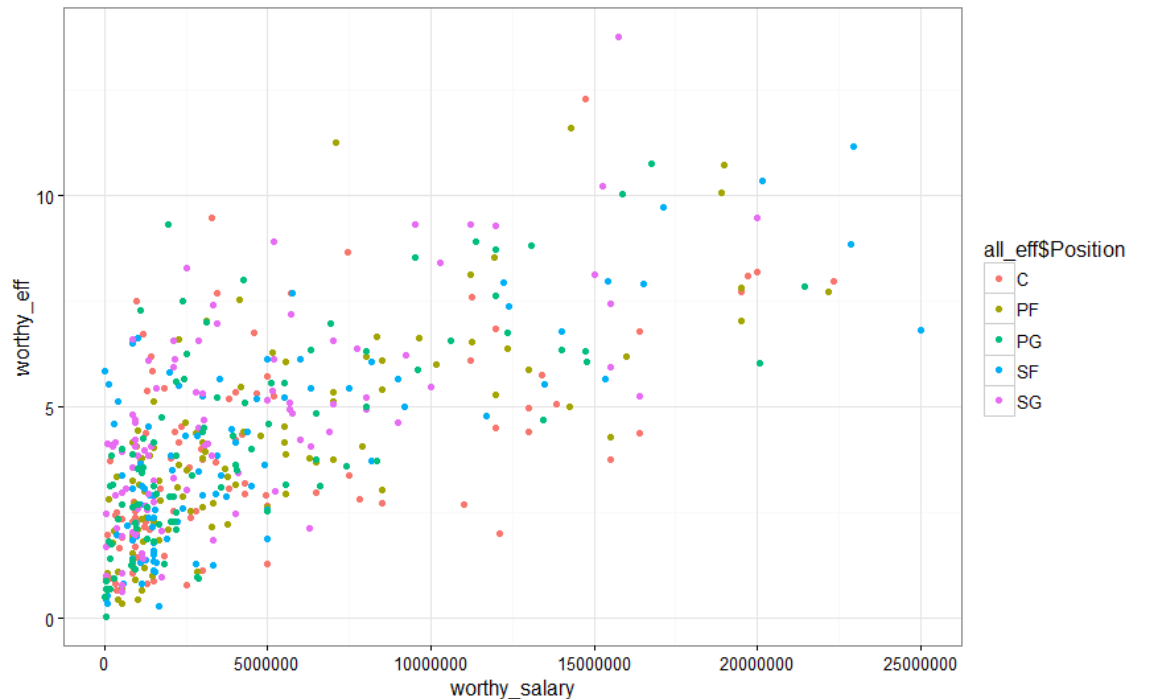


Figure 25, 26. These two show regression between EFF and salary to check worthiness of salary.

We made two graphs to see if players are worthy of their salary: a plot graph with a regression line, and a colored plot graph based on positions. We assume that the regression line indicates the adequate line between salary and efficiency. Although most of the players are positioned close to the regression line, there are some outliers. As the above graph shows, the numbers above and below the regression line are similar. The points above the line are considered “worthier” players than the clubs pay for, and the points below the line are considered “less worthy” players than the clubs pay for. It is easily shown in the position-colored graph that many centers and small forwards (and slightly the point forwards) are less worthy, and many shooting guards (and slightly the point guards) are worthier. This graph indirectly shows that most of the forwards are earning much higher salaries than the guards who have similar stats.

5. Are there any undervalued or overvalued players?

The value of player is defined by the efficiency and salary. If a player gets paid more than his efficiency, we can consider him as an overvalued player. Conversely, if a player gets paid less than he should, we can consider him as an undervalued player. We have already calculated the value that is efficiency divided with the salary, and the volume of values represent whether the player is getting too much money or too less money. In order to sort undervalued and overvalued players, I ordered all players by the their value and extracted 30 players with top 30 values and 30 players with bottom 30 values.



Figure 27. It shows top 30 overvalued and 30 undervalued players

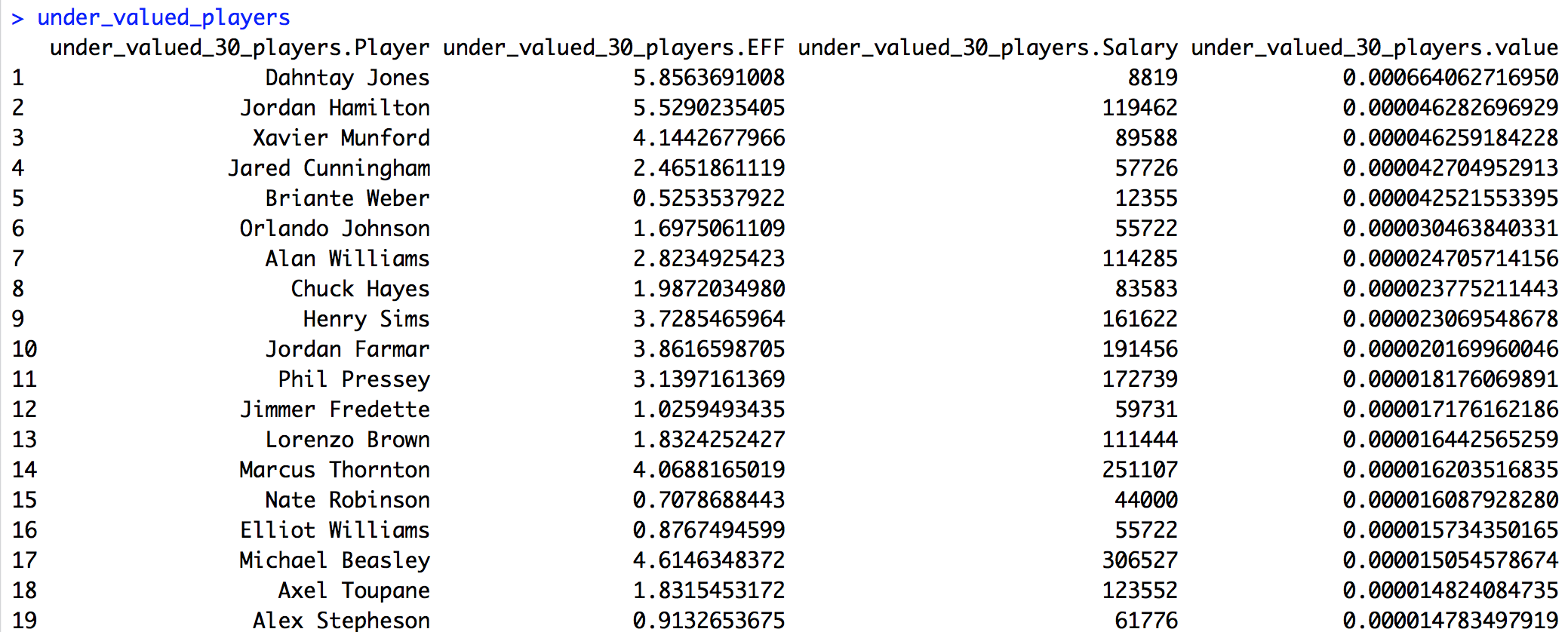


Figure 28. It is the list of under-valued players.

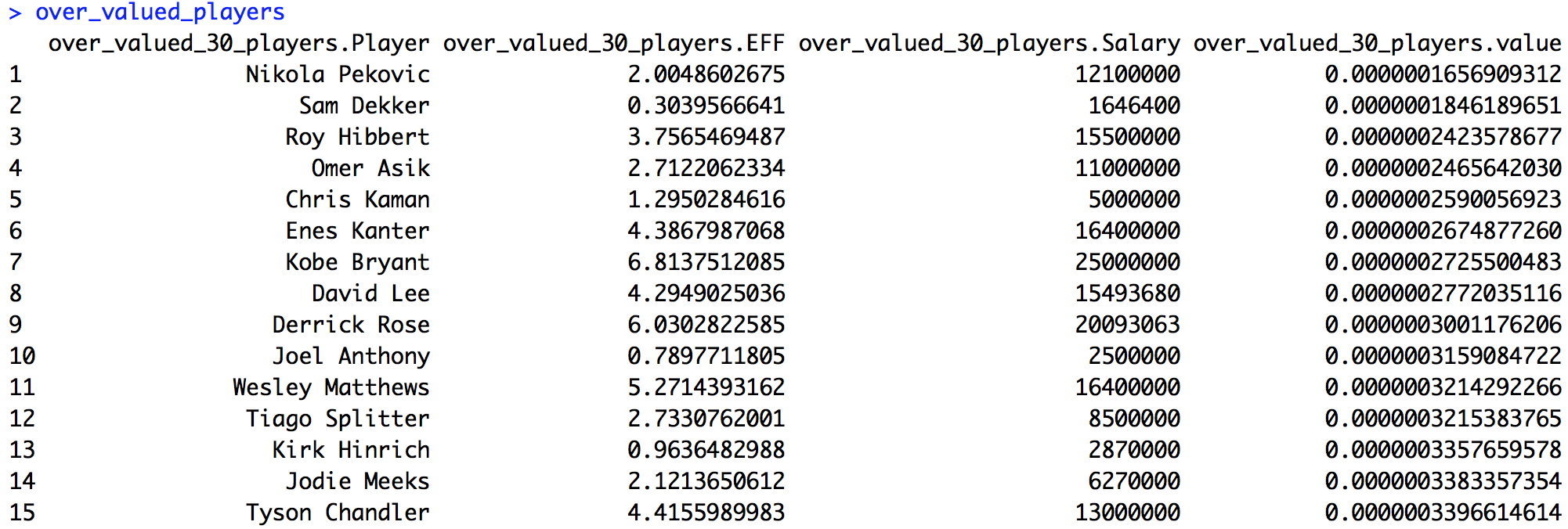


Figure 29. It is the list of over-valued-players.

Note: All the codes for the analysis are belong to the *compute-efficiency-script.R* script.

The interesting thing we found is that the NBA has different relationships than other sports, for example, soccer. As we demonstrated in our analysis, most of the positions’ salaries have the least correlation with the number of games they played, but in soccer, the games they played definitely have a lot to do with their salaries. The games they played are great proof, from soccer players, that they are important in the team; thus, the more games soccer players play, the higher the salaries they usually earn. Also we were surprised that the most related factor was total point among all positions, although three out of five positions showed that the number of assists was the most related factor. Thus, we assume that the total point of power forward and center has a higher influence the correlation than the number of assist does.

Conclusion

We found that points, missed field goals, and rebounds are the top three correlation factors for salary; however, blocks, games played, and missed free throws are the bottom three correlation factors for salary in general. Although we do not consider position a skill, we did find that position is one of the biggest factors that affects the salary of the players. Each player’s position has different skills that are most / least correlated with their salaries. Here, we found that points and assists are important factors for their salaries, but the number of games played is not that related to salary. We found that shooting guards have the highest efficiency, and point guards have the lowest efficiency. Finally, there are many shooting forwards who are worthier than what they are paid, and many centers and small forwards that are less worthy than what they are paid. As a result, our analysis would be beneficial to the owner of basketball team. The analysis will provide the value of each player, so that the owner can set the amount of salary strategically. However, we are wondering if top three skills that are highly related to the amount of salary are also related to the percentage of victories. In the future, we would like to investigate the factors that influence the percentage of winning, because the ultimate goal of a game would be winning the game. Also, the owner of a team will be able to reward players properly through our analysis.

From our analysis, although there is usually a positive relation between players' salaries and their performance and skills, i.e. usually players who get higher salaries have better performances than other players do, such as more total points, and fewer missed field goals and missed free throws, there do exist some exceptions. In the pictures in Shiny App we showed before, there are some players who have a high salary but who have missed a lot of field goals, and some players who earn an average or low salary actually perform well. So basketball teams should use an accurate statistical method to analyze players' performances as a standard to determine their salaries. They need to see which players are more or less worthy than the others, and distribute salaries more properly.

Reference

@bball\_ref. "Basketball Statistics and History | Basketball-Reference.com." Basketball-Reference.com. SPORTS REFERENCE LLC, 2016. Web. 27 Nov. 2016.