Final Analysis

Zhuo Chen

2018/12/16

What make a good player

Dataset

The dataset I chose is a "csv" file that contains the stats of all the NBA players in the last season. Variables include player names, salary, position, points, rebounds, assists and so on (variable categories and values totally differnt from the NBA file we used in class).

Objective

"What define a good basketabll player" is always a controversial question, as there are so many aspects and factors that affect people's views towards basketball players. To answer this question is not easy, but indexes including "on-court stats", "salary" and "position" should never be neglected. So I try to get on these factors first and study the relationship between them.

1. Get the dataset and fix it

Import the dataset and Get a felling of what this dataset might look like

```
get_data <- function()
{
  nba_file <- "../math110/NBA.csv"
  nba <- data.table::fread(nba_file, header=T,sep=",", quote="\"",
                           dec=".", fill=T, stringsAsFactors = F)
  return(nba)
}

nba <- get_data()
head(nba)
```

```
##            Player PayRnk      pay                         college team pos
## 1:  Stephen Curry      1 34682550          University of Kentucky  GSW  PG
## 2:   LeBron James      2 33285709        University of Notre Dame  CLE  PF
## 3:   Paul Millsap      3 31269231 North Carolina State University  DEN  PF
## 4: Gordon Hayward      4 29727900          University of Kentucky  BOS  SF
## 5:  Blake Griffin      5 29512900            Marquette University  DET  PF
## 6:     Kyle Lowry      6 28703704            University of Kansas  TOR  PG
##    age games point ThrP ThrPA   eFG FreeThrow rebound assist
## 1:  29    51  28.2  4.2   9.8 0.618     0.921     5.1    6.1
## 2:  33    82  28.6  1.8   5.0 0.590     0.731     8.6    9.1
## 3:  32    38  17.0  1.0   3.0 0.509     0.696     6.4    2.8
## 4:  27     1   7.3  0.0   1.0 0.500     0.000     1.0    0.0
## 5:  28    58  19.6  1.9   5.6 0.493     0.785     7.4    5.8
## 6:  31    78  19.5  3.1   7.6 0.553     0.854     5.6    6.9
```

As there are five positions on court, we devide them into two groups, Inside players and Outside players. I do this by adding a column "InOut" to the dataset.

```
get_InOut <- function(nba)
{
  rownum <- nrow(nba)
  for (i in 1:rownum) {
    if(nba$pos[i]=="PF"|nba$pos[i]=="C"){
      nba$InOut[i] <- "Inside"
    }else{
      nba$InOut[i] <- "Outside"
    }
  }
  return(nba)
}

nba <- get_InOut(nba)
head(nba)
```

```
##            Player PayRnk      pay                         college team pos
## 1:  Stephen Curry      1 34682550          University of Kentucky  GSW  PG
## 2:   LeBron James      2 33285709        University of Notre Dame  CLE  PF
## 3:   Paul Millsap      3 31269231 North Carolina State University  DEN  PF
## 4: Gordon Hayward      4 29727900          University of Kentucky  BOS  SF
## 5:  Blake Griffin      5 29512900            Marquette University  DET  PF
## 6:     Kyle Lowry      6 28703704            University of Kansas  TOR  PG
##    age games point ThrP ThrPA   eFG FreeThrow rebound assist   InOut
## 1:  29    51  28.2  4.2   9.8 0.618     0.921     5.1    6.1 Outside
## 2:  33    82  28.6  1.8   5.0 0.590     0.731     8.6    9.1  Inside
## 3:  32    38  17.0  1.0   3.0 0.509     0.696     6.4    2.8  Inside
## 4:  27     1   7.3  0.0   1.0 0.500     0.000     1.0    0.0 Outside
## 5:  28    58  19.6  1.9   5.6 0.493     0.785     7.4    5.8  Inside
## 6:  31    78  19.5  3.1   7.6 0.553     0.854     5.6    6.9 Outside
```

2. The percentage of number of players in every position

To study the relationship between position and on-court stats (point, rebound and assist), we first want to know the percentage of players in every position. Because different positions are good at different things. For examplem, Centers generally do better in rebound while Point Guards do better in assist.

```
get_position_percent <- function(position)
{
  nba_pos <- nba[nba$pos== position, ]

  num_pos <- nrow(nba_pos)
  num_nba <- nrow(nba)
  fraction_pos <- num_pos/num_nba
  return(fraction_pos)
}

get_position_percent("PG")
```

```
## [1] 0.185941
```

```
get_position_percent("SG")
```

```
## [1] 0.2154195
```

```
get_position_percent("PF")
```

```
## [1] 0.2040816
```

```
get_position_percent("SF")
```

```
## [1] 0.1609977
```

```
get_position_percent("C")
```

```
## [1] 0.2267574
```

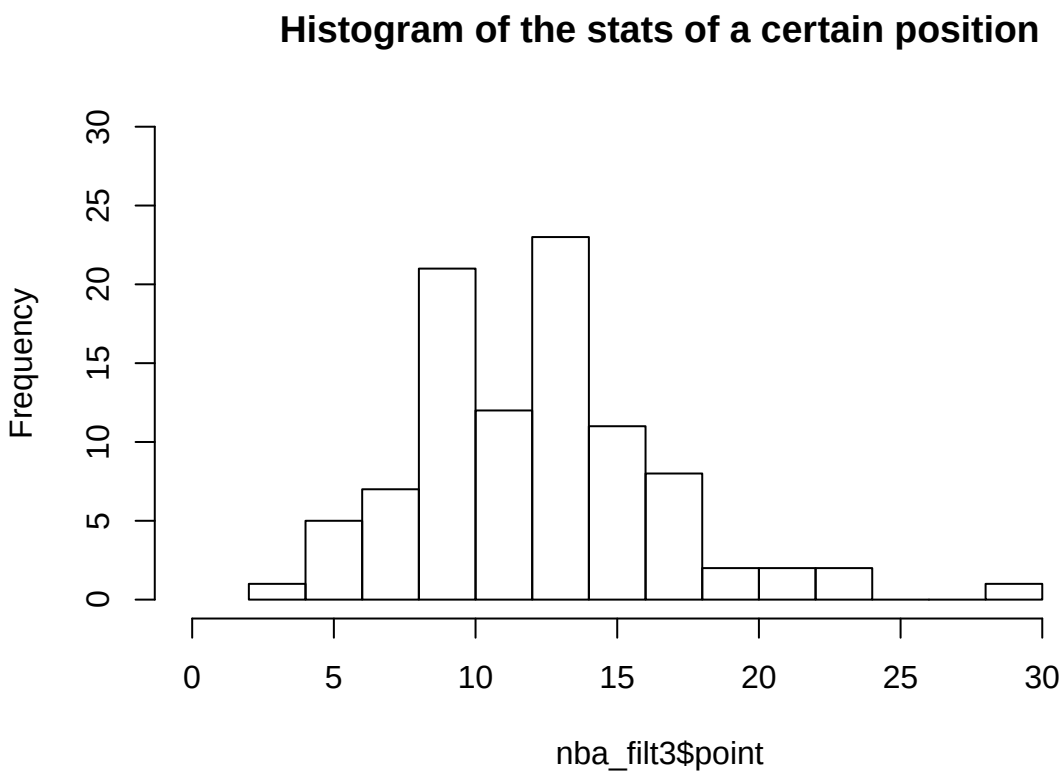
3. Get the On-court stats histogram of differnt positions

As players in different positions are good at different things, we want to study the relationship between position and points/ rebounds/ assists. To show the results clearly, I use histogram to display the frequency of different levels of values of a position.

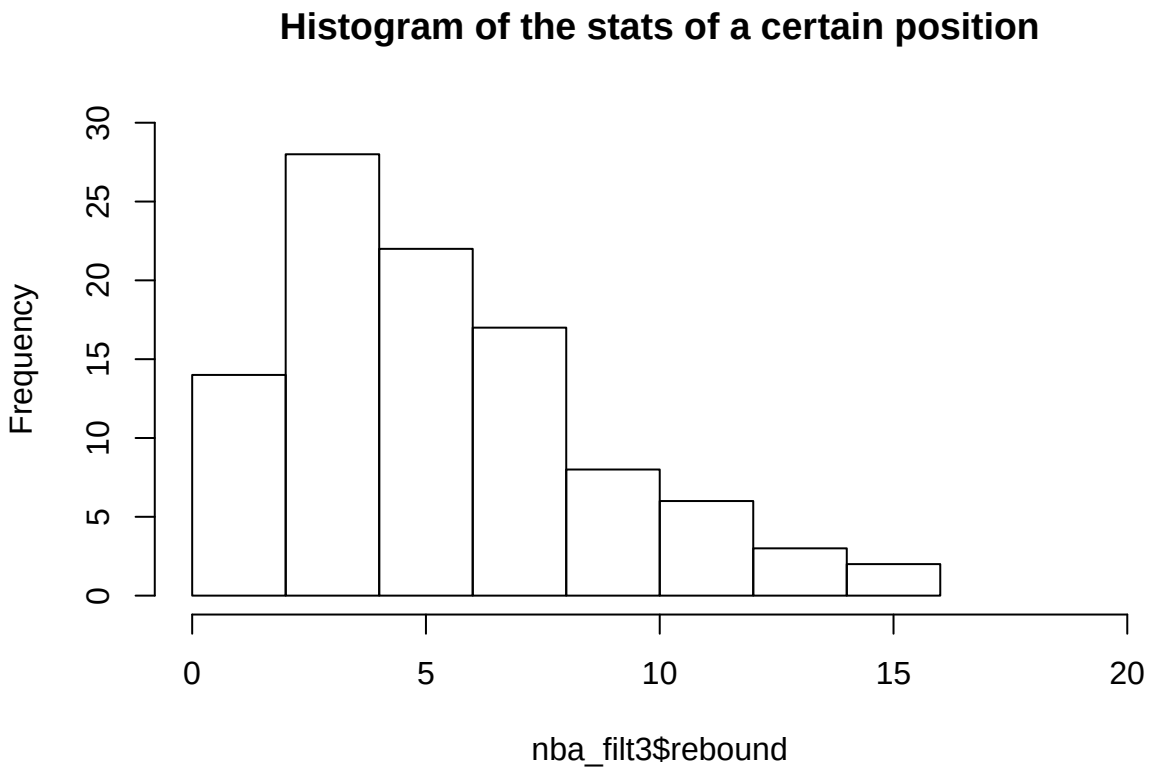
```
get_point <- function(position)
{
  nba_filt3 <- nba[nba$pos == position,]

  hist(nba_filt3$point,xlim = c(0,33),breaks=10, ylim = c(0,30),
       main = "Histogram of the stats of a certain position")
}

get_rebound <- function(position)
{
  nba_filt3 <- nba[nba$pos == position,]

  hist(nba_filt3$rebound,xlim = c(0,20),breaks=10, ylim = c(0,30),
       main = "Histogram of the stats of a certain position")
}

get_assist <- function(position)
{
  nba_filt3 <- nba[nba$pos == position,]

  hist(nba_filt3$assist,xlim = c(0,13),breaks=10, ylim = c(0,20),
       main = "Histogram of the stats of a certain position")
}
```

Some samples

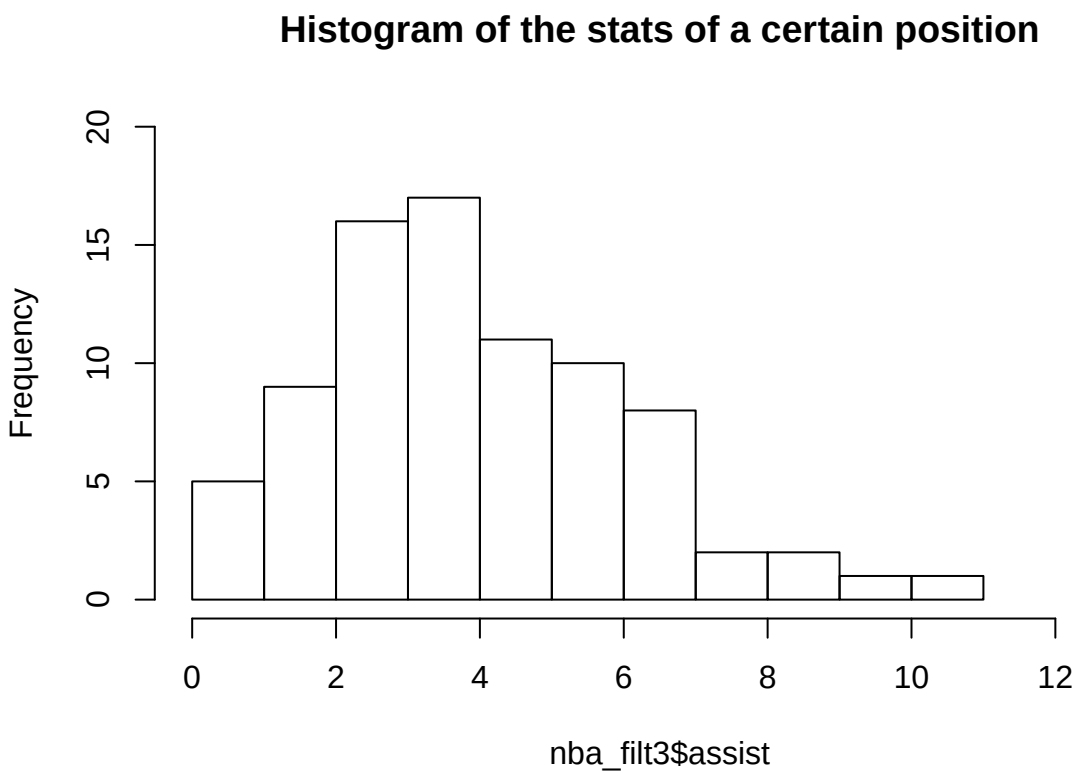
```
get_point("SG")
```




```
get_rebound("C")
```



```
get_assist("PG")
```



4.SALARY

Besides on-court stats and position, salary is another way of measuring the valur of a player. Below I will study the pay of players from all different aspects, and try to get a relationship between salary and influencing factors.

Clear dataset further

Sometimes there are unusual situations happening. Like players may get injured, so their attendance and stats would be drastically influenced. To get an unbiased judgement, we clear out players who play less than 20 games during this year. Besides, for convenience, we convert "PG-SG" and "SF-SG" into "SG".

```
clear_players <- function(nba)
{
  NBA <- nba[nba$games >= 20,]
  number_row <- nrow(NBA)
  for (i in 1:number_row) {
    if(NBA$pos[i]=="PG-SG"|NBA$pos[i]=="SF-SG"){
      NBA$pos[i] <- "SG"
    }

  }
  return(NBA)
}
```

```
NBA <- clear_players(nba)
head(NBA)
```

```
##               Player PayRnk      pay                         college team
## 1:     Stephen Curry      1 34682550          University of Kentucky  GSW
## 2:      LeBron James      2 33285709        University of Notre Dame  CLE
## 3:      Paul Millsap      3 31269231 North Carolina State University  DEN
## 4:     Blake Griffin      5 29512900            Marquette University  DET
## 5:        Kyle Lowry      6 28703704            University of Kansas  TOR
## 6: Russell Westbrook      7 28530608               Temple University  OKC
##    pos age games point ThrP ThrPA   eFG FreeThrow rebound assist   InOut
## 1:  PG  29    51  28.2  4.2   9.8 0.618     0.921     5.1    6.1 Outside
## 2:  PF  33    82  28.6  1.8   5.0 0.590     0.731     8.6    9.1  Inside
## 3:  PF  32    38  17.0  1.0   3.0 0.509     0.696     6.4    2.8  Inside
## 4:  PF  28    58  19.6  1.9   5.6 0.493     0.785     7.4    5.8  Inside
## 5:  PG  31    78  19.5  3.1   7.6 0.553     0.854     5.6    6.9 Outside
## 6:  PG  29    80  24.7  1.2   4.1 0.477     0.737    10.1   10.3 Outside
```

As we know, age affects the athletic levle of a player, thus affecting his salary level.

Below I will create a data.frame with columns age, players, topSalary. The age column should contain all the unique age in NBA. The players column should, for a given age, contain the number of players in that age. The topSalary column should, for a given age, contain the highest salary of that age. And write the data.frame into a csv file.

```
get_salary_age <- function(NBA)
{
  NBA_filt4 <- NBA[, c("age","Player","pay")]
  NBA_filt4_age <- sort(unique(NBA_filt4$age))
  age_number <- length(NBA_filt4_age)
  NBA_filt4_players <- rep(NA, age_number)
  NBA_filt4_topSalary <- rep(NA, age_number)

  for (i in 1:age_number) {
    NBA_c <- NBA[NBA_filt4$age == NBA_filt4_age[i],]
    NBA_filt4_players[i] <- length(NBA_c$Player)
    NBA_filt4_topSalary[i] <- NBA_c$pay[1]

  }

  salary_age <- data.frame(age=NBA_filt4_age,
                           players=NBA_filt4_players,
                           topSalary=NBA_filt4_topSalary,
                           stringsAsFactors = F)
  write.csv(salary_age, "salary_age.csv")
  return(salary_age)
}

salary_age <- get_salary_age(NBA)
salary_age
```

```
##    age players topSalary
## 1   19       5   5645400
## 2   20      20   6286560
## 3   21      24   6168840
```

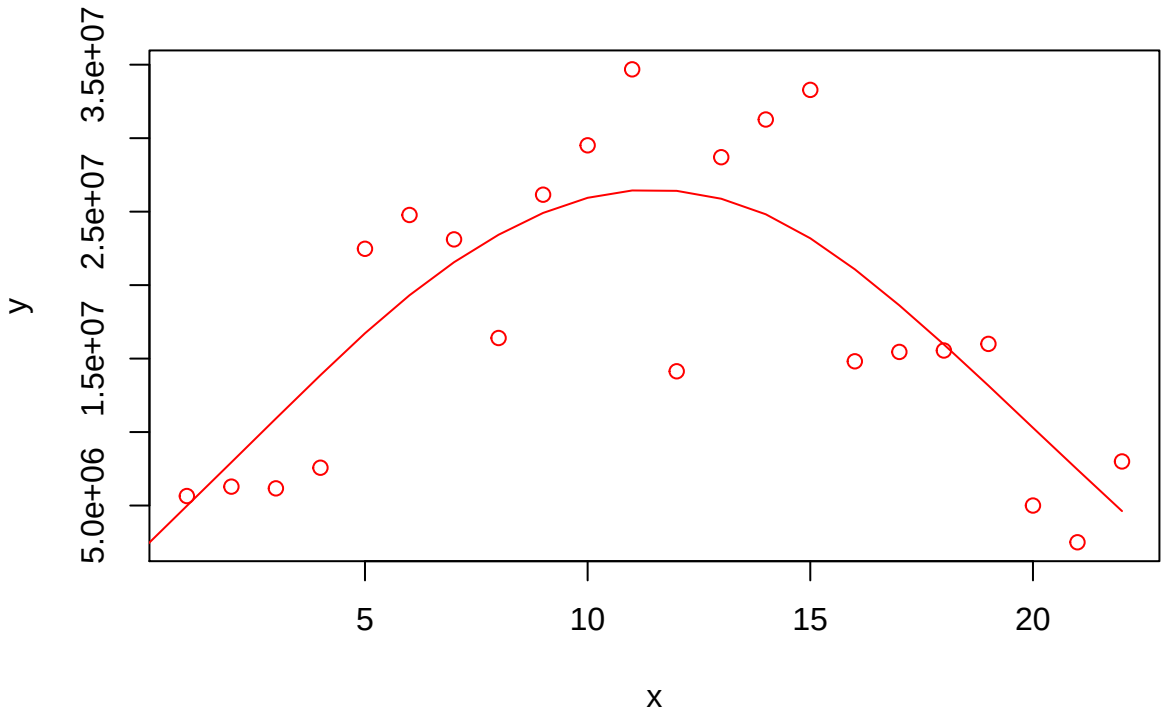
```
## 4   22      29   7574322
## 5   23      39  22471910
## 6   24      35  24773250
## 7   25      36  23112004
## 8   26      29  16400000
## 9   27      29  26153057
## 10  28      27  29512900
## 11  29      30  34682550
## 12  30      20  14136364
## 13  31      22  28703704
## 14  32      15  31269231
## 15  33      13  33285709
## 16  34       4  14814815
## 17  35       4  15453126
## 18  36       6  15550000
## 19  37       4  16000000
## 20  39       1   5000000
## 21  40       2   2500000
## 22  41       1   8000000
```

Use Graph to show the relationship between topSalary and age

```
plot_graph <- function(base_name)
{
  file_name <- paste(base_name, ".csv", sep="")
  salary_a <- read.csv(file_name, header=T,
                  stringsAsFactors = F)
  plot(salary_a$age, salary_a$topSalary, xlim = c(19, 41), ylim = c(2300000,35000000),
       xlab = "Age", ylab = "Top Salary Gained", main = "Age Salary Plot",
       col = "gold", pch = 16)
  lines(salary_a$age, salary_a$topSalary, col = "goldenrod1")
  legend("bottomright", legend = c("TopSalry"),
         col = c("gold"), pch = 16, bty = "n")
}
plot_graph("salary_age")
```

Try to fit a regression line

```
regression_graph <- function(salary_age)
{
  x <- 1:length(salary_age$age)
  y <- salary_age$topSalary
  plot(x,y, col="red")
  s <- stats::smooth.spline(x, y, nknots=15)
  many_x <- seq(0,length(salary_age$age),1)
  y_smooth <- predict(s, many_x)
  lines(y_smooth$x, y_smooth$y, col="red")
}
regression_graph(salary_age)
```



From these two graphs we can conclude that an age around 29 is the best for a player, and players under this age is likely to make a big amount of money in a season.

Get age with most players

We then naturally think that ages around 29 should have the largest number of players as well. We will test this thought below. We use a function to find the age with most players.

```
Age_mostPlayers <- function(salary_age)
{
  most_age <- which.max(salary_age$players)
  age1 <- salary_age$age[most_age]
  return(age1)
}
Age_mostPlayers(salary_age)
```

```
## [1] 23
```

The age with the most players is 23. It is different from the age of 29 as we have guessed. This inconsistency may be caused by the fact that several ages share similar numbers of players.

Average salary in every age

The mismatch between the age with the greatest top salary and the age with the most players makes me to reconsider my index selection. Perhaps it would be better to use average salary instead of top salary.

```
get_meanSalary <- function(NBA)
{
  NBA_filt_age <- sort(unique(NBA$age))
  h_average <- rep(NA, length(NBA_filt_age))
```

```
  for (i in 1:length(NBA_filt_age)) {
    NBA_agefilted <- dplyr::filter(NBA, age == NBA_filt_age[i])
    sum_pay <- sum(NBA_agefilted$pay)
    h_average[i] <- sum_pay/length(NBA_agefilted$pay)

  }

  Mean_Salary <- data.frame(age=NBA_filt_age,
                            average_salary=h_average,
                            stringsAsFactors = F)
  return(Mean_Salary)

}
Mean_Salary <- get_meanSalary(NBA)
```

```
## Warning: package 'bindrcpp' was built under R version 3.4.4
```

```
Mean_Salary
```

```
##    age average_salary
## 1   19        3109944
## 2   20        3125637
## 3   21        2402792
## 4   22        2705363
## 5   23        2525579
## 6   24        6742756
## 7   25        8566440
## 8   26        6083625
## 9   27       11226015
## 10  28       10886663
## 11  29       13253635
## 12  30        7088683
## 13  31       10523541
## 14  32       12319919
## 15  33       11434947
## 16  34        7241731
## 17  35        9166694
## 18  36        8134716
## 19  37        7942854
## 20  39        5000000
## 21  40        2414326
## 22  41        8000000
```

Then we use a function to find the age with greatest average salary

```
Age_mostMean <- function(Mean_Salary)
{
  most_salary <- which.max(Mean_Salary$average_salary)
  age2 <- Mean_Salary$age[most_salary]
  return(age2)
}
Age_mostMean(Mean_Salary)
```

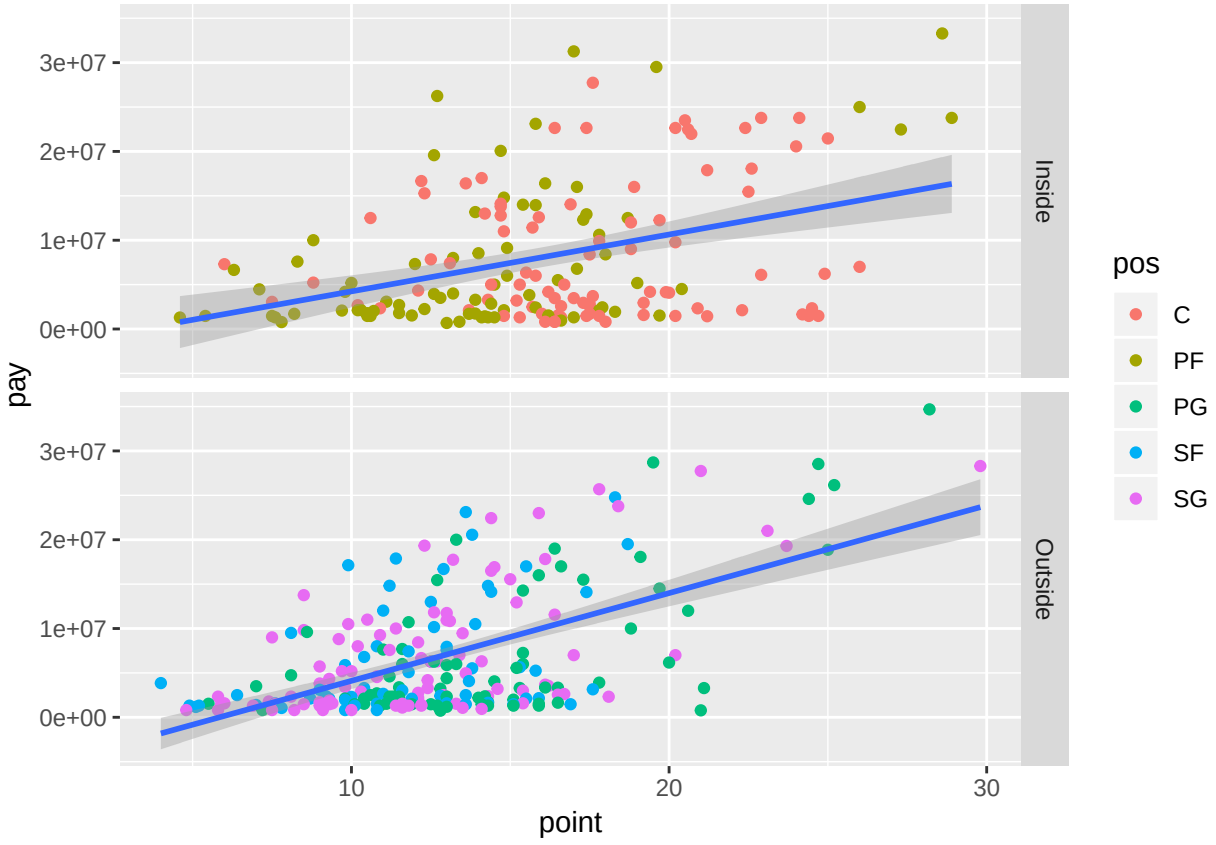
```
## [1] 29
```

Age 29 has the largest average salary. This supports our precious conclusion that players of ages around 29 can make the most money.

5. Relationship between salary and on-court stats

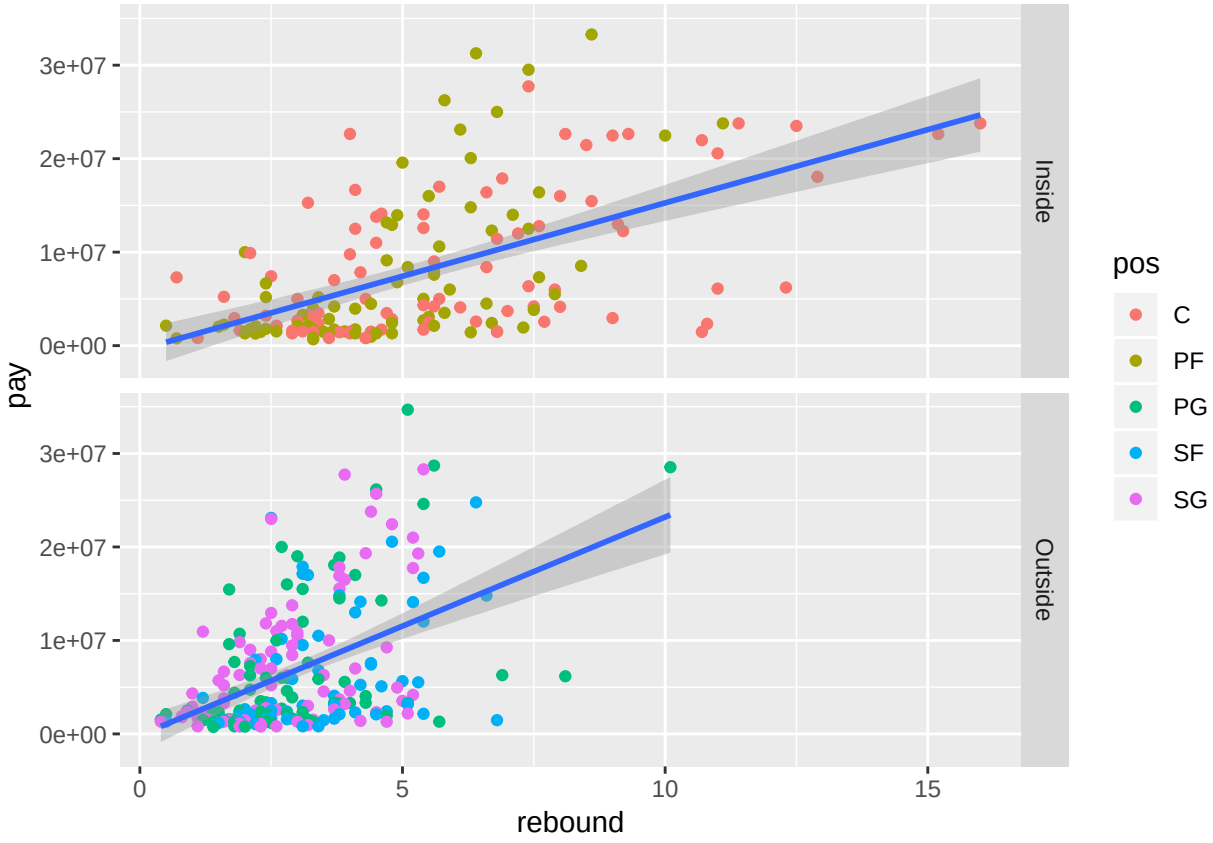
Then we want to study the relationship between salary and stats (including Ponits/ Rebounds/ Assists) ##A graph of the relationship between salary and points. To compare, we separate inside players and outside players.

```
point_salary_graph <- function(NBA)
{
  p <- ggplot()
  p <- p + geom_point(mapping=aes(x=point, y=pay,
                                  color=pos),
                      data=NBA)
  p + geom_smooth(mapping=aes(x=point, y=pay,),
                  data=NBA, method="lm")+ facet_grid(InOut ~ .)

}
point_salary_graph(NBA)
```



A graph of the relationship between salary and rebounds.

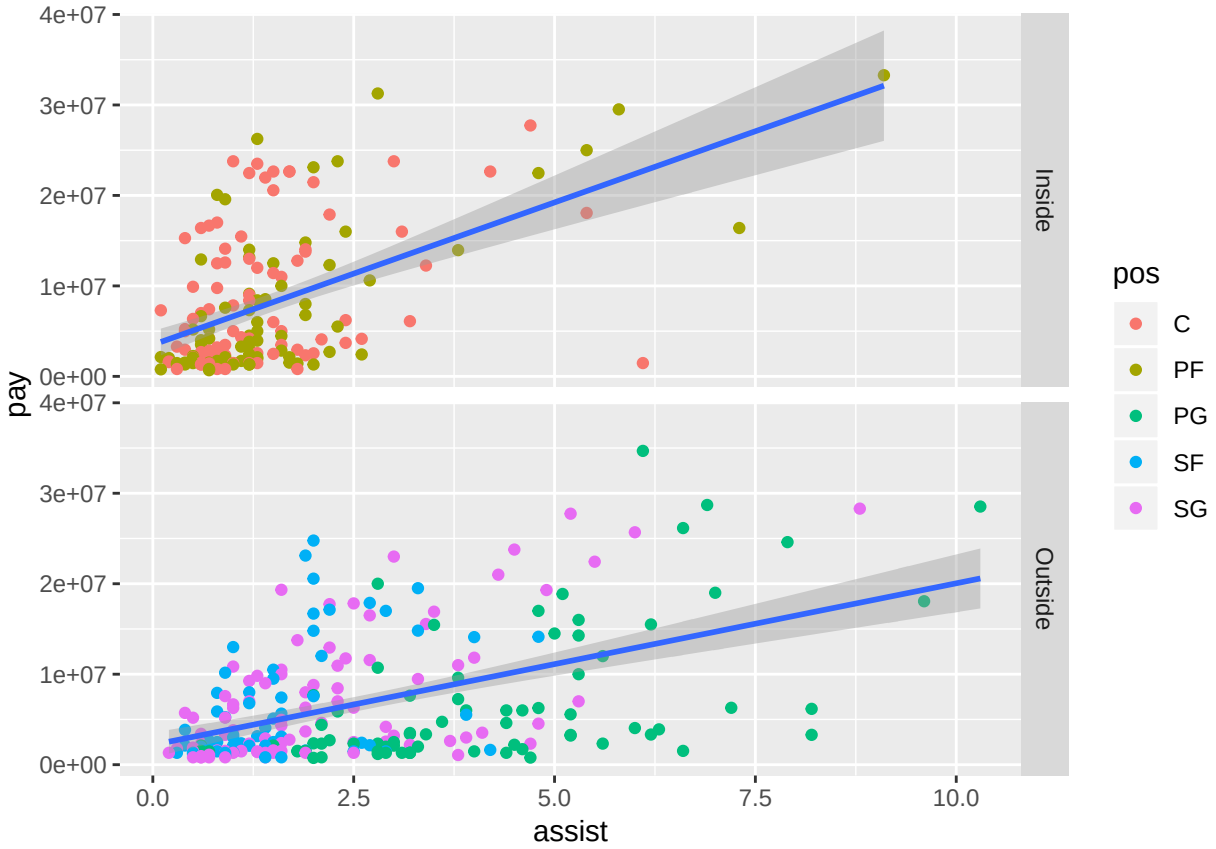
```
rebound_salary_graph <- function(NBA)
{
  p <- ggplot()
  p <- p + geom_point(mapping=aes(x=rebound, y=pay,
                                  color=pos),
                      data=NBA)
  p + geom_smooth(mapping=aes(x=rebound, y=pay,),
                  data=NBA, method="lm")+ facet_grid(InOut ~ .)

}
rebound_salary_graph(NBA)
```



A graph of the relationship between salary and assists.

```
assist_salary_graph <- function(NBA)
{
  p <- ggplot()
  p <- p + geom_point(mapping=aes(x=assist, y=pay,
                                  color=pos),
                      data=NBA)
  p + geom_smooth(mapping=aes(x=assist, y=pay,),
                  data=NBA, method="lm")+ facet_grid(InOut ~.)
```

```
}
assist_salary_graph(NBA)
```



From these graphs we can see that players with higher points, rebounds and assists will get a better salary, which is consistent with our common sense.

6. Colleges and teams

There are some other interesting factors worth studying

As we know, NBA players come from different colleges, and some of the colleges are famous of basketball, like Duke, Georgetown and so on. To find out the influence of college on NBA, I try to give a function, with which, you input a NBA team name and a college name, and get the names of the players in both that team and that collage.

```
get_college_team <- function(NBA, College, Team)
{
  NBA_filt6 <- dplyr::filter(NBA, college == College & team == Team)
  NBA_filted6 <- NBA_filt6$Player
  return (NBA_filted6)
}
#Samples
get_college_team(NBA, "Georgetown University", "CHO")
```

```
## [1] "Miles Plumlee"
```

```
get_college_team(NBA, "Temple University", "OKC")
```

```
## [1] "Russell Westbrook"
```

```
get_college_team(NBA, "Yale University", "MIL")
```

```
## [1] "Amir Johnson"
```

7. Best offensive team

From above we can see that players with greater average point are better. So we can generate the idea of "Best offensive team" based on the thought that a team with a higher average point generally do better in offensive side.

Below I offer a function that gives the total score of a team inputted. Round the number into a integer.

```
get_total_point <- function(NBA, Team)
{
  NBA_filt7 <- dplyr::filter(NBA, team == Team)
  t <- NBA_filt7$point
  leng <- length(t)
  tp <- sapply(t, function(t)
  {
    if((t-floor(t)) >= 0.5){
      tp <- (floor(t)+1)
    }else{
      tp <- floor(t)
    }
    return(tp)
  }
  )
  total <- sum(tp)
  return(total)
}
#Samples
get_total_point(NBA, "CHI")
```

```
## [1] 214
```

```
get_total_point(NBA, "OKC")
```

```
## [1] 189
```

To get the "Best offensive team"

```
get_best_offensive <- function(NBA)
{
  teams <- sort(unique(NBA$team))
  team_num <- length(teams)
  total_score <- rep(NA, team_num)
  for (i in 1:team_num) {
    team_score <- get_total_point(NBA, teams[i])
    total_score[i] <- team_score
  }
  score_team <- data.frame(team=teams, score=total_score,
                           stringsAsFactors = F)
```

```
  most_score <- which.max(score_team$score)
  team_good <- score_team$team[most_score]
  return(team_good)
}
get_best_offensive(NBA)
```

```
## [1] "BRK"
```

So we can see that team "BRK" did best in offensive side during the last season.

Conclusion:

There are multiple ways to define a good player, including salary, points, rebounds, assists and so on. When analyzing this, we may also take other factors like colleges and positions into consideration. Only with diverse methods can we get an unbiased result.