# *Building a team for Chicago Bulls*

**Report made by - Kanishka Goyal - u3219632**

## A. Introduction

### Background

Basketball is a popular sport worldwide, with the NBA being one of the most popular professional sports leagues in the United States. NBA teams consist of five players assigned to specific positions - point guard, shooting guard, small forward, power forward, or center. Each position has unique roles and responsibilities, with point guards typically responsible for ball-handling and passing, centers being the tallest and most dominant players on the court, and each player's performance measured by key metrics such as points, rebounds, assists, steals, and blocks.

### Scenario

In this scenario, the Chicago Bulls NBA team finished 27th out of 30 teams in the previous season, and their budget for player contracts for the upcoming season is $118 million, which ranks 26th out of 30 teams. This project aims to identify the best five starting players for the team, while ensuring that they remain within the budget.This project aims to improve the team's overall performance by selecting the best players for the starting lineup, while also providing valuable insights to the management for future team selections.

### Aim

The aim of this project is to analyze NBA player statistics and salaries to identify the best five starting players for the Chicago Bulls while staying within budget constraints.

### Importance (Justification)

This project can provide valuable insights into player performance and the overall competitive landscape of the NBA. By conducting a thorough analysis of player statistics and salaries, this project can identify key trends and patterns in player performance that can be used to inform future team selections and strategies. This can provide the team with a competitive advantage, as they will have access to data-driven insights that can inform their decision-making processes.

### Budgeting

Chicago Bulls Budget for 2019-2020 Season as per requirements should be roughly divided like this :-   Total Budget: $118 million   5 main Player Salaries: $64 million (based on 5 starting players and 7 additional players)   7 additional Player Salaries: $30 million (7 additional players)   Coaching Staff: $6 million (head coach and 3 assistant coaches)   Medical Staff: $4 million (team physician, athletic trainer, and other medical personnel)   Travel Expenses: $10 million (transportation, lodging, and meals for the team and staff)   Arena Expenses: $4 million (rental fees, maintenance, and utilities for the team's home arena)   Note: This budget assumes that there are no additional unexpected expenses.

Based on this budget, we can allocate $90 million towards player salaries.

## B. Reading & cleaning raw data

### Data preprocessing

Data pre-processing is a very important aspect of a data science project and accounts for almost half of the work in many cases. It helps to manage missing data, detect outliers, change variable names etc. In data pre-processing the data is cleaned in a series of steps to make it easily and readily available for later stages of the project.

### Data Loading

```
# Loading data sets into working directory--------------------------------------

player_salaries <- read.csv("data/raw/2018-19_nba_player-salaries.csv")
player_statistics <- read.csv("data/raw/2018-19_nba_player-statistics.csv")
team_statistics_1 <- read.csv("data/raw/2018-19_nba_team-statistics_1.csv")
team_statistics_2 <- read.csv("data/raw/2018-19_nba_team-statistics_2.csv")
team_payroll <- read.csv("data/raw/2019-20_nba_team-payroll.csv")
```

### Data inspection

```
head(player_salaries)
```

```
##   player_id  player_name   salary
## 1         1 Alex Abrines  3667645
## 2         2   Quincy Acy   213948
## 3         3 Steven Adams 24157304
## 4         4 Jaylen Adams   236854
## 5         5  Bam Adebayo  2955840
## 6         6    Deng Adel    77250
```

```
head(player_statistics)
```

```
##    player_name Pos Age  Tm  G GS   MP  FG FGA   FG. X3P X3PA  X3P. X2P X2PA
## 1 Alex Abrines  SG  25 OKC 31  2  588  56 157 0.357  41  127 0.323  15   30
## 2   Quincy Acy  PF  28 PHO 10  0  123   4  18 0.222   2   15 0.133   2    3
## 3 Jaylen Adams  PG  22 ATL 34  1  428  38 110 0.345  25   74 0.338  13   36
## 4 Steven Adams   C  25 OKC 80 80 2669 481 809 0.595   0    2 0.000 481  807
## 5  Bam Adebayo   C  21 MIA 82 28 1913 280 486 0.576   3   15 0.200 277  471
## 6    Deng Adel  SF  21 CLE 19  3  194  11  36 0.306   6   23 0.261   5   13
##    X2P.  eFG.  FT FTA   FT. ORB DRB TRB AST STL BLK TOV  PF  PTS
## 1 0.500 0.487  12  13 0.923   5  43  48  20  17   6  14  53  165
## 2 0.667 0.278   7  10 0.700   3  22  25   8   1   4   4  24   17
## 3 0.361 0.459   7   9 0.778  11  49  60  65  14   5  28  45  108
## 4 0.596 0.595 146 292 0.500 391 369 760 124 117  76 135 204 1108
## 5 0.588 0.579 166 226 0.735 165 432 597 184  71  65 121 203  729
## 6 0.385 0.389   4   4 1.000   3  16  19   5   1   4   6  13   32
```

```
colnames(player_statistics)
```

```
##  [1] "player_name" "Pos"         "Age"         "Tm"          "G"
##  [6] "GS"          "MP"          "FG"          "FGA"         "FG."
## [11] "X3P"         "X3PA"        "X3P."        "X2P"         "X2PA"
## [16] "X2P."        "eFG."        "FT"          "FTA"         "FT."
## [21] "ORB"         "DRB"         "TRB"         "AST"         "STL"
## [26] "BLK"         "TOV"         "PF"          "PTS"
```

```
colnames(team_payroll)
```

```
## [1] "team_id" "team"    "salary"
```

```
str(player_salaries)
```

```
## 'data.frame':    576 obs. of  3 variables:
##  $ player_id  : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ player_name: chr  "Alex Abrines" "Quincy Acy" "Steven Adams" "Jaylen Adams" ...
##  $ salary     : int  3667645 213948 24157304 236854 2955840 77250 5285394 77250 2000000 22347015 ...
```

```
str(player_statistics)
```

```
## 'data.frame':    708 obs. of  29 variables:
##  $ player_name: chr  "Alex Abrines" "Quincy Acy" "Jaylen Adams" "Steven Adams" ...
##  $ Pos        : chr  "SG" "PF" "PG" "C" ...
##  $ Age        : int  25 28 22 25 21 21 25 33 21 23 ...
##  $ Tm         : chr  "OKC" "PHO" "ATL" "OKC" ...
##  $ G          : int  31 10 34 80 82 19 7 81 10 38 ...
##  $ GS         : int  2 0 1 80 28 3 0 81 1 2 ...
##  $ MP         : int  588 123 428 2669 1913 194 22 2687 120 416 ...
##  $ FG         : int  56 4 38 481 280 11 3 684 13 67 ...
##  $ FGA        : int  157 18 110 809 486 36 10 1319 39 178 ...
##  $ FG.        : num  0.357 0.222 0.345 0.595 0.576 0.306 0.3 0.519 0.333 0.376 ...
##  $ X3P        : int  41 2 25 0 3 6 0 10 3 32 ...
##  $ X3PA       : int  127 15 74 2 15 23 4 42 12 99 ...
##  $ X3P.       : num  0.323 0.133 0.338 0 0.2 0.261 0 0.238 0.25 0.323 ...
##  $ X2P        : int  15 2 13 481 277 5 3 674 10 35 ...
##  $ X2PA       : int  30 3 36 807 471 13 6 1277 27 79 ...
##  $ X2P.       : num  0.5 0.667 0.361 0.596 0.588 0.385 0.5 0.528 0.37 0.443 ...
##  $ eFG.       : num  0.487 0.278 0.459 0.595 0.579 0.389 0.3 0.522 0.372 0.466 ...
##  $ FT         : int  12 7 7 146 166 4 1 349 8 45 ...
##  $ FTA        : int  13 10 9 292 226 4 2 412 12 60 ...
##  $ FT.        : num  0.923 0.7 0.778 0.5 0.735 1 0.5 0.847 0.667 0.75 ...
##  $ ORB        : int  5 3 11 391 165 3 1 251 11 3 ...
##  $ DRB        : int  43 22 49 369 432 16 3 493 15 20 ...
##  $ TRB        : int  48 25 60 760 597 19 4 744 26 23 ...
##  $ AST        : int  20 8 65 124 184 5 6 194 13 25 ...
##  $ STL        : int  17 1 14 117 71 1 2 43 1 6 ...
##  $ BLK        : int  6 4 5 76 65 4 0 107 0 6 ...
##  $ TOV        : int  14 4 28 135 121 6 2 144 8 33 ...
##  $ PF         : int  53 24 45 204 203 13 4 179 7 47 ...
##  $ PTS        : int  165 17 108 1108 729 32 7 1727 37 211 ...
```

```
str(team_statistics_1)
```

```
## 'data.frame':    30 obs. of  25 variables:
##  $ Rk    : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Team  : chr  "Milwaukee Bucks" "Golden State Warriors" "Toronto Raptors" "Utah Jazz" ...
##  $ Age   : num  26.9 28.4 27.3 27.3 29.2 26.2 24.9 25.7 25.7 27 ...
##  $ W     : int  60 57 58 50 53 53 54 49 49 48 ...
##  $ L     : int  22 25 24 32 29 29 28 33 33 34 ...
##  $ PW    : int  61 56 56 54 53 51 51 52 50 50 ...
##  $ PL    : int  21 26 26 28 29 31 31 30 32 32 ...
##  $ MOV   : num  8.87 6.46 6.09 5.26 4.77 4.2 3.95 4.44 3.4 3.33 ...
##  $ SOS   : num  -0.82 -0.04 -0.6 0.03 0.19 0.24 0.24 -0.54 0.15 -0.57 ...
##  $ SRS   : num  8.04 6.42 5.49 5.28 4.96 4.43 4.19 3.9 3.56 2.76 ...
##  $ ORtg  : num  114 116 113 111 116 ...
##  $ DRtg  : num  105 110 107 106 111 ...
##  $ NRtg  : num  8.6 6.4 6 5.2 4.8 4.2 4.1 4.4 3.3 3.4 ...
##  $ Pace  : num  103.3 100.9 100.2 100.3 97.9 ...
##  $ FTr   : num  0.255 0.227 0.247 0.295 0.279 0.258 0.232 0.215 0.266 0.242 ...
##  $ X3PAr : num  0.419 0.384 0.379 0.394 0.519 0.339 0.348 0.381 0.347 0.292 ...
##  $ TS.   : num  0.583 0.596 0.579 0.572 0.581 0.568 0.558 0.567 0.545 0.561 ...
##  $ eFG.  : num  0.55 0.565 0.543 0.538 0.542 0.528 0.527 0.534 0.514 0.53 ...
##  $ TOV.  : num  12 12.6 12.4 13.4 12 12.1 11.9 11.5 11.7 12.4 ...
##  $ ORB.  : num  20.8 22.5 21.9 22.9 22.8 26.6 26.6 21.6 26 21.9 ...
##  $ FT.FGA: num  0.197 0.182 0.198 0.217 0.221 0.21 0.175 0.173 0.19 0.182 ...
##  $ DRB.  : num  80.3 77.1 77.1 80.3 74.4 77.9 78 77 78.2 76.2 ...
##  $ X     : logi  NA NA NA NA NA NA ...
##  $ X.1   : logi  NA NA NA NA NA NA ...
##  $ X.2   : logi  NA NA NA NA NA NA ...
```

```
str(team_payroll)
```

```
## 'data.frame':    30 obs. of  3 variables:
##  $ team_id: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ team   : chr  "Miami " "Golden State " "Oklahoma City " "Toronto " ...
##  $ salary : chr  "$153,171,497 " "$146,291,276 " "$144,916,427 " "$137,793,831 " ...
```

### Setting my standard theme for graphs

```
my_theme <- theme(
  panel.background = element_rect( fill = "ivory2" , color ="gray"),
  plot.background = element_rect( color = "black"),
  plot.title = element_text(size = 14, face = "bold", hjust = 0.5),
  axis.title.x = element_text(size = 12, hjust = 0.5),
  axis.title.y = element_text(size = 12, hjust = 0.5))
```

## Data cleaning

### 1. Missing data management

In the data set, there are two datasets that have missing data. All these values need to be handled, either by imputing or by removing the missing values.

#### Number of missing values for all datasets

```
# Player Salaries --------------------------------------------------------------
colSums(is.na(player_salaries))
```

```
##   player_id player_name      salary
##           0           0           0
```

```
# zero missing values

# Team statistics 2 ------------------------------------------------------------
colSums(is.na(team_statistics_2))
```

```
##   Rk Team    G   MP   FG  FGA  FG.  X3P X3PA X3P.  X2P X2PA X2P.   FT  FTA  FT.
##    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
##  ORB  DRB  TRB  AST  STL  BLK  TOV   PF  PTS
##    0    0    0    0    0    0    0    0    0
```

```
# zero missing values

# Team statistics 2 ------------------------------------------------------------
colSums(is.na(team_payroll))
```

```
## team_id    team  salary
##       0       0       0
```

```
# Player Statistics ------------------------------------------------------------
colSums(is.na(player_statistics))
```

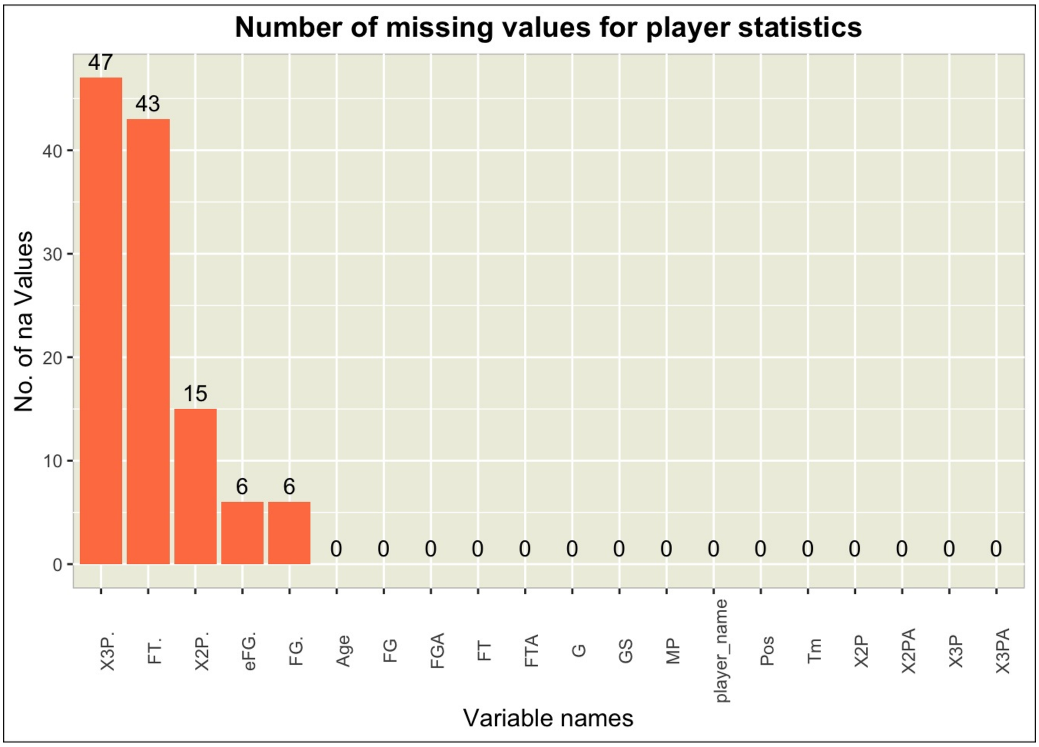
```
## player_name         Pos         Age          Tm           G          GS
##           0           0           0           0           0           0
##          MP          FG         FGA         FG.         X3P        X3PA
##           0           0           0           6           0         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

```
# Team statistics 1 ------------------------------------------------------------
colSums(is.na(team_statistics_1))
```

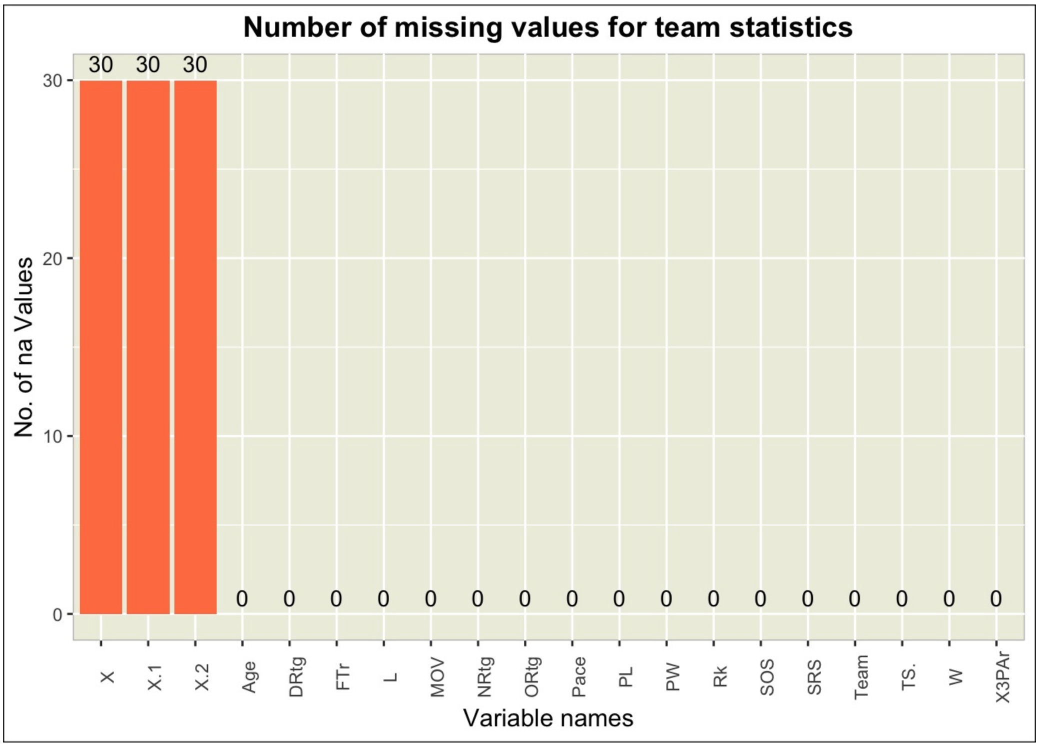
```
##     Rk   Team    Age      W      L     PW     PL    MOV    SOS    SRS   ORtg
##      0      0      0      0      0      0      0      0      0      0      0
##   DRtg   NRtg   Pace    FTr  X3PAr    TS.   eFG.   TOV.   ORB. FT.FGA   DRB.
##      0      0      0      0      0      0      0     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

```
# Player statistics has missing values
```

In the **plot** shown below, you can see the with maximum number of NA values for Player Statistics.




In the **plot** shown below, you can see the with maximum number of NA values for Team Statistics.




**a.** Handling missing values for player_statistics. The first step is to remove the variables that are 100% empty.

```
## [1] "All the empty columns removed."
```

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

**b.** Handling missing values for team_statistics. Removing rows for those columns where more less than 10% values are missing.

```
# Handling missing values for player statistics -------------------------------

prop_missing <- colMeans(is.na(player_statistics))
cols_to_keep <- names(prop_missing[prop_missing < 0.1])
player_statistics <- player_statistics[complete.cases(player_statistics[, cols_to_keep]),]

# check again if removed -------------------------------------------------------

colSums(is.na(player_statistics))
```

```
## player_name         Pos         Age          Tm           G          GS
##           0           0           0           0           0           0
##          MP          FG         FGA         FG.         X3P        X3PA
##           0           0           0           0           0           0
##        X3P.         X2P        X2PA        X2P.        eFG.          FT
##           0           0           0           0           0           0
##         FTA         FT.         ORB         DRB         TRB         AST
##           0           0           0           0           0           0
##         STL         BLK         TOV          PF         PTS
##           0           0           0           0           0
```

## 2. Handling errors and duplicate values -

In the player statistics, there are duplicate rows of players. Those players need to be sorted and only the players with the maximum cumulative value for games played variable will be selected. As during a season, player could change teams, they have created this list in that way which considers matches for each team they played.

**a.** Renaming column names for player statistics

```
colnames(player_statistics) <- c("player_name",
                                 "position",
                                 "age",
                                 "team",
                                 "games",
                                 "games_started",
                                 "minutes_played",
                                 "field_goals",
                                 "fg_attempts",
                                 "fg_percentage",
                                 "three_pointers",
                                 "threep_attempts",
                                 "threep_percentage",
                                 "two_pointers",
                                 "twop_attempts",
                                 "twop_percentage",
                                 "effective_fg",
                                 "free_throws",
                                 "ft_attempts",
                                 "ft_percentage",
                                 "offense_rebounds",
                                 "defence_rebounds",
                                 "total_rebounds",
                                 "assists",
                                 "steals",
                                 "blocks",
                                 "turnovers",
                                 "fouls",
                                 "points" )
```

**b.** Renaming column names for team statistics

```
colnames(team_statistics_1) <- c("rank",
"age",
"wins",
"losses",
"pythagorean_wins",
"pythagorean_loss",
"victory_margin",
"schedule_strength",
"simple_rating",
"offensive_rating",
"defensive_rating",
"net_rating",
"pace_factor",
"free_throw_attempt",
"threep_attempt",
"true_shoot",
"effective_fg",
"turnover_percentage",
"offensive_per",
"ft_per_fg",
"defensive_per"
)
```

**c.** Handling duplicates for player statistics

```
dim(player_statistics)
```

```
## [1] 629  29
```

```
player_statistics <- player_statistics %>%
  group_by(player_name) %>%
  filter(`games` == max(`games`)) %>%
  ungroup()

dim(player_statistics)
```

```
## [1] 475  29
```

**d.** # Download clean files back into separate data folder to be used further

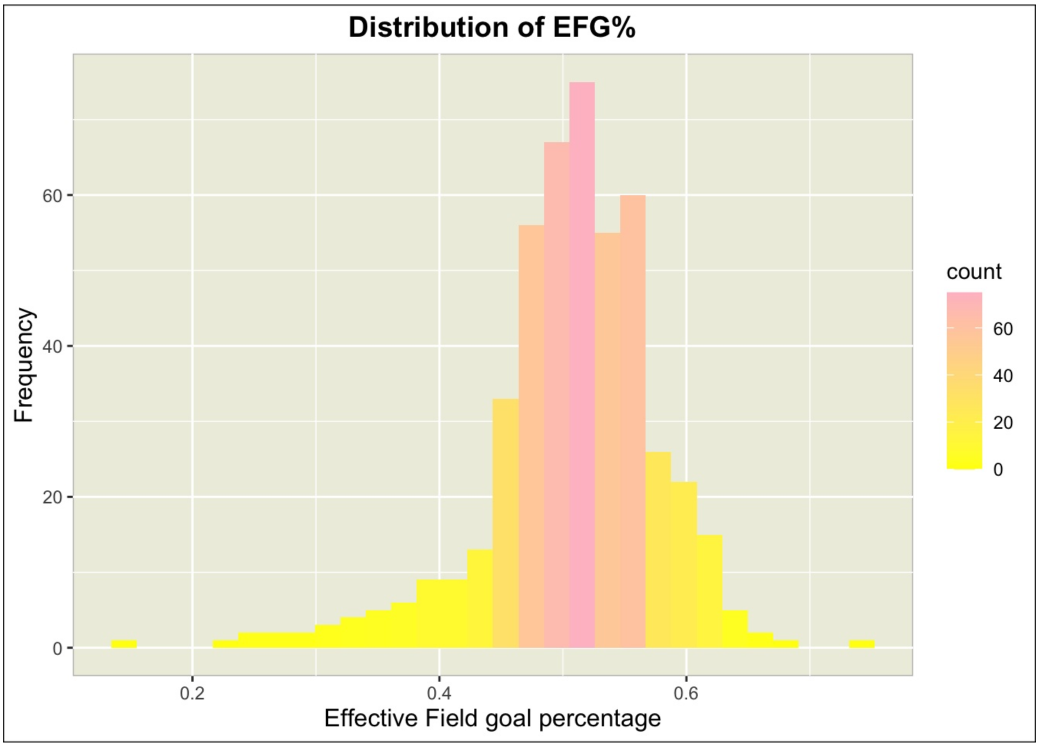
```
write.csv(player_salaries, "data/processed/player_salaries.csv")
write.csv(player_statistics, "data/processed/player_statistics.csv")
write.csv(team_payroll, "data/processed/team_payroll.csv")
write.csv(team_statistics_1, "data/processed/team_statistics_1.csv")
write.csv(team_statistics_2, "data/processed/team_statistics_2.csv")
```

# C. Exploratory data analysis

EDA provides data visualization methodologies which help to understand and summarize a data set without prior assumptions. It is crucial for gaining insight into data. Analysis Problems identified at the start of the project will be discussed here-

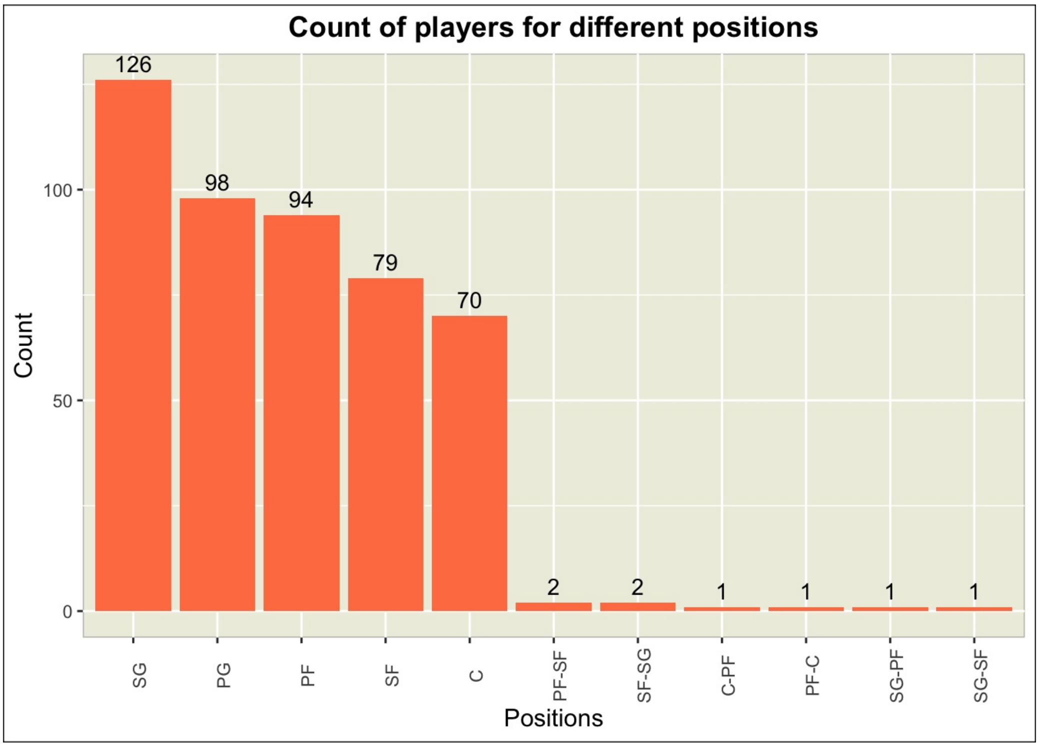
### Example 1 - Distribution of field goal attempts

#### Field goal attempts



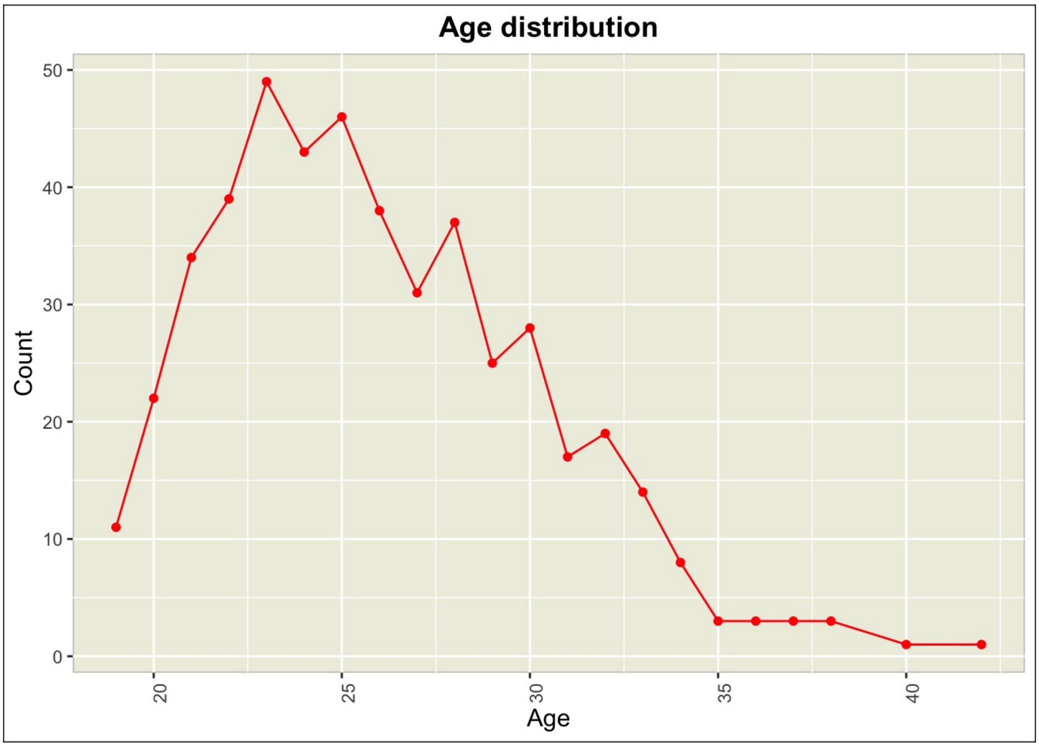

### Count of players for different positions

#### Player’s position on field

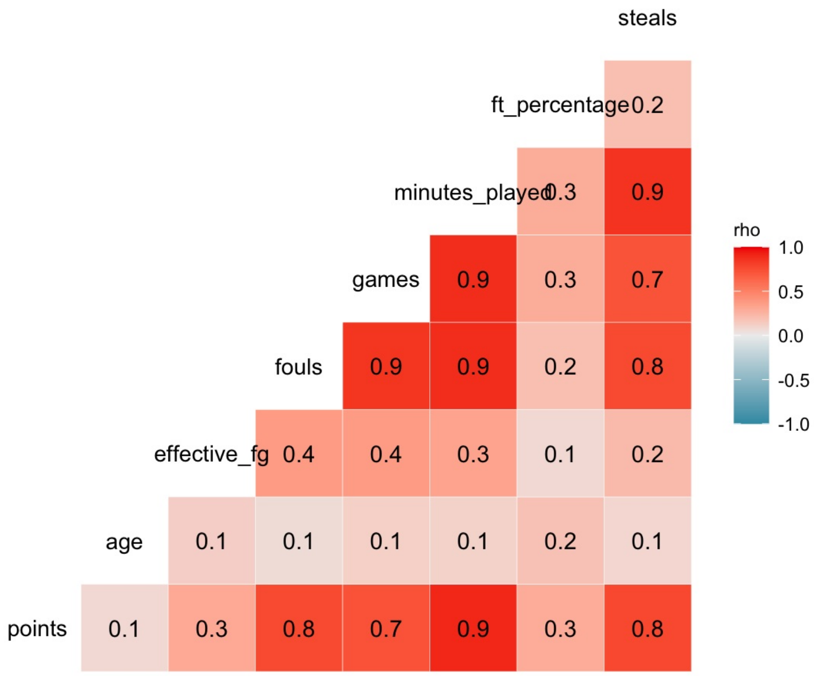



## Age distribution

### Player’s age




## Correlation Plot




### Attaching player salaries to player statistics for better analysis

```
player_statistics <-
  merge(player_statistics,
        player_salaries[,c("player_name","salary")], by = "player_name", all.x = TRUE)
```

### Diving datasets based on positions

```
# Dividing datasets for each position

point_guard_df <- subset(player_statistics, position == "PG")
shooting_guard_df <- subset(player_statistics, position == "SG")
small_forward_df <- subset(player_statistics, position == "SF")
power_forward_df <- subset(player_statistics, position == "PF")
center_df <- subset(player_statistics, position == "C")
```

# D. Data Modelling & prediction for selecting players

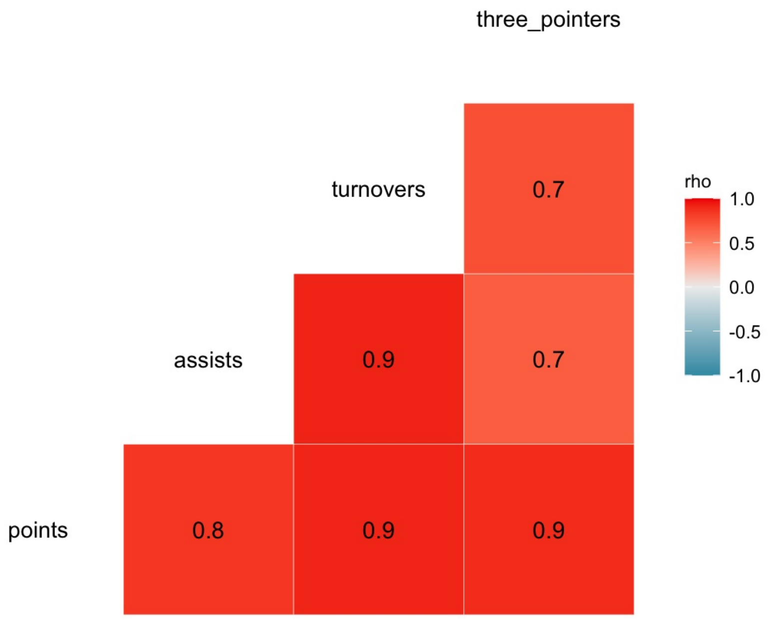
Data modeling is a crucial aspect of data analysis that involves the process of creating a model based on a specific dataset to uncover relationships, that can aid in making better decisions. In this project, data modeling has been used to predict the performance of basketball players based on specific attributes for each position.

The modeling process involved selecting key characteristics for each position based on research. These characteristics were then used to build separate models for each position using regression techniques. The models were trained on historical data to predict the expected performance of a player based on their attributes.

After building the models, a new variable was created using the predicted values. This variable represents the expected performance of a player for a given position based on their attributes. A budget threshold was set for each position, and players were selected based on their predicted performance within that budget.

### 1. Selecting a player for point guard

Creating a correlation plot using the variables required for the point guard position.



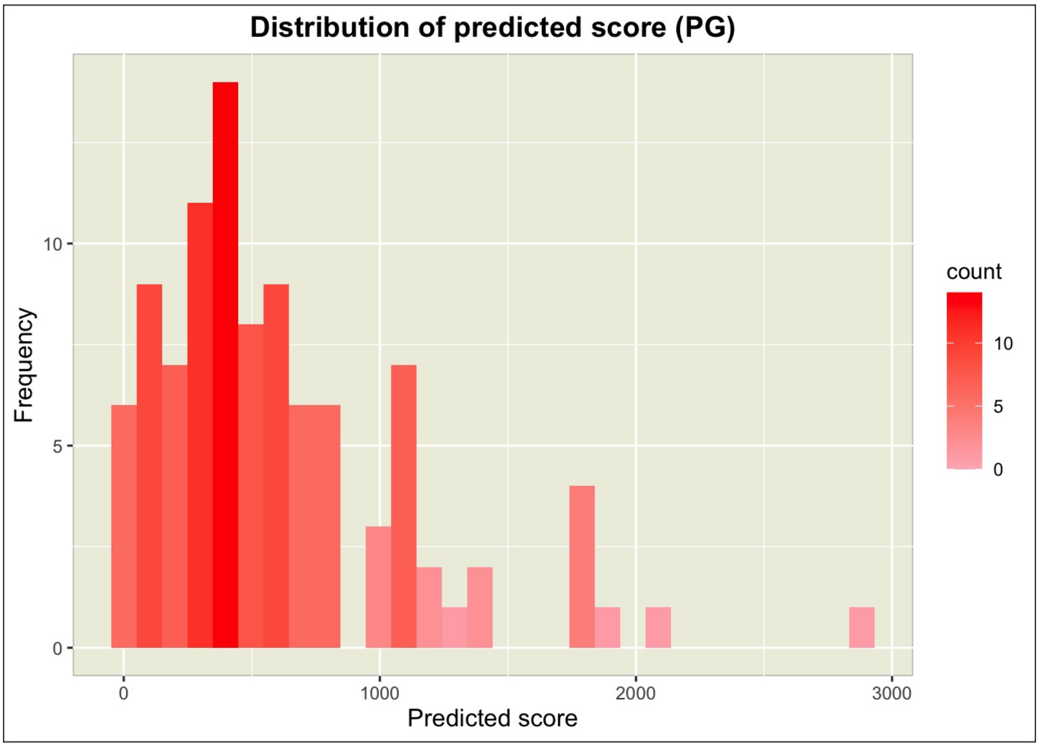

### Model summary for Point Guard

```
##
## Call:
## lm(formula = points ~ assists + turnovers + three_pointers, data = point_guard_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -425.34  -71.38  -12.63   75.11  379.59
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -2.3272    23.2987  -0.100    0.921
## assists          0.2499     0.2033   1.229    0.222
## turnovers        3.4863     0.5061   6.888 6.33e-10 ***
## three_pointers   3.6829     0.2940  12.525  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 137.1 on 94 degrees of freedom
## Multiple R-squared:  0.938,  Adjusted R-squared:  0.936
## F-statistic: 473.7 on 3 and 94 DF,  p-value: < 2.2e-16
```

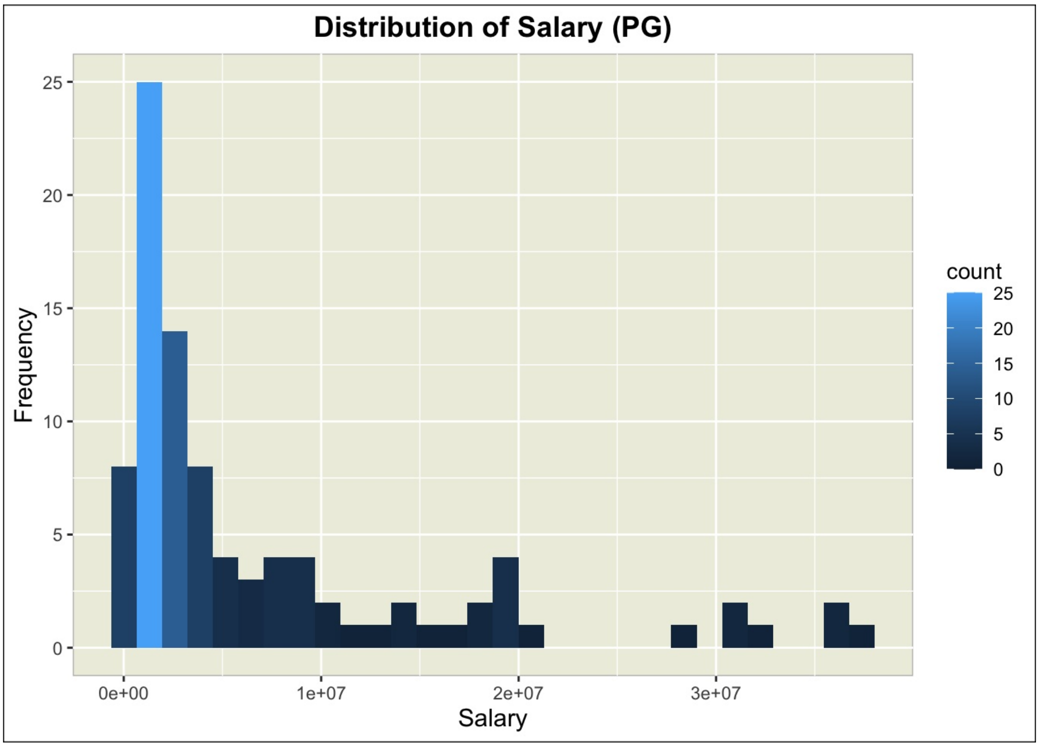
### Prediction for point guard

```
# Prediction of score variable just based on those specific parameters

output_pg <- point_guard_df[c("assists", "turnovers", "three_pointers","player_name", "salary")]

predicted_values <- predict(pg_model,output_pg)

output_pg$predicted_values <- predicted_values
```

### Prediction score for point guard




**Salary distribution for point guard**




### POINT GUARD SELECTION

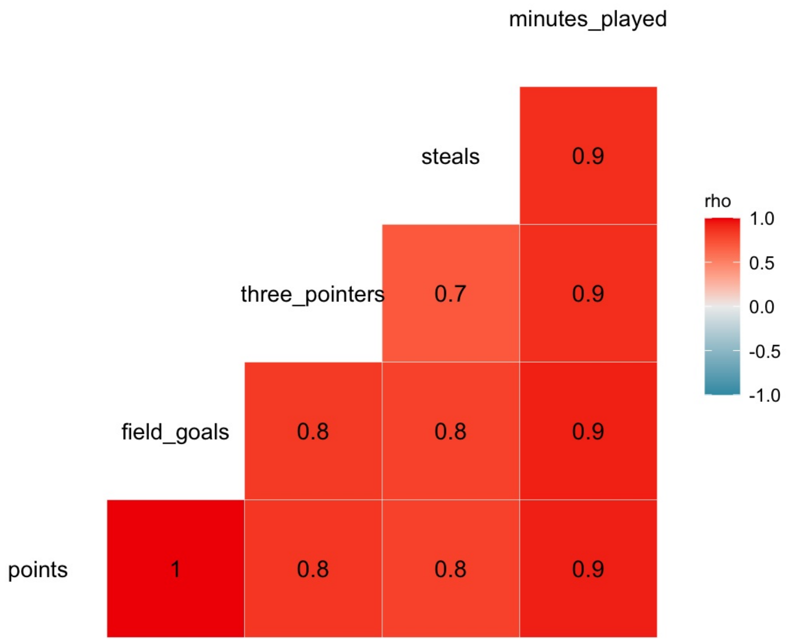
The chart shows that we should select the player with maximum score and and considering the budget salary less than 18 million.

```
output_pg %>%
  filter(salary < 18000000) %>%
  arrange(desc(predicted_values)) %>%
  slice(1)
```

```
##   assists turnovers three_pointers      player_name  salary predicted_values
## 1     563       253            234 D'Angelo Russell 7019698         1882.202
```

## 2. Selecting a player for Shooting guard

Creating a correlation plot using the variables required for the shooting guard position.



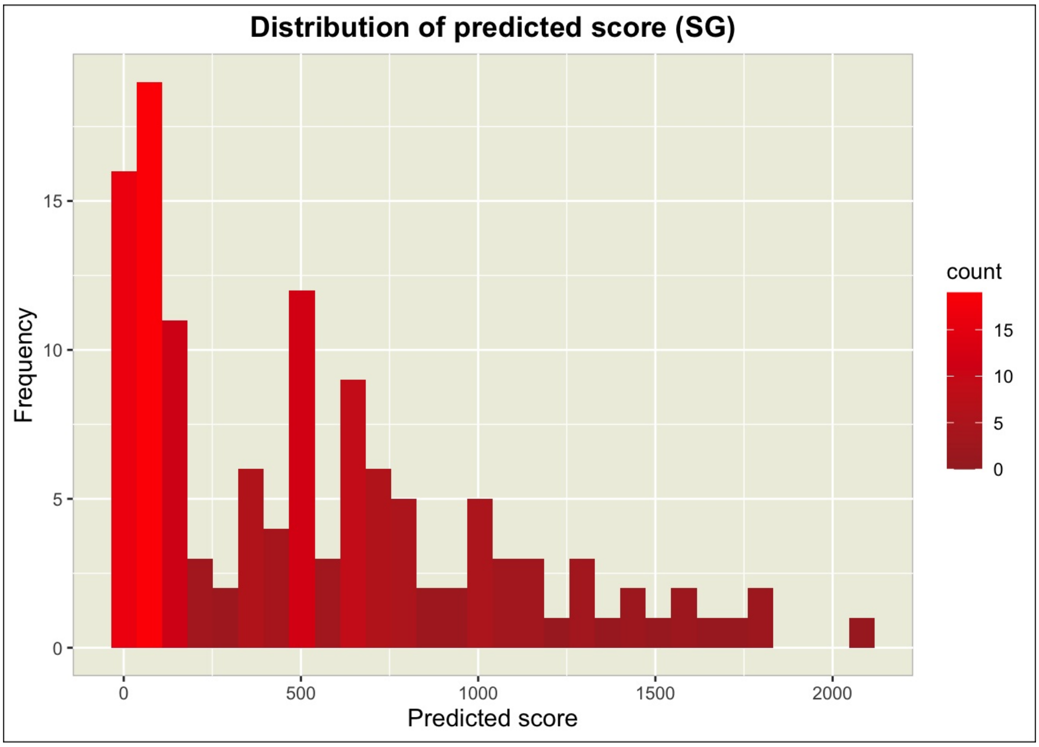

### Model summary for Shooting guard

```
##
## Call:
## lm(formula = points ~ field_goals + steals + three_pointers +
##     minutes_played, data = shooting_guard_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -134.852  -14.293   -0.563   10.997  189.606
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -1.094555   5.928388  -0.185 0.853829
## field_goals     2.644633   0.050743  52.119  < 2e-16 ***
## steals          0.055276   0.243309   0.227 0.820664
## three_pointers  0.434685   0.114985   3.780 0.000245 ***
## minutes_played -0.009687   0.016009  -0.605 0.546230
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 38.49 on 121 degrees of freedom
## Multiple R-squared:  0.9943, Adjusted R-squared:  0.9941
## F-statistic:  5244 on 4 and 121 DF,  p-value: < 2.2e-16
```

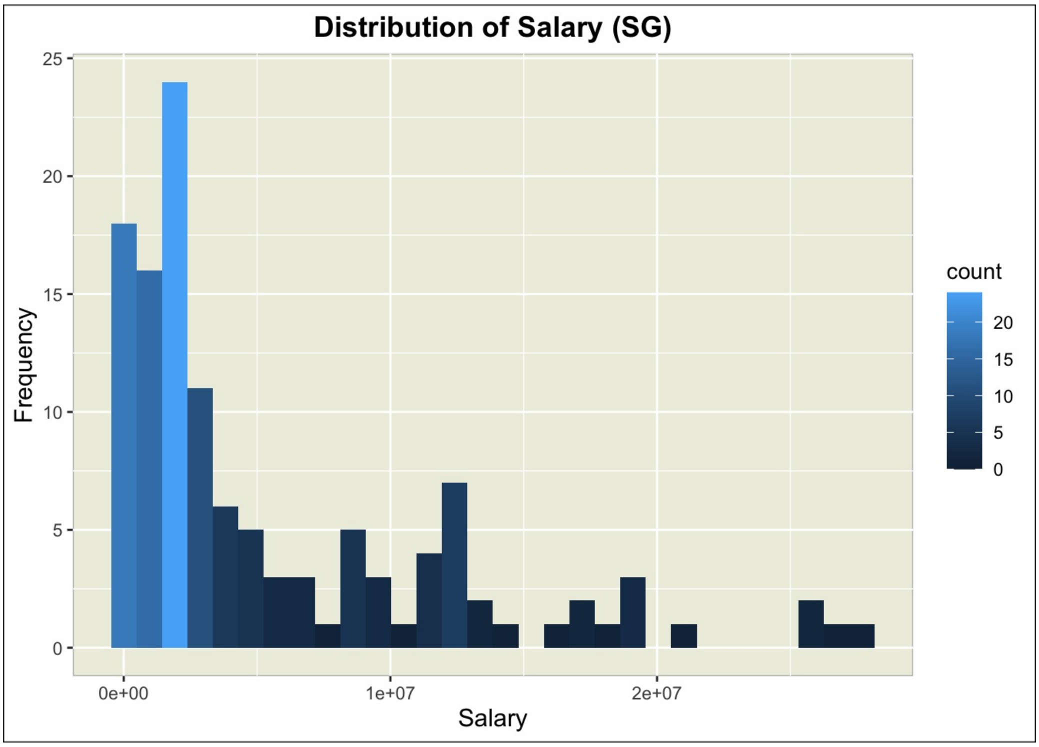
### Prediction for shooting guard

```
# Prediction of score variable just based on those specific parameters

output_sg <- shooting_guard_df[c("field_goals","three_pointers",
                              "steals", "minutes_played","player_name", "salary")]

predicted_sg <- predict(sg_model,output_sg)

output_sg$predicted_values<- predicted_sg
```

### Prediction score for Shooting guard




**Salary distribution for shooting guard**




**SHOOTING GUARD SELECTION**

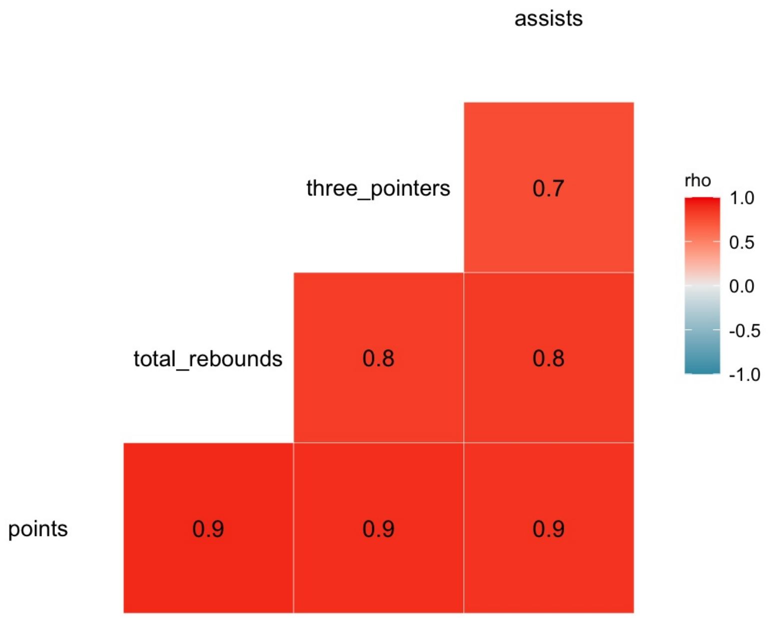
The chart shows that we should select the player with maximum score and and considering the budget salary less than 20 million.

```
# This shows that we should select the player with score > 1500 and
# and considering the budget salary less than 20 million.

output_sg %>%
  filter(salary < 20000000) %>%
  arrange(desc(predicted_values)) %>%
  slice(1)
```

```
##   field_goals three_pointers steals minutes_played   player_name   salary
## 1         655            241     84           2652 Klay Thompson 18988725
##   predicted_values
## 1         1814.852
```

## 3. Selecting a player for Small forward

Creating a correlation plot using the variables required for the small forward position.



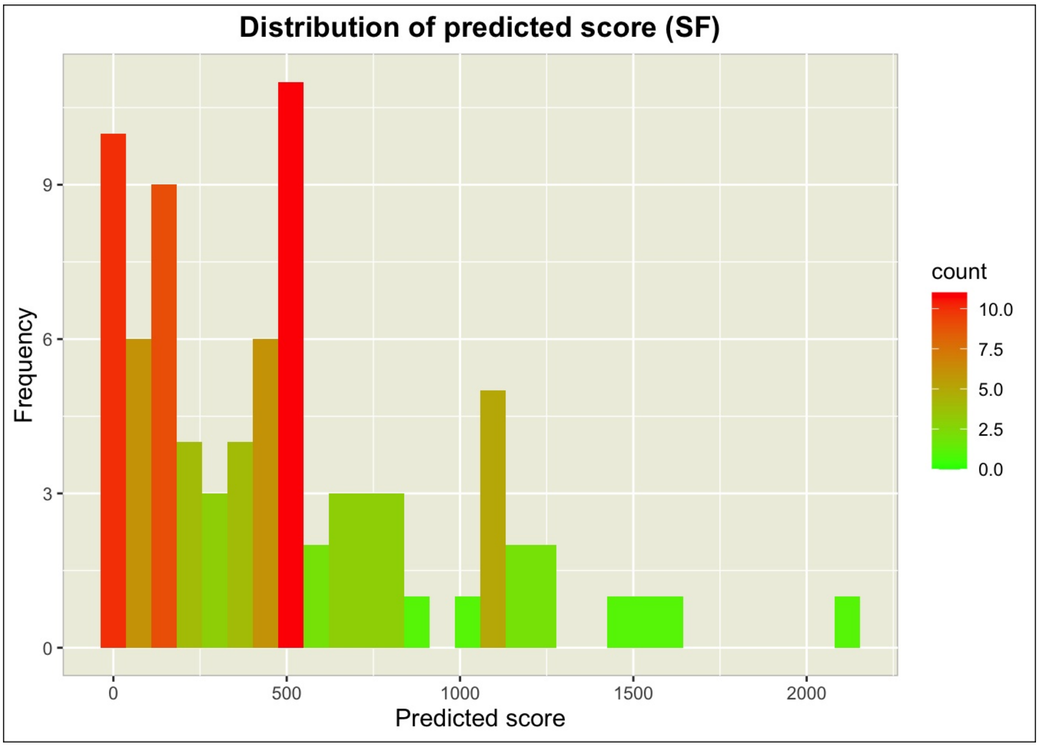

### Model summary for Small forward

```
##
## Call:
## lm(formula = points ~ total_rebounds + assists + three_pointers,
##     data = small_forward_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -429.49  -60.73    7.34   44.42  483.02
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -45.6453    30.4193  -1.501 0.137674
## total_rebounds   1.2205     0.2672   4.568 1.89e-05 ***
## assists          1.2978     0.3357   3.866 0.000233 ***
## three_pointers   3.2549     0.5755   5.656 2.68e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 156 on 75 degrees of freedom
## Multiple R-squared:  0.8962, Adjusted R-squared:  0.8921
## F-statistic:   216 on 3 and 75 DF,  p-value: < 2.2e-16
```

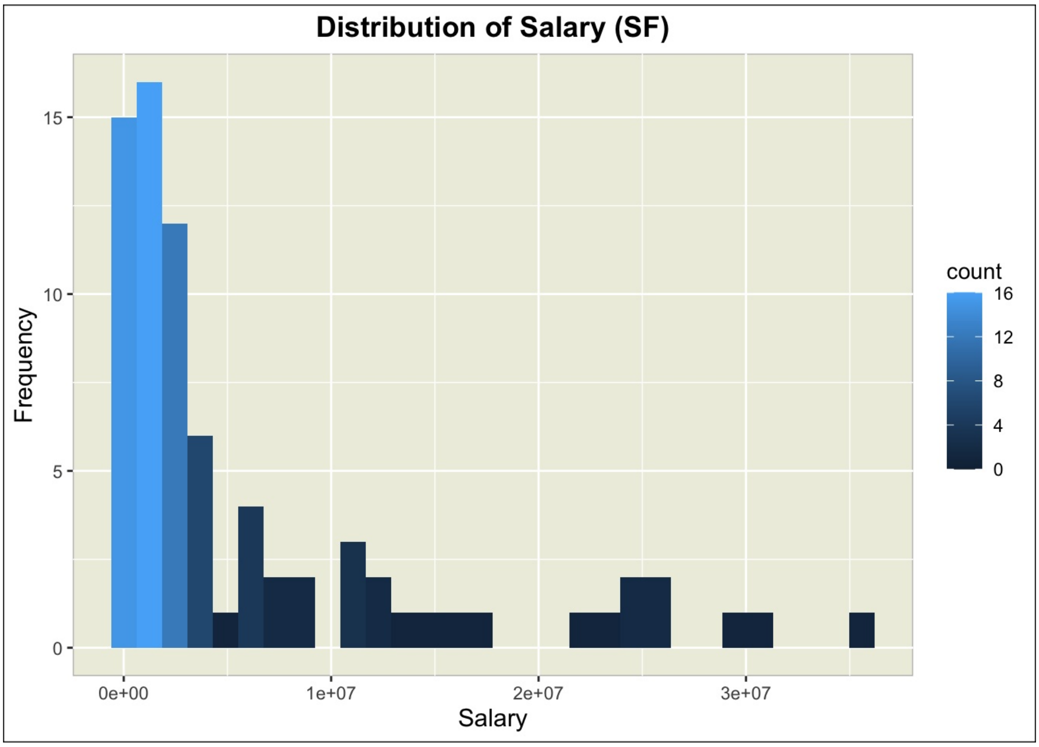
### Prediction for small forward

```
# Prediction of score variable just based on those specific parameters

output_sf <- small_forward_df[c("total_rebounds","three_pointers", "assists","player_name", "salary")]

predicted_sf <- predict(sf_model,output_sf)

output_sf$predicted_values<- predicted_sf
```

### Prediction score for Small Forward




**Salary distribution for small forward**




### SMALL FORWARD SELECTION

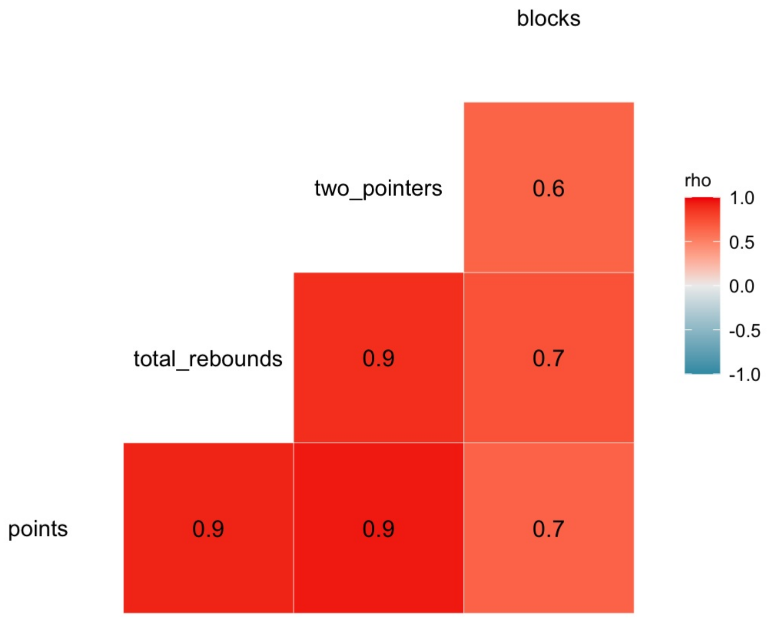
The chart shows that we should select the player with maximum score and and considering the budget salary less than 20 million.

```
# This shows that we should select the player with score > 1500 and
# and considering the budget salary less than 20 million.

output_sf %>%
  filter(salary < 20000000) %>%
  arrange(desc(predicted_values)) %>%
  slice(1)
```

```
##   total_rebounds three_pointers assists     player_name   salary
## 1            461            179     331 Khris Middleton 13000000
##   predicted_values
## 1          1529.22
```

## 4. Selecting a player for Power forward

Creating a correlation plot using the variables required for the power forward position.



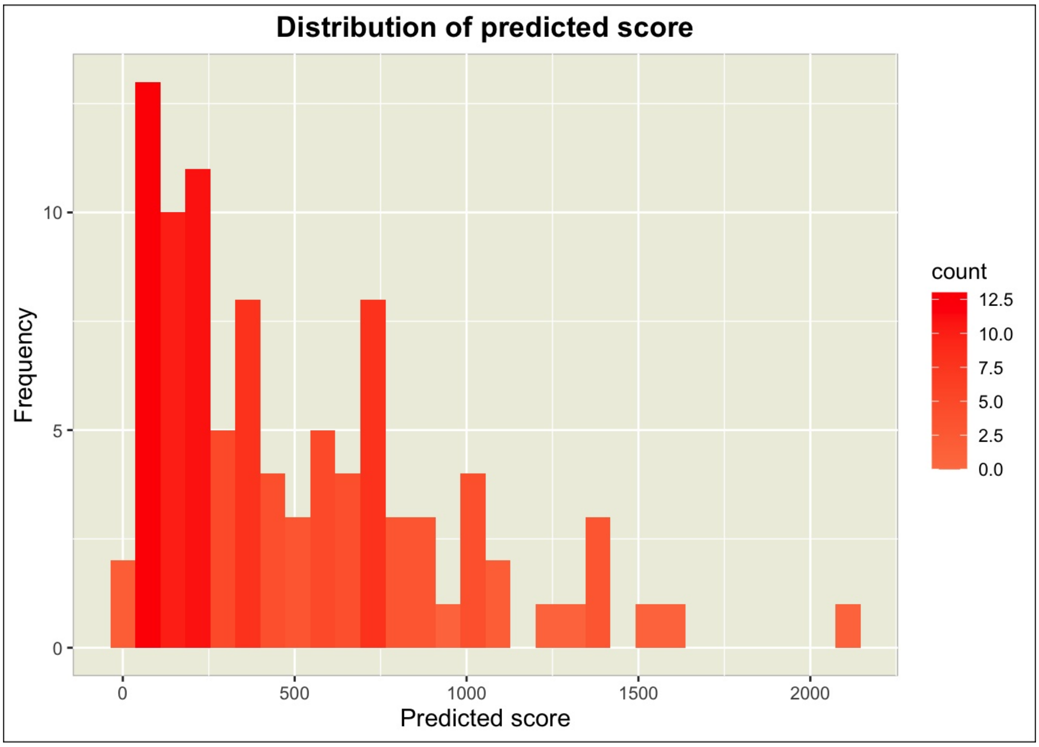

### Model summary for power forward

```
##
## Call:
## lm(formula = points ~ total_rebounds + two_pointers + blocks,
##     data = power_forward_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -258.11  -62.52  -21.08   46.23  446.61
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     19.8578    21.8313   0.910    0.365
## total_rebounds   0.9140     0.1554   5.883 6.77e-08 ***
## two_pointers     2.0351     0.1998  10.187  < 2e-16 ***
## blocks          -0.6064     0.7934  -0.764    0.447
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 122.8 on 90 degrees of freedom
## Multiple R-squared:  0.9263, Adjusted R-squared:  0.9239
## F-statistic: 377.2 on 3 and 90 DF,  p-value: < 2.2e-16
```

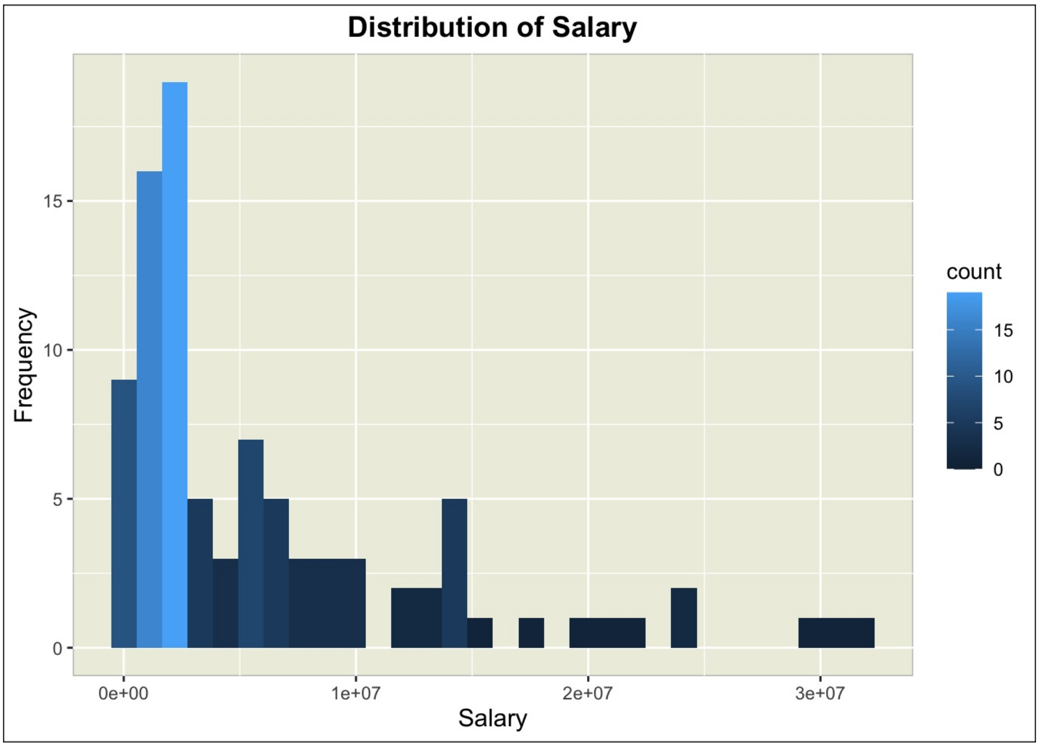
### Prediction for Power forward

```
# Prediction of score variable just based on those specific parameters

output_pf <- power_forward_df[c("total_rebounds","two_pointers", "blocks","player_name", "salary")]

predicted_pf <- predict(pf_model,output_pf)

output_pf$predicted_values<- predicted_pf
```

### Prediction score for Power forward




**Salary distribution for power forward**



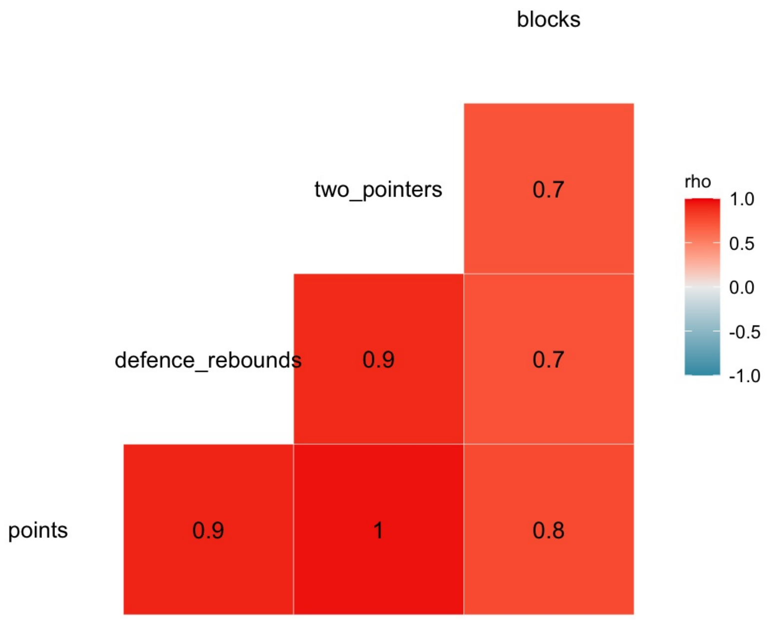

**POWER FORWARD SELECTION**

```
# This shows that we should select the player with score > 1500 and
# and considering the budget salary less than 20 million.

output_pf %>%
  filter(salary < 20000000) %>%
  arrange(desc(predicted_values)) %>%
  slice(1)
```

```
##   total_rebounds two_pointers blocks   player_name  salary predicted_values
## 1            634          504     45 Julius Randle 8641000         1597.747
```

## 5. Selecting a player for Center

Creating a correlation plot using the variables required for the Center position.



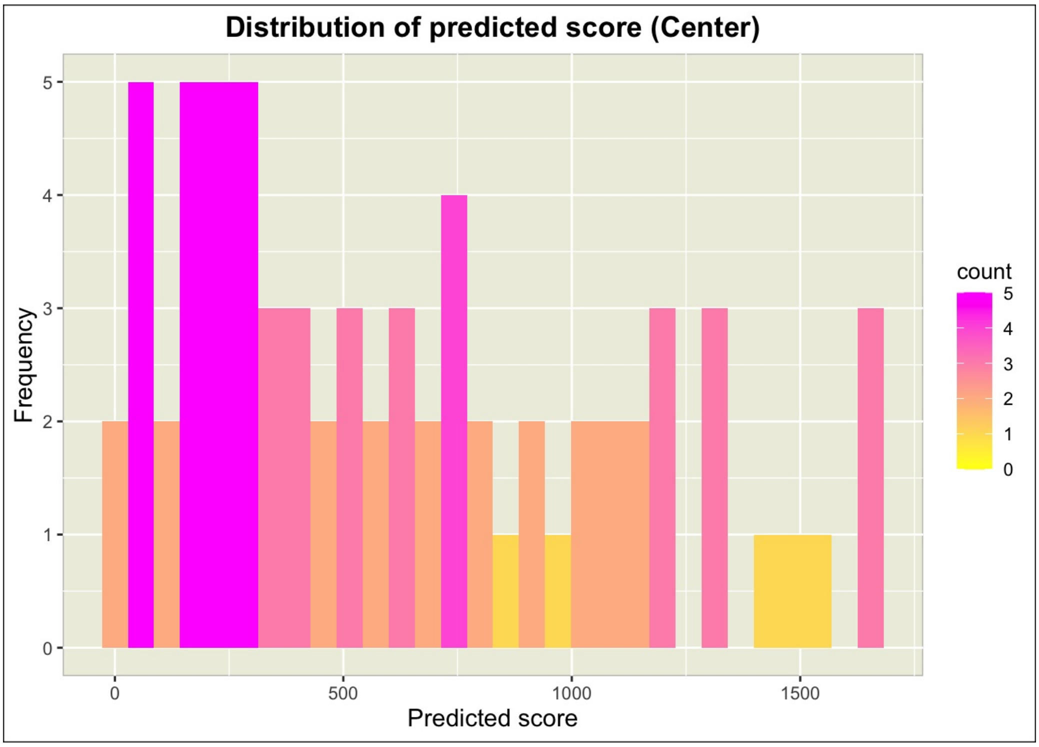

### Model summary for Center

```
##
## Call:
## lm(formula = points ~ defence_rebounds + two_pointers + blocks,
##     data = center_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -296.75  -49.39   -2.88   36.19  351.83
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -5.0780    24.1443  -0.210  0.83407
## defence_rebounds   0.5481     0.1636   3.351  0.00134 **
## two_pointers       1.8670     0.1802  10.361 1.78e-15 ***
## blocks             1.1852     0.4459   2.658  0.00986 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 114.5 on 66 degrees of freedom
## Multiple R-squared:  0.9463, Adjusted R-squared:  0.9438
## F-statistic: 387.5 on 3 and 66 DF,  p-value: < 2.2e-16
```

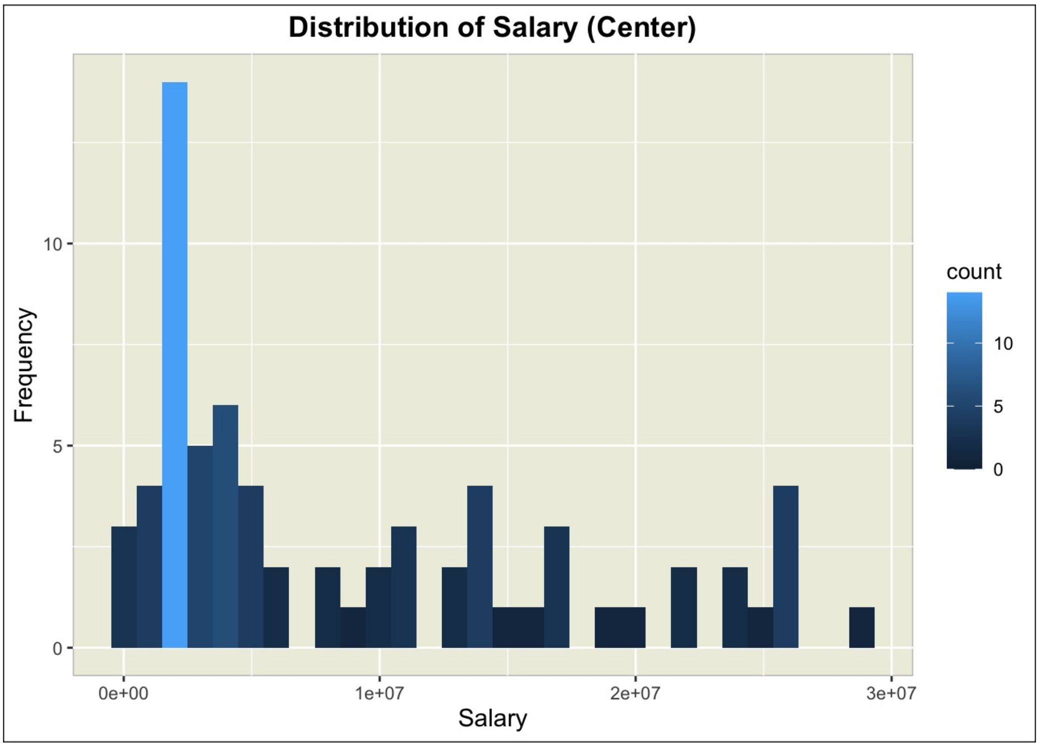
### Prediction for center

```
# Prediction of score variable just based on those specific parameters

output_ct <- center_df[c("defence_rebounds","two_pointers", "blocks","player_name", "salary")]

predicted_ct <- predict(ct_model,output_ct)

output_ct$predicted_values <- predicted_ct
```

### Prediction score for Center




**Salary distribution for center**




### CENTER SELECTION

The chart shows that we should select the player with maximum score and and considering the budget salary less than 20 million.

```
# This shows that we should select the player with score > 1500 and
# and considering the budget salary less than 20 million.

output_ct %>%
  filter(salary < 20000000) %>%
  arrange(desc(predicted_values)) %>%
  slice(1)
```

```
##   defence_rebounds two_pointers blocks    player_name   salary predicted_values
## 1              736          617     89 Nikola Vucevic 12750000         1655.797
```

# E. Player recommendations

Point guard

1. I choose **Angelo Russell** for **Point guard**. Salary - 7019698

Shooting guard

2. I choose **Klay Thompson** for **Shooting guard** Salary - 18988725

Small forward

3. I choose **Khris Middleton** for **Small forward** Salary - 13000000

Power forward

4. I choose **Julius Randle** for **Power forward** Salary - 8641000

Center

5. I choose **Nikola Vucevic** for **Center** Salary - 12750000

So the total salary for all 5 players is 61.37 million (approx) which is under the budget specifically assigned for main players.

# F. Summary

In this project, I was tasked with finding the best five starting players (one from each position) for the Chicago Bulls basketball team for the upcoming season. The team's budget for player contracts was $118 million, so I had to find players who could perform well on the court without exceeding the budget.

To do this, I first obtained data on NBA players from the past season and cleaned it up, filtering out unnecessary columns and missing values. Then, I explored the data using visualizations to gain insights on how different variables were related to each other and to player performance.

Next, I performed a linear regression analysis on the data to build a model that could predict a player's score based on their performance statistics. Using this model, I identified the players who were likely to score the highest and selected the best player for each position, considering their predicted scores and their salaries.

After analyzing the data and modeling the players, I have decided to select Angelo Russell as the Point Guard, Klay Thompson as the Shooting Guard, Khris Middleton as the Small Forward, Julius Randle as the Power Forward and Nikola Vucevic as the Center for the Chicago Bulls basketball team. I believe that these players will perform well on the court and help the team to improve its ranking in the upcoming season.

# G. References


1. Wikipedia. Basketball positions. [cited 7 May 2023]. Available from: https://en.wikipedia.org/wiki/Basketball_positions (https://en.wikipedia.org/wiki/Basketball_positions)

2. Red Bull. Basketball Positions: What Each Player Does [Internet]. Red Bull; [cited 2023 May 10]. Available from: https://www.redbull.com/us-en/basketball-positions-what-each-player-does (https://www.redbull.com/us-en/basketball-positions-what-each-player-does)

3. Golliver, B. (2018, September 21). Breaking Down NBA Teams' Revenue, Spending by Market Size. Sports Illustrated. https://www.si.com/nba/2018/09/21/nba-teams-revenue-spending-breakdown-small-large-market (https://www.si.com/nba/2018/09/21/nba-teams-revenue-spending-breakdown-small-large-market)

4. Smith J. 2022 Ranking: Top 20 NBA Players Right Now. NBC Sports Washington [Internet]. 2022 [cited 2023 May 07]. Available from: https://www.nbcsports.com/washington/wizards/2022-ranking-top-20-nba-players-right-now (https://www.nbcsports.com/washington/wizards/2022-ranking-top-20-nba-players-right-now)