Transfer Market of Football Player

Mohammed Akib Iftakher (20215282)

2022-04-13

1. Introduction

This analysis is an assessment of a data collection of Transfer market in order to estimate the player values. This study is intended to be an example of the many forms of analysis that can be performed using Transfermarkt Football Statistics. Information such as such as scores, assists, results, statistics, transfer news, and fixtures usually determines the player values.

1.1 Library Installation

1. **dplyr**- a structure of data manipulation that provides a uniform set of verbs.
2. **ggplot2** - a framework for making graphics declaratively.
3. **tidyverse** - provide key data transformation functions in a single package.
4. **caret** - used for Data preparation, model construction, and model assessment.
5. **corrr** - used for exploring correlations.
6. **Amelia** - employed in the repeated imputation of multivariate incomplete data.
7. **highcharter** - a package that uses shortcut functions to visualize R objects
8. **corrplot** - a visual exploration tool for correlation matrices.
9. **corrgram** - creates graphical display of a correlation matrix.
10. **grid**- represents graphical objects or grobs.
11. **gridExtra** - adds functionality to the grid system.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.1.3

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble  3.1.6     v purrr   0.3.4
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(corrr)
```

```
## Warning: package 'corrr' was built under R version 4.1.3
```

```
library(Amelia)
```

```
## Warning: package 'Amelia' was built under R version 4.1.3

## Loading required package: Rcpp

## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.8.0, built: 2021-05-26)
## ## Copyright (C) 2005-2022 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

```
library(highcharter)
```

```
## Warning: package 'highcharter' was built under R version 4.1.3

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(grid)
library(gridExtra)
```

```
## Warning: package ’gridExtra’ was built under R version 4.1.3
```

```
##
## Attaching package: ’gridExtra’
```

```
## The following object is masked from ’package:dplyr’:
##
##     combine
```

```


players <- read.csv("D:\\personal\\R_programming\\Football_transfer_market\\players.csv")
games <-  read.csv("D:\\personal\\R_programming\\Football_transfer_market\\games.csv")
appearances <-  read.csv("D:\\personal\\R_programming\\Football_transfer_market\\appearances.csv")
clubs <-  read.csv("D:\\personal\\R_programming\\Football_transfer_market\\clubs.csv")
competitions <-  read.csv("D:\\personal\\R_programming\\Football_transfer_market\\competitions.csv")
```

1.2 Importing Dataset

2 Data Preparation

2.1 Merging Dataset

The merge function has been used to horizontally merge two data frames (datasets). In most situations, one or more unique key variables connect two data frames.

```
# Merging datesets players and appearances
market_value <- merge(players, appearances, by  = "player_id")

# Removing column named url
market_value <- subset(market_value, select = -c(url) )


# Renaming a specific column (competition_id)
competitions <- competitions %>%
  rename(competition_code = competition_id)

# Merging datesets games and competitions
games <- merge(games, competitions, by  = "competition_code")


# Merging datesets games and market_value
market_value <- merge(market_value, games, by = "game_id" )

# Renaming a specific column (club_id)
clubs <- clubs %>%
  rename(current_club_id = club_id)
```

```
# Merging datesets clubs and market_value
market_value <- merge(market_value, clubs, by  = "current_club_id")
```

2.2 Dropping Column

For the competitive model, the following independent variables are commonly used: age, experience (number of league appearances), goal record, position played, international appearances, selling club's status and performances, divisional standing, and amount of goals [1].

To build the models for estimation, the columns that are unnecessary or duplicated must first be removed based on the model requirements.

1. **player_club_id** - It is same as club_id. So it has been dropped.
2. **name.x** / **pretty_name.x/ name** / **pretty_name.y** / **name.y-** The estimation doesn't depends on the individual player name.
3. **country_of_birth** / **country_of_citizenship** - The estimation of transfer market doesn't rely on the country of birth.
4. **home_club_id,away_club_id** - The correlation between home_club_id and away_club_id with market_value_in_gbp is very negligible.
5. **url.x/url.y/url** - It is merely the link the data were retrieved.
6. **foot** - Player value doesn't depend on foot.
7. **referee** / **squad_size** / **coach_name** / **country_name** - These information doesn't have high impact on the player's estimation.
8. **Stadium** / **attendance** / **stadium_name** / **stadium_seats** - Information related to stadium has small correlation values.
9. **date_of_birth** - It is same as the age. So, it is dropped.

```
# Dropping column
market_value <- subset(market_value, select = -c(player_club_id, name.x, pretty_name.x, country_of_birt

# Bring played_id in the front
market_value <- market_value %>%
  select(player_id,last_season, everything())
```

Based on the data, it can be concluded that the majority of the undesirable columns have been removed.

2.3 Cleaning Data

All of the data that was not in use has been deleted. Because each data point refers to a unique player. As a result, assigning a mean value based on neighbors or average values will be difficult. Furthermore, the data-set has been filtered to include only data relating to the position **Attack**.

```
# Removing rows which are not available
market_value <- na.omit(market_value)

# Filtering the data associated with only position "Attack"
market_value <- market_value %>% filter(position== "Attack")
```

2.4 Features Engineering

Feature engineering is the most important strategy for constructing machine learning models. The term "feature engineering" refers to a wide range of techniques performed on variables (features) in order to fit them into the algorithm. In this scenario, Feature Construction is being used to produce extra features based on the original variables in order to improve the accuracy of the prediction model.

2.4.1 Goal_in_UCL

The price of a player connected with the team's attacking side is typically determined by goal scores in big competitions such as the UCL or Europa League.

```
for (i in 1:length(market_value$player_id)) {
  if (market_value$competition_code[i] == "CL" && market_value$goals[i] != 0 ) {
    market_value$G_UCL[i] <- market_value$goals[i]
  }
   else {
    market_value$G_UCL[i] <- 0
  }
}
```

```
for (i in 1:length(market_value$player_id)) {
  if (market_value$competition_code[i] == "EL" && market_value$goals[i] != 0 ) {
    market_value$G_EL[i] <- market_value$goals[i]
  }
   else {
    market_value$G_EL[i] <- 0
  }
}
```

2.4.2 Goal_in_EL

2.4.3 Assist_in_UCL

In this part of the process, two new column have been generated which corresponds to the assist in the UCL. According to the **Uefa**, on account of the assist numbers, price usually go high [2].

```
for (i in 1:length(market_value$player_id)) {
  if (market_value$competition_code[i] == "CL" && market_value$assists[i] != 0 ) {
    market_value$A_CL[i] <- market_value$assists[i]
  }
   else {
    market_value$A_CL[i] <- 0
  }
}
```

```
for (i in 1:length(market_value$player_id)) {
  if (market_value$competition_code[i] == "EL" && market_value$assists[i] != 0 ) {
    market_value$A_EL[i] <- market_value$assists[i]
  }
   else {
    market_value$A_EL[i] <- 0
  }
}
```

2.4.4 Assist_in_EL

2.5 Row Transformation

In this step, a single row has been set for each individual players. It will reduce number of row size from 260411 to 4218.

```
market_value1 <- market_value # Duplicating the dataset

market_value1 <- subset(market_value1, select = -c(position, season,competition_code))

market_value1 <- market_value1 %>% group_by(player_id,last_season, current_club_id, sub_position, heigh
```

```
## ‘summarise()‘ has grouped output by ’player_id’, ’last_season’,
## ’current_club_id’, ’sub_position’, ’height_in_cm’, ’market_value_in_gbp’,
## ’highest_market_value_in_gbp’. You can override using the ‘.groups‘ argument.
```

2.5.1 Creating New features

New features has been created which calculates the player goal per minutes.

```
market_value1$G_per_min <- market_value1$goals/market_value1$minutes_played
```

2.5.2 Renaming and Relocating The Column

In this section, the columns’ name has been renamed in order to wrap the names perfectly in the market_value1 dataset.

```
market_value1 <- market_value1 %>%
  rename(
    MV_GBP = market_value_in_gbp,
    hi_MV = highest_market_value_in_gbp,
    home_G = home_club_goals,
    away_G = away_club_goals,
    height = height_in_cm,
    min_play = minutes_played,
    age = average_age,
    Y_C = yellow_cards,
    id  = player_id,
    L_S = last_season,
    club_id = current_club_id,
    sub_ps = sub_position
    )
market_value1 <- market_value1 %>% relocate(MV_GBP, .after = last_col()) # Relocating MV_GBP at the end
```

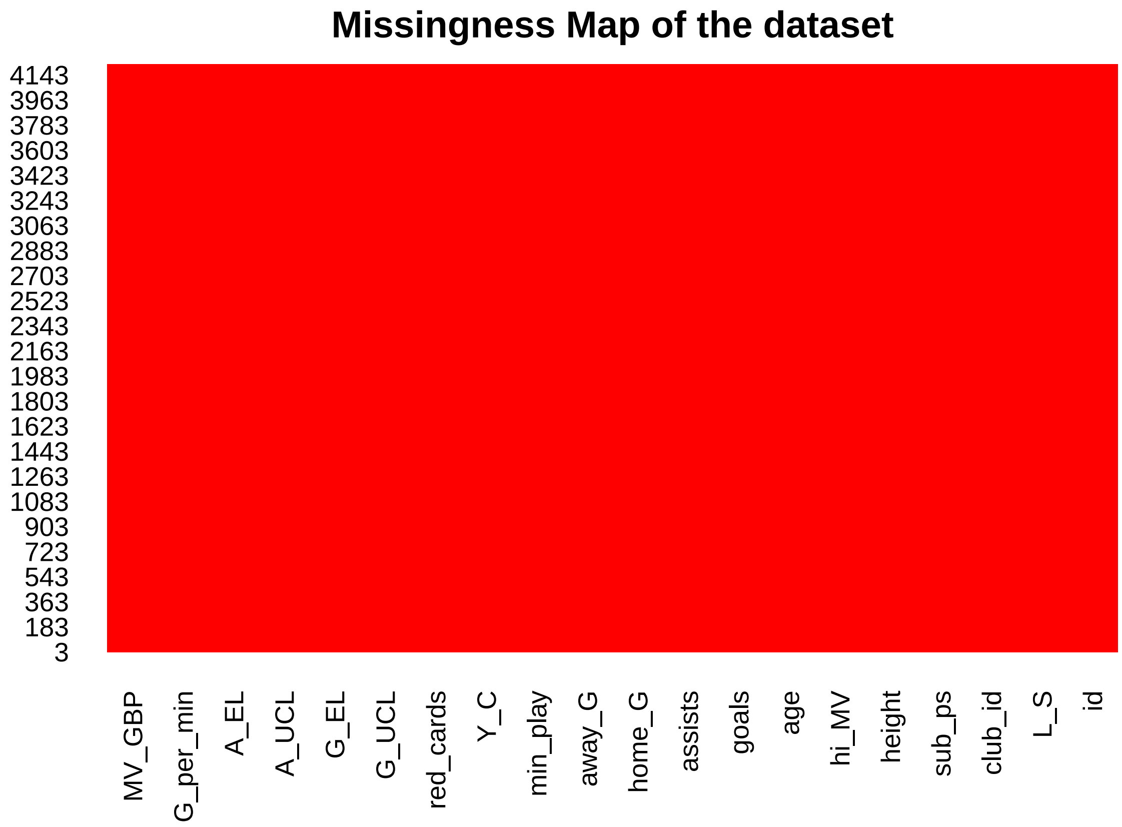
3. Exploratory Data Analysis

3.1 Analysing Missing Values

A Missigness map is plotted which will show values which are missing in the data. Color red denotes value presence and blue will denotes the missing patches.

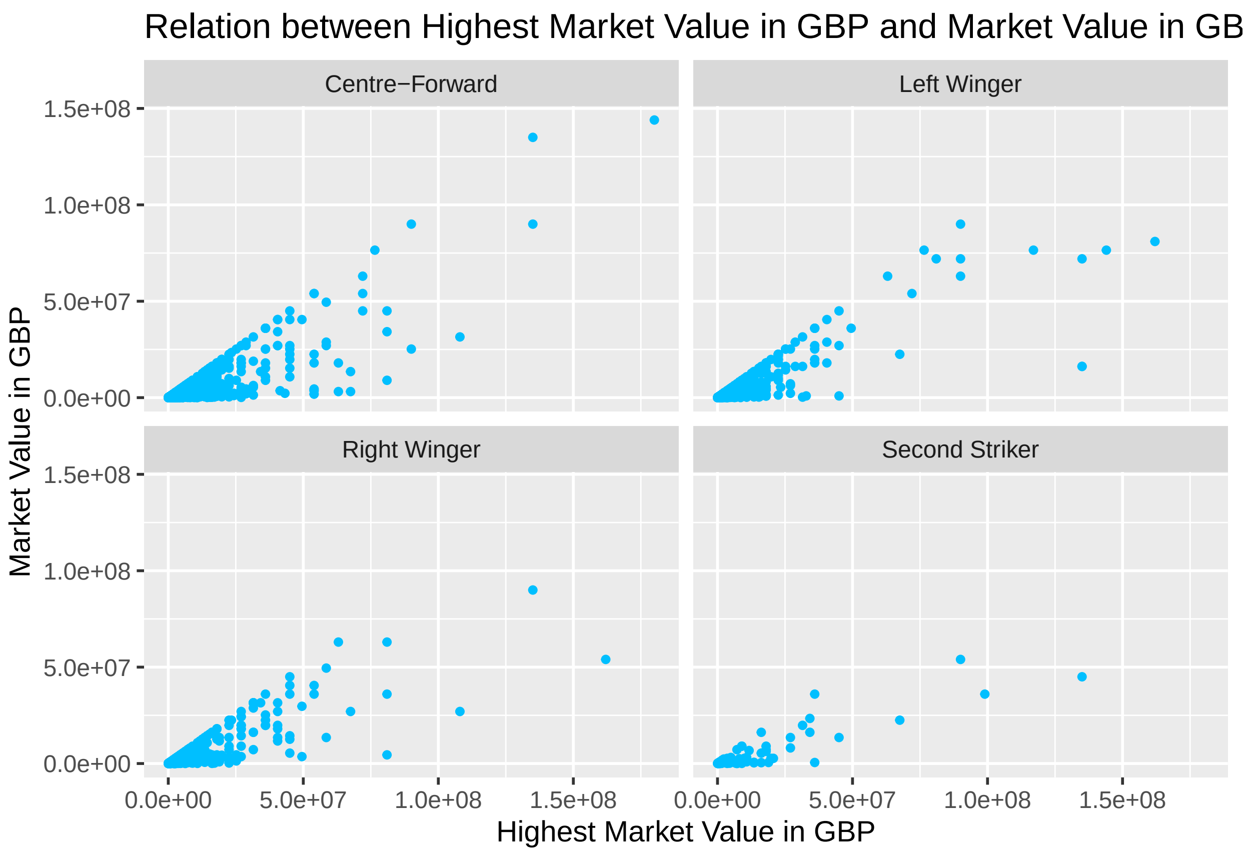
```
missmap(market_value1, col = c("blue", "red"), legend = F, main = ("Missingness Map of the dataset"))
```

```
## Warning: Unknown or uninitialised column: ‘arguments‘.
## Unknown or uninitialised column: ‘arguments‘.

## Warning: Unknown or uninitialised column: ‘imputations‘.
```



It can be seen that no data is being missed from the market_value1 dataset.

```
ggplot(market_value1, aes(x = hi_MV, y = MV_GBP)) +
  ggtitle("Relation between Highest Market Value in GBP and Market Value in GBP") +
  geom_point(colour = "deepskyblue", size = 1) +
  facet_wrap(~sub_ps)+
  labs(x = "Highest Market Value in GBP", y = "Market Value in GBP")
```

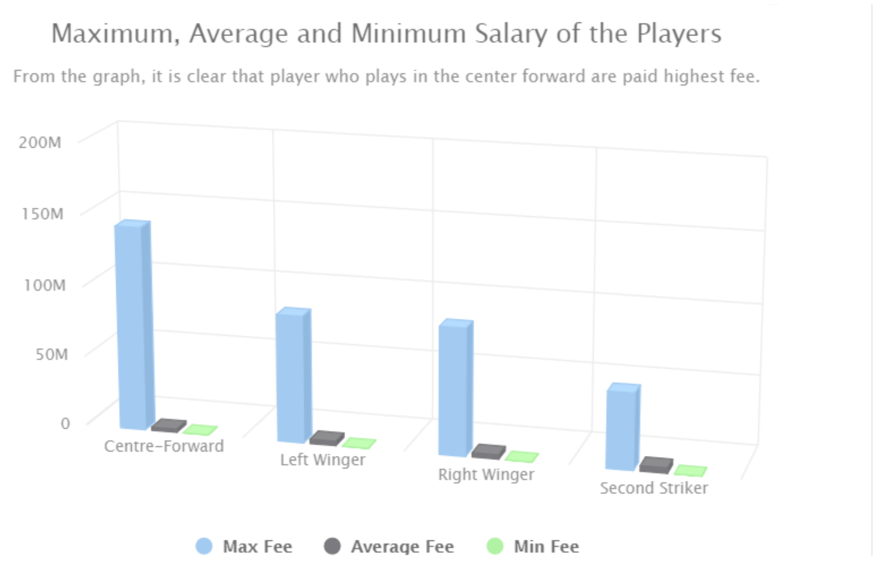


```
  #
```

3.2 Market Value VS Sub Position

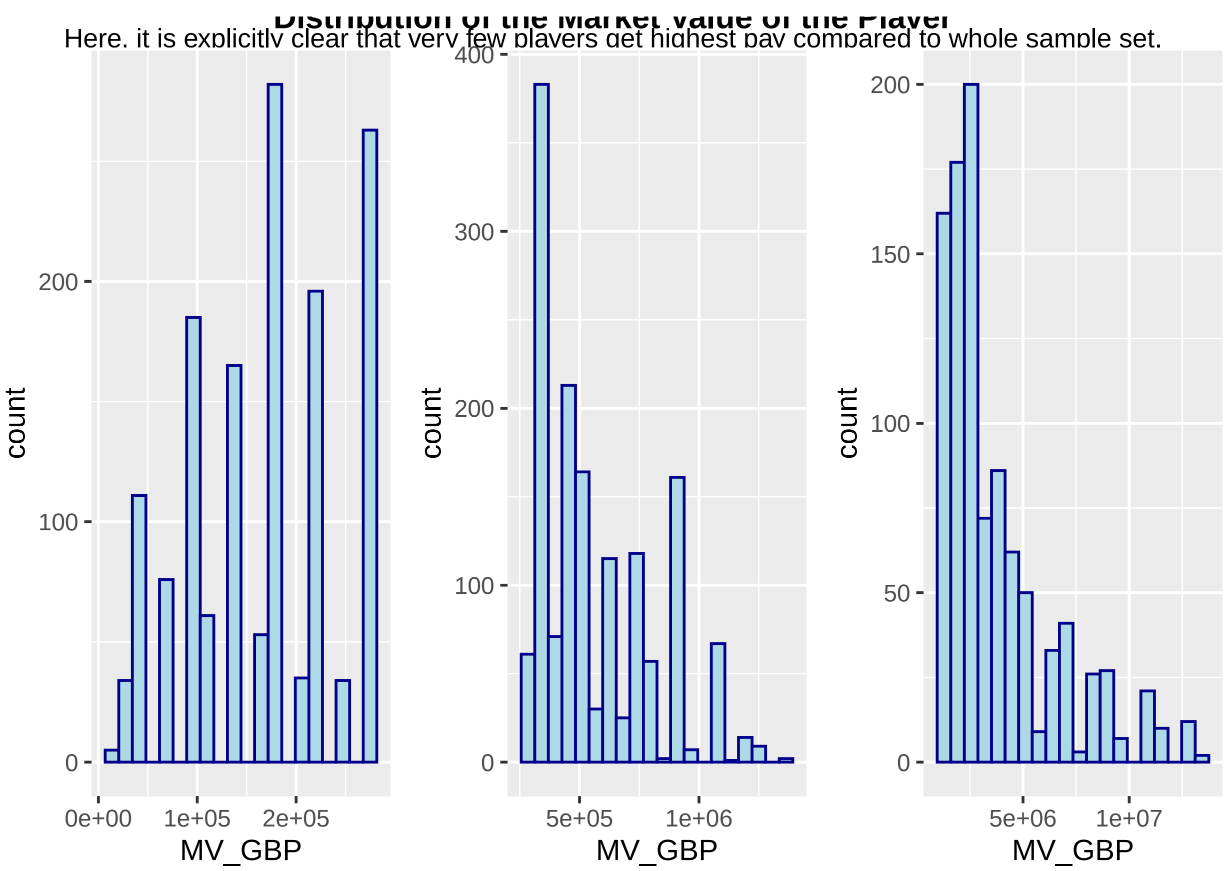
In this plot, it has been illustrated the maximum, average and minimun fees corresponds to the player in the front.

```
df = market_value1%>%
  filter(!is.na(sub_ps))%>%
  filter(!is.na(MV_GBP))%>%
  group_by(sub_ps)%>%
  summarise( min = min(MV_GBP), max = max(MV_GBP), Count = n(), Average = mean(MV_GBP))%>%
  arrange(desc(Count))%>%
  head(15)

  highchart()%>%
     hc_title(
    text = "Maximum, Average and Minimum Salary of the Players"
    ) %>%
    hc_subtitle(
    text = "From the graph, it is clear that player who plays in the center forward are paid highest fe
    ) %>%
  hc_xAxis(categories = df$sub_ps)%>%
  hc_add_series(name = "Max Fee", data = df$max)%>%
    hc_add_series(name = "Average Fee", data = df$Average)%>%
   hc_add_series(name = "Min Fee", data = df$Count)%>%
  hc_chart(type = "column",
           options3d = list(enabled = TRUE, beta = 15, alpha = 15))
```



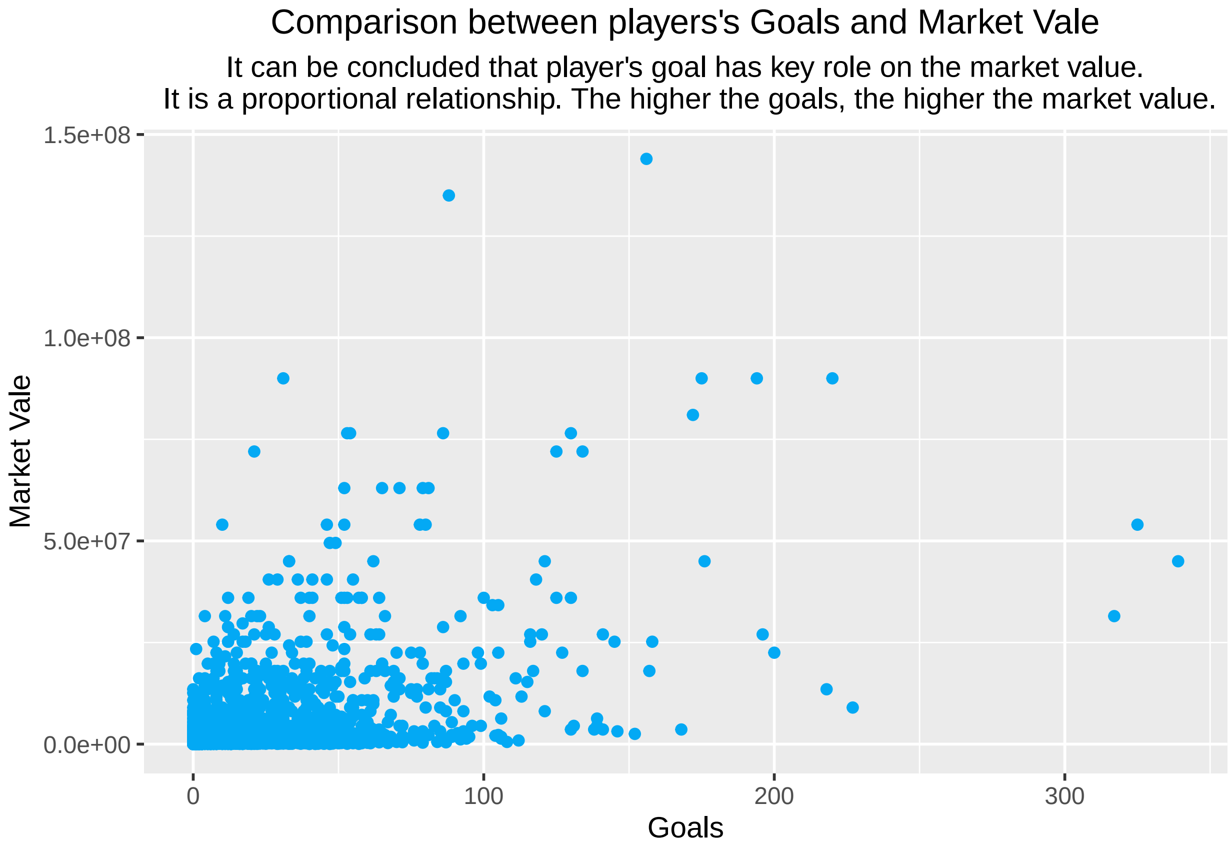
3.3 Distribution of the Market Value in GBP

```
hist_D <- market_value1[order(market_value1$MV_GBP),]

hist_1 <- hist_D[1:1500, ]
c1 = ggplot(data = hist_1, aes(x = MV_GBP)) + geom_histogram(color="darkblue", fill="lightblue",bins = 

hist_2 <- hist_D[1501:3000, ]
c2= ggplot(data = hist_2, aes(x = MV_GBP)) + geom_histogram(color="darkblue", fill="lightblue",bins = 2

hist_3 <- hist_D[3001:4000, ]
c3= ggplot(data = hist_3, aes(x = MV_GBP)) + geom_histogram(color="darkblue", fill="lightblue",bins = 2

tg <- textGrob("Distribution of the Market Value of the Player", gp = gpar(fontface = "bold", cex = 1))
sg <- textGrob('Here, it is explicitly clear that very few players get highest pay compared to whole sa

lt <- list(tg, sg)
heights <- do.call(unit.c, lapply(lt, function(.g) 1*grobHeight(.g)))
titles <- gtable::gtable_matrix('title',
                                grobs = matrix(lt, ncol=1),
                                widths = unit(1,'npc'),
                                heights = heights)


grid.arrange(c1,c2,c3, ncol= 3, top = titles)
```



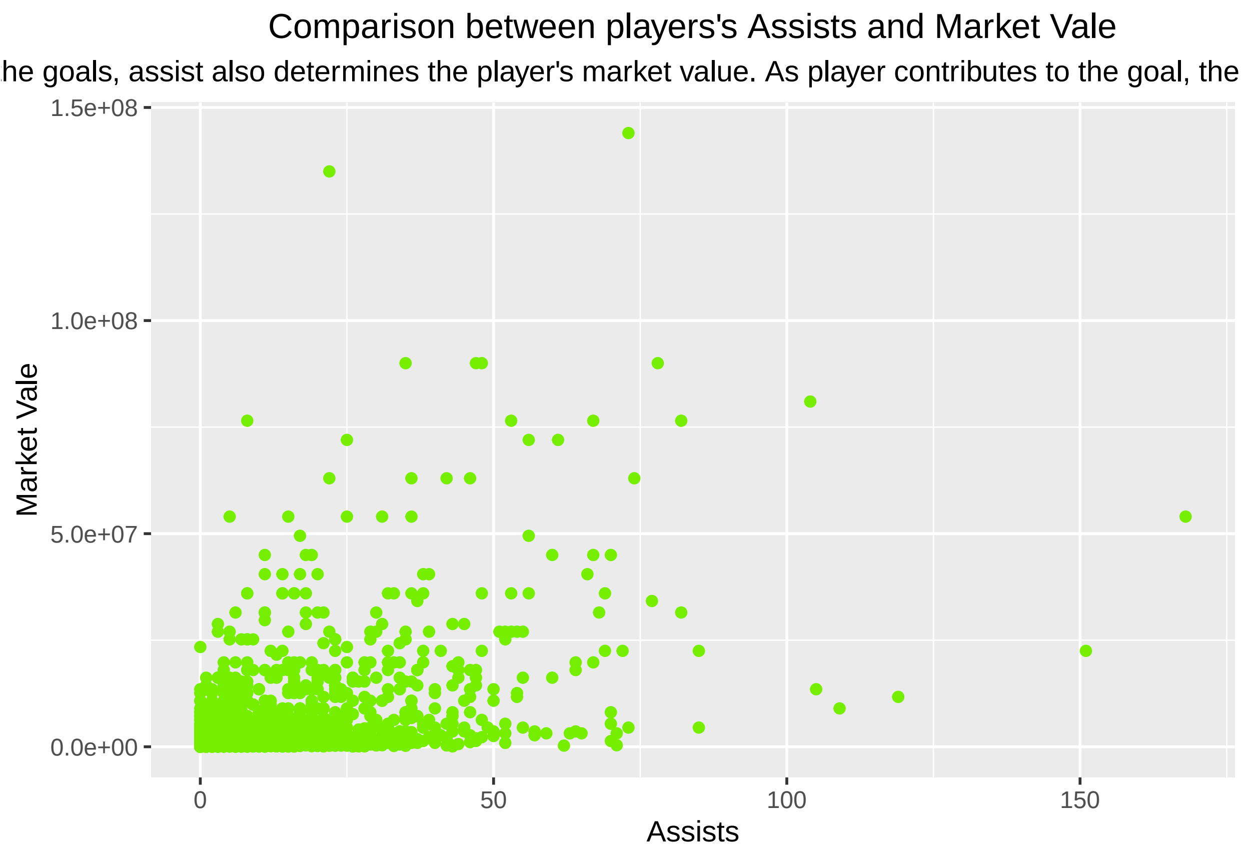
3.4 Relation between Goals and Market Value in GBP

```
ggplot(market_value1, aes(x = goals, y = MV_GBP)) +
  geom_point(colour = "#03A9F4") +
  labs(title = "Comparison between players's Goals and Market Vale",
       subtitle="It can be concluded that player's goal has key role on the market value.\n It is a prop
       x = "Goals", y = "Market Vale")+
  theme(plot.title=element_text(hjust=0.5),
        plot.subtitle=element_text(hjust=0.5))
```



3.5 Relation between Goals and Market Value in GBP

```
ggplot(market_value1, aes(x = assists, y = MV_GBP)) +
  geom_point(colour = "chartreuse2") +
  labs(title = "Comparison between players's Assists and Market Vale",
       subtitle="Similar to the goals, assist also determines the player's market value. As player conti
       x = "Assists", y = "Market Vale")+
  theme(plot.title=element_text(hjust=0.5),
        plot.subtitle=element_text(hjust=0.5))
```



3.6 Correlation Graph

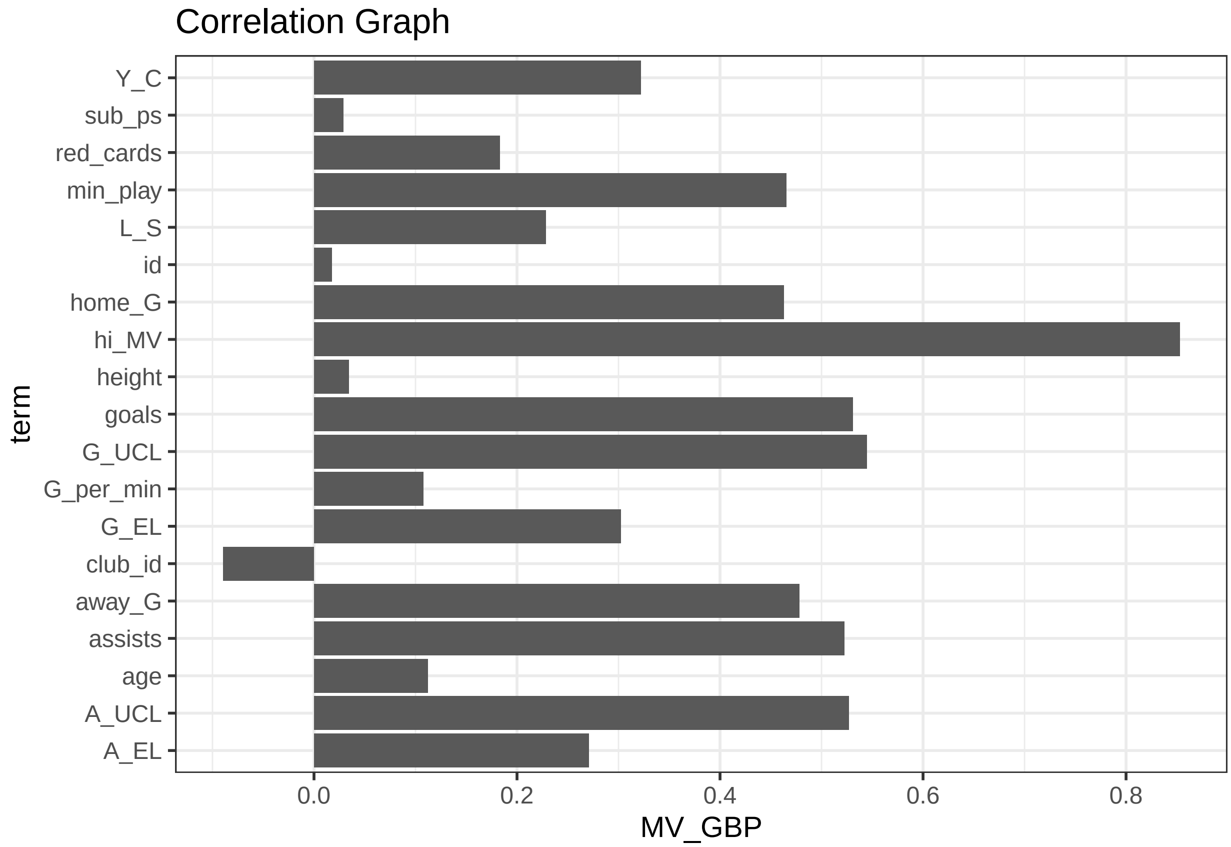
```
market_value1$sub_ps <- as.numeric(factor(market_value1$sub_ps))- 1

dummy <- dummyVars(" ~ .", data=market_value1)

#performing one-hot encoding on data frame
final_df <- data.frame(predict(dummy, newdata=market_value1))

corr <- correlate(final_df)
```

```
##
## Correlation method: 'pearson'
## Missing treated using: 'pairwise.complete.obs'
```

```
# Extract the correlation
corr %>%
  focus(MV_GBP)
```

```
## # A tibble: 19 x 2
##    term       MV_GBP
##    <chr>       <dbl>
##  1 id         0.0177
##   2 L_S         0.228
##   3 club_id    -0.0895
##   4 sub_ps      0.0289
##   5 height      0.0342
##   6 hi_MV       0.853
##   7 age         0.112
##   8 goals       0.531
##   9 assists     0.523
##  10 home_G      0.463
##  11 away_G      0.478
##  12 min_play    0.465
##  13 Y_C         0.322
##  14 red_cards   0.183
##  15 G_UCL       0.545
##  16 G_EL        0.302
##  17 A_UCL       0.527
##  18 A_EL        0.271
##  19 G_per_min   0.108
```

```
corr %>%
  focus(MV_GBP) %>%
  mutate(rowname = reorder(term, MV_GBP)) %>%
  ggplot(aes(term, MV_GBP)) +
    geom_col() + coord_flip() +
  theme_bw()  + ggtitle("Correlation Graph")
```



```
round(cor(subset(market_value1, select=c(hi_MV,goals, assists, Y_C, home_G,away_G, min_play, G_UCL, A_U
```

```
##          hi_MV goals assists  Y_C home_G away_G min_play G_UCL A_UCL A_EL
## hi_MV     1.00  0.71    0.70 0.45   0.63   0.64     0.62  0.71  0.71 0.36
## goals     0.71  1.00    0.81 0.69   0.86   0.85     0.87  0.69  0.58 0.45
## assists   0.70  0.81    1.00 0.69   0.88   0.88     0.89  0.59  0.66 0.53
## Y_C       0.45  0.69    0.69 1.00   0.80   0.79     0.82  0.30  0.34 0.41
## home_G    0.63  0.86    0.88 0.80   1.00   0.99     0.97  0.47  0.49 0.52
## away_G    0.64  0.85    0.88 0.79   0.99   1.00     0.97  0.47  0.48 0.51
## min_play  0.62  0.87    0.89 0.82   0.97   0.97     1.00  0.47  0.47 0.50
## G_UCL     0.71  0.69    0.59 0.30   0.47   0.47     0.47  1.00  0.81 0.21
## A_UCL     0.71  0.58    0.66 0.34   0.49   0.48     0.47  0.81  1.00 0.23
## A_EL      0.36  0.45    0.53 0.41   0.52   0.51     0.50  0.21  0.23 1.00
## MV_GBP    0.85  0.53    0.52 0.32   0.46   0.48     0.47  0.54  0.53 0.27
##          MV_GBP
## hi_MV      0.85
## goals      0.53
## assists    0.52
## Y_C        0.32
## home_G     0.46
## away_G     0.48
## min_play   0.47
## G_UCL      0.54
## A_UCL      0.53
## A_EL       0.27
## MV_GBP     1.00
```

From the correlation diagram and chart, it appears that data related to **hi_MV (Highest market value in GBP), goals, assists, Y_C (Yellow Cards_, home_G (Home goal), away_G (Away goal), min_play (Minutes played), G_UCL (Goals in UCL), A_UCL (Assists in UCL)**are the main correlated factor to estimate a player's financial worth. So, these nine features are being considered for the rest of the process.

4. Metric Selection

Here, three metrics have been chosen for the model accuracy.

4.1 R-Squared

R-Squared, also referred as the Coefficient of Determination, is a value ranging from 0 to 1 that reflects how well the regression line fits the data. R-Squared can be interpreted as the proportion of variation in the dependent variable explained by the model. The model will predict the dependent variable better if R-Squared is near to one or one hundred percent [3].

$$\begin{aligned} R^2 &= 1 - \frac{\text{sum squared regression (SSR)}}{\text{total sum of squares (SST)}}, \\ &= 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}. \end{aligned}$$

4.2 Adjusted R-squared

Adjusted R-squared is a variant of R-squared that accounts for the number of predictors in the model. When the new term improves the model more than can be expected by coincidence, the adjusted R-squared rises. It drops when a predictor improves the model by less than projected. The adjusted R-squared is normally positive rather than negative. It is never greater than R-squared.

$$Adjusted\ R^2 = \{1 - [\frac{(1-R^2)(n-1)}{(n-k-1)}]\}$$

Here, n signifies the number of data points in the dataset, k represents the number of independent variables, and R signifies the R-squared values of the model [4].

4.3 F-Test

The F-Test of overall significance in regression is used to test if a linear regression model matches a dataset better than a model with no predictor variables.

The F-Test of overall significance supports the following two hypotheses:

Null hypothesis (H0) : The model with no predictor variables (also referred to as an intercept-only model) fits the data and the regression model.

Alternative hypothesis (HA) : The regression model outperforms the intercept-only model in terms of data fit.

When a regression model has been fit to a dataset,a regression table will be created containing the F-statistic and the matching p-value for that F-statistic.

If the p-value is smaller than the significance threshold which has been chosen (***typical values include .01, .05, and .10***), it can be inferred that the regression model fits the data better than the intercept-only model.

- **If the p-value associated with the F-statistic is >= 0.05:** Then there is no relationship between ANY of the independent variables and Y
- **If the p-value associated with the F-statistic < 0.05:** Then, AT LEAST 1 independent variable is related to Y [1] [6]^.

5. Training and testing data sets

70% of the data will be used to train the model and the rest **30%** of the data will be utilized to evaluate its performance.

```
n_train <- round(0.7 * nrow(market_value1))
train_indices <- sample(1:nrow(market_value1), n_train)
df_train <- market_value1[train_indices, ]
df_test <- market_value1[-train_indices, ]
```

[1] 5

6. Linear Regression Model

Here, it can be seen that only 9 features has been considered as they are more correlated to market value.

```
lm.fit <- lm(MV_GBP ~ Y_C + goals + G_UCL+ away_G + home_G + assists + A_UCL+ min_play+hi_MV, data = df_

round(summary(lm.fit)$coefficients, 2)
```

```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  125632.17  100574.15    1.25     0.21
## Y_C           -8581.74   13510.53   -0.64     0.53
## goals        -32410.21    8355.77   -3.88     0.00
## G_UCL        -49316.98   64112.19   -0.77     0.44
## away_G        48580.54    5464.53    8.89     0.00
## home_G       -44218.80    4583.87   -9.65     0.00
## assists        5484.30   17050.65    0.32     0.75
## A_UCL       -777696.40  108435.83   -7.17     0.00
## min_play        105.04      83.45    1.26     0.21
## hi_MV             0.63       0.01   60.37     0.00
```

```
cat('R-Squared:', round(summary(lm.fit)$r.sq, 2),'   ', 'Adjusted R^2:', round(summary(lm.fit)$adj.r.sq
```

```
## R-Squared: 0.73     Adjusted R^2: 0.73     F-Test: 906.08 9 2943
```

The Yellow Card (Y_C) of the players is the sole variable that is not strongly connected to Charges. This variable will be removed in order to enhance the model.

```
lm.fit <- lm(MV_GBP ~  goals + G_UCL+ away_G + home_G + assists + A_UCL+ min_play+hi_MV, data = df_trai

round(summary(lm.fit)$coefficients, 2)
```

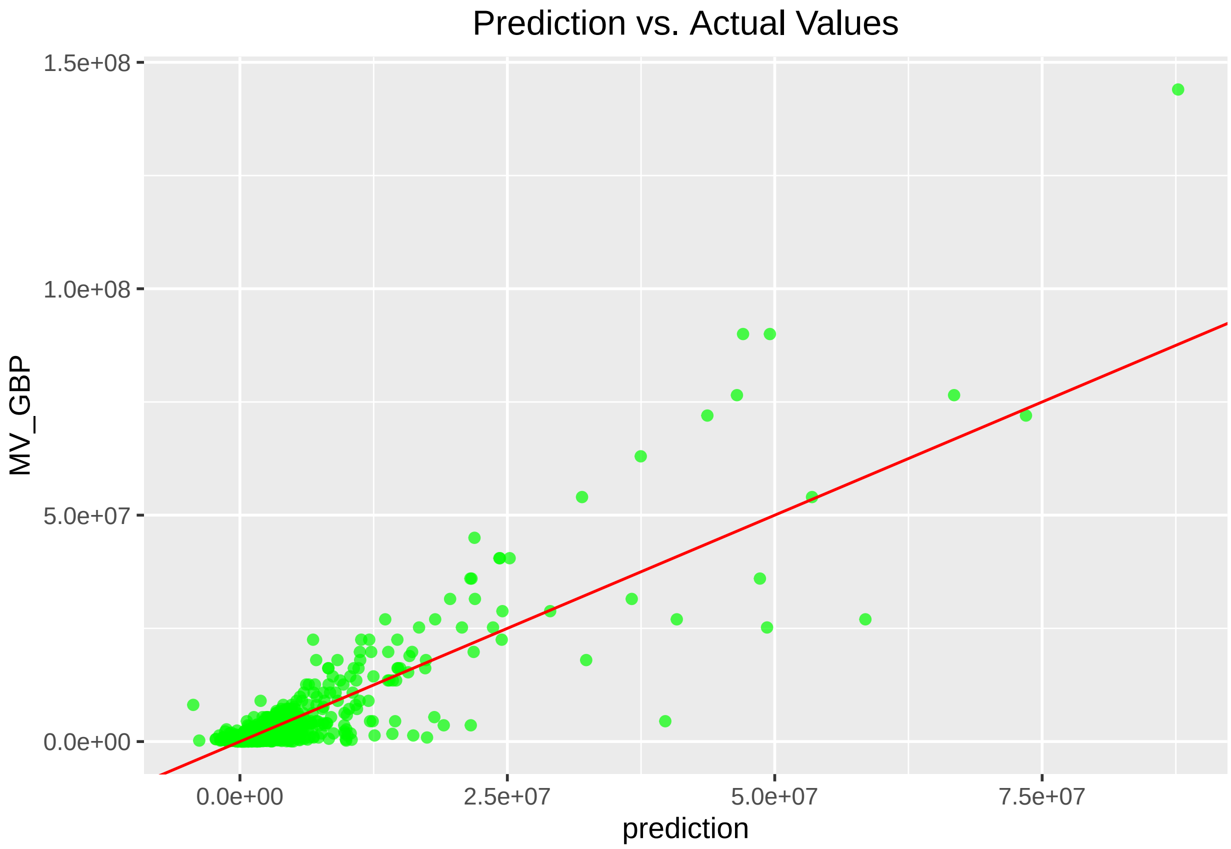
```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  119955.69  100166.17    1.20     0.23
## goals        -32507.27    8353.52   -3.89     0.00
## G_UCL        -44376.49   63632.19   -0.70     0.49
## away_G        48818.14    5451.16    8.96     0.00
## home_G       -44395.98    4574.91   -9.70     0.00
## assists        6410.80   16986.42    0.38     0.71
## A_UCL       -786244.01  107586.70   -7.31     0.00
## min_play         88.57      79.31    1.12     0.26
## hi_MV             0.63       0.01   60.43     0.00
```

```
cat('R-Squared:', round(summary(lm.fit)$r.sq, 2),'Adjusted R^2:', round(summary(lm.fit)$adj.r.sq, 2),'
```

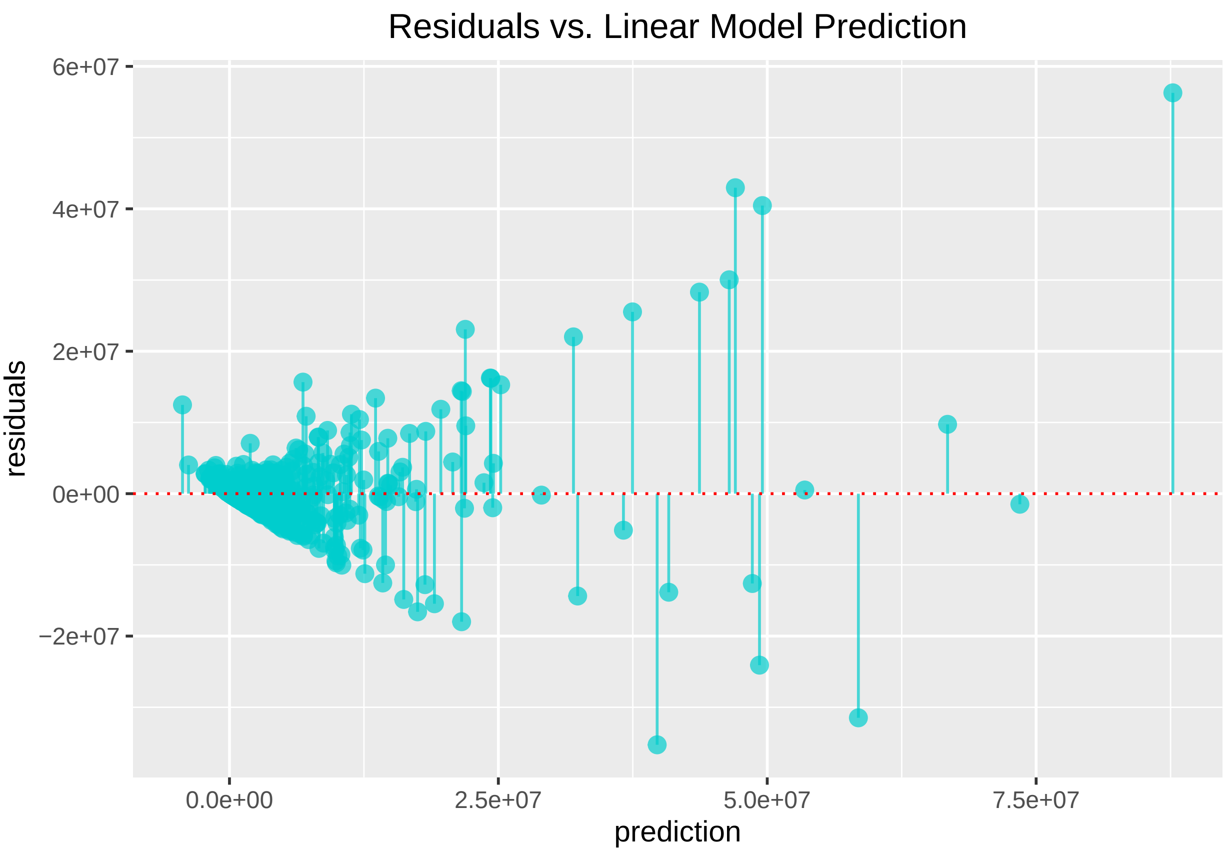
```
## R-Squared: 0.73 Adjusted R^2: 0.73     F-Test: 1019.49 8 2944
```

After dropping the Yellow Card (Y_C), the Adjusted R^2 stayed the same while the F-Statistic greatly increased. The model is now simpler.

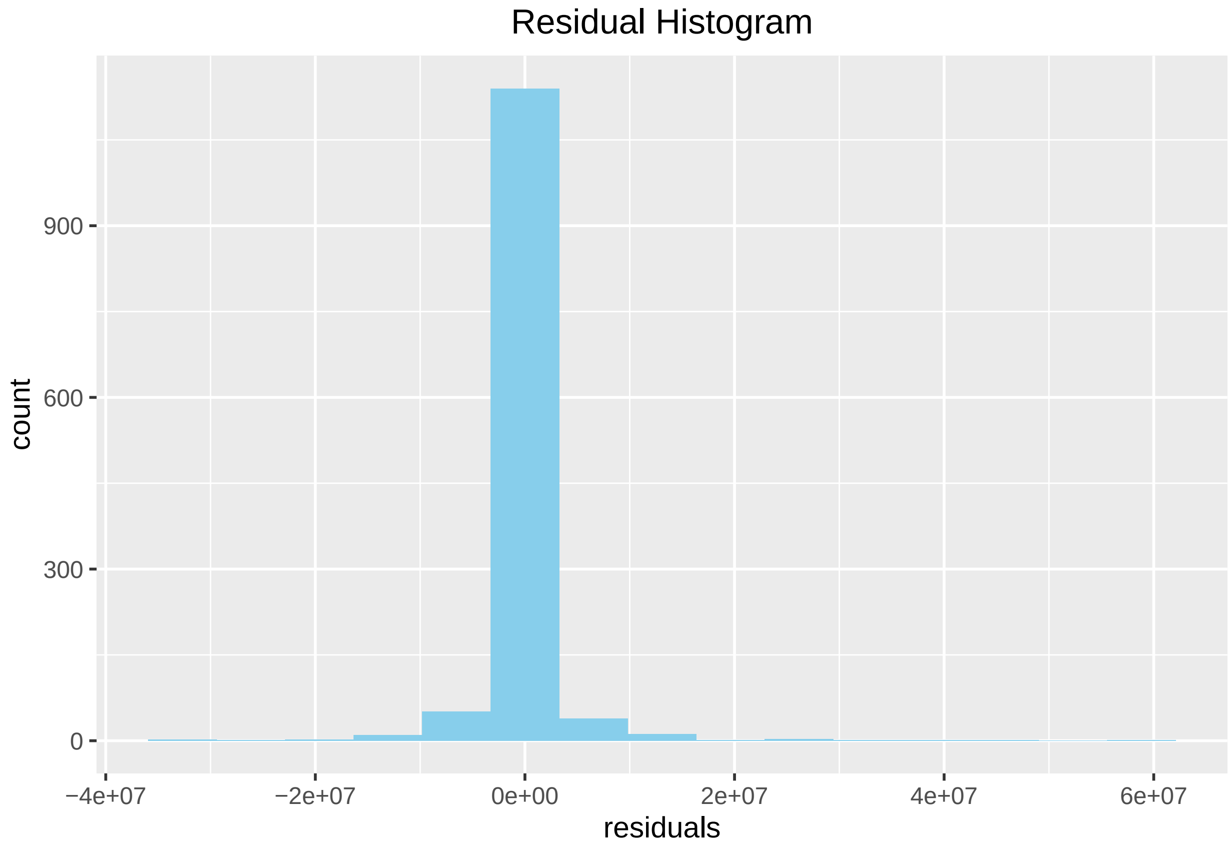
```r
df_test$prediction <- predict(lm.fit, newdata = df_test)
ggplot(df_test, aes(x = prediction, y = MV_GBP)) +
  geom_point(color = "green", alpha = 0.7) +
  geom_abline(color = "red") +
  ggtitle("Prediction vs. Actual Values")+
  theme(plot.title = element_text(hjust = 0.5))
```



```r
df_test$residuals <- df_test$MV_GBP - df_test$prediction

ggplot(data = df_test, aes(x = prediction, y = residuals)) +
  geom_pointrange(aes(ymin = 0, ymax = residuals), color = "cyan3", alpha = 0.7) +
  geom_hline(yintercept = 0, linetype = 3, color = "red") +
  ggtitle("Residuals vs. Linear Model Prediction")+
  theme(plot.title = element_text(hjust = 0.5))
```



```
ggplot(df_test, aes(x = residuals)) +
  geom_histogram(bins = 15, fill = "skyblue") +
  ggtitle("Residual Histogram")+
  theme(plot.title = element_text(hjust = 0.5))
```



The residuals are heavily concentrated around 0, which is a good sign the model is accurate.

7. Conclusion

Our goal is to predict player's transfer fee as accurately as possible. minimize the false negatives. So, three metrics have been selected to improve the model. These are **R-Squared, Adjusted R-Squared and F1 score.** By dropping 1 features, it is found that the F1-score has been improved while the rest two metrics remained same.

Overall this data set shows that players market value mainly depends on goals (in Overall and high ranked competition), assists (same as goals), playing times and highest market value.

8. Reference

[1] Ezzeddine, M. (2020). *Pricing football transfers : determinants, inflation, sustainability, and market impact : finance, economics, and machine learning approaches* [Doctoral Dissertation]. https://tel.archives-ouvertes.fr/tel-03171642

[2] UEFA.com. (2021, October 18). *Champions League Fantasy Football price changes – who has risen in value?* UEFA.com. https://www.uefa.com/uefachampionsleague/news/026e-136e4fd0c1e7-7ed92dcff837-1000--fantasy-price-changes/

[3] Investopedia. (2019). *What is the Difference Between R-Squared and Adjusted R-Squared?* Investopedia. https://www.investopedia.com/ask/answers/012615/whats-difference-between-rsquared-and-adjusted-rsquared.asp

[4] aniruddha. (2020, July 7). *Difference Between R-Squared and Adjusted R-Squared*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/07/difference-between-r-squared-and-adjusted-r-squared/#h2_2

[5] Choueiry, G. (2020, March). *Understand the F-statistic in Linear Regression – Quantifying Health*. Quantifying Health. https://quantifyinghealth.com/f-statistic-in-linear-regression/

[6] Zach. (2019, March 26). *A Simple Guide to Understanding the F-Test of Overall Significance in Regression*. Statology. https://www.statology.org/a-simple-guide-to-understanding-the-f-test-of-overall-significance-in-regression/