Final Project

2023-12-07

Final Project: Analysis of Soccer Player Market Values

Author: Emiliano Colin-Diaz

Introduction

Soccer is the sport that generates the most money worldwide. With the amount of revenue generated comes huge transfer fees. In todays market it is not uncommon to see players being sold for 100 million dollars. A fee that a decade ago would have been record breaking. In the summer transfer window of 2023 FIFA reported that a total of 7.36 billion US dollars were spent worldwide on soccer players (https://theathletic.com/4844507/2023/09/08/summer-transfer-window-2023-record/). Out of these 7.36 billion, 2.9 billion US dollars came from England. This comes to no surprise to fans of the sport as the English Premier League is the largest league in the world and last year reported a total revenue of 6.96 billion US dollars across all 20 teams (https://www.reuters.com/sports/soccer/premier-league-clubs-post-record-revenues-europe-recovers-covid-19-impact-2023-06-14/). This rise of expenditure has led to many debates regarding the value of a player and whether a club has overspent on a player or not. *The purpose of this project is to analyze the different factors that determine player market value and to try to model these relationships between various factors and market value.*

The data set I'm using is from Kaggle and it consists of data from Transfermarkt. Most analyses on Kaggle using this data set pertain to a certain soccer team, to scouting players, or just an explanatory data analysis in which someone tries to find interesting trends. Transfermarkt is the most popular soccer website for viewing player valuations and other information (https://www.nytimes.com/2021/08/12/sports/soccer/soccer-football-transfermarkt.html). This data set consists of various csv files that each contain relevant information pertaining to player statistics and valuations The three I will be using in this are

players - contains data about all players in Transfermarkt's database. Relevant columns from this table are name, country of citizenship, date of birth, position, sub position, foot, height, market value, highest market value, and current club. All of these contain information that could help in the prediction of player market values.

player_valuations - contains data about all player valuations ever entered in Transfermarkt's database. The relevant column in this table is "last season" as using it I can filter out only the players that have played in 2023, therefore filtering out older players as player values used to be lower in the past.

appearances - contains data about every player appearance ever recorded in Transfermarkt's database. This will be useful as I will extract goals and assists statistics from it.

Additionally, I imported a dataset from FIFA that contains the most recent nation rankings. This table only consists of two main columns, rank and country name.

For this project I will begin with a simple linear regression model in regards to a players nationality and market value. I will then attempt to make the most efficient multiple regression model using model selection techniques. Finally, I will conduct a nested model comparison with a model that contains additional factors besides nationality

Initial Data Wrangling

After importing the CSV files using read.csv() I then created the *player_values* data frame which is the data set that I will be using for all future visualizations and models. I filtered it so that *player_values* only has players that have played in an English team within the last year and that have a valid integer for their market value. I then selected the following variables:

player_id - primary id for players.csv and reference key in appearances.csv and player_valuations.csv name country_of_citizenship - as players may have multiple citizenships this column shows the country they represent internationally date_of_birth sub_position - the specific name of the position they play position - the category of their position, for example, defence or attack foot - their stronger foot height_in_cm market_value_in_eur - their most recent market valuation in euros highest_market_value_in_eur current_club_name - the team they play for

Additionally, I created a goals, an assists, and a total goals + assists column using the data from the *appearances* data. I made sure to filter out so only goals scored in the last season are included. I also filtered out all missing values and replaced NA for 0. Then I merged them by matching the player_id from both tables. Finally, I removed all rows that did not have left or right as their stronger foot.

Part 1: Simple Linear Regression Using player nationalities

Intro

As player market values become discussed more and more among the footballing world, there have been many debates regarding whether a players country influences their value. With many claiming that English players for example are overhyped and overpriced as players from countries with worse soccer team such as Morocco or Mexico might be undervalued. Therefore this initial analysis will be a linear regression model to assess the impact of a players nationality on their market value. For this analysis I will use a country's rank. FIFA determines a nations rank based on their match results and these rankings can be used to guage how good a player's nation is at soccer. It is important to note that as nation rank increases then the nation gets worse as the best team is ranked at 1. Here are the null and alternative hypotheses

$H0 : \beta_1 = 0; HA : \beta_1 < 0$

In words: the null hypothesis is that there is no relationship between a players nation rank and their market value, while the alternative hypothesis is that as a player's nation rank increases then their market value decreases.

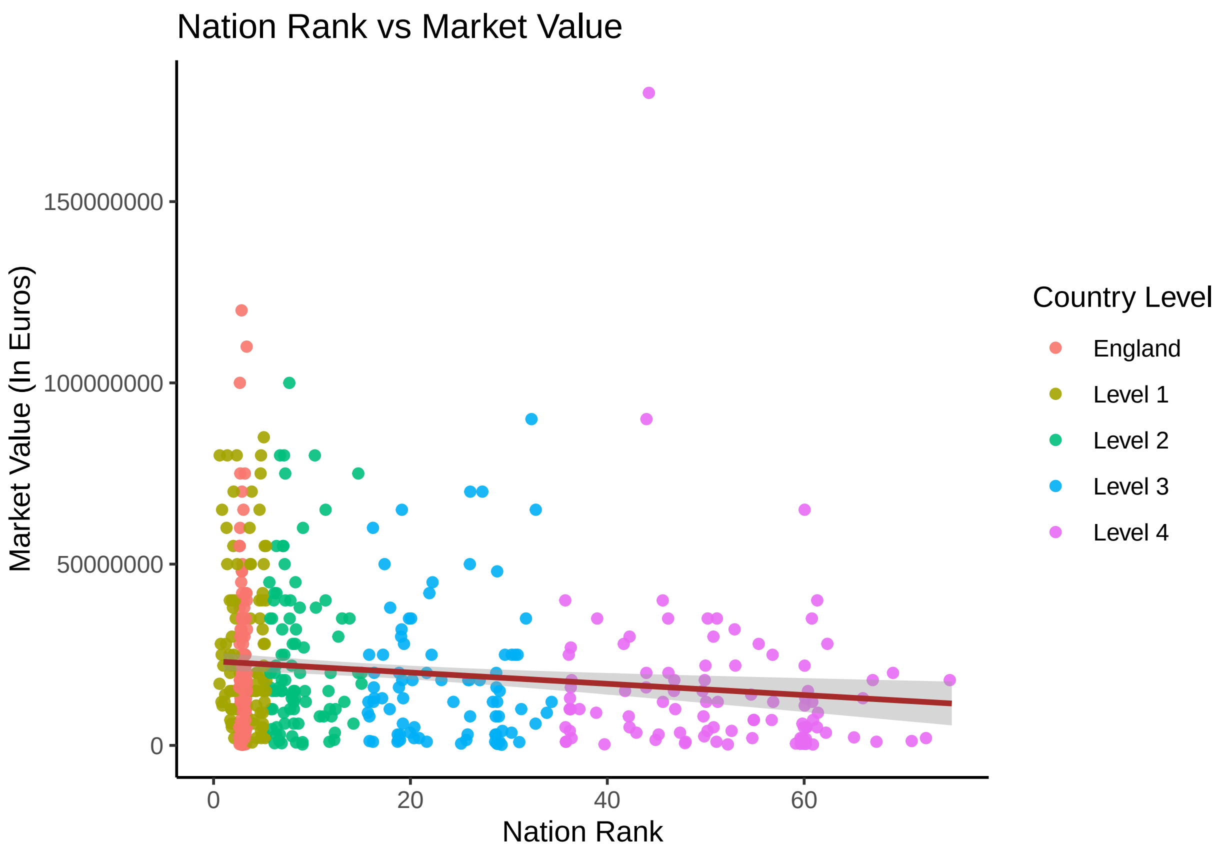
Data Wrangling

I began the data wrangling for this analysis by importing the FIFA nation rankings table. I fixed some of the nation names to match the names in the *player_values* data. I then merged with *player_values* and created a new column called Rank. I also created another column called country_level in which each nation rank corresponds to a different level. This is to help with visualizations.

Data Visualizations

I began by plotting out the nation rank as the independent variable and market value as the dependent variable. From the graph there is a visible relationship between nation rank and market value as the line of best fit seems to have a negative slope. Something else interesting to not is the outlier point around rank 45 and over 150,000,000 market value. Despite this maybe having an effect on the data, it should not be removed as it corresponds to Haaland, the best soccer player in the world currently and therefore despite being Norwegian it is no mistake that he is valued so highly.

```r
ggplot(player_values, aes(x = Rank, y = market_value_in_eur, color= country_level)) +
  geom_jitter(alpha = 0.9) +
  geom_smooth(method='lm', formula= y ~ x, color= 'brown') +
  theme_classic() +
  xlab("Nation Rank") +
  ylab("Market Value (In Euros)") +
  ggtitle("Nation Rank vs Market Value")  +
  xlim(c(0,75)) +
  labs(color = "Country Level")
```

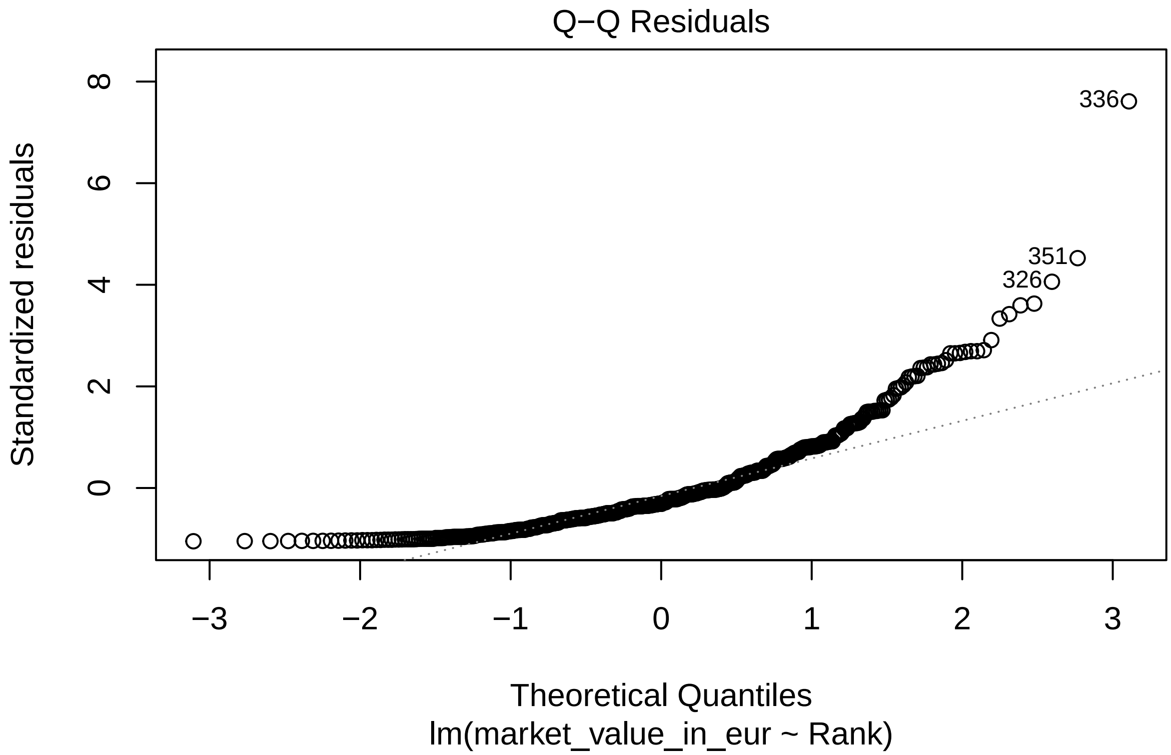


Checking Normality

I began by creating the linear regression model with nation rank as the independent variable and market value as the dependent variable. I then plotted the QQ-plot which showed that the data was not normally distributed.

```
lm_rank <- lm(market_value_in_eur ~ Rank, data= player_values)

plot(lm_rank, which= 2)
```



Transforming Data

As the data was not normal I took the log transformation of the response variable $\hat{y}$, in this case the market value. After this I called the summary() command and found the p-value of the model to be 0.0005727 which is indeed below the significance level of 0.05 meaning that it is statistically significant. Also I found that the β_1 value is -0.009711. Therefore we can accept the alternative hypothesis and market value does decrease as nation rank increases.

```r
lm_rank2 <- lm(log(market_value_in_eur) ~ Rank, data= player_values)
```

Linear regression equation =

$\hat{Y} = \beta_0 + \beta_1 * NationRank$

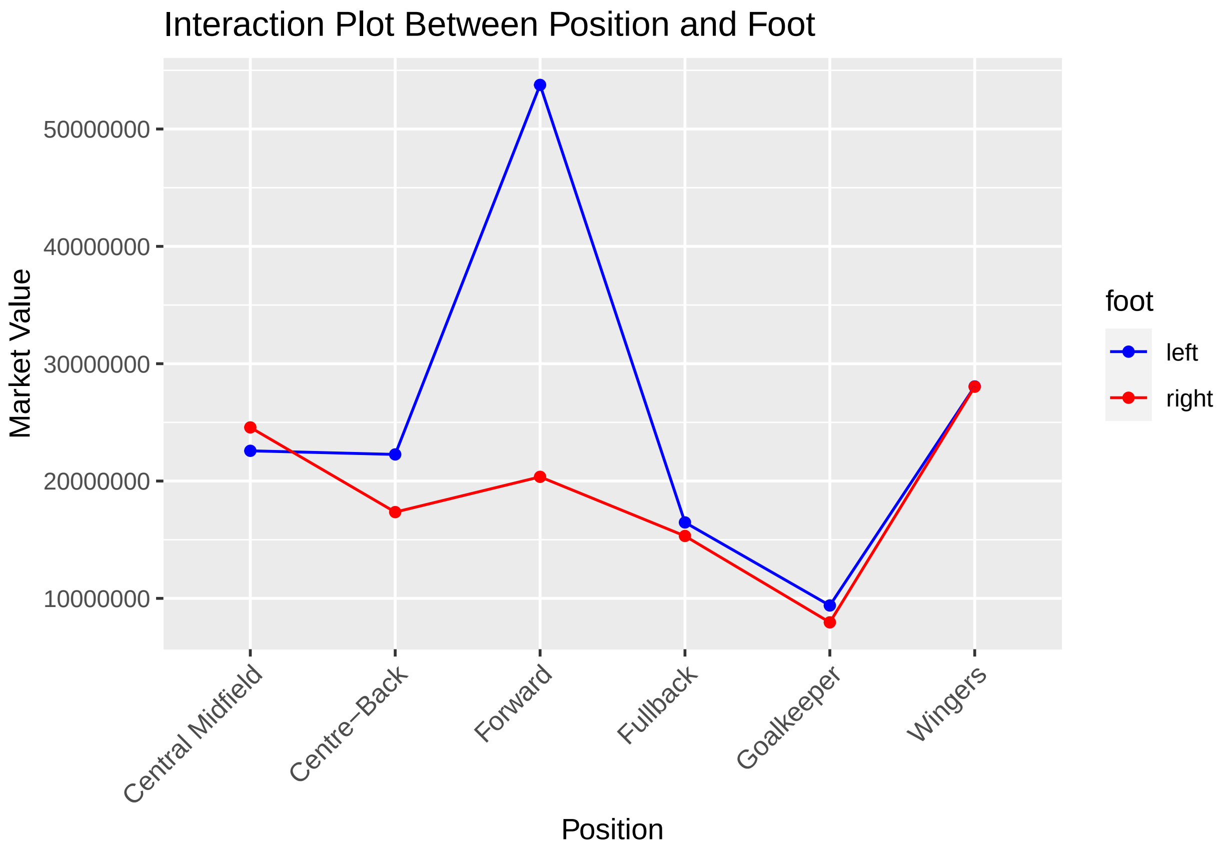
With Y hat being the estimated player market value.

Part 2: Model Selection for Multiple Regression Model

Checking for Interaction Effects

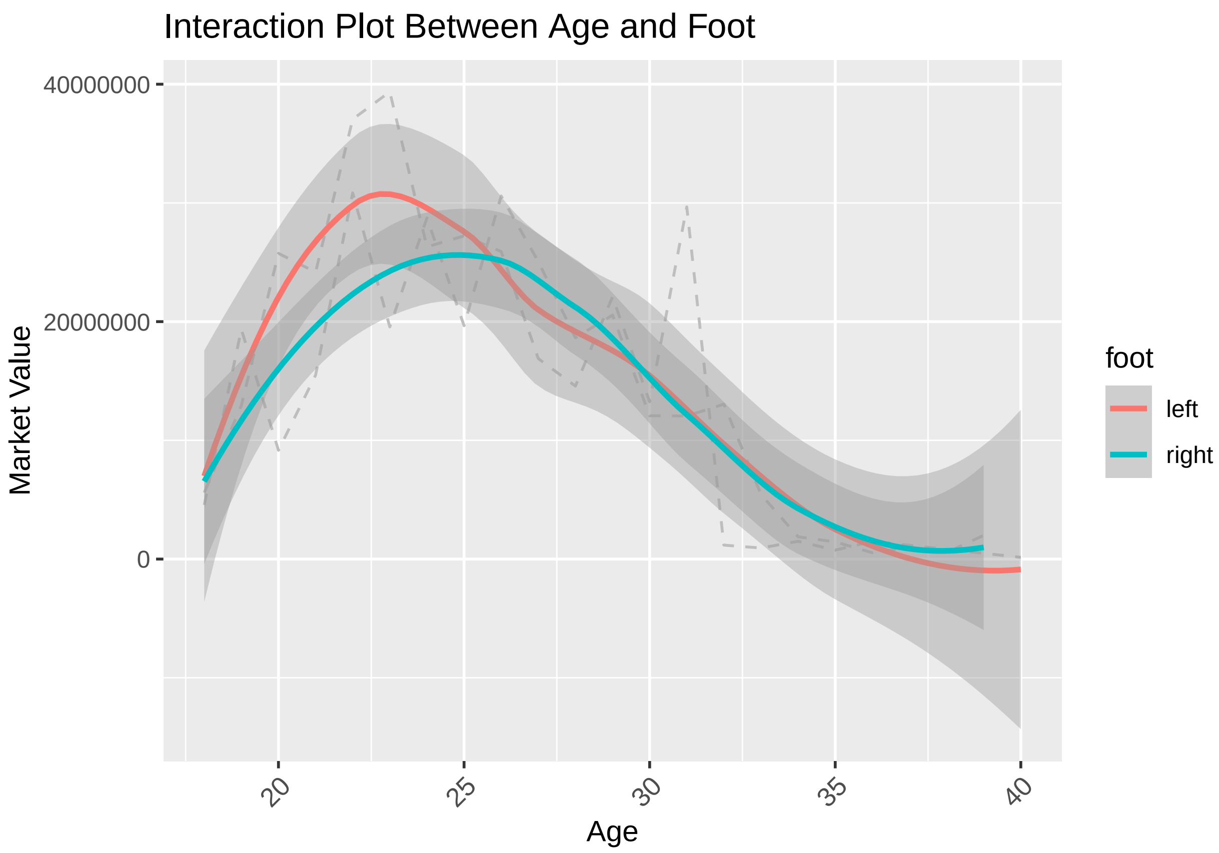
First I plotted an interaction plot between position and foot with market value as the y value. This initial graph demonstrates that there is likely not an interaction between position and foot as both lines seem to follow a similar trend. The only notable difference being the sharp spike for the left foot, forward which can be explained by Haaland, the same outlier in the previous linear regression model. It does show that there is likely a main effect from position however as both lines change in market value depending on the position.

```
player_values |>
  group_by(foot, sub_position) |>
  summarize(mean_market_value = mean(market_value_in_eur)) |>
  ggplot(aes(sub_position, mean_market_value, color = foot)) +
  geom_point() +
  geom_line(aes(group = foot), linetype = "solid") +
  ylab("Market Value") +
  xlab("Position") +
  ggtitle("Interaction Plot Between Position and Foot") +
  theme(axis.text.x = element_text(size = 10, angle = 45, hjust = 1)) +
  scale_color_manual(values = c("left" = "blue", "right" = "red"))
```



I then conducted a second isualizations to try to gather insight into the relationship between a players age and their strong foot. This seemed to repeat a similar result to the last graph with no clear interaction effect present although there does seem to be a main effect for age. At first this graph was hard to decipher however using the geom_smooth() line helped make the relationships easy to see. Around the age of 23 there is a spike for left foot but this again can be explained by Haaland and is not a sign of an interaction effect.

```
player_values |>
  group_by(foot, age) |>
  summarize(mean_market_value = mean(market_value_in_eur)) |>
  ggplot(aes(age, mean_market_value, color = foot)) +
  geom_line(aes(group = foot), linetype = "dashed", color= 'gray') +
  ylab("Market Value") +
  xlab("Age") +
  ggtitle("Interaction Plot Between Age and Foot") +
  theme(axis.text.x = element_text(size = 10, angle = 45, hjust = 1)) +
  geom_smooth()
```



To finish analyzing possible interaction effects I decided to run a factorial ANOVA with all possible variables. Here are the main takeaways. Firstly, 5 of the variables had a p-value less than 0.05 and therefore are statistically significant with foot being the only factor that is not statistically significant. The only interaction effects seen in this plot involved total goals and assists, however as explained earlier on these results are likely because of Haaland who is an outlier due to his large amount of goals, therefore the p-values are most likely not statistically significant (full summary in appendix)

```
lm_all <- lm(log(market_value_in_eur) ~ sub_position * age * foot * Rank * height_in_cm * total_ga, data
```

Cross-Validation of different models

As I conducted the previous tests I have concluded the best model in terms of which variables should have significant interaction and main effects. However, I want to make sure that I am not over fitting with my model. Therefore I will be using cross-validation to test different models with varying number of predictors to then calculate which one has the lowers MSPE.

I began by splitting up the player_values data into training and testing data with a 70-30 split.

```
total_num_points <- dim(player_values)[1]
num_training_points <- floor(0.7 * nrow(player_values))

training_data <- player_values[1:num_training_points, ]
test_data <- player_values[(num_training_points + 1):total_num_points, ]
```

I then fit the three different models each with different predictors, I started with all 5 predictors and then removed them one by one according to the highest p-values.

```
lm_fit1 <- lm(log(market_value_in_eur) ~ sub_position + total_ga + height_in_cm + Rank + age, data= trai

lm_fit2 <- lm(log(market_value_in_eur) ~ sub_position + total_ga + Rank + age, data= training_data)

lm_fit3 <- lm(log(market_value_in_eur) ~ sub_position + Rank + age, data= training_data)
```

```
test_1 <- predict(lm_fit1, newdata = test_data)

test_2 <- predict(lm_fit2, newdata = test_data)

test_3 <- predict(lm_fit3, newdata = test_data)

MSPE_1 <- mean((test_data$market_value_in_eur - test_1)^2)

MSPE_2 <- mean((test_data$market_value_in_eur - test_2)^2)

MSPE_3 <- mean((test_data$market_value_in_eur - test_3)^2)

which.min(c(MSPE_1, MSPE_2, MSPE_3))
```

```
## [1] 3
```

According to the MSPEs the most effective model is the third one with the predictors of position, nation rank, and age.

```
lm_fit <- lm(log(market_value_in_eur) ~ sub_position + Rank + age, data= player_values)

anova(lm_fit)
```

```
## Analysis of Variance Table
##
## Response: log(market_value_in_eur)
##                Df Sum Sq Mean Sq F value                Pr(>F)
## sub_position    5 159.77  31.954  19.444 < 0.00000000000000022 ***
## Rank            1  22.34  22.342  13.595             0.0002505 ***
## age             1  38.59  38.586  23.480           0.000001667 ***
## Residuals     522 857.85   1.643
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Nested Model Comparison

Now I will be conducting a nested model comparison to see if the model with the 5 predictors is a better fit than the nation rank linear regression model from part 1. I will use the formula $F = \frac{(SSRes_{Reduced} - SSRes_{Full})/q}{SSRes_{Full}/(n-k-1)}$

The null and alternative hypothesis are

$H0 : \beta_{position} = \beta_{age} = \beta_{nationrank} = 0$

HA : at least one of $\beta_{position}, \beta_{age}, \beta_{nationrank} \neq 0$

```
SSRes_reduced <- anova(lm_rank2)[2, "Sum Sq"]

SSRes_full <- anova(lm_fit)[4, "Sum Sq"]

n <- nrow(player_values)

numerator <- (SSRes_reduced - SSRes_full) / 2

denominator <- (SSRes_full) / (n - 3 - 1)

numerator/denominator
```

```
## [1] 60.30868
```

```
anova(lm_rank2, lm_fit)
```

```
## Analysis of Variance Table
##
## Model 1: log(market_value_in_eur) ~ Rank
## Model 2: log(market_value_in_eur) ~ sub_position + Rank + age
##   Res.Df     RSS Df Sum of Sq     F                Pr(>F)
## 1    528 1054.57
## 2    522  857.85  6    196.71 19.95 < 0.000000000000000022 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

I also ran an ANOVA table with the two models. Using the F-statistic and the p-value which is less than 0.05 then we can reject the null hypothesis. Meaning that we accept the alternative hypothesis and the full model is more accurate than the reduced one.

Conclusion

In conclusion, the linear regression model did prove that a player's nationality has an effect on their market value. It would be interesting to pose this model in other countries that have less foreign players and seeing if the relationship changes. However, it certainly is interesting that a player's country can contribute to their market value and perhaps this has to do with biases. Player values would be more accurate if there biases were removed altogether. As for the multiple regression model I found that sometimes too many predictors can cause overfitting, as was shown by the cross validation. Another notable takeaway was the lack of interaction effects, perhaps with access to more in-depth soccer statistics then we would see more interaction between variables. Finally, the nested model helped confirm that the final lm_fit model is the most accurate and that adding more predictors to the initial linear regression is beneficial All in all, through these analyses I learned more about the soccer player transfer market and the various factors that are taken into consideration when Tranfermarkt values players.

Reflection on Canvas

Appendix

Data Wrangling

Importing CSV Files

```
appearences <- read.csv("appearances.csv")

player_valuations <- read.csv("player_valuations.csv")

players <- read.csv("players.csv")
```

Creating player_values data frame

```
active_players <- player_valuations |>
                  filter(last_season == '2023')

player_values <- players |>
                    filter(current_club_domestic_competition_id == 'GB1') |>
                    filter(!is.na(market_value_in_eur)) |>
                    filter(player_id %in% active_players$player_id) |>
                    select(player_id, name, country_of_citizenship, date_of_birth, sub_position, positi
```

Creating a Rank column in player_values using a table from FIFA that contains the nation rankings. Also had to fix some country naming differences between the two data sets

```
nation_rankings <- read.csv("FIFA_rankings.csv")

nation_rankings <- nation_rankings |>
                    rename("country_of_citizenship" = "Nation")

nation_rankings <- nation_rankings|>
    mutate(country_of_citizenship = ifelse(row_number() == 60, "Ireland", country_of_citizenship)) |>
    mutate(country_of_citizenship = ifelse(row_number() == 12, "United States", country_of_citizenship)
    mutate(country_of_citizenship = ifelse(row_number() == 21, "Iran", country_of_citizenship)) |>
    mutate(country_of_citizenship = ifelse(row_number() == 23, "South, Korea", country_of_citizenship))
    mutate(country_of_citizenship = ifelse(row_number() == 39, "Czech Republic", country_of_citizenship
    mutate(country_of_citizenship = ifelse(row_number() == 50, "Cote d'Ivoire", country_of_citizenship)
    mutate(country_of_citizenship = ifelse(row_number() == 67, "DR Congo", country_of_citizenship)) |>
    mutate(country_of_citizenship = ifelse(row_number() == 66, "Bosnia-Herzegovina", country_of_citizens
    mutate(country_of_citizenship = ifelse(row_number() == 104, "New Zealand", country_of_citizenship))

player_values <- merge(player_values, nation_rankings, by = "country_of_citizenship", all.x = TRUE)
```

Creating a levels column based on the rank of a players nation

```
player_values <- player_values |>
  mutate(country_level = case_when(
           Rank %in% c(1, 2, 4, 5) ~ "Level 1",
           Rank == 3 ~ "England",
           Rank <= 15 ~ "Level 2",
           Rank <= 35 ~ "Level 3",
           Rank >= 36 ~ "Level 4"))
```

Creating goals column in player_values using appearances table

```
goals <- appearences |>
            filter(player_id %in% player_values$player_id) |>
            select(player_id, goals) |>
            group_by(player_id) |>
            summarize(total_goals = sum(goals))
```

```
player_values <- merge(player_values, goals, by = "player_id", all.x = TRUE)
```

Creating assists column in player_values using appearances table

```
assists <- appearences |>
              filter(player_id %in% player_values$player_id) |>
              select(player_id, assists) |>
              group_by(player_id) |>
              summarize(total_assists = sum(assists))
```

```
player_values <- merge(player_values, assists, by = "player_id", all.x = TRUE)
```

Creating total goals + assists colum

```
player_values <- player_values |>
            mutate(total_ga = total_goals + total_assists)
```

Filtering out NA values and players without a strong foot

```
layer_values <- player_values |>
  filter(foot %in% c("left", "right"))

player_values <- player_values |>
                  mutate(total_goals = ifelse(is.na(total_goals), 0, total_goals)) |>
                  mutate(total_assists = ifelse(is.na(total_assists), 0, total_assists))
```

I used an R package called lubridate and with the help of several online forums (such as this one https://stackoverflow.com/questions/70531616/how-can-i-convert-birth-date-to-age). I was able to mutate the date_of_birth column to create a new column called age.

```
player_values <- player_values |>
                    mutate(date_of_birth = as.Date(date_of_birth, format = "%Y-%m-%d"),
                      age = as.numeric(difftime(Sys.Date(), date_of_birth, units = "days")) %/% 365.25)
```

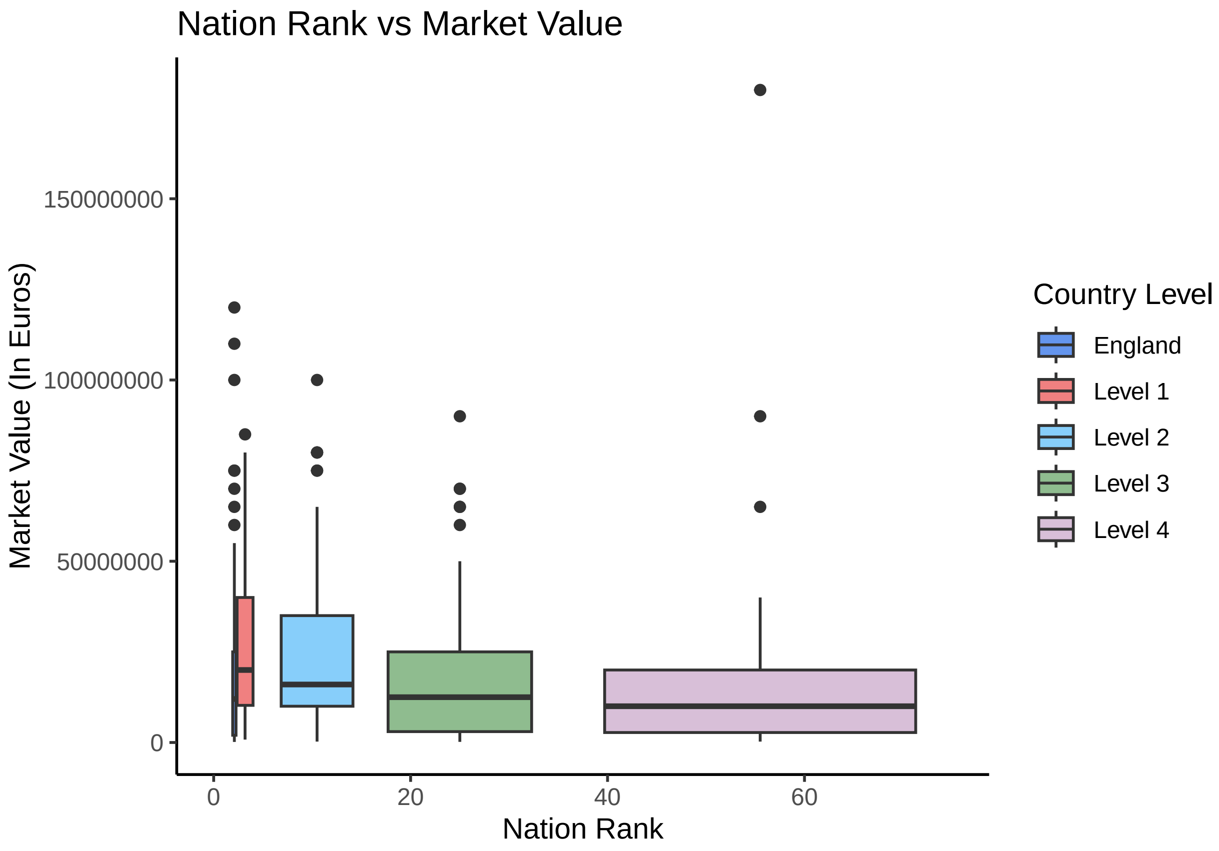
Changing sub_position to more encompassing categories

```
player_values <- player_values |>
          mutate(sub_position = case_when(
              sub_position %in% c("Central Midfield", "Defensive Midfield", "Attacking Midfield") ~ "Cen
              sub_position %in% c("Left-Back", "Right-Back") ~ "Fullback",
              sub_position %in% c("Left Midfield", "Right Midfield", "Left Winger", "Right Winger") ~ "W
              sub_position %in% c("Centre-Forward", "Second Striker", "Forward") ~ "Forward",
              sub_position == "Centre-Back" ~ "Centre-Back",
              sub_position == "Goalkeeper" ~ "Goalkeeper",
              TRUE ~ as.character(sub_position)
          ))
```

Appendix: Data Visualizations

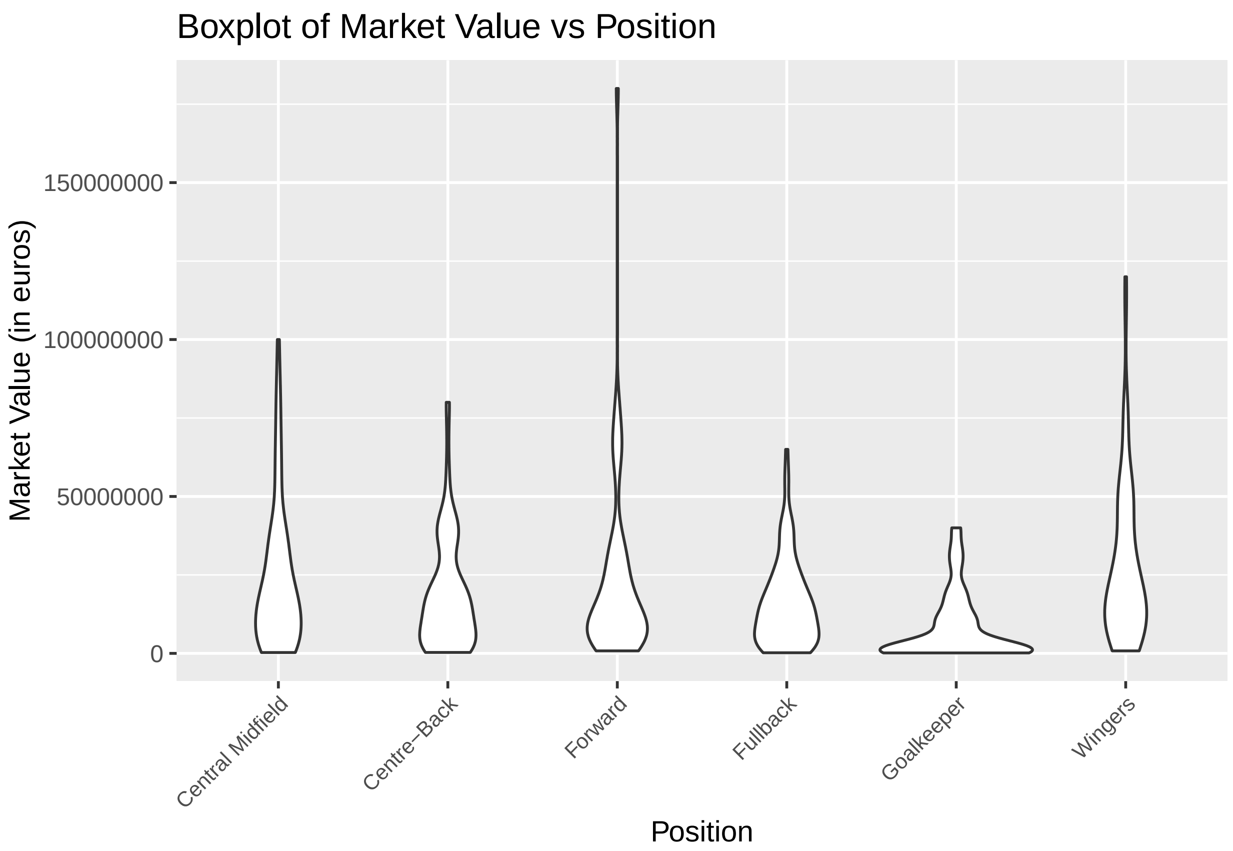
Boxplot of country levels with nation rank as x and market value as y

```
ggplot(aes(x = Rank, y = market_value_in_eur, fill = country_level), data = player_values) +
    geom_boxplot() +
    scale_fill_manual(values = c("cornflowerblue", "lightcoral","lightskyblue","darkseagreen","thistle")
    scale_color_manual(values = c("cornflowerblue", "lightcoral","lightskyblue","darkseagreen","thistle
    theme_classic() +
    xlab("Nation Rank") +
    ylab("Market Value (In Euros)") +
    ggtitle("Nation Rank vs Market Value") +
    xlim(c(0,75)) +
    labs(fill= 'Country Level')
```



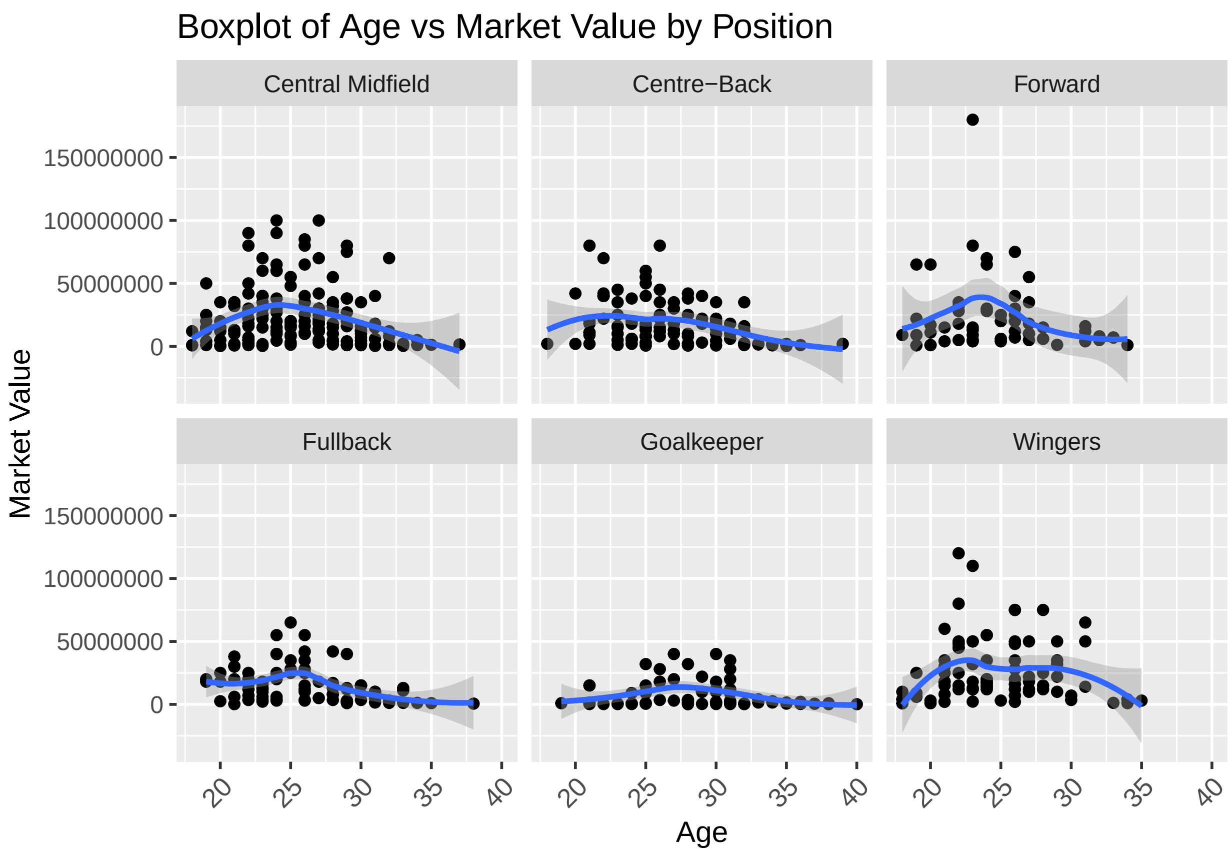
Comparison of the market values of different positions

```
ggplot(player_values, aes(x= sub_position, y= market_value_in_eur)) +
    geom_violin() +
    xlab("Position") +
    ylab("Market Value (in euros)") +
    ggtitle("Boxplot of Market Value vs Position") +
    theme(axis.text.x = element_text(size = 7.5, angle = 45, hjust = 1))
```



The relationship between age and market value depending on a player's position

```
ggplot(player_values, aes(x= age, y= market_value_in_eur)) +
    geom_point() +
    facet_wrap(~sub_position) +
    xlab("Age") +
    ylab("Market Value") +
    ggtitle("Boxplot of Age vs Market Value by Position") +
    theme(axis.text.x = element_text(size = 10, angle = 45, hjust = 1)) +
    geom_smooth()
```



Appendix: Model Selection Statistics

Summary of all variables to analyze possible interaction effects

```
summary(aov(log(market_value_in_eur) ~ sub_position * age * foot * Rank * height_in_cm * total_ga, data=
```

```
##                                                   Df Sum Sq Mean Sq F value
## sub_position                                       5  115.2   23.04  20.508
## age                                                1   52.5   52.50  46.728
## foot                                               3    8.3    2.76   2.460
## Rank                                               1   20.3   20.31  18.080
## height_in_cm                                       1    9.6    9.57   8.514
## total_ga                                           1  123.0  122.98 109.456
## sub_position:age                                   5    6.4    1.28   1.135
## sub_position:foot                                 12   17.4    1.45   1.292
## age:foot                                           3    4.7    1.56   1.390
## sub_position:Rank                                  5    2.3    0.47   0.417
## age:Rank                                           1    0.2    0.20   0.181
## foot:Rank                                          2    0.7    0.36   0.325
## sub_position:height_in_cm                          5    3.7    0.73   0.654
## age:height_in_cm                                   1    0.0    0.05   0.041
## foot:height_in_cm                                  3    1.9    0.64   0.571
## Rank:height_in_cm                                  1    0.1    0.13   0.116
## sub_position:total_ga                              5   62.5   12.49  11.118
## age:total_ga                                       1   59.9   59.91  53.322
## foot:total_ga                                      2    5.5    2.76   2.461
## Rank:total_ga                                      1    0.3    0.30   0.263
## height_in_cm:total_ga                              1    8.4    8.41   7.488
## sub_position:age:foot                              5    2.5    0.50   0.447
## sub_position:age:Rank                              5   12.0    2.39   2.127
## sub_position:foot:Rank                             5    7.9    1.57   1.399
## age:foot:Rank                                      1    0.0    0.00   0.000
## sub_position:age:height_in_cm                      5    2.1    0.42   0.370
## sub_position:foot:height_in_cm                     5    9.4    1.88   1.677
## age:foot:height_in_cm                              1    0.9    0.94   0.836
## sub_position:Rank:height_in_cm                     5   11.1    2.23   1.983
## age:Rank:height_in_cm                              1    0.5    0.50   0.444
## foot:Rank:height_in_cm                             1    0.2    0.20   0.174
## sub_position:age:total_ga                          5   11.3    2.26   2.014
## sub_position:foot:total_ga                         5    9.3    1.85   1.649
## age:foot:total_ga                                  1    1.0    0.99   0.882
## sub_position:Rank:total_ga                         5    7.8    1.57   1.395
## age:Rank:total_ga                                  1    1.5    1.45   1.291
## foot:Rank:total_ga                                 1    1.7    1.71   1.519
## sub_position:height_in_cm:total_ga                 5    3.8    0.76   0.677
## age:height_in_cm:total_ga                          1    0.0    0.02   0.014
## foot:height_in_cm:total_ga                         1    0.9    0.94   0.837
## Rank:height_in_cm:total_ga                         1    1.1    1.06   0.945
## sub_position:age:foot:Rank                         4    5.7    1.42   1.264
## sub_position:age:foot:height_in_cm                 3    1.7    0.57   0.506
## sub_position:age:Rank:height_in_cm                 5    2.8    0.56   0.501
## sub_position:foot:Rank:height_in_cm                3    1.3    0.44   0.390
## age:foot:Rank:height_in_cm                         1    0.9    0.87   0.778
## sub_position:age:foot:total_ga                     3    1.2    0.40   0.360
## sub_position:age:Rank:total_ga                     5    8.0    1.59   1.417
## sub_position:foot:Rank:total_ga                    3    1.5    0.51   0.451
## age:foot:Rank:total_ga                             1    0.0    0.00   0.000
## sub_position:age:height_in_cm:total_ga             5    5.9    1.17   1.044
## sub_position:foot:height_in_cm:total_ga            3    1.3    0.44   0.388
## age:foot:height_in_cm:total_ga                     1    0.0    0.04   0.035
## sub_position:Rank:height_in_cm:total_ga            5    2.1    0.41   0.367
## age:Rank:height_in_cm:total_ga                     1    0.0    0.02   0.016
## foot:Rank:height_in_cm:total_ga                    1    0.4    0.42   0.378
## sub_position:age:foot:Rank:height_in_cm            3    4.0    1.32   1.172
## sub_position:age:foot:Rank:total_ga                3    2.5    0.83   0.741
## sub_position:age:foot:height_in_cm:total_ga        3    1.5    0.50   0.441
## sub_position:age:Rank:height_in_cm:total_ga        5    5.7    1.14   1.014
## sub_position:foot:Rank:height_in_cm:total_ga       3    1.1    0.36   0.323
## age:foot:Rank:height_in_cm:total_ga                1    0.1    0.05   0.047
## sub_position:age:foot:Rank:height_in_cm:total_ga   3    1.6    0.53   0.470
## Residuals                                        345  387.6    1.12        
##                                                                 Pr(>F)    
## sub_position                                      < 0.0000000000000002 ***
## age                                                   0.00000000003711 ***
## foot                                                           0.06259 .  
## Rank                                                  0.00002730903601 ***
## height_in_cm                                                   0.00376 ** 
## total_ga                                          < 0.0000000000000002 ***
## sub_position:age                                               0.34142    
## sub_position:foot                                              0.22114    
## age:foot                                                      0.24570
## sub_position:Rank                                             0.83681
## age:Rank                                                      0.67072
## foot:Rank                                                     0.72288
## sub_position:height_in_cm                                     0.65879
## age:height_in_cm                                              0.83951
## foot:height_in_cm                                             0.63424
## Rank:height_in_cm                                             0.73357
## sub_position:total_ga                                0.00000000060357 ***
## age:total_ga                                         0.00000000000197 ***
## foot:total_ga                                                 0.08686 .
## Rank:total_ga                                                 0.60857
## height_in_cm:total_ga                                         0.00653 **
## sub_position:age:foot                                         0.81542
## sub_position:age:Rank                                         0.06176 .
## sub_position:foot:Rank                                        0.22415
## age:foot:Rank                                                 0.99568
## sub_position:age:height_in_cm                                 0.86912
## sub_position:foot:height_in_cm                                0.13948
## age:foot:height_in_cm                                         0.36111
## sub_position:Rank:height_in_cm                                0.08051 .
## age:Rank:height_in_cm                                         0.50546
## foot:Rank:height_in_cm                                        0.67706
## sub_position:age:total_ga                                     0.07617 .
## sub_position:foot:total_ga                                    0.14641
## age:foot:total_ga                                             0.34825
## sub_position:Rank:total_ga                                    0.22550
## age:Rank:total_ga                                             0.25674
## foot:Rank:total_ga                                            0.21868
## sub_position:height_in_cm:total_ga                            0.64136
## age:height_in_cm:total_ga                                     0.90677
## foot:height_in_cm:total_ga                                    0.36086
## Rank:height_in_cm:total_ga                                    0.33174
## sub_position:age:foot:Rank                                    0.28387
## sub_position:age:foot:height_in_cm                            0.67860
## sub_position:age:Rank:height_in_cm                            0.77537
## sub_position:foot:Rank:height_in_cm                           0.76013
## age:foot:Rank:height_in_cm                                    0.37835
## sub_position:age:foot:total_ga                                0.78185
## sub_position:age:Rank:total_ga                                0.21760
## sub_position:foot:Rank:total_ga                               0.71679
## age:foot:Rank:total_ga                                        0.99411
## sub_position:age:height_in_cm:total_ga                        0.39176
## sub_position:foot:height_in_cm:total_ga                       0.76185
## age:foot:height_in_cm:total_ga                                0.85249
## sub_position:Rank:height_in_cm:total_ga                       0.87095
## age:Rank:height_in_cm:total_ga                                0.89935
## foot:Rank:height_in_cm:total_ga                               0.53907
## sub_position:age:foot:Rank:height_in_cm                       0.32019
## sub_position:age:foot:Rank:total_ga                           0.52807
## sub_position:age:foot:height_in_cm:total_ga                   0.72396
## sub_position:age:Rank:height_in_cm:total_ga                   0.40899
## sub_position:foot:Rank:height_in_cm:total_ga                  0.80900
## age:foot:Rank:height_in_cm:total_ga                           0.82778
## sub_position:age:foot:Rank:height_in_cm:total_ga              0.70306
## Residuals
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 17 observations deleted due to missingness
```

References

https://www.kaggle.com/datasets/davidcariboo/player-scores/data

https://www.fifa.com/fifa-world-ranking/men

https://www.nytimes.com/2021/08/12/sports/soccer/soccer-football-transfermarkt.html