Football Transfer Market Data Analysis

Data Science Final Project - January 2024

Farzam Manafzadeh and Sepehr Mousavian

Introduction

This project is based on the 'Football Data from Transfermarkt Football (Soccer) data' on Kaggle, scraped from **Transfermark website**.

This project is done in two main parts. One part, the first part is related to the analysis of football data and drawing graphs and obtaining information based on the available data.

In the second part, it is about building machine learning models and building a comprehensive data set to predict the price of football players based on their performance in 2023.

The data came from: **https://www.kaggle.com/datasets/davidcariboo/player-scores** by **https://www.kaggle.com/davidcariboo**

These data are included with this shape. Some of them were used in the first part and some others were used in the second part.

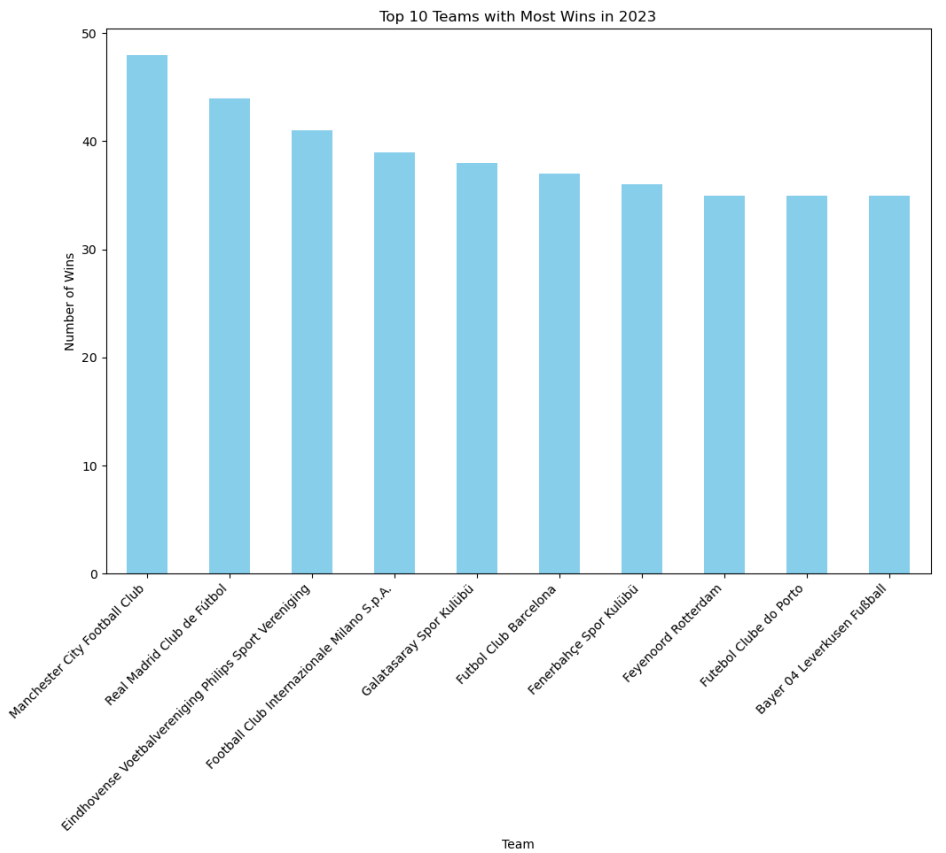
Data Frame	Rows	Columns
appearances_df	1524941	13
clubs_df	426	17
club_games_df	132390	11
competitions_df	43	10
games_df	66195	23
game_events_df	681952	10
game_lineups_df	2304615	9
players_df	30396	23
player_valuations_df	464869	5

1. Football data analysis

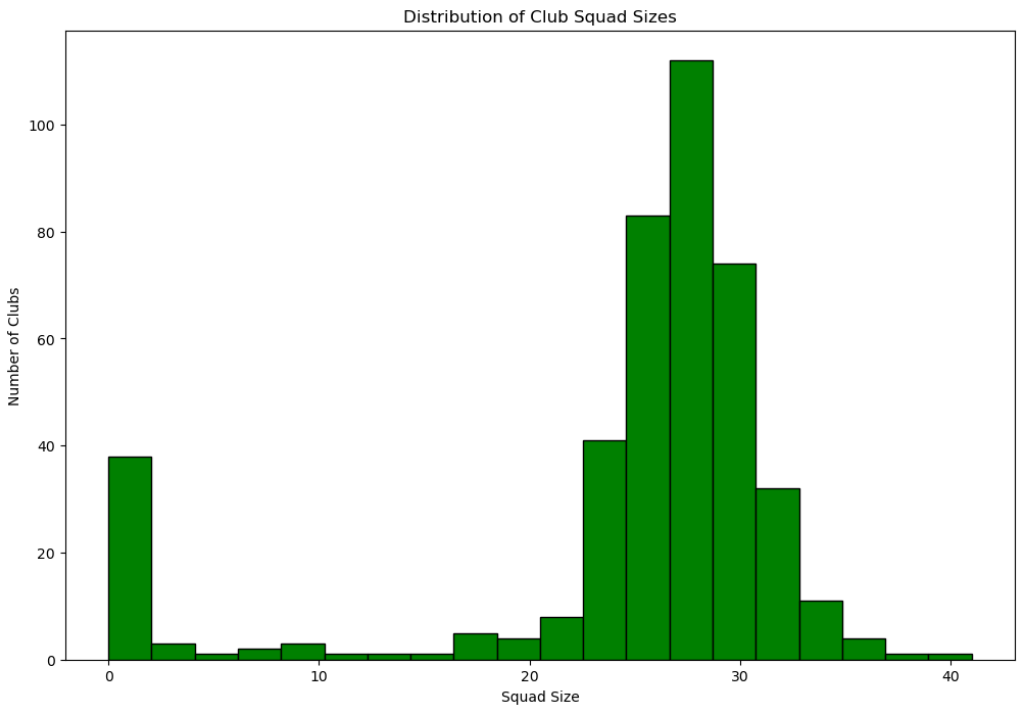
In this part of the project, some questions were asked and their answers were found in the data set, and relevant analyzes and graphs were drawn.

Which team has won more in the year 2023?

The team with the most wins in 2023 is Manchester City Football Club with 48.0 wins.



Visualizing Club Squad Sizes

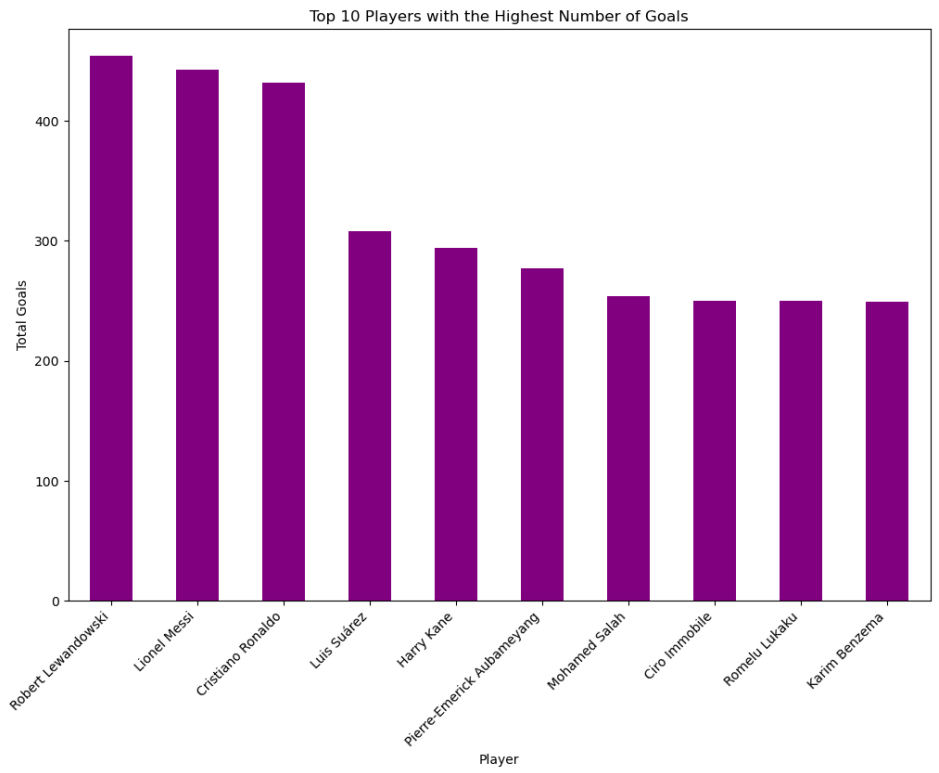


- According to the result of the plot, most teams have a squad size of 25 to 30.

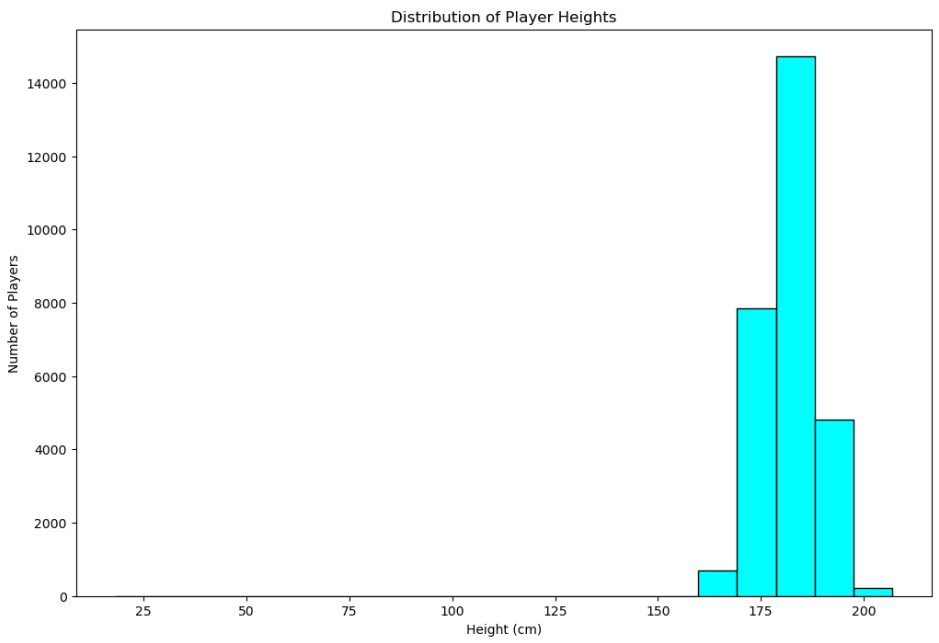
Which player has the highest number of goals in a Match?

The player with the highest number of goals is Arkadiusz Milik with 6 goals.

Highest number of goals in domestic leagues

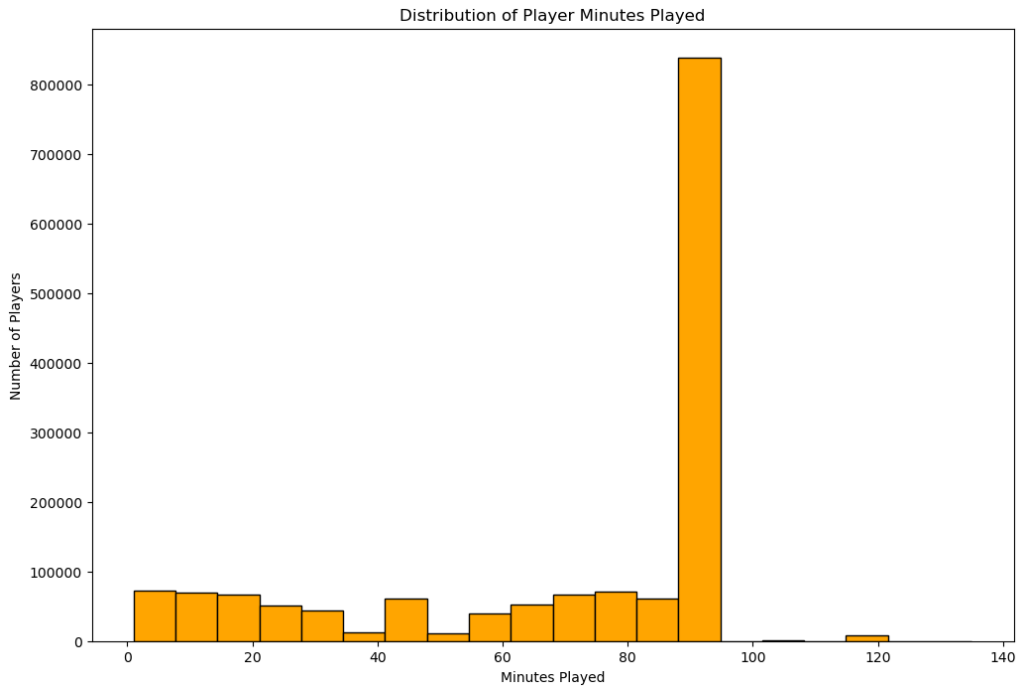


Player height distribution



- As can be seen, most players have the height between 175 and 185.

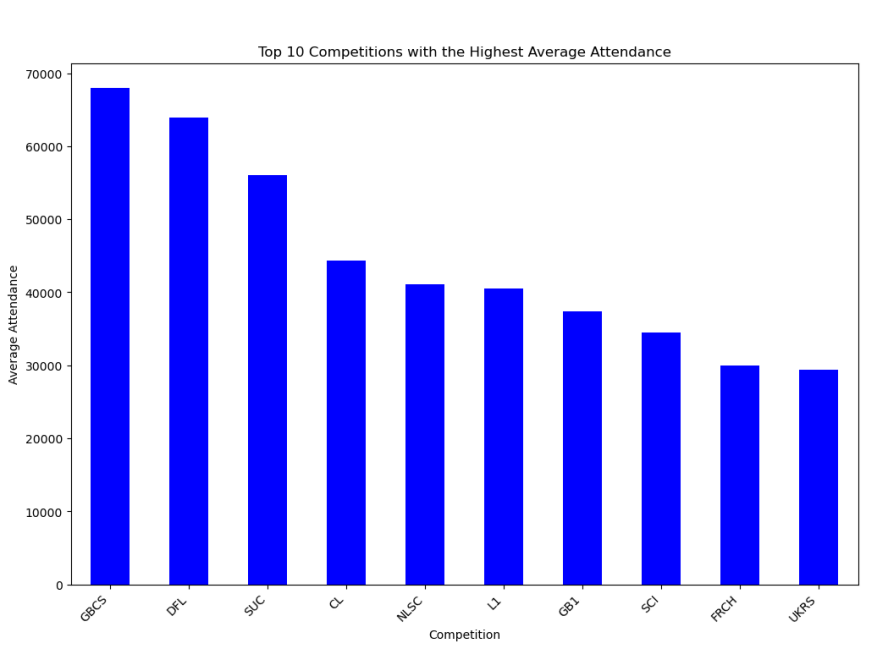
Visualizing Player Minutes Played



- As expected! Because in football we have 3 substitutions (in 90 minutes), so the majority of players play 90 minutes.

Which competition had the highest average attendance?

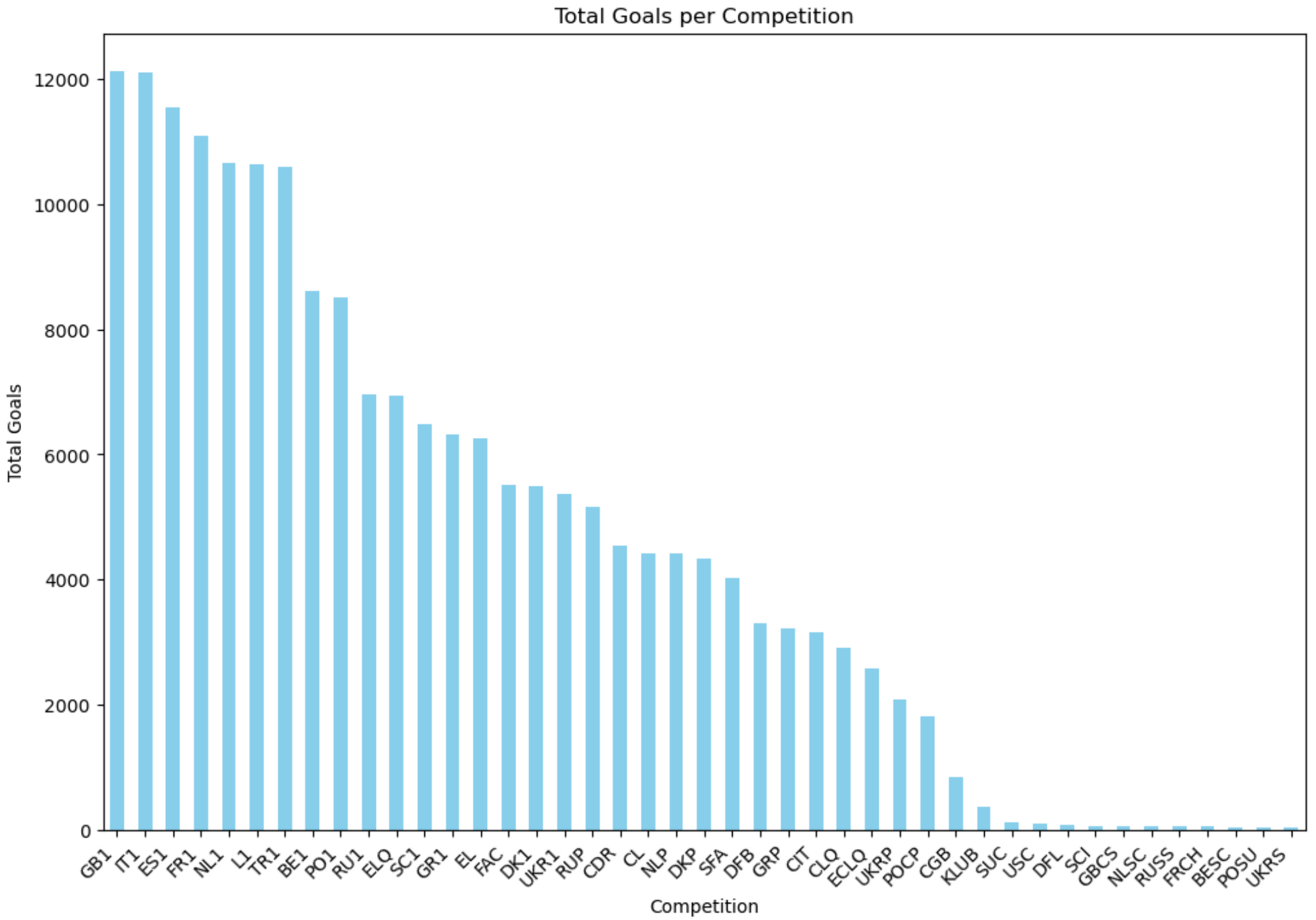
The competition with the highest average attendance is GBCS.



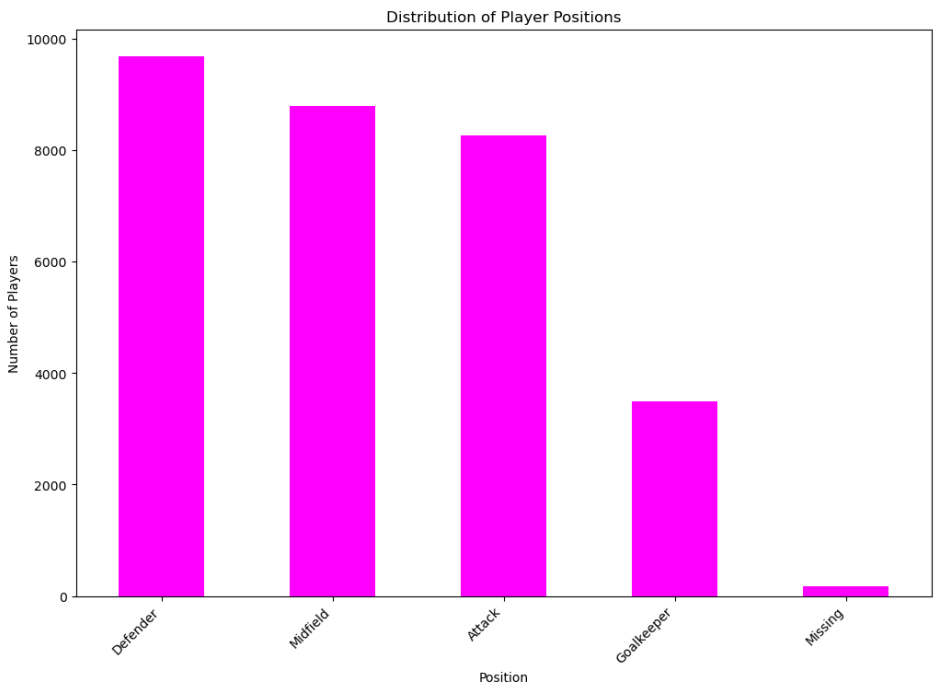
- in this plot, 10 competitions with the highest average attendance have been shown.

Which competition had the highest total number of goals scored?

The competition with the highest total number of goals scored is GB1.

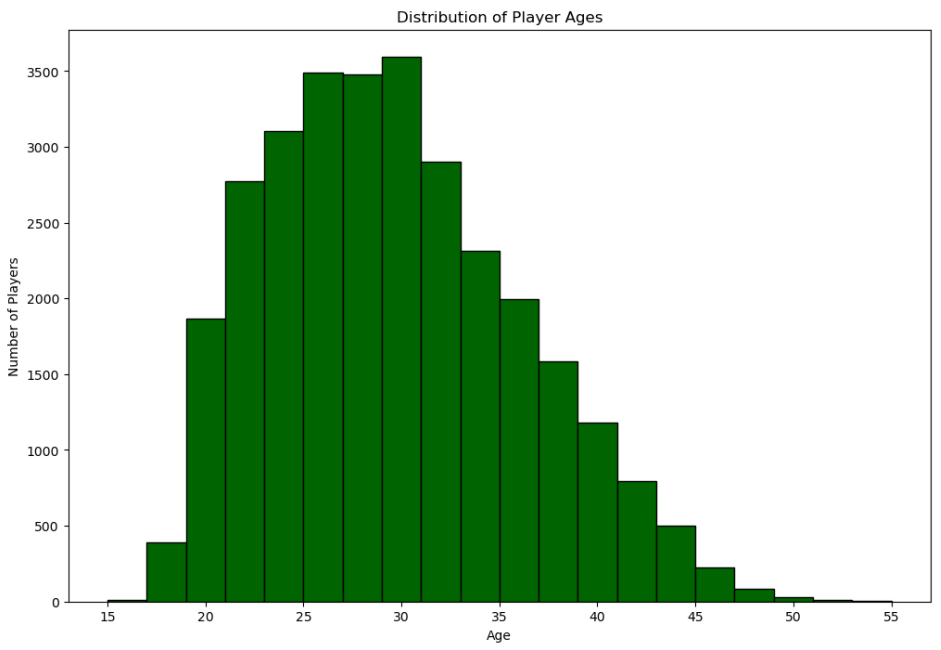


Visualizing Player Positions Distribution

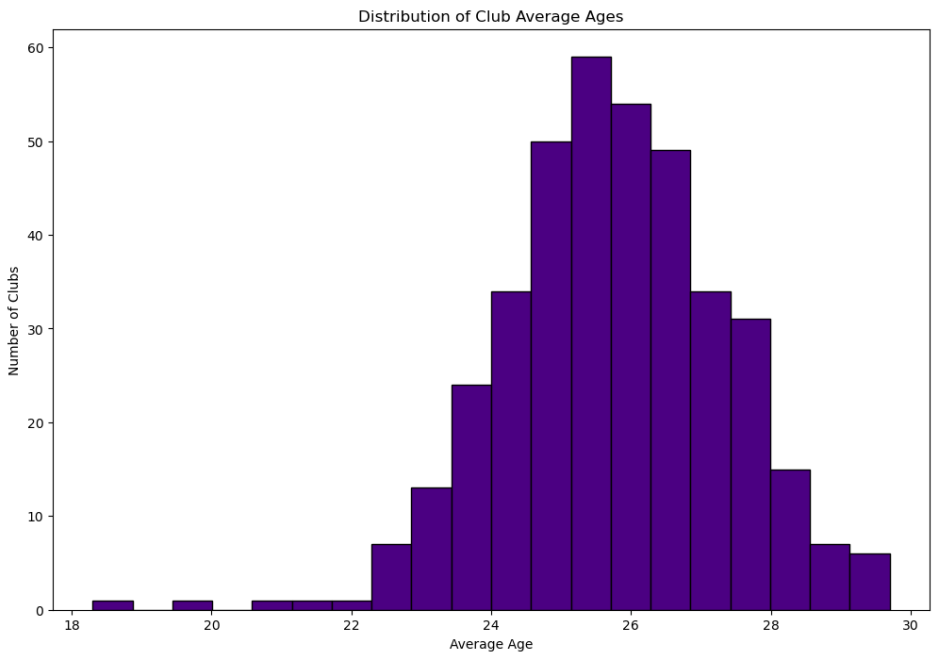


- Defenders are the majority! And some Players have "Missing" position. So in the modeling this issue should be care of.

Visualizing Player Age Distribution

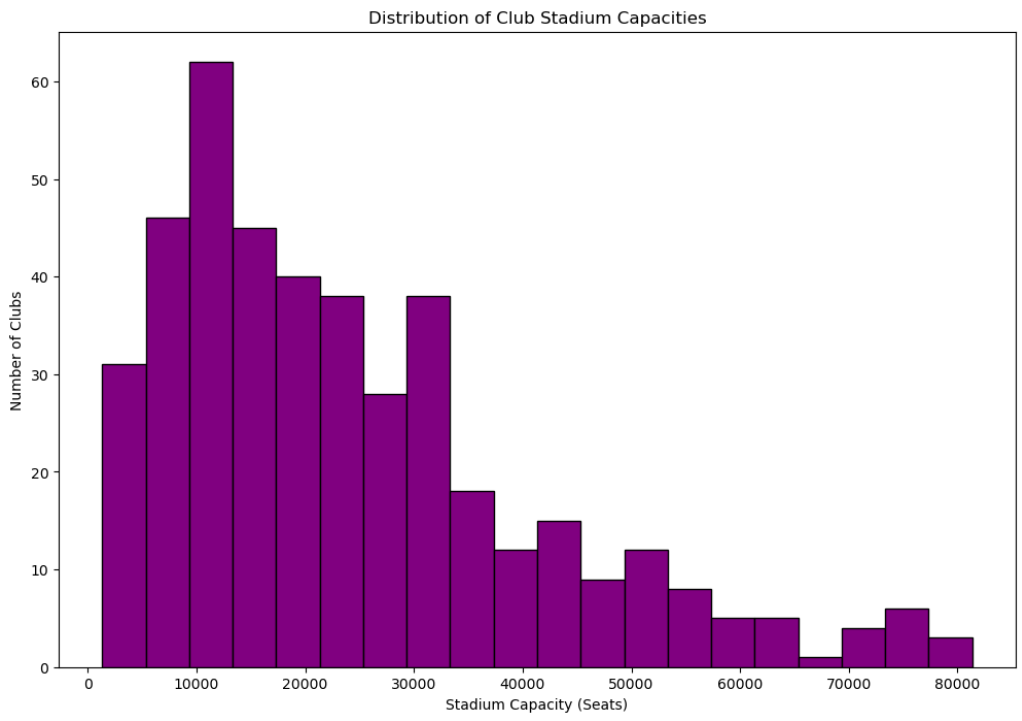


Visualizing Club Average Age Distribution



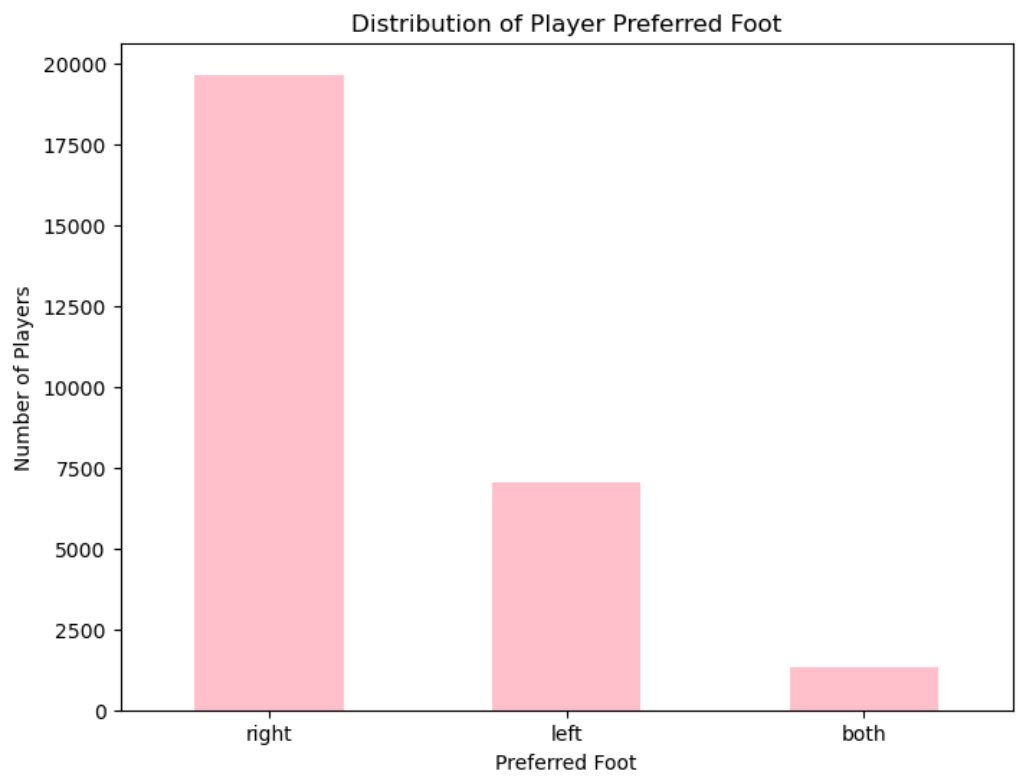
- This two graphs have different distributions. because some players in the **players_df** are not located in the **clubs_df**!

Visualizing Club Stadium Capacity Distribution



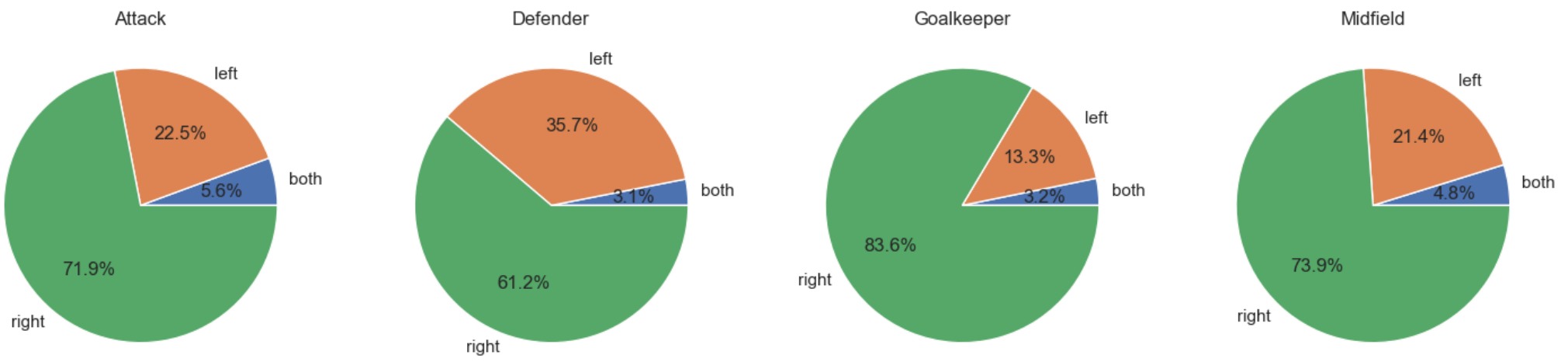
- As expected there are few stadium with high capacity like Camp Nou.

Visualization for Player Foot Distribution



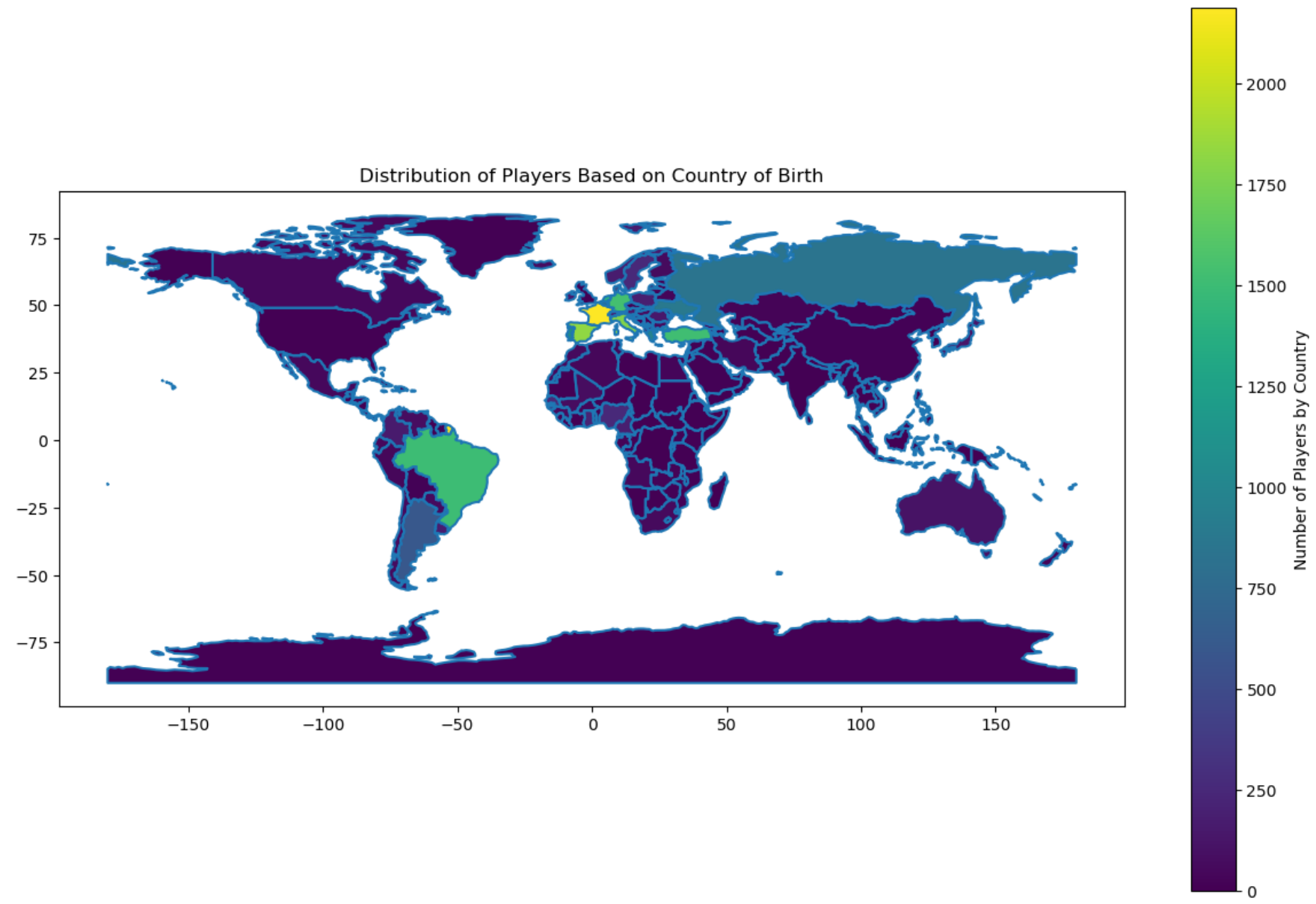
- Most players are right-footed and there is a Hypothesis that players with both feet are more valuable!

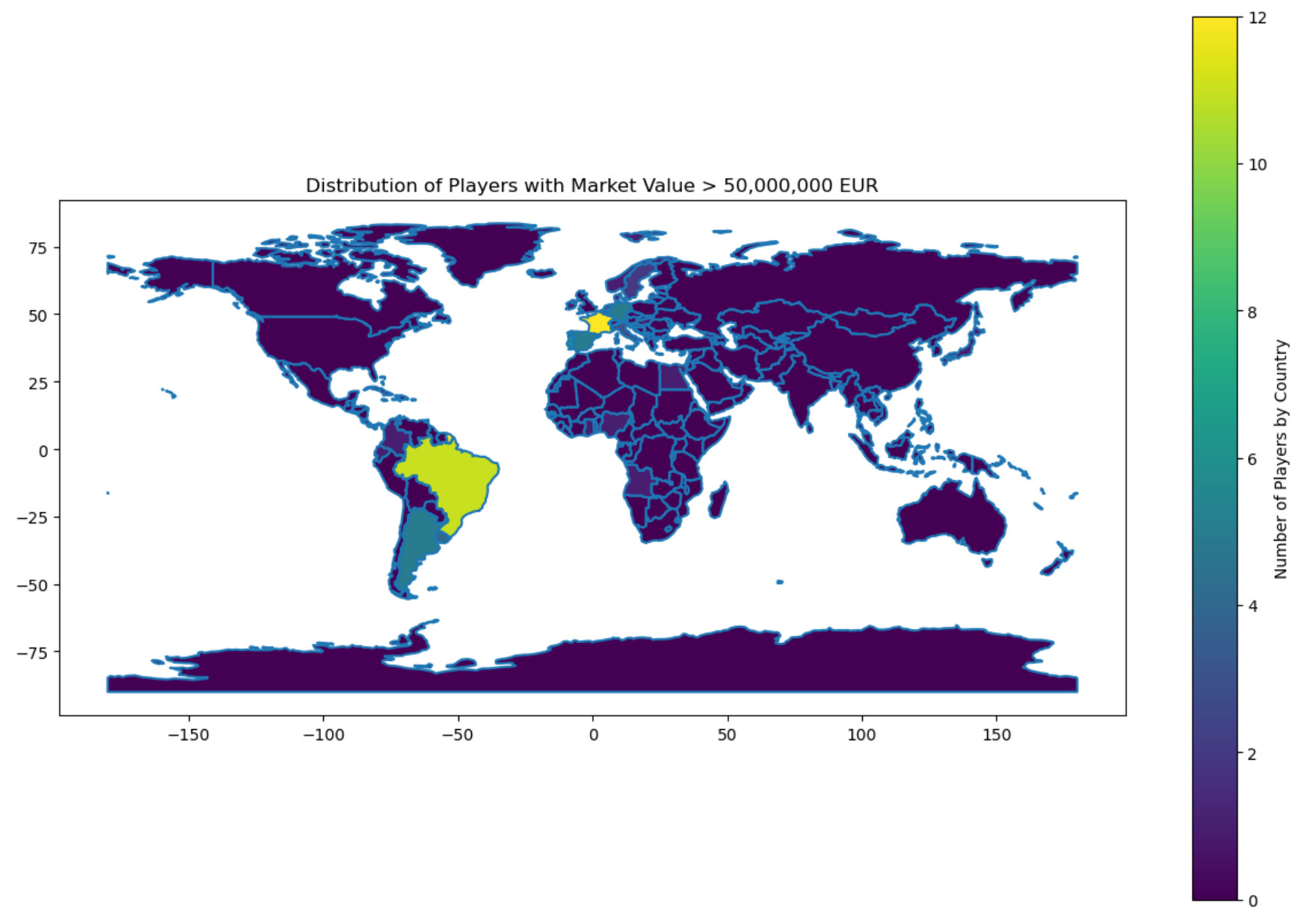
How frequent left footers are versus right footers across positions?

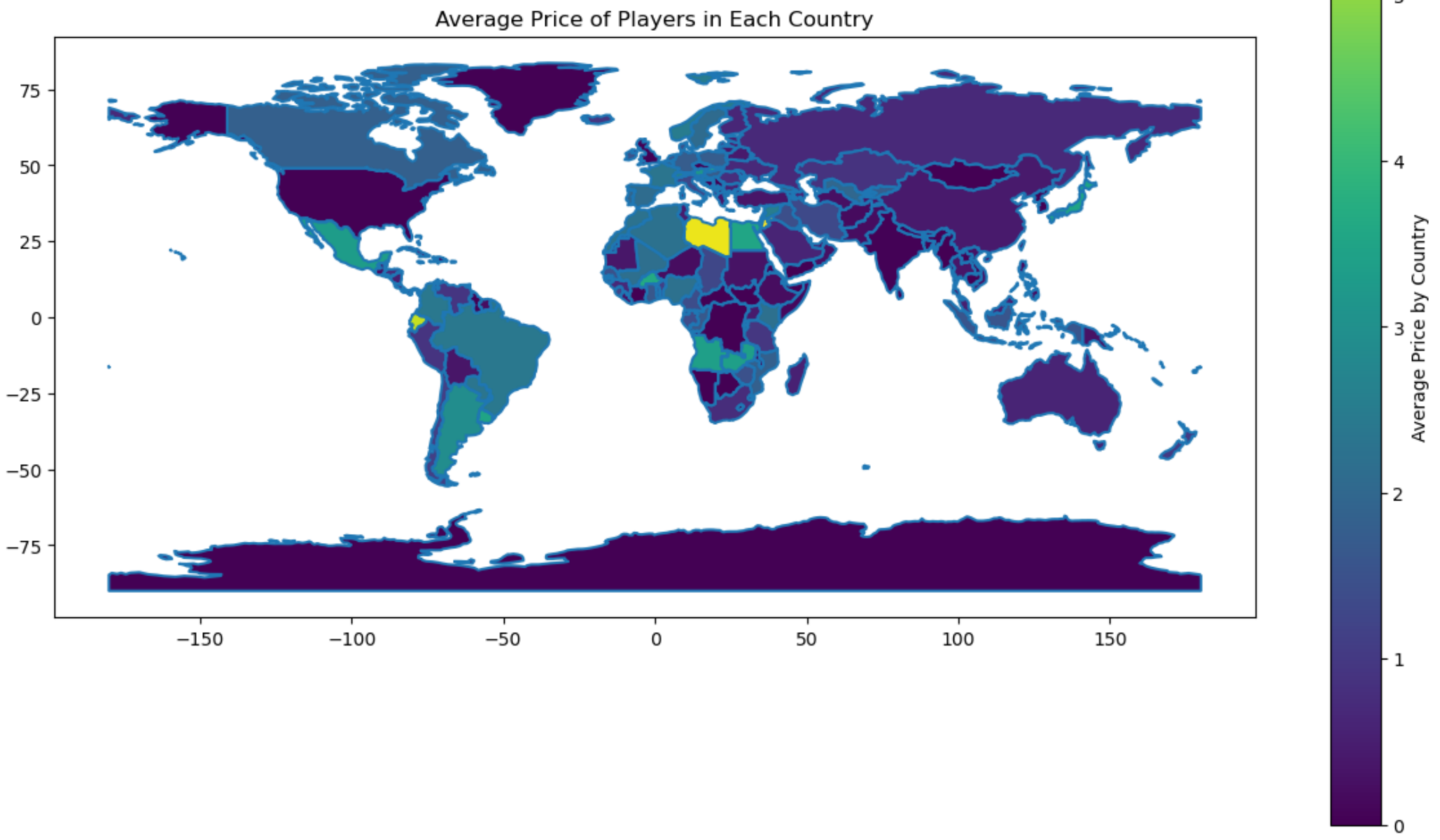


Below are the diagrams related to the map

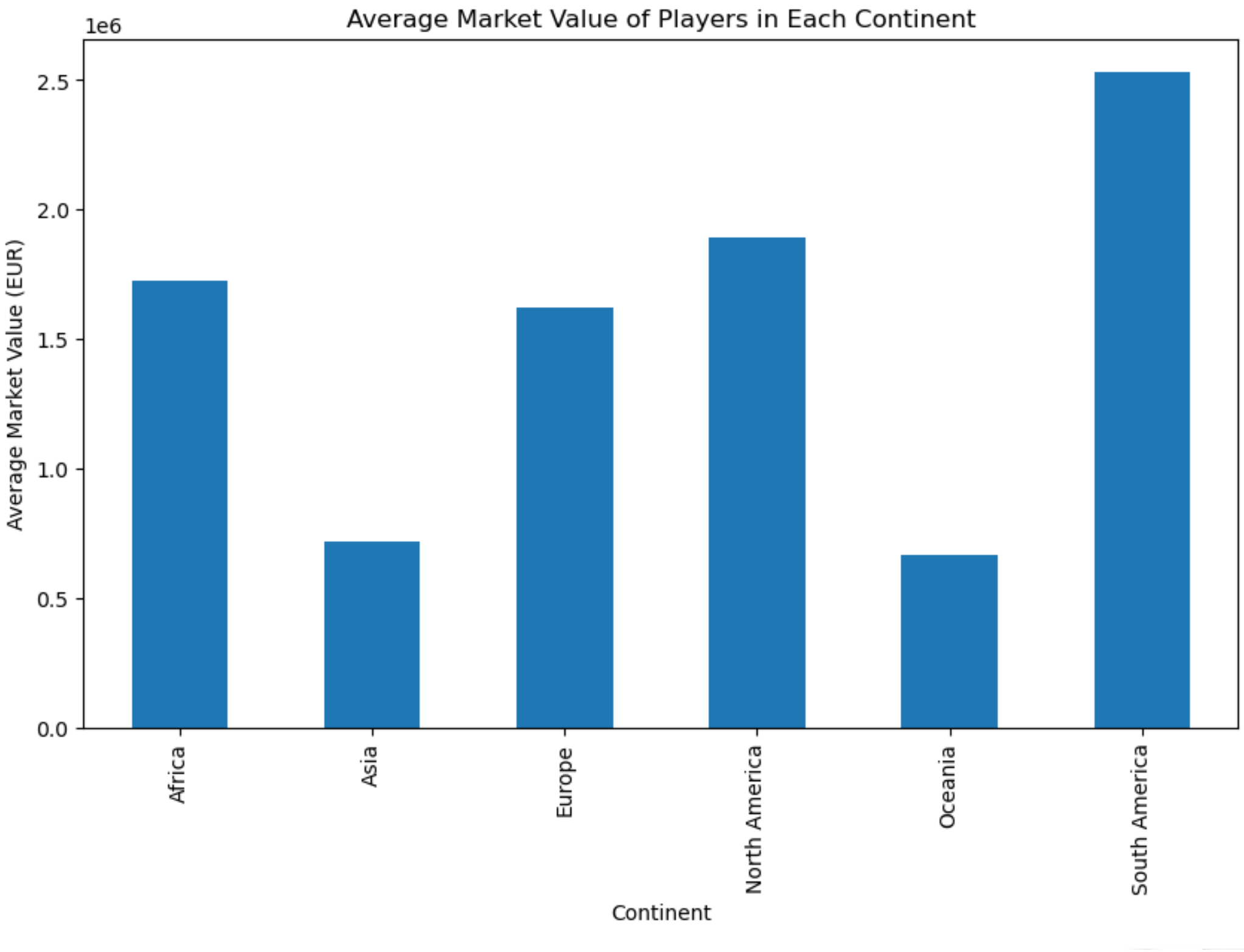
- It shows that Europe and South America have more players
- Also, Europe and South America have a larger share of expensive players





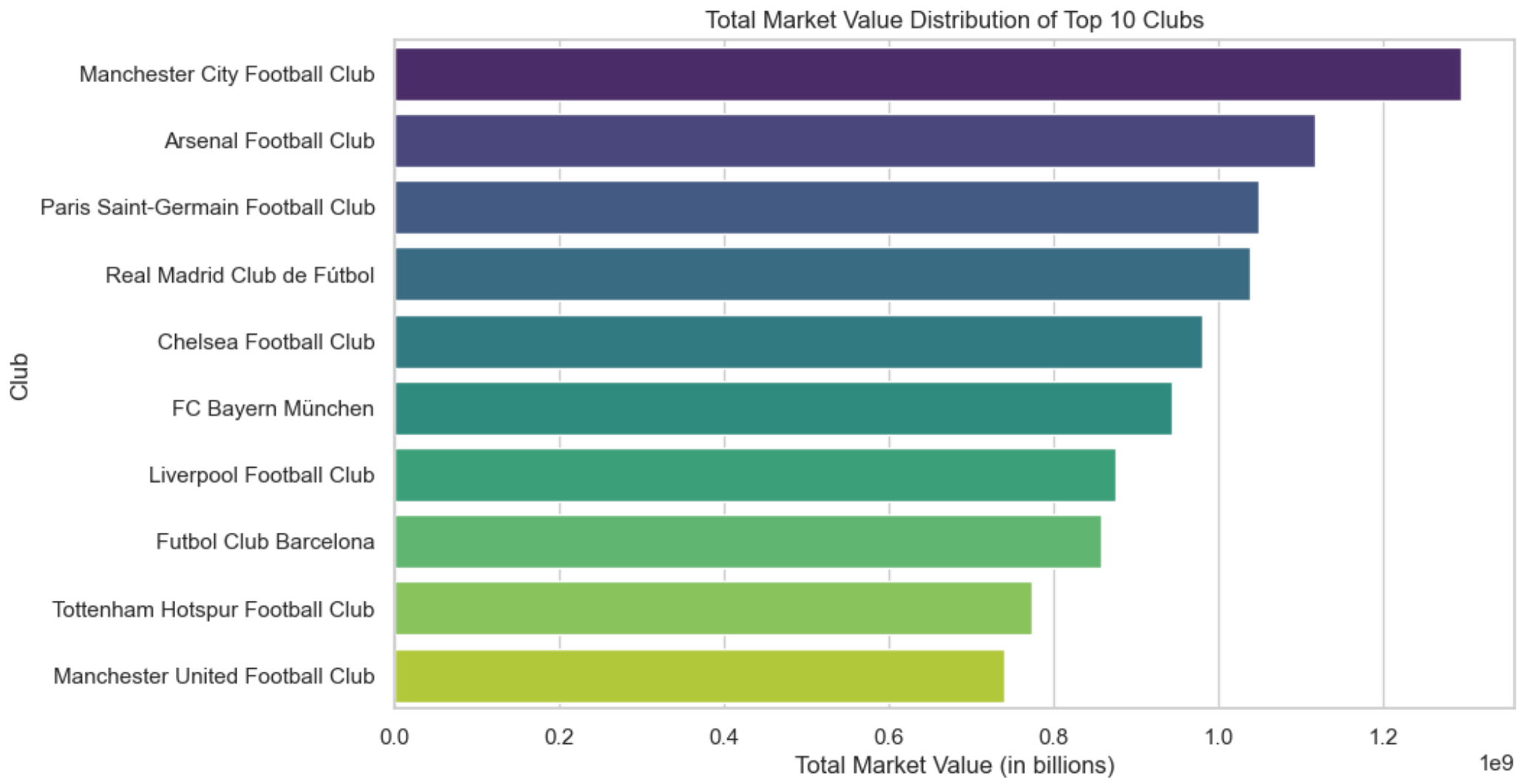


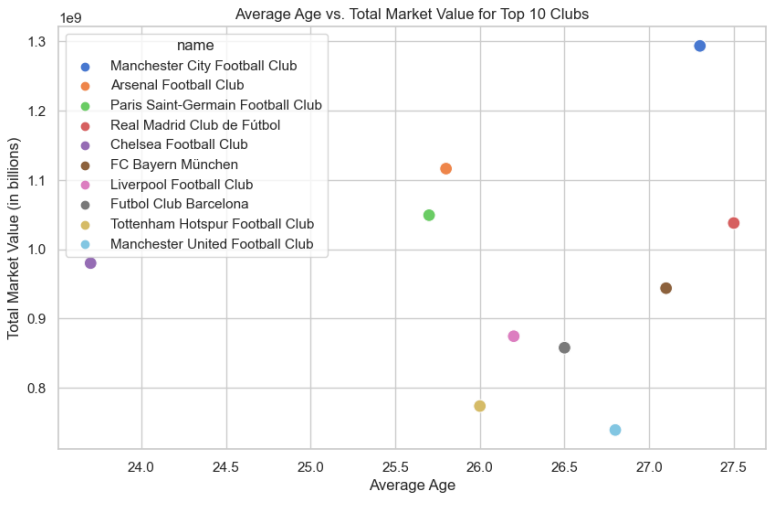
- Because South Africa has few registered players in the leagues, that is why the average of one of those countries "Libya" is high.

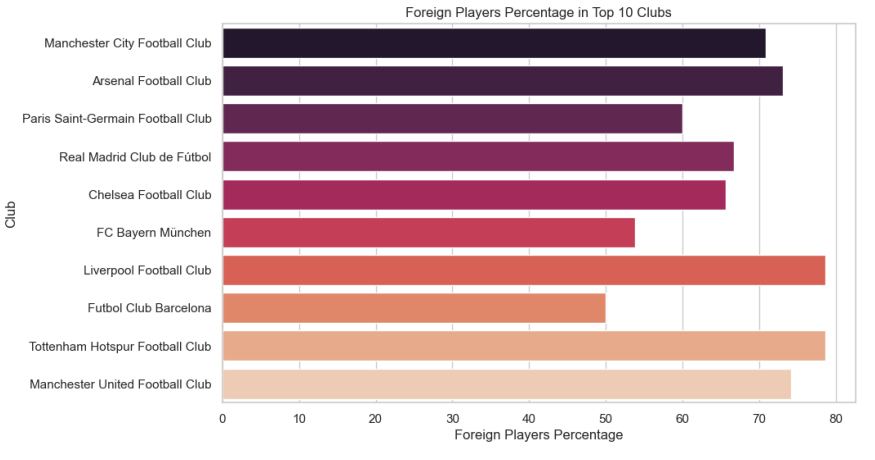


Club Compare

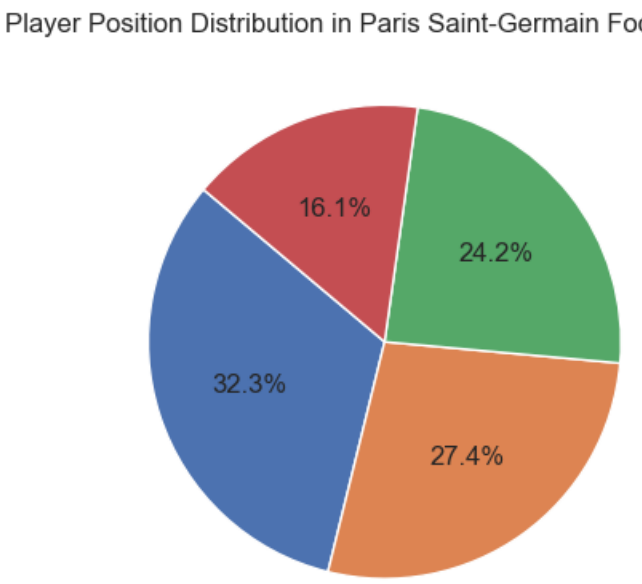
After looking at **clubs_df**, it can be undrastd that this data frame have null values for **total_market_value**. so based on **players_df** and the season = 2023, this column has been calculated.

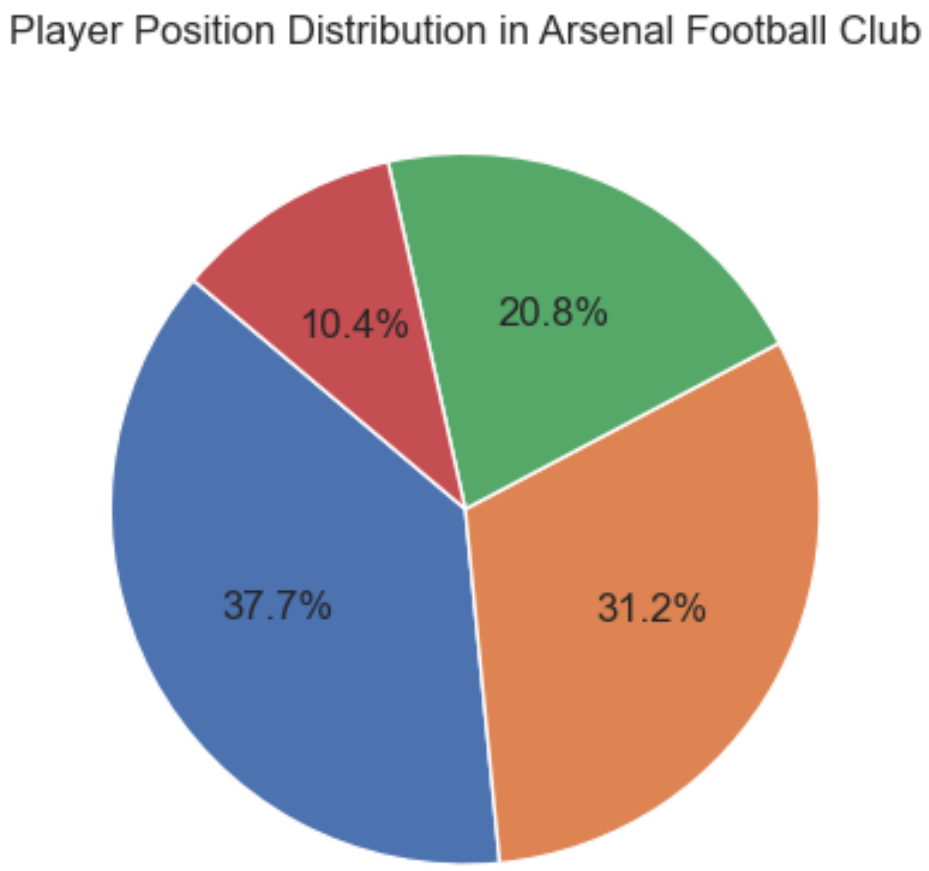


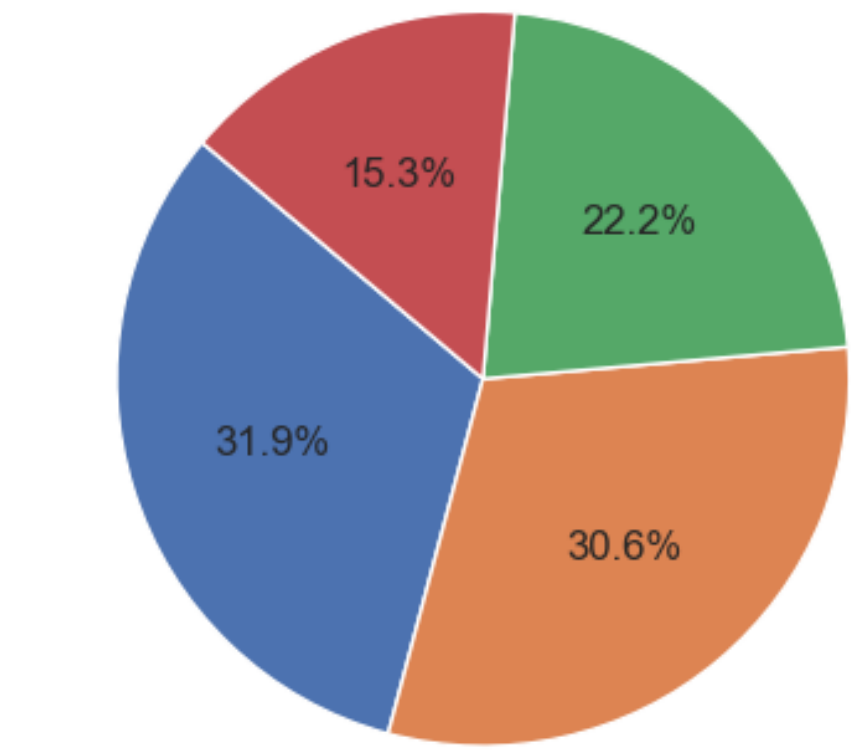


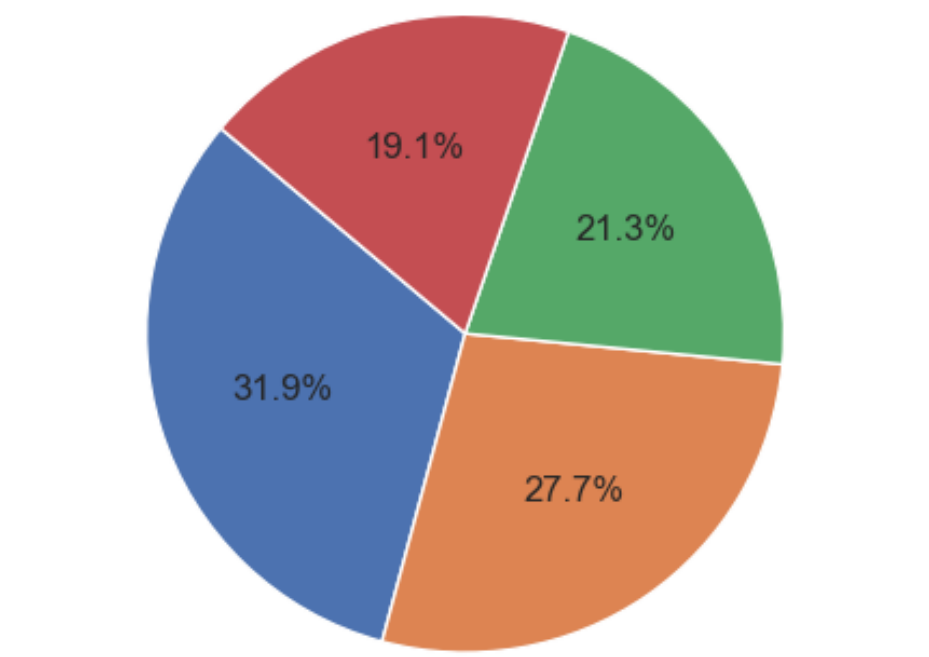


Pie chart showing the distribution of player positions in each club









Iranian Players

Top 10 Iranian Players based on Highest Market Value:

	name	highest_market_value_in_eur	last_season
11441	Sardar Azmoun	25000000.0	2023
17245	Mehdi Taremi	20000000.0	2023
13154	Alireza Jahanbakhsh	18000000.0	2023
12803	Saman Ghoddos	5000000.0	2023
18145	Milad Mohammadi	5000000.0	2023
16911	Majid Hosseini	3500000.0	2023
25831	Mohammad Mohebi	1800000.0	2023
6550	Ehsan Hajsafi	1500000.0	2023
14731	Saeid Ezatolahi	1500000.0	2023
14736	Ali Alipour	1500000.0	2023

2. Predict the price of players

In the next part, we will get information through data and by doing **EDA** and **FEATURE ENGINEERING**, we will test the machine learning **MODELS** for predicting the players' prices.

2.1 Data processing

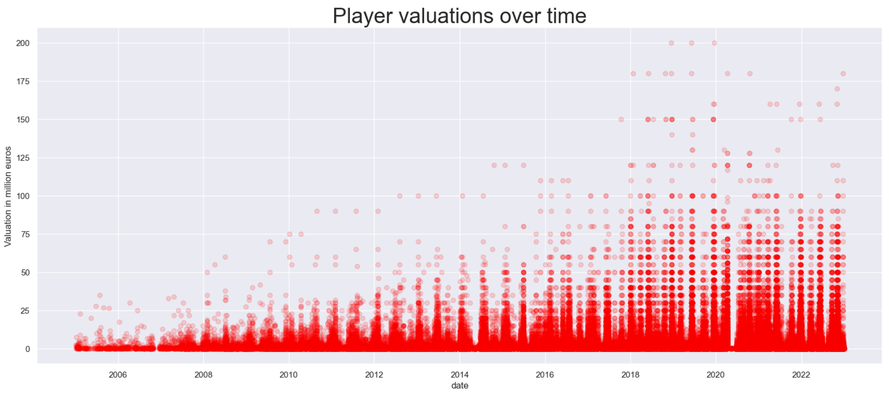
Before that, some Data processing on data set :

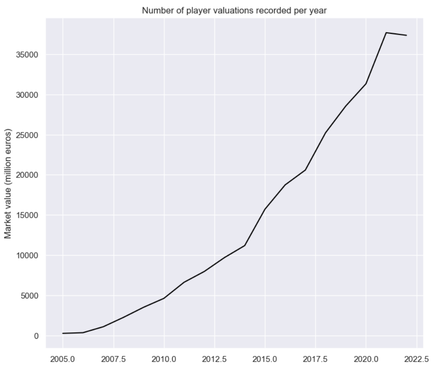
- Calculate the age of each player
- drop players with no date of birth
- Calculate the contract remaining of each player
- drop players with no date of birth
- add year to player valuations
- add year to player appearances
- add position to player valuations
- add position to appearances

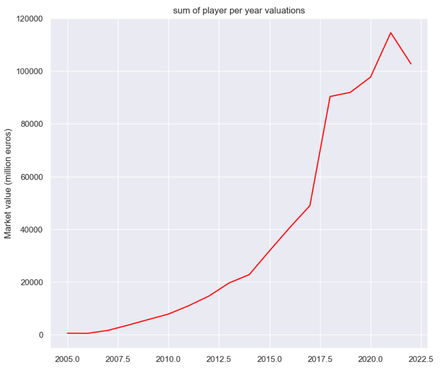
Data processing completed.

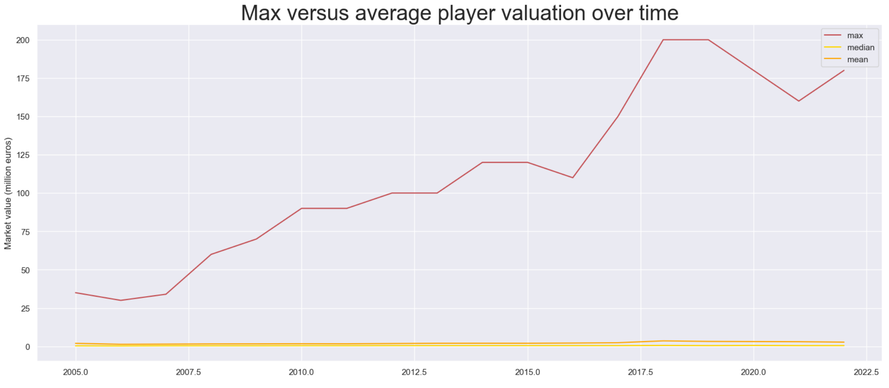
2.2 Visualization

Player Valuation Data Visualization:







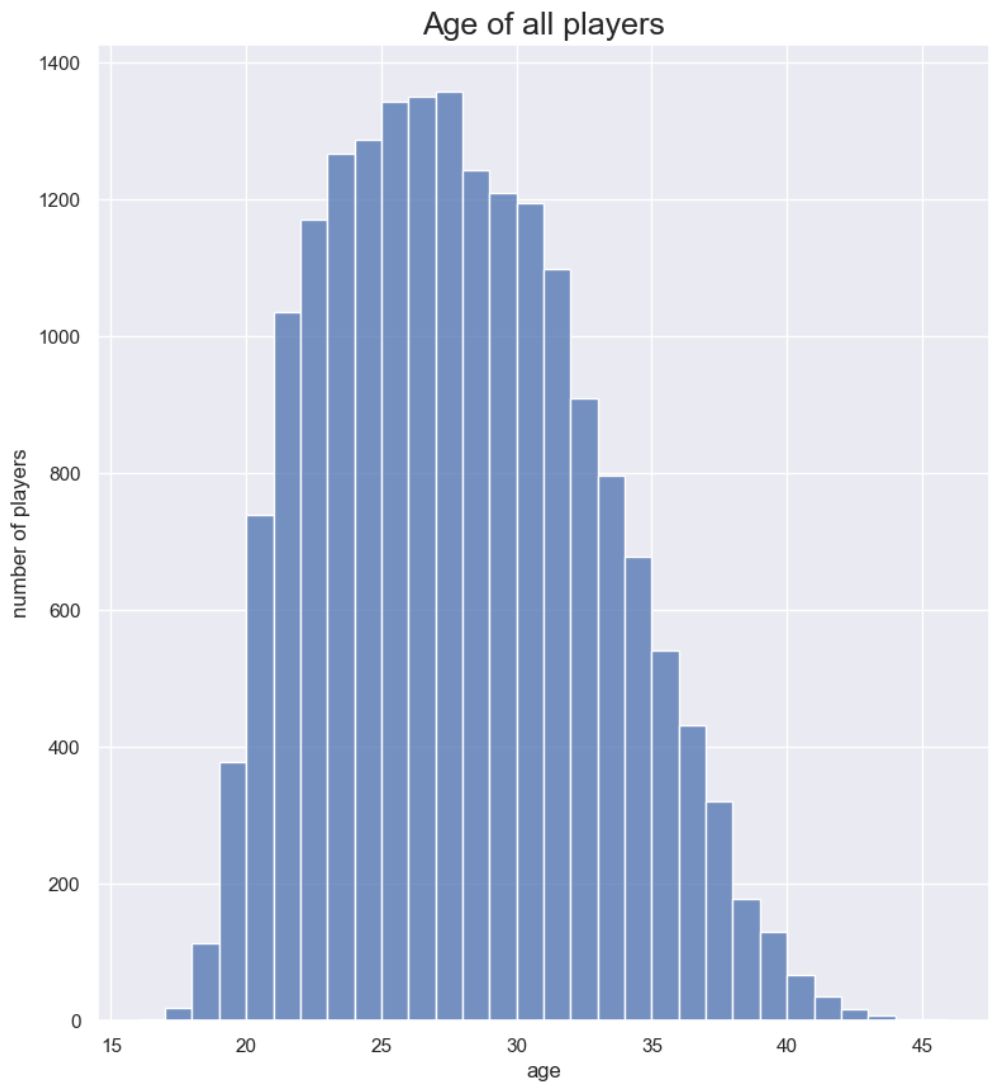


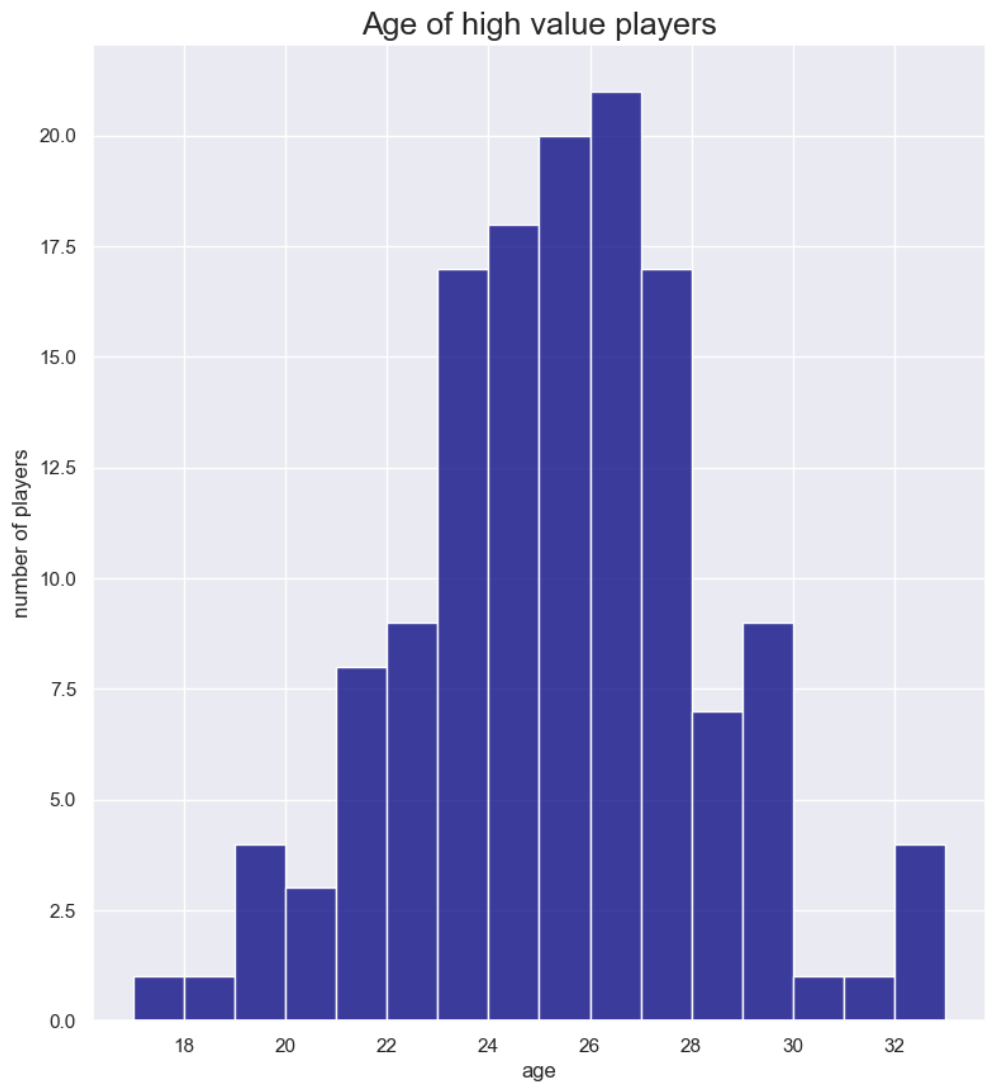
Observations on market value timeline data

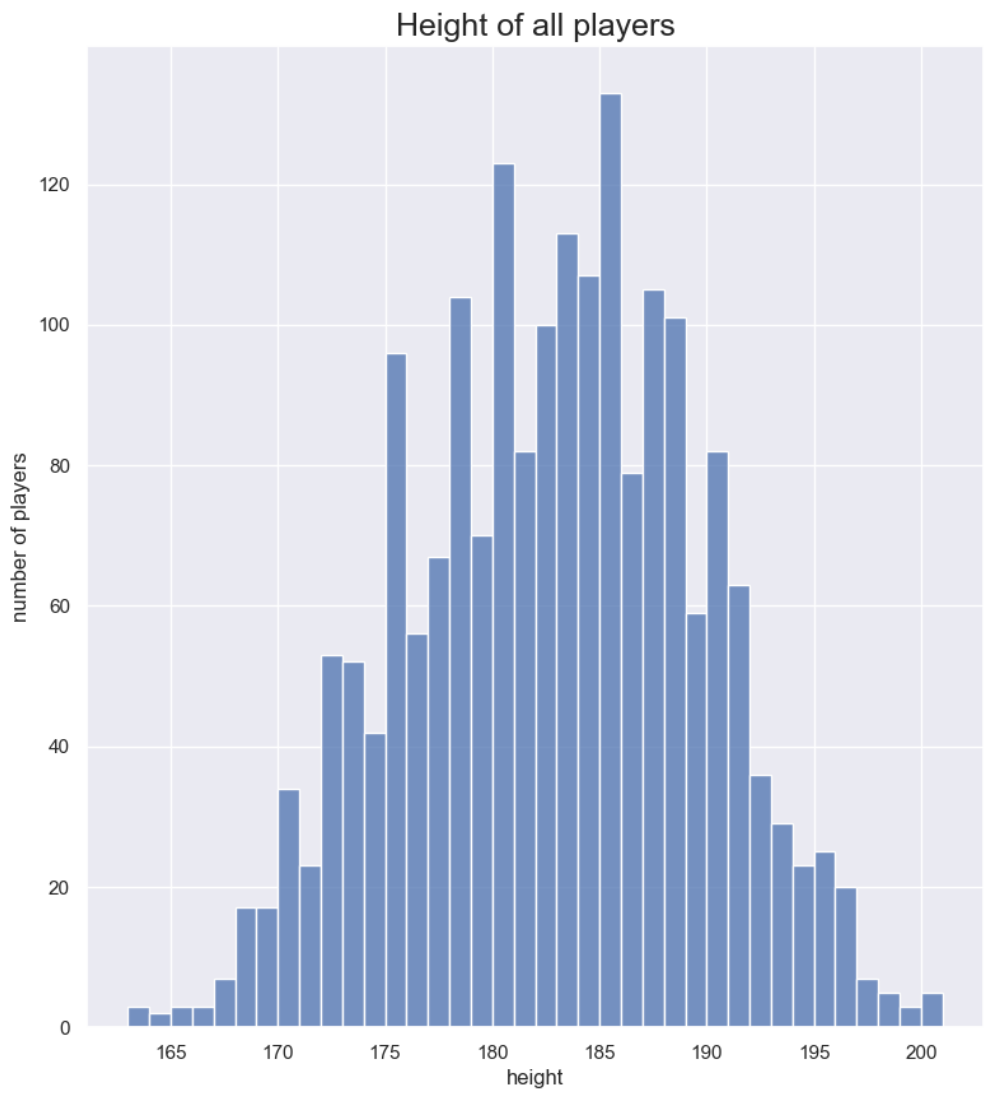
> The scatter plot of player valuations over the period 2006 to 2022 shows that the value of players has generally risen over time. The plot shows that the vast majority of players are at the lower end of the scale below the 50 million euros mark.
>
> Separating player values in attack,midfield,defence and goal keeper positions shows attacking positions generally achieve the highest values while goal keepers command lower values.
>
> There appears to be a shift around 2018 in where values at the higher end of the scale in most positions rose dramatically, this appears to have peaked and come down a little to 2022 rates.

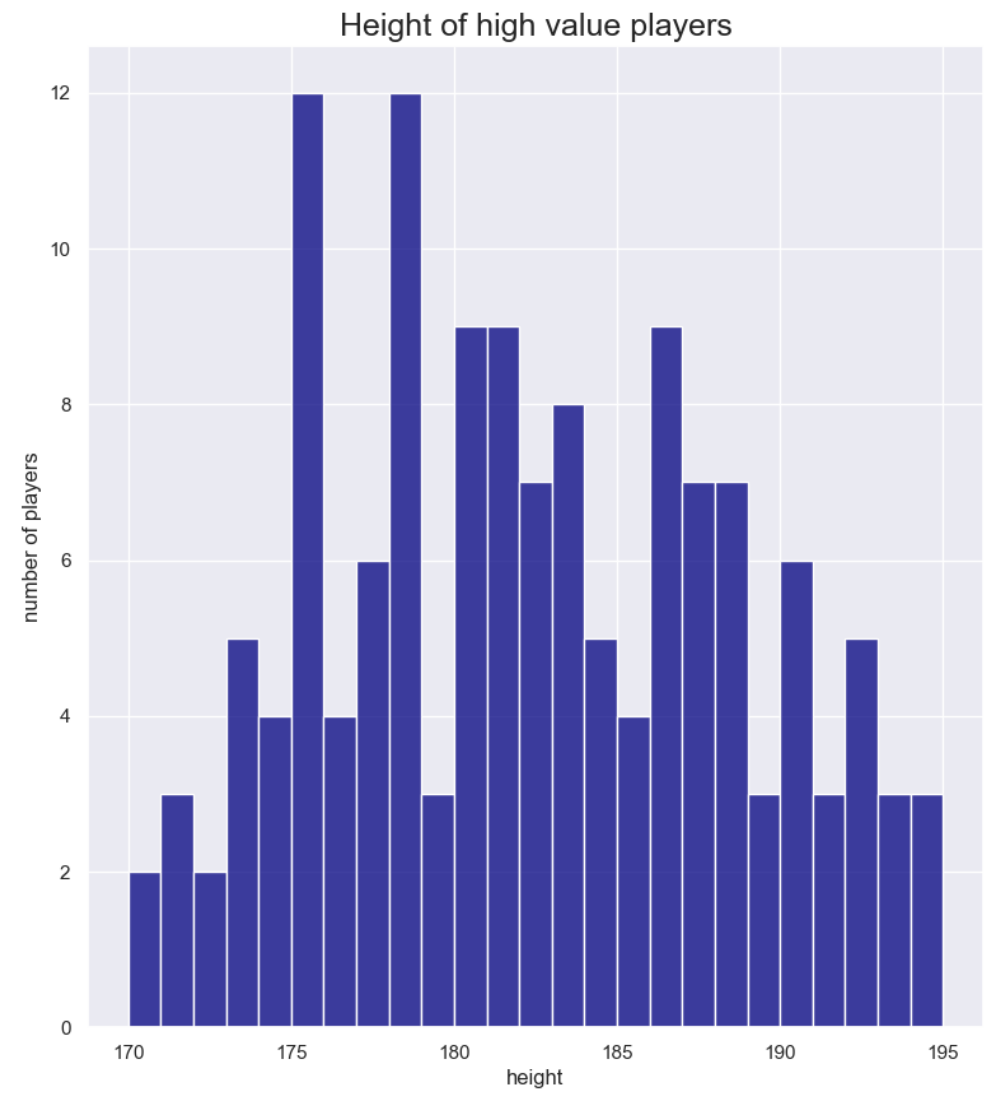
Player Data Visualization

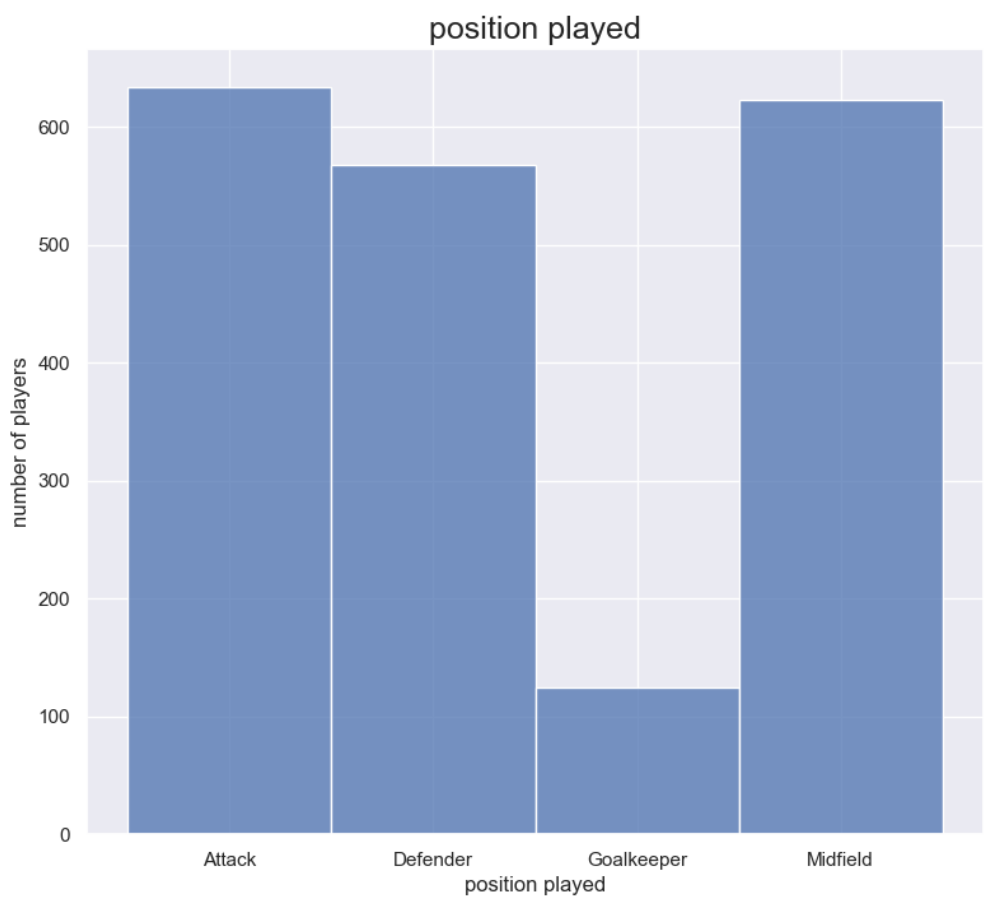
Exploratored data visualizations relating to players who's last season played was 2022 or 2023 with a market value of greater that 15 million euros.

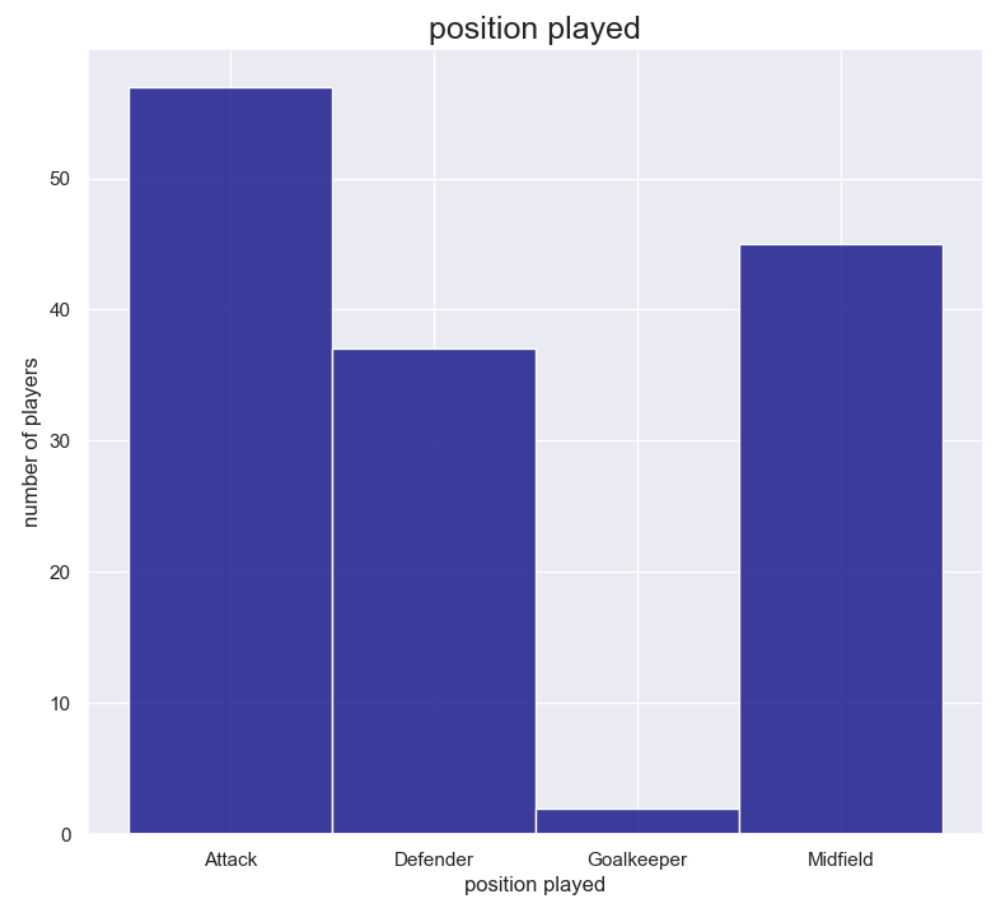


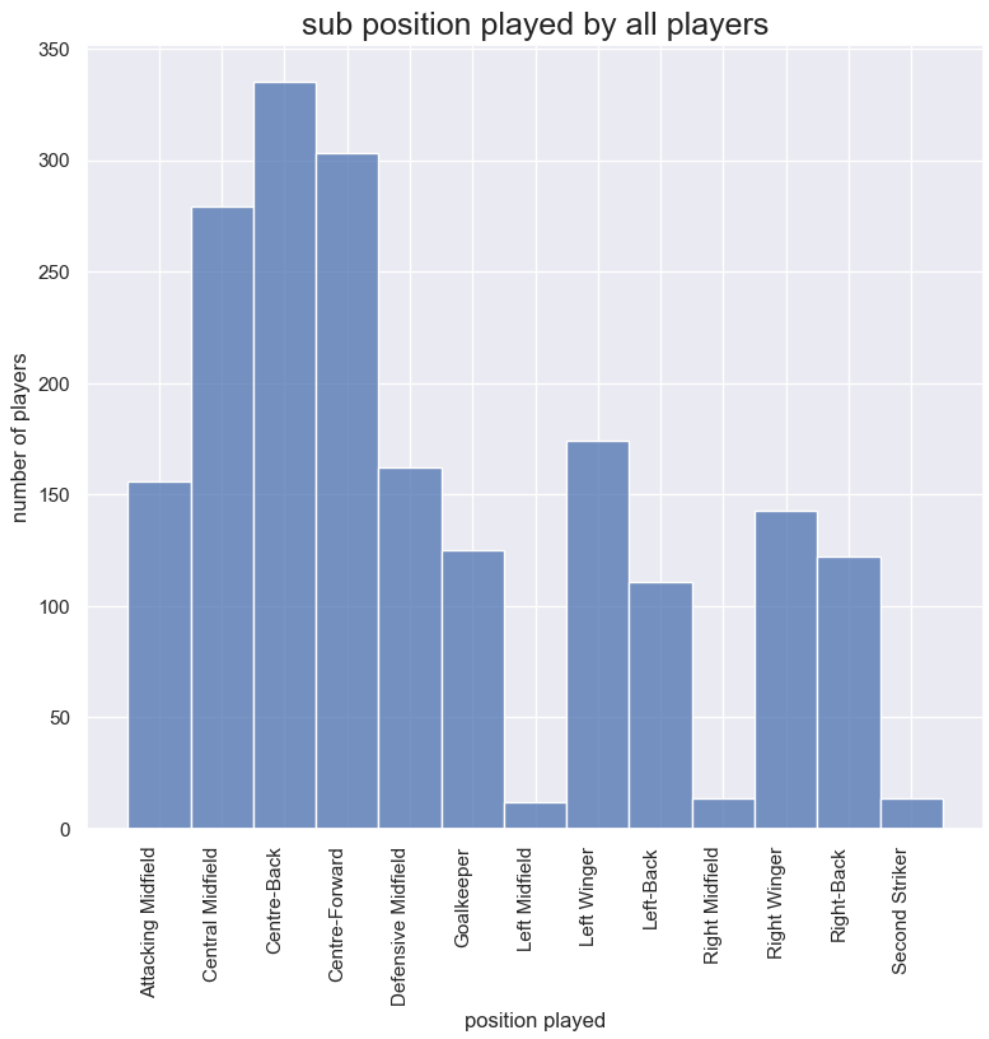


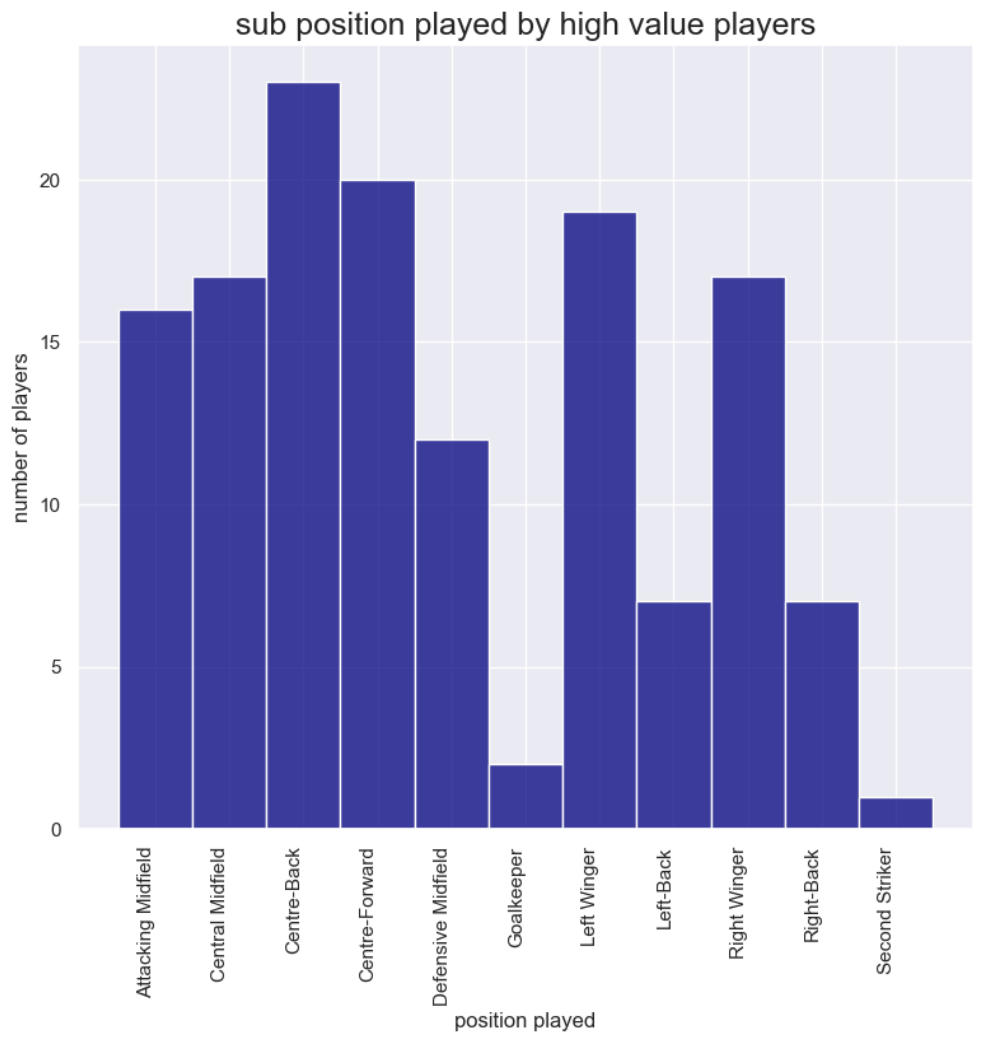


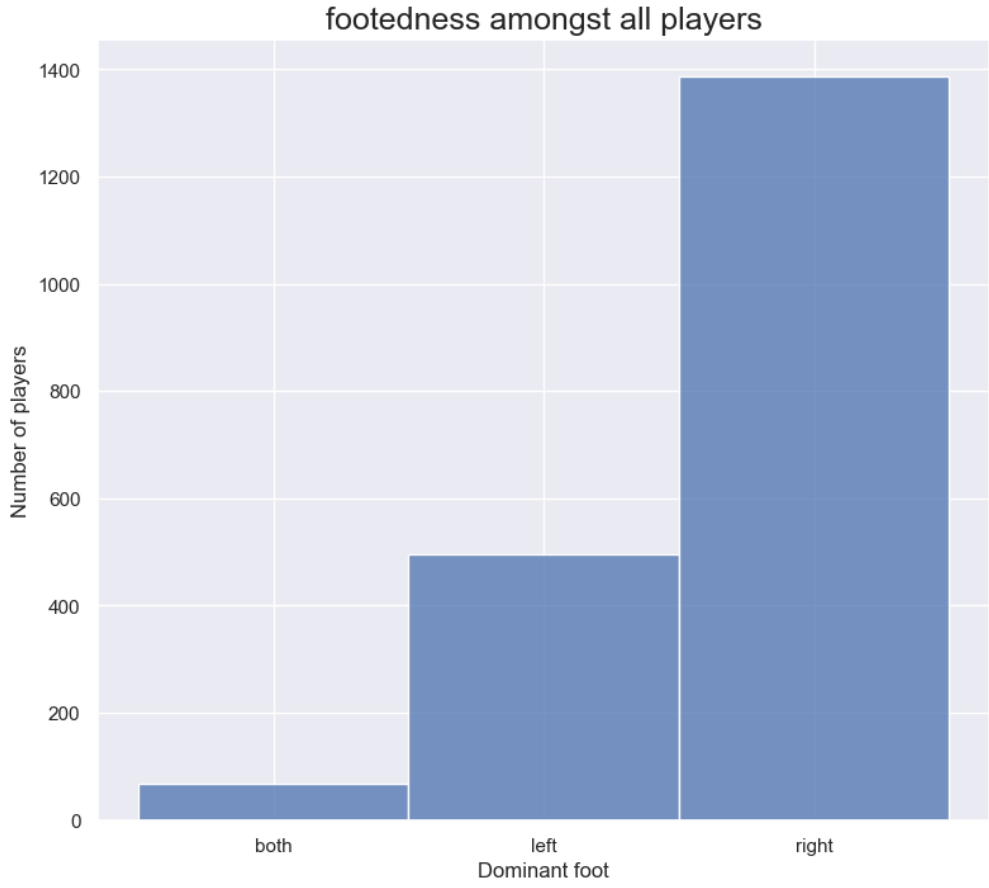


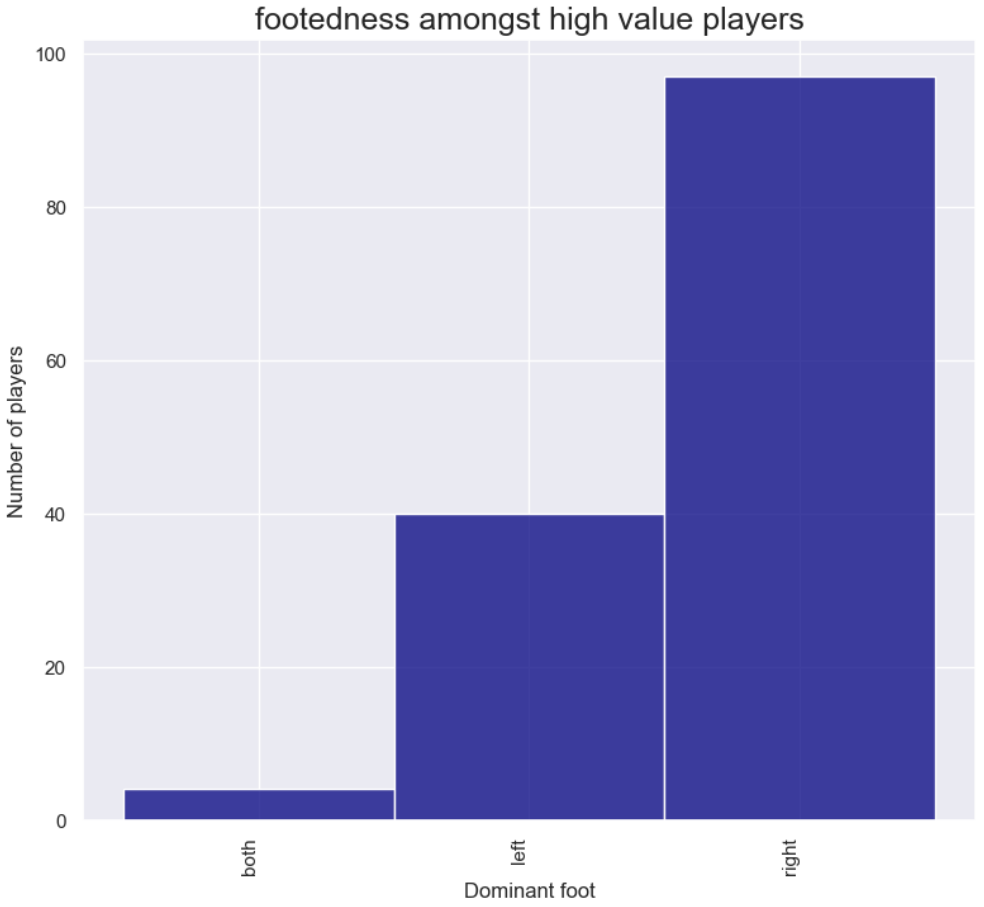








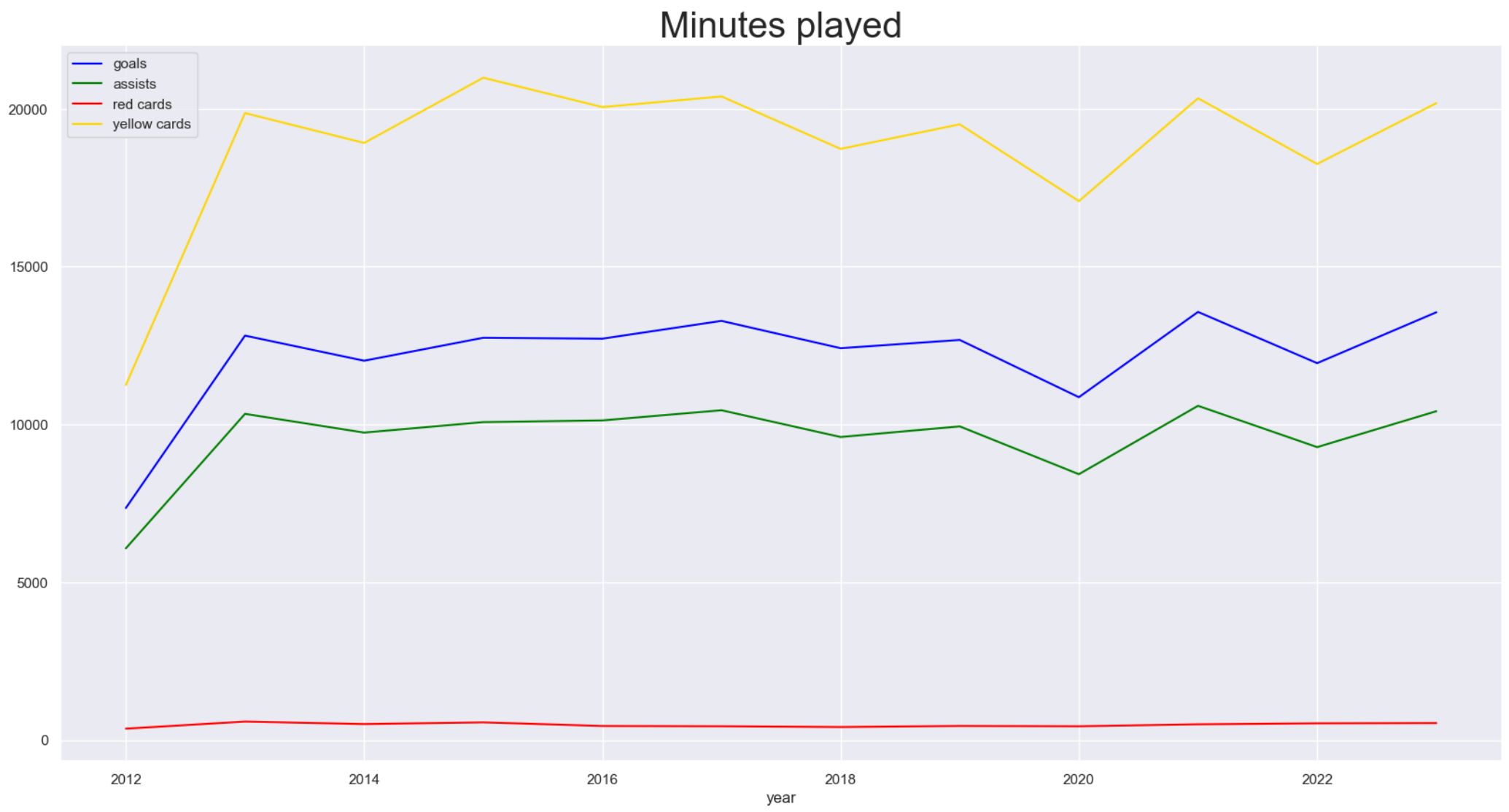


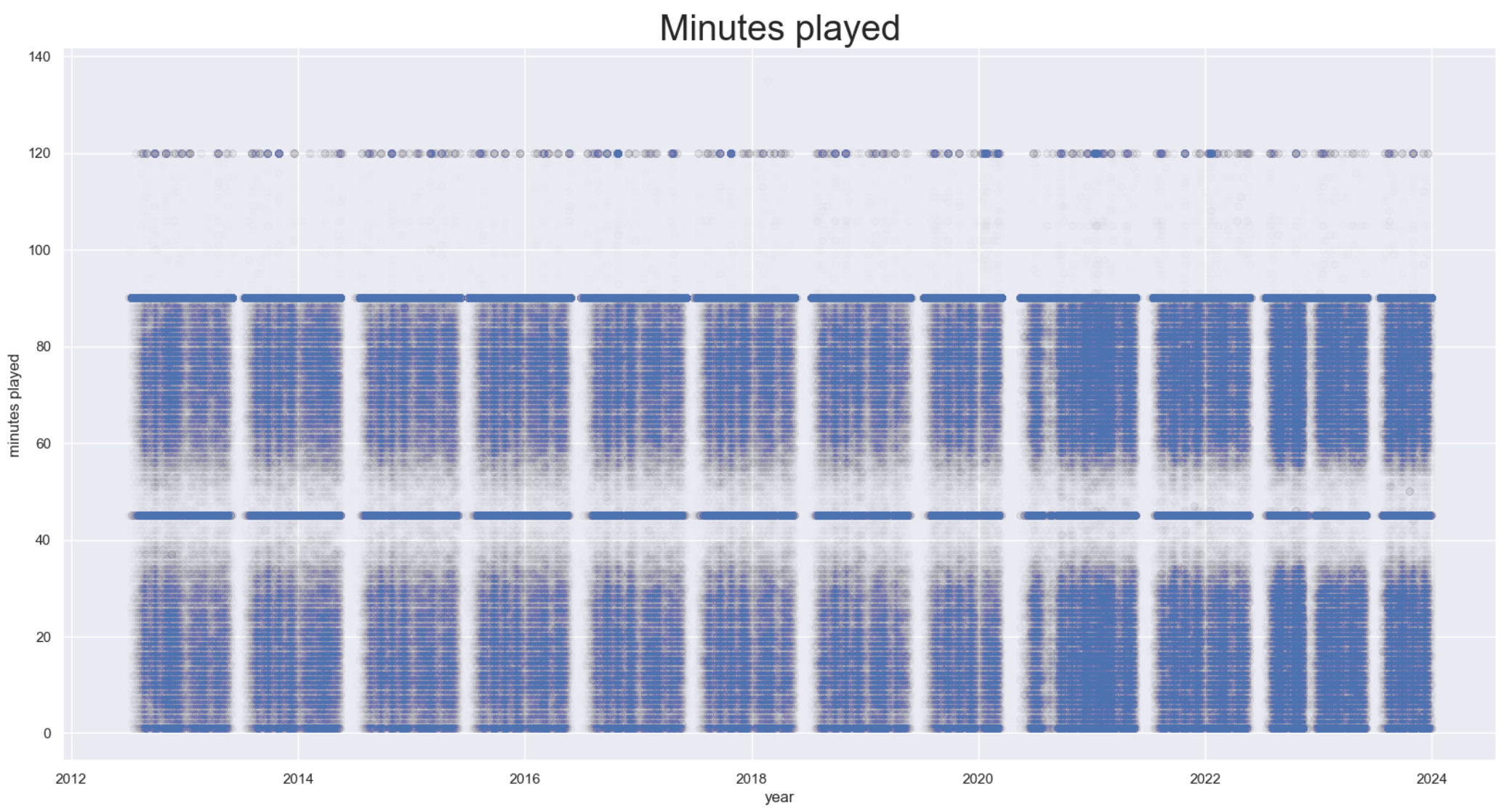


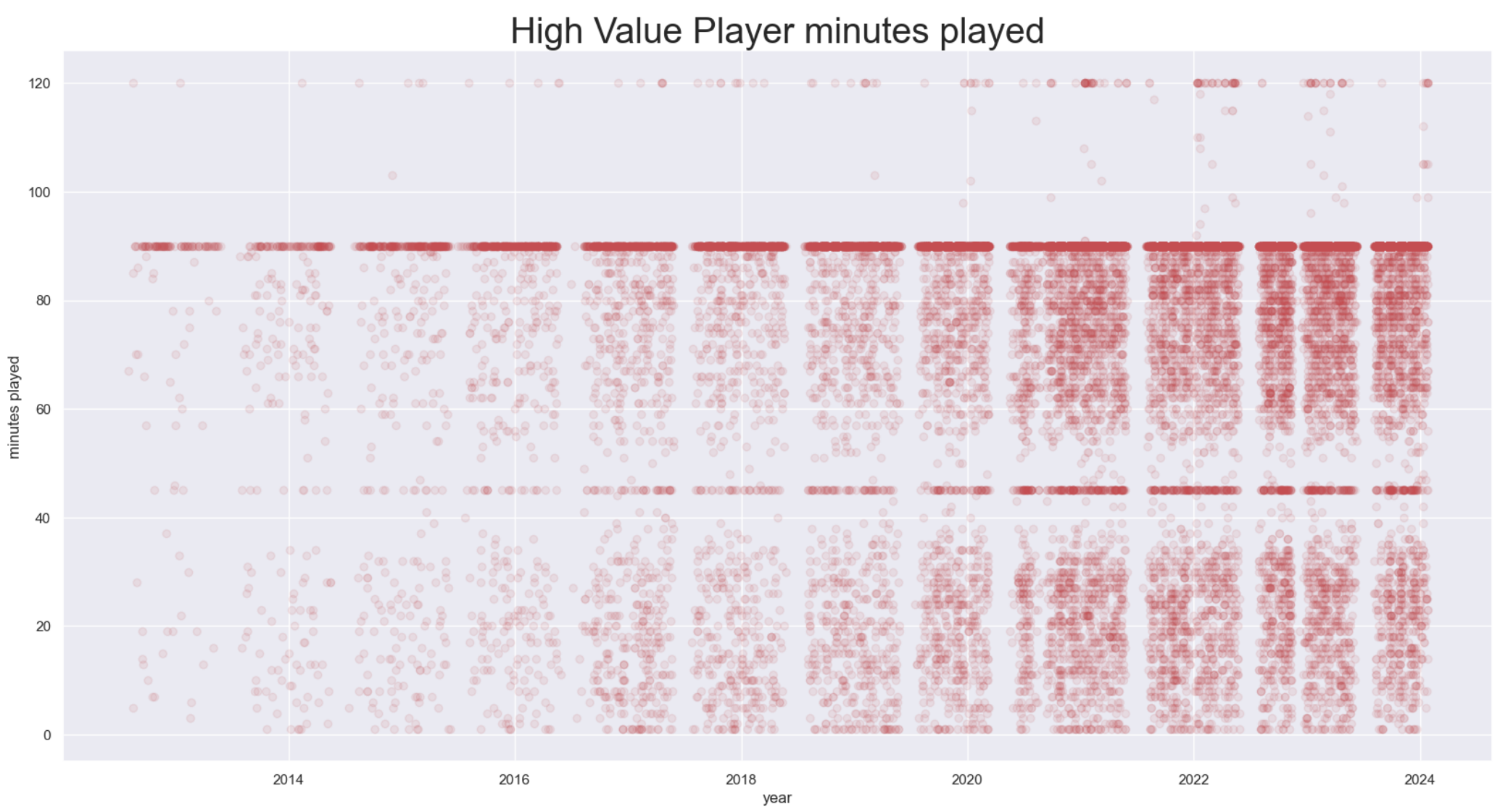
Observations on player data visualizations

> Generally the distibutions of age is a skewed normal distribution and height is a normal distribution. Distibutions amongst high value players appear to be reflective of values within the entire dataset.

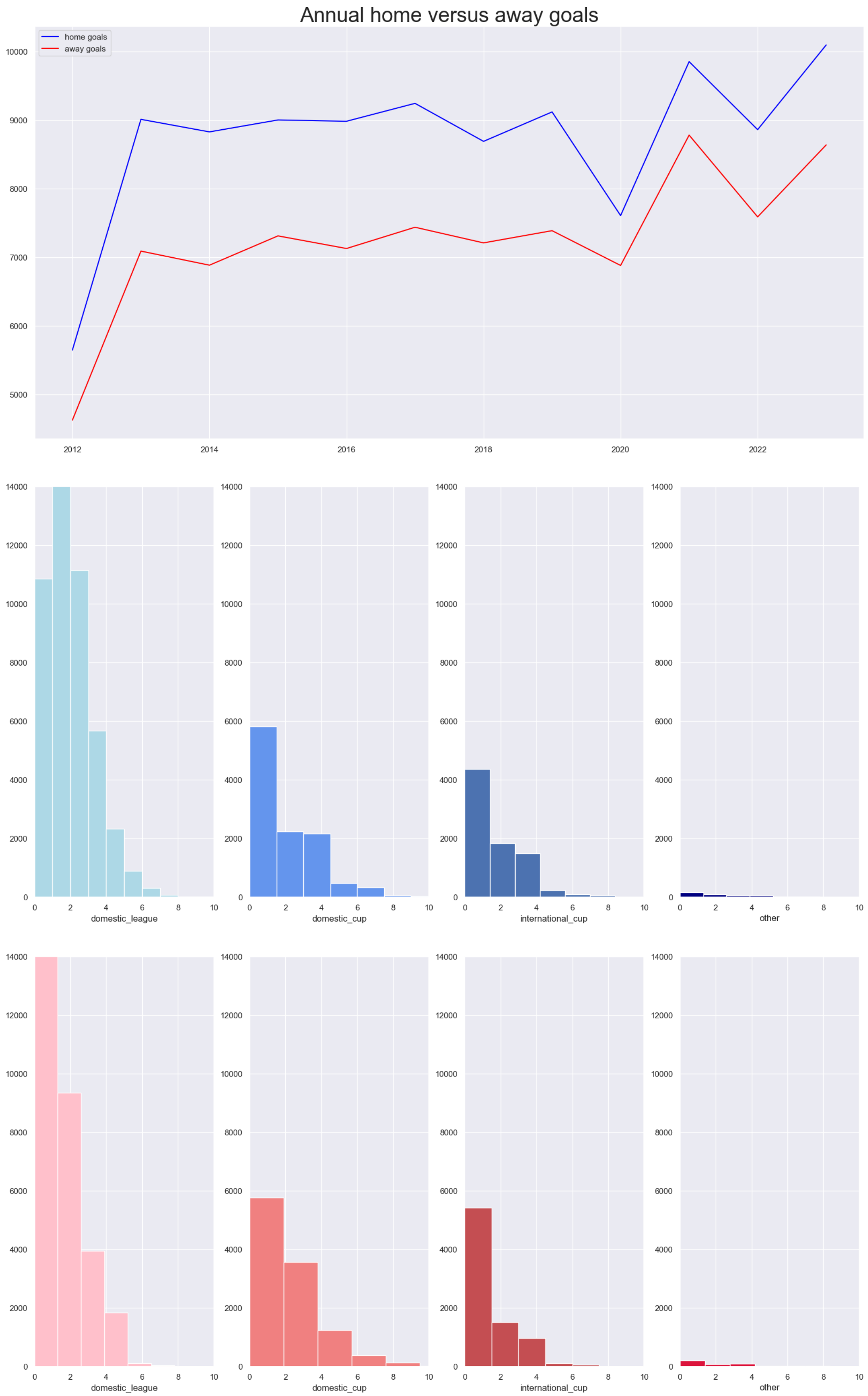
Appearance Visualizations





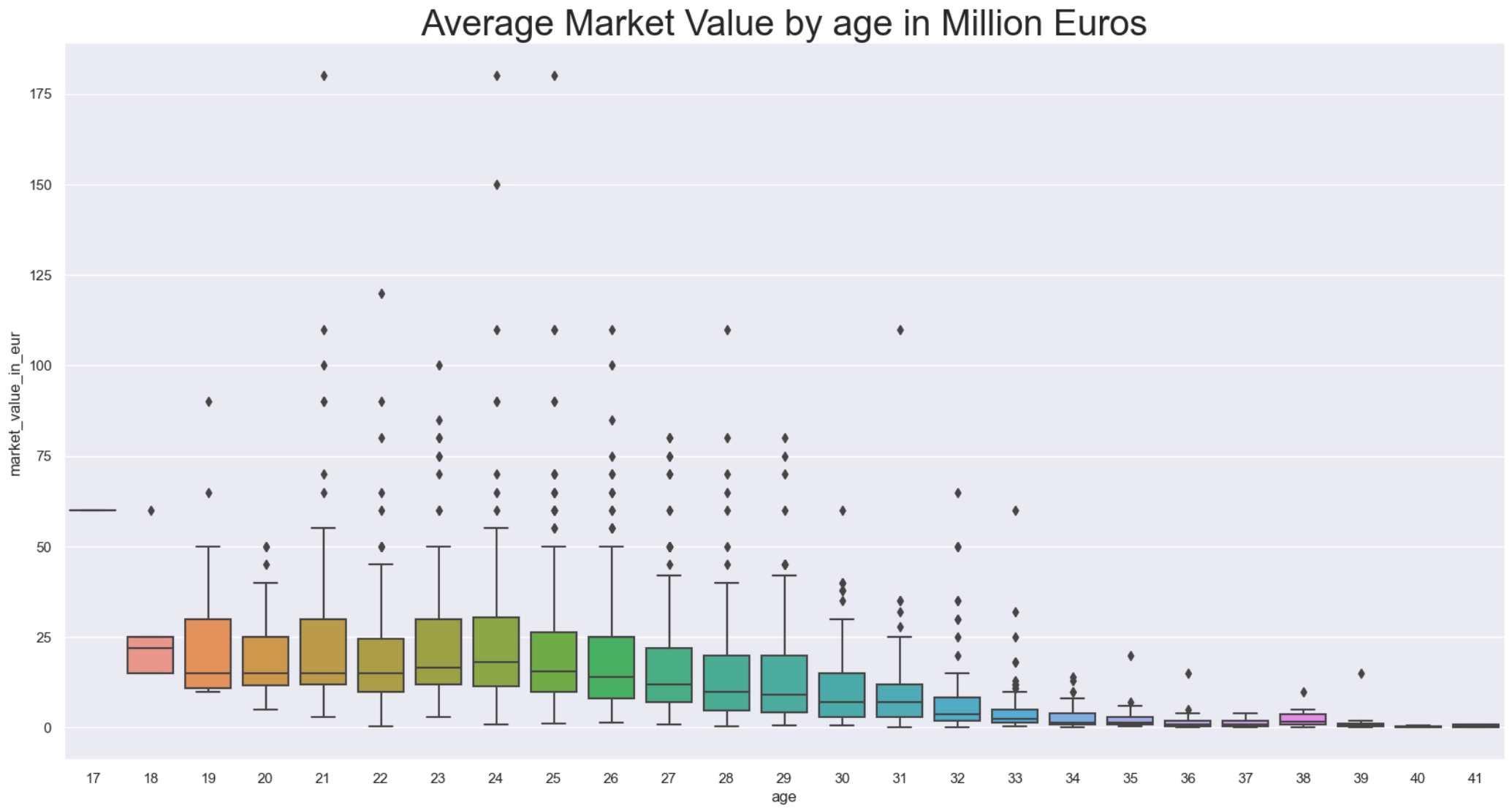


Game Visualizations



- As can be seen, usually the home teams score more goals. So hosting a match has an impact on the outcome of the game.

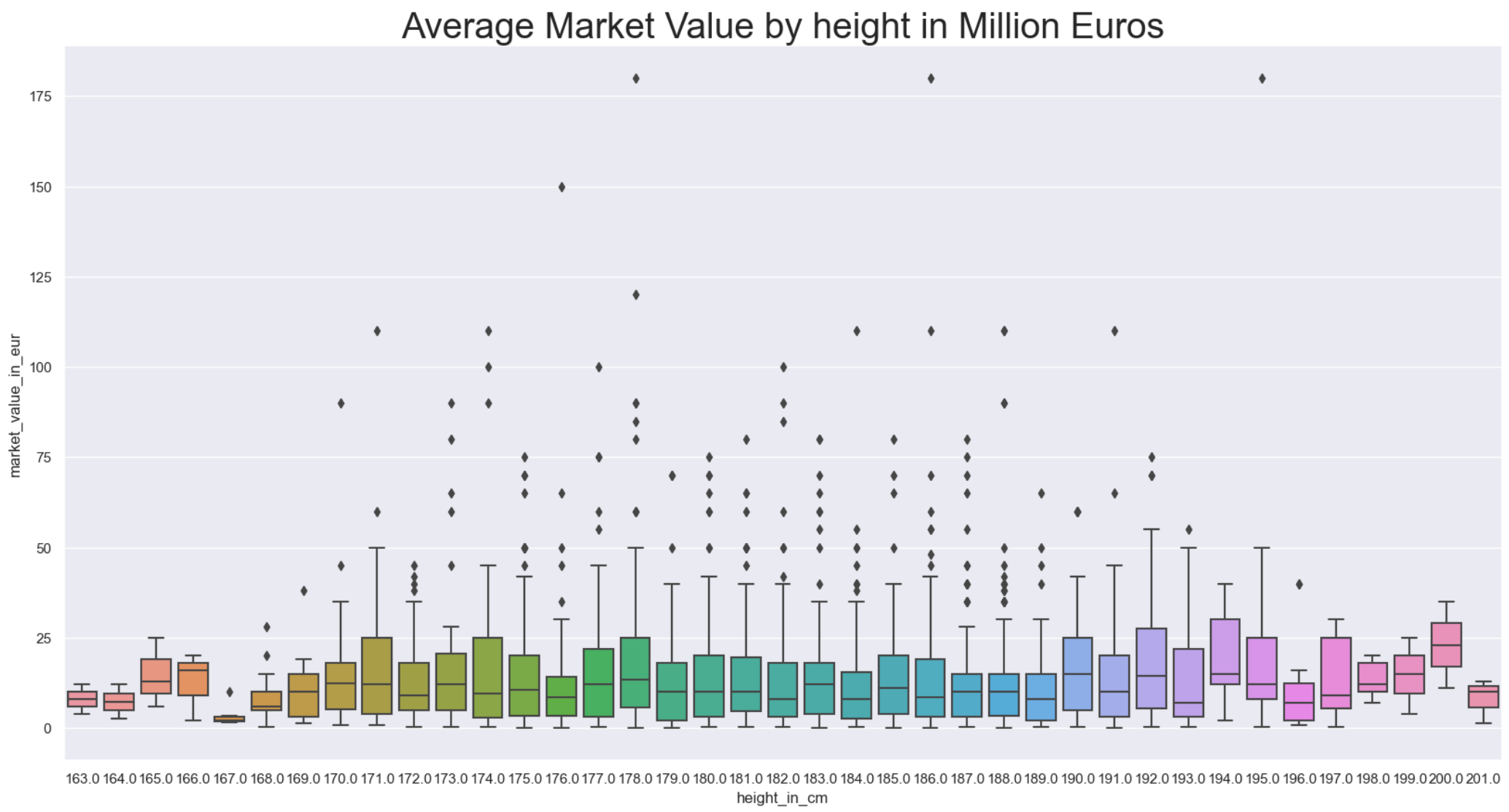
Data vizualisation of market value by age

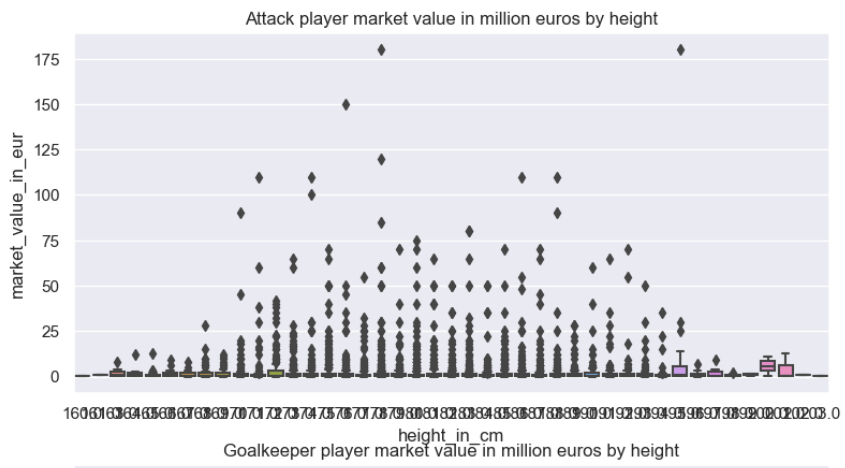


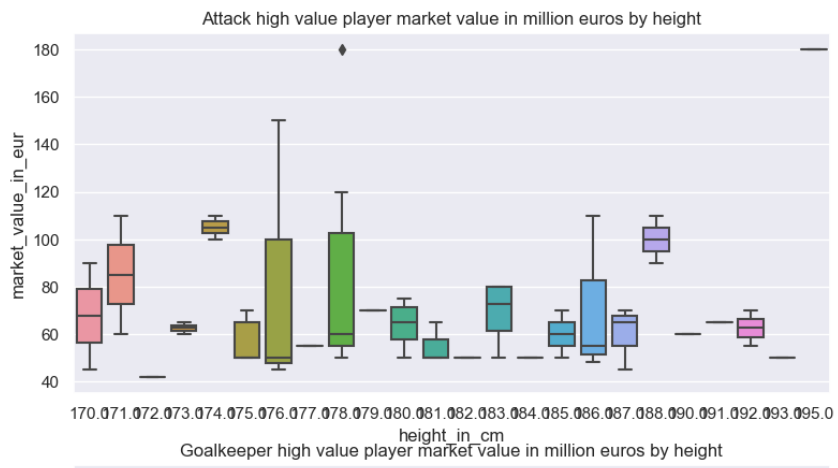
Observations on age

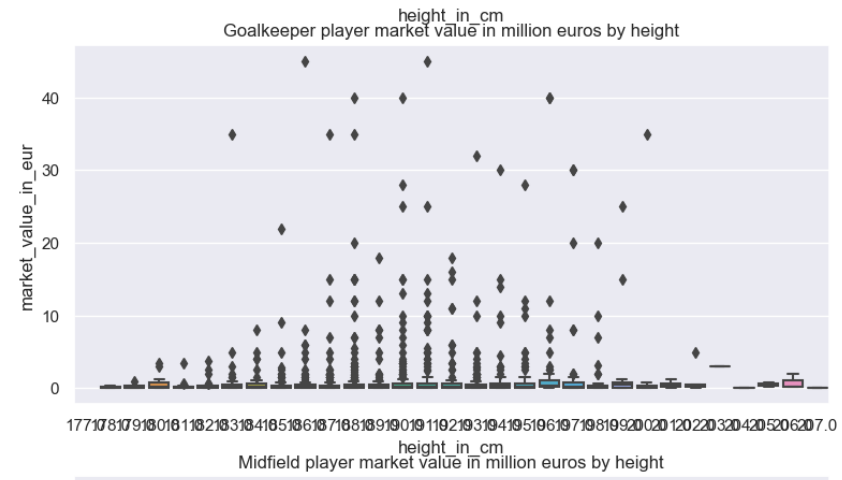
> The age distribution of players is a skewed normal distribution starting at 17 years of age and going up 50 with an average age of 29. Players with a market value to over 400,000 euros form a normal distribution ranging from 19 to 32 years of age, with a significant number at age 25.

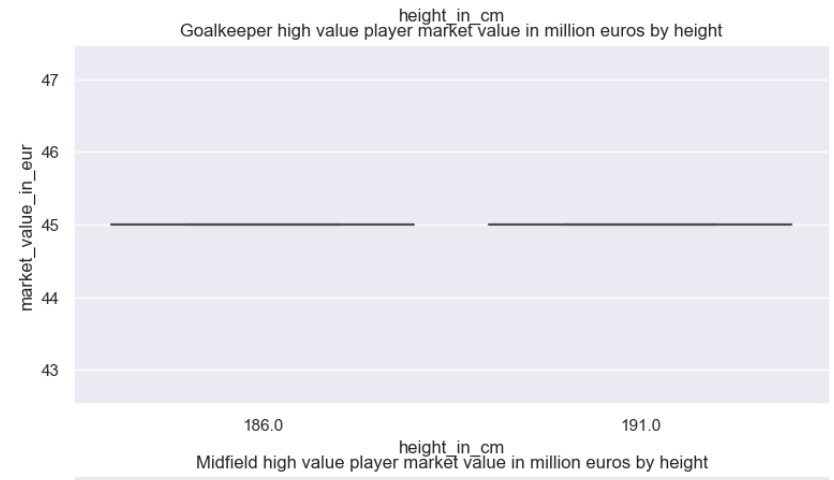
Data visualisation of market value by height

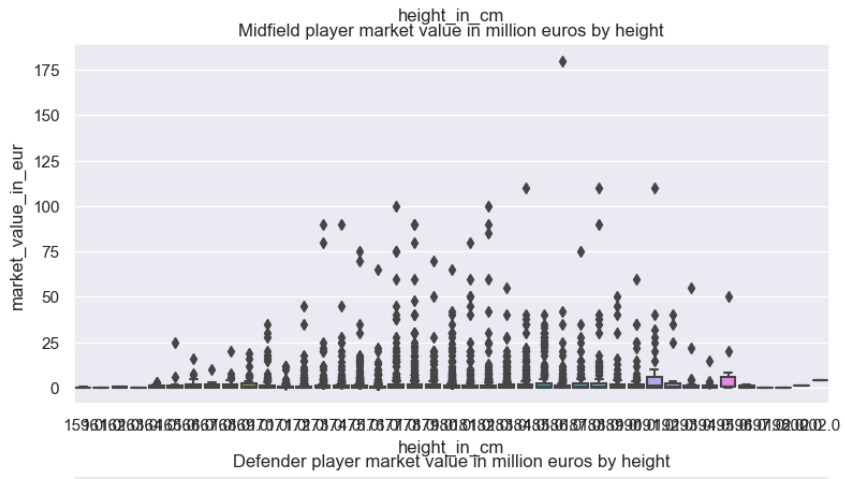


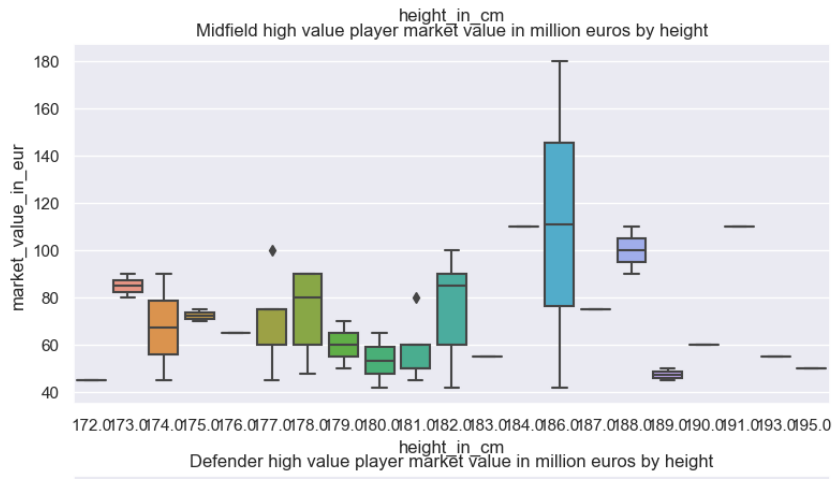


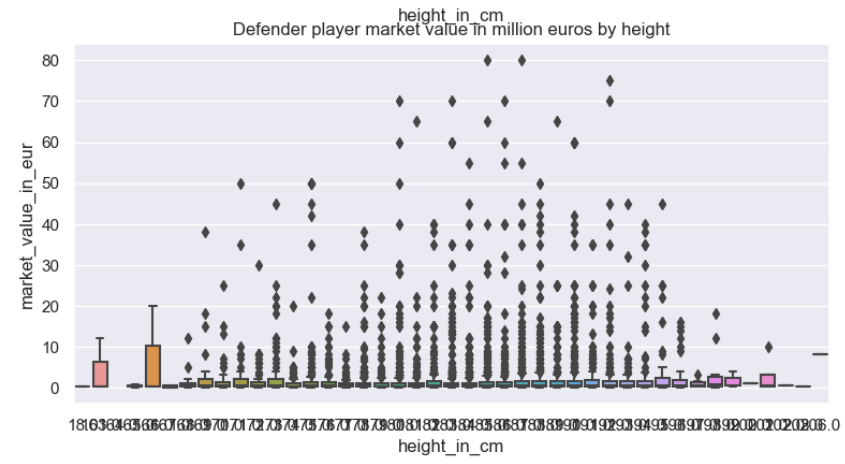


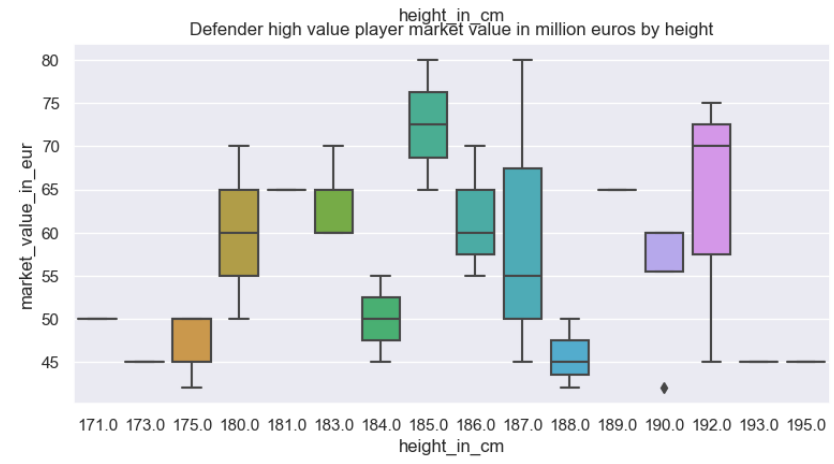












Observations on height

> The height distribution of players is a normal distribution ranging from 160cm to 200cm with an average age of 181cm. The average height for heigh value players is about 1cm taller that the general average.
>
> There are relatively fee high value goal keepers, but they are all tall ranging froom 188 to 200cm.
>
> There appears to be a normal like relationship between height and market value for each position when you look at all players, but does not appear to be any significant relationship between height and market value in the most valuable players.

Top players by Market Value

	name	market_value_in_eur	last_season
21474	Erling Haaland	180000000.0	2023
25392	Jude Bellingham	180000000.0	2023
18682	Kylian Mbappé	180000000.0	2023
19862	Vinicius Junior	150000000.0	2023
21891	Bukayo Saka	120000000.0	2023
9419	Harry Kane	110000000.0	2023
21158	Lautaro Martínez	110000000.0	2023
21013	Victor Osimhen	110000000.0	2023

	name	highest_market_value_in_eur	last_season
18682	Kylian Mbappé	200000000.0	2023
5941	Neymar	180000000.0	2022
21474	Erling Haaland	180000000.0	2023
25392	Jude Bellingham	180000000.0	2023
9496	Raheem Sterling	160000000.0	2023
7171	Kevin De Bruyne	150000000.0	2023
19862	Vinicius Junior	150000000.0	2023

2.3 Collating All Player Data

Having look at each the data in each of the data files, going to pull all of the data together for each player, so that we can look at feature importance and start to model transfer values.

after filtering players with the last season of 2023, and dropping some columns, ((['current_club_id', 'city_of_birth', 'date_of_birth','first_name', 'last_name', 'player_code', 'image_url', 'url']) **merged_players_df** have been created.

games_df and **appearances_df** have been merged together. and the result is the final data frame that we will run the machine learning models on it.

Select the top 10 players based on the sum of goals :

```
      player_name  goals
2004   Erling Haaland    51
3716    Kylian Mbappé    36
5608  Robert Lewandowski    33
1954    Enner Valencia    32
2565      Harry Kane    32
3486    Karim Benzema    31
4514     Mehdi Taremi    31
6621    Victor Osimhen    31
4696    Mohamed Salah    29
3763   Lautaro Martínez    28
```

After merging all data frames together the output is **merged_players_df**. it contains this columns:

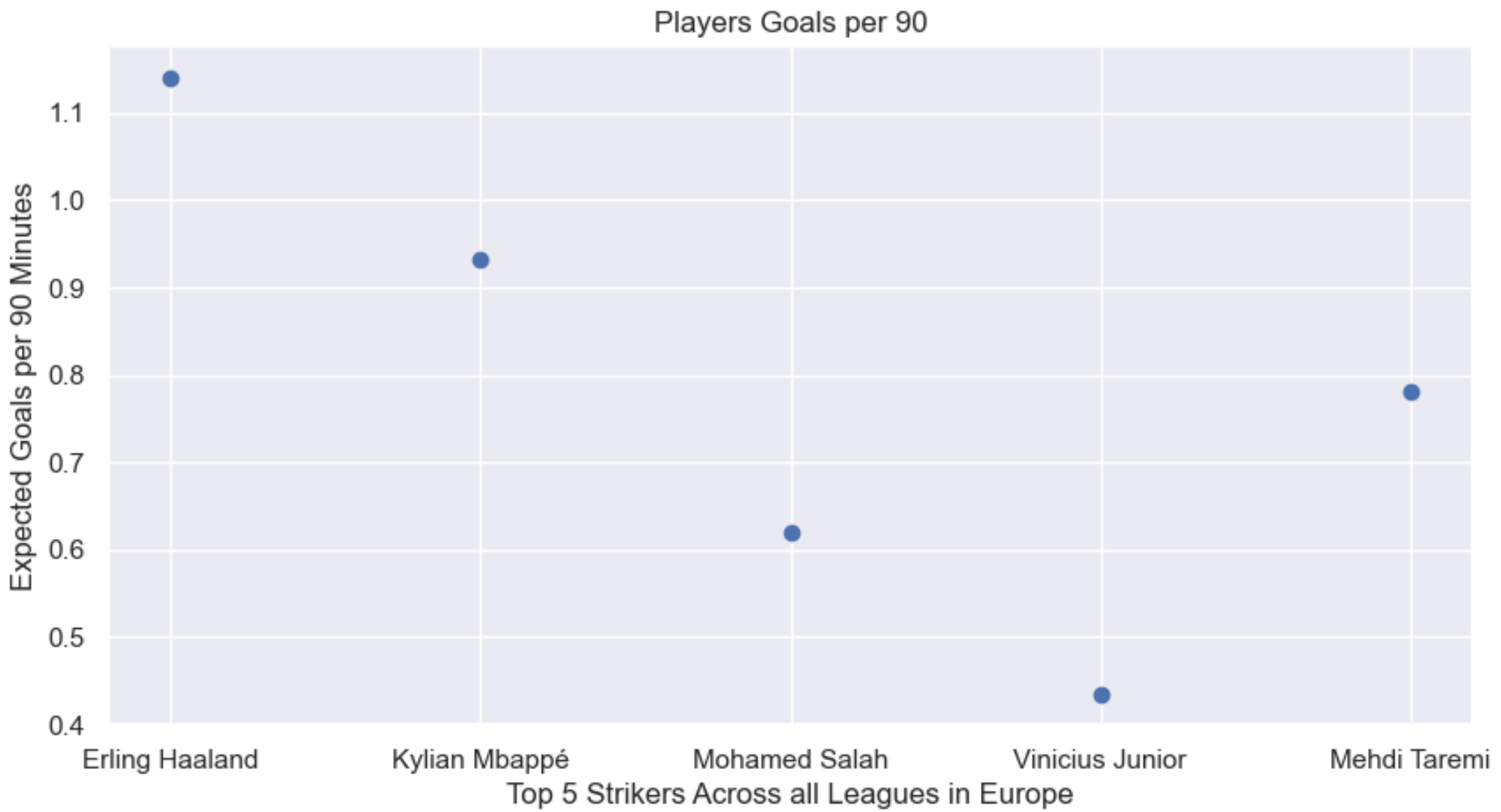
```
Index(['games_2022', 'minutes_played_2022', 'goals_2022', 'assists_2022', 'goals_against_2022',
'goals_for_2022', 'clean_sheet_2022', 'position', 'sub_position', 'last_season', 'foot', 'height_in_cm', 'age',
'country_of_birth', 'club_value', 'squad_size', 'current_club_domestic_competition_id',
'term_days_remaining', 'market_value_in_eur', 'highest_market_value_in_eur', 'yellow_cards_2022',
'red_cards_2022', 'direct_goal_contribution2022', 'goals_per_90', 'score_2022', 'price_last_year'],
dtype='object')
```

further feature engineering will be done on the data set to get ready for the models.

2.4 feature engineering

2.4.1 direct goal contribution , 2.4.2 goals per 90 min

These two features were made according to the previous features. based on total goals and assist for direct goal contribution, and minutes played and goals for creating goals per 90 min.



2.4.3 score_2022

this feature created based on this metrics:

Position	Minutes Played	Goals	Assists	Goals Against	Clean Sheet	Yellow Cards	Red Cards
GK	0.1	0	0	0	0.4	-0.1	-0.2
MD	0.2	0.3	0.3	0	0	-0.1	-0.2
DE	0.2	0	0	-0.6	0.3	-0.1	-0.2
AT	0.2	5	0.2	0	0	0	0

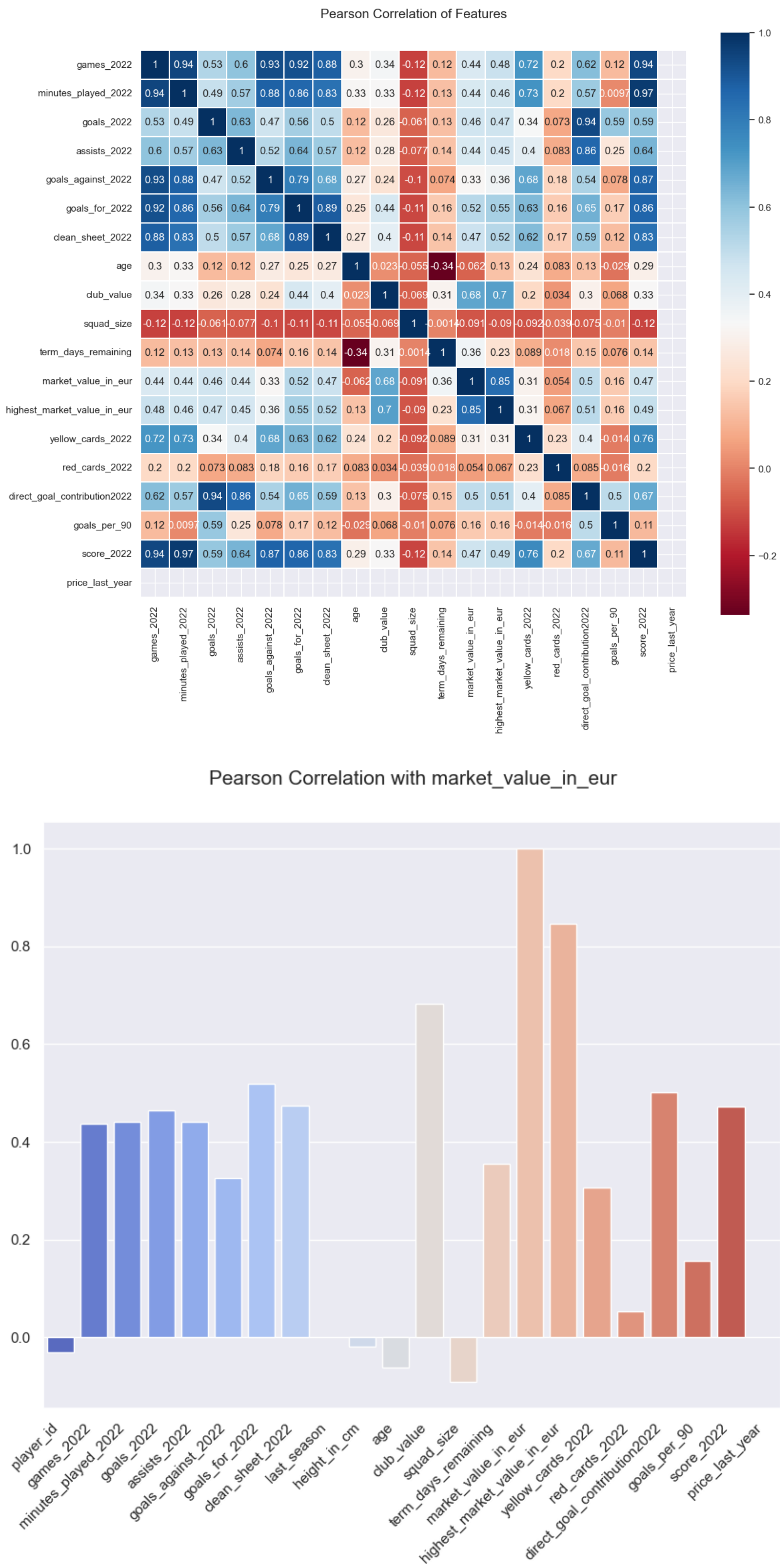
```
                name  position  score_2022
19862  Vinicius Junior    Attack      1071.0
21474   Erling Haaland    Attack      1061.6
9419       Harry Kane    Attack      1009.6
10048    Mohamed Salah    Attack       990.4
21158  Lautaro Martínez    Attack       965.8
3253  Robert Lewandowski    Attack       957.4
14313   Bruno Fernandes  Midfield       949.8
21719   Pedro Gonçalves    Attack       940.0
15238   Marcus Rashford    Attack       908.6
9909          Danilo  Defender       906.4
19742  Federico Valverde  Midfield       885.7
```

The correlation between 'score_2022' and 'market_value_in_eur' is: 0.4719641270796478 5

> It shows that feature should be useful in model!

2.5 Statistical Test

Pearson Correlation of Features



> this two plots show correlations between columns, first for **games_2022,** and the second one is for **market_value_in_eur.**

Hypothesis testing

---

1. test whether there is a significant difference in the 'market_value_in_eur' between players with different foot preferences (left and right foot)

F-statistic: 1.2185291313582847
P-value: 0.29573427267093705
Fail to reject the null hypothesis: There is no significant difference in market values between foot preference groups.

2.Analysis of Variance (ANOVA) test to investigate whether there is a significant difference in the average market values between players of different positions.

ANOVA Results:
F_onewayResult(statistic=19.506551702364632, pvalue=5.724310966553086e-16)

There is a significant difference in the average market values between players of different positions.

3. market values of players across different age groups

ANOVA Results for Age Groups:
F_onewayResult(statistic=36.884733657931115, pvalue=1.5666088444352123e-30)

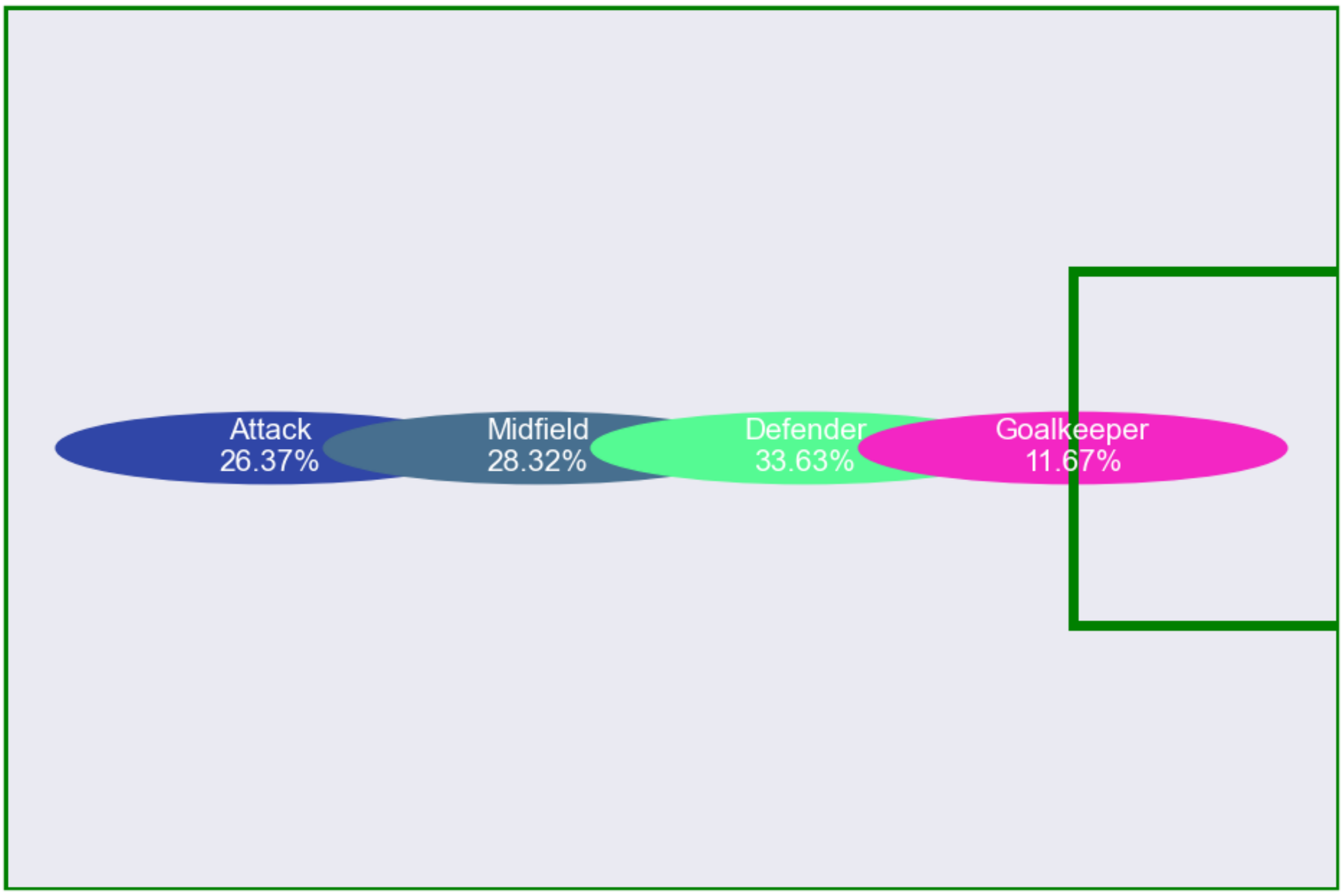
There is a significant difference in the market values of players across different age groups.

2.6 Handling missing values

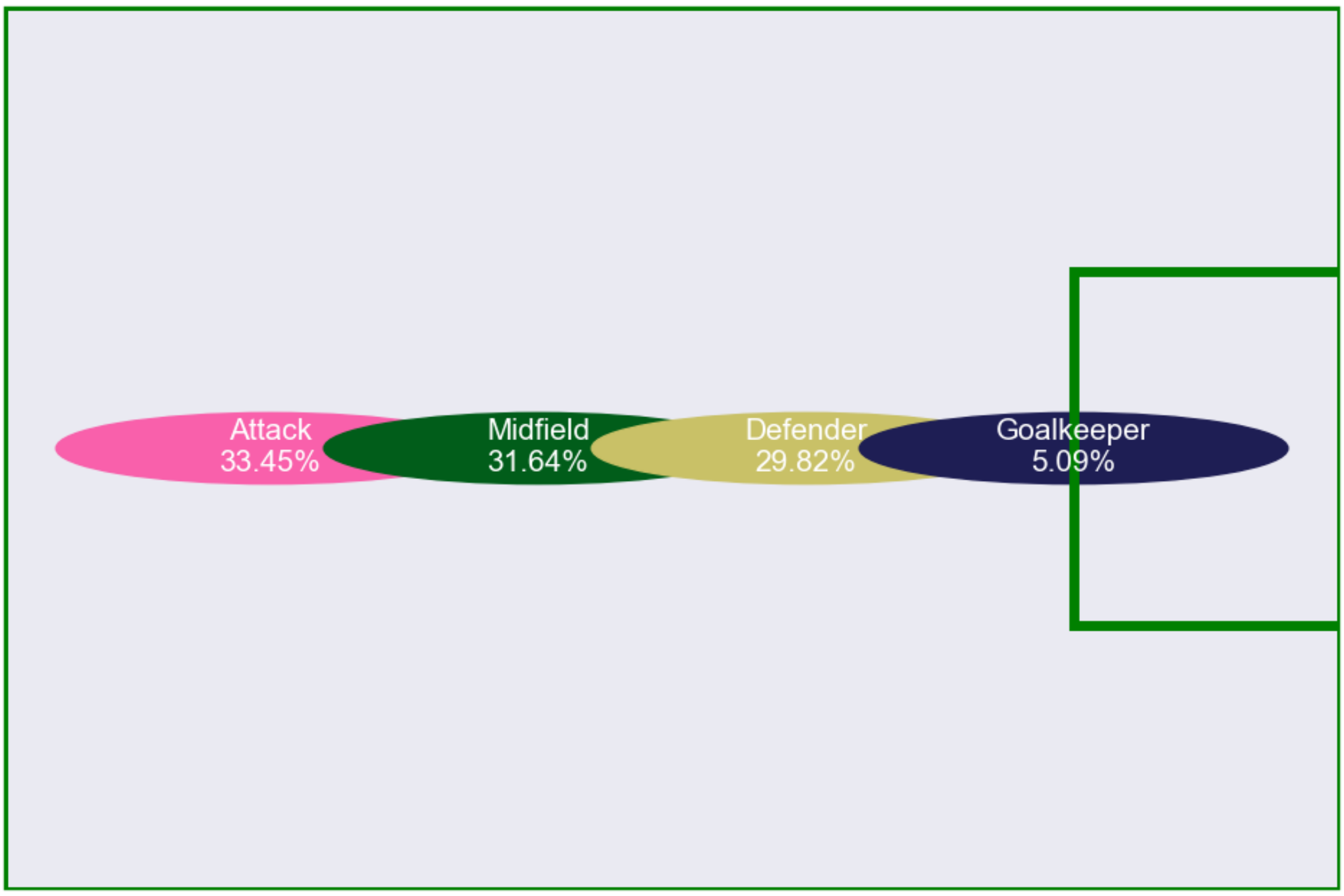
> Handling missing values by imputing the mean for numerical columns and the mode (most common value) for categorical columns.

Distribution of players based on positions :

It is not without grace looking at this position distribution. it gives insightful idea of distribution of players in the football pitch.



Distribution of players with 30.000.000 and more market values:



2.7 Encoding Categorical Variables:

This columns are categorical. so we done a one hot-encoding, to use them in or model. because they are valuable columns!

['position', 'sub_position', 'foot', 'country_of_birth', 'current_club_domestic_competition_id']

> Now after splitting data in **test_size=0.2**, we are ready to predict the market price of players!

2.8 Models

Model: Linear Regression R-squared Score: 0.7430768046370413

Model: Decision Tree R-squared Score: 0.686818655804631

Model: Random Forest R-squared Score: 0.8572988112124682

Model: Gradient Boosting R-squared Score: 0.8623650559226523

Model: Support Vector Machine R-squared Score: -0.1634947452556612

Observation on model

The negative R-squared score for the Support Vector Machine indicates that this model is not performing well on your data. It's possible that the SVM model may not be suitable for your regression task, or there might be an issue with the model's hyperparameters. Consider adjusting the parameters or trying different models to improve performance.

In contrast, both Random Forest and Gradient Boosting seem to be performing well, with high R-squared scores.

It's clear that the SVM model is not performing well, as indicated by the negative R-squared score and high error metrics. The other models (Linear Regression, Decision Tree, Random Forest, and Gradient Boosting) are providing better results, with Random Forest and Gradient Boosting showing particularly strong performance.

Model: Neural Network

- Neural Network Structure: A simple neural network architecture was employed with two hidden layers containing 100 and 50 neurons, respectively. The activation function used is the default rectified linear unit (ReLU).
- Training Parameters: The model was trained for a maximum of 500 iterations, and the random state was set to 42 for reproducibility.

Model: Neural Network
R-squared Score: 0.7373565166708334

> A neural network regression model with two hidden layers (100, 50 neurons) achieved an R-squared score of 0.7374 on the test set. The model was trained for 500 iterations using standardized data.

Model: Neural Network (More Complex)

R-squared Score

The R-squared score for the more complex neural network regression model on the test set is approximately 0.7557.

> The more complex neural network model demonstrates improved performance, as indicated by the higher R-squared score compared to the previous model. This suggests that increasing the model's complexity by adding more layers and neurons has contributed to better predictive capabilities.

Best hyperparameters for Random Forest and Gradient Boosting:

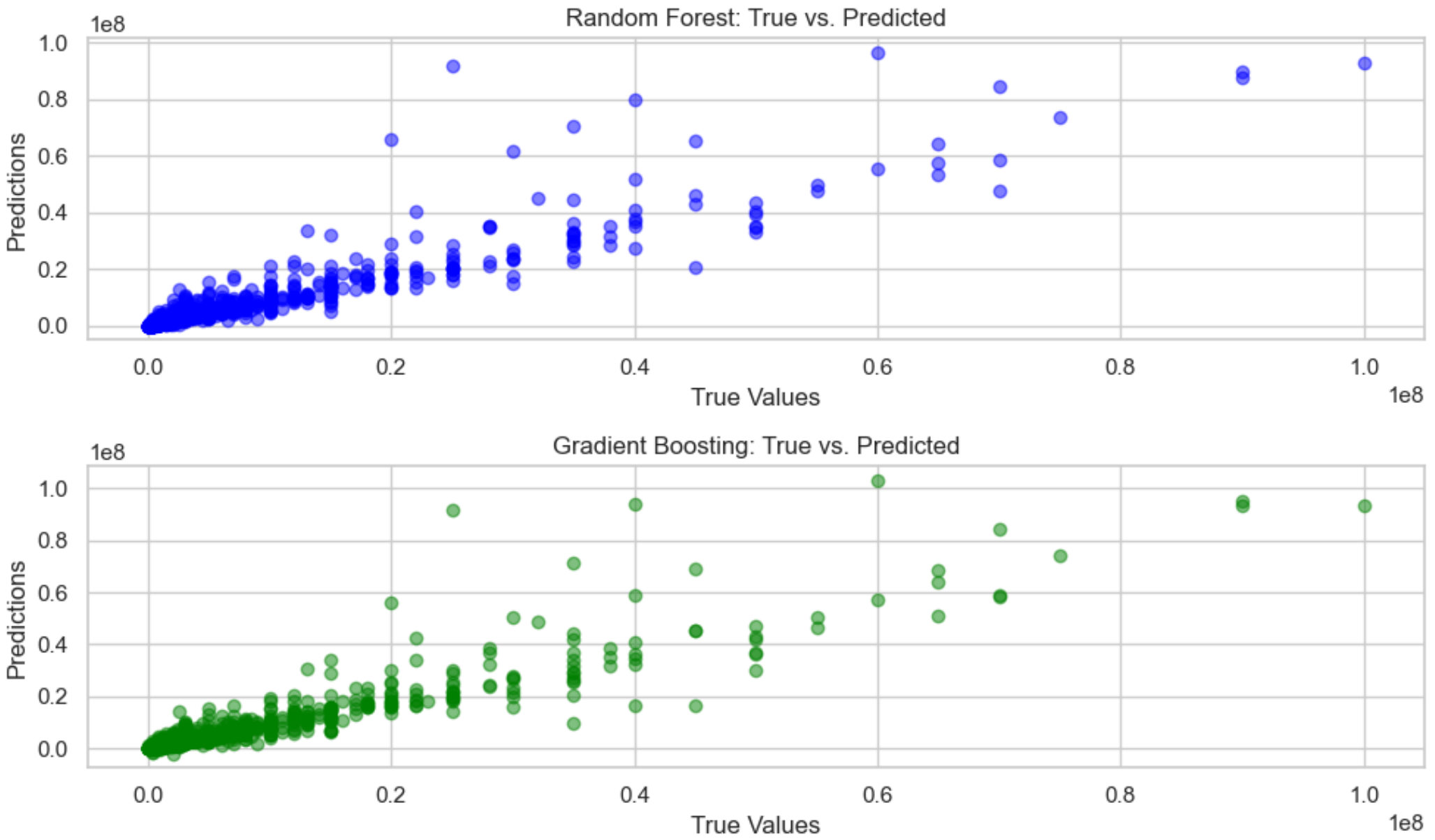
After performing **Grid Search** we found the best possible Models.

Best Hyperparameters for Random Forest:
{'max_depth': 10, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 200}

Best Hyperparameters for Gradient Boosting:
{'learning_rate': 0.1, 'max_depth': 5, 'min_samples_leaf': 1, 'min_samples_split': 10, 'n_estimators': 200}

Random Forest R-squared Score with Best Hyperparameters: 0.8787846261124929
Gradient Boosting R-squared Score with Best Hyperparameters: 0.8834115304962495

> These results indicate that after hyperparameter tuning, both Random Forest and Gradient Boosting models have improved R-squared scores, suggesting better predictive performance.



Random Forest Metrics:

Mean Squared Error: 17583657502782.309

Mean Absolute Error: 1490477.8322360215

Gradient Boosting Metrics:

Mean Squared Error: 18959108041409.75

Mean Absolute Error: 1524660.2525385667

> It looks like the Mean Squared Error (MSE) and Mean Absolute Error (MAE) for both Random Forest and Gradient Boosting models are quite large, indicating a significant level of error in the predictions. Large values for these metrics suggest that the models may not be performing optimally on the given data.

Conclusion

In general, a lot of work was done on the data. New features were made, statistical tests were done, many maps and plots were drawn, and finally, Machine learning models were used to predict players' prices!

Comparison of these models showed that **Random Forest** and **Gradient Boosting** performed well, Because of the tree nature! But in other metrics like **MES**, they were weak!

As a result, finding and estimating the price of players is a difficult task because it depends on several factors, one of the most important of which is sponsorship, because if a player has a good performance, but if he is in a weak league, the price is not high compared to the other players so that make mistake on the model. for example if we compare each leagues players together, the model would have a better result.

another reason that models have high MSE because prices have many distance between together. for example **Messi** have 180.000.000 price and the next player have 160.000.000 and they are in million scale! this is in much big scale. so validating models based on **MSE** is not a good metric to conclusion models result.