Data Preprocessing

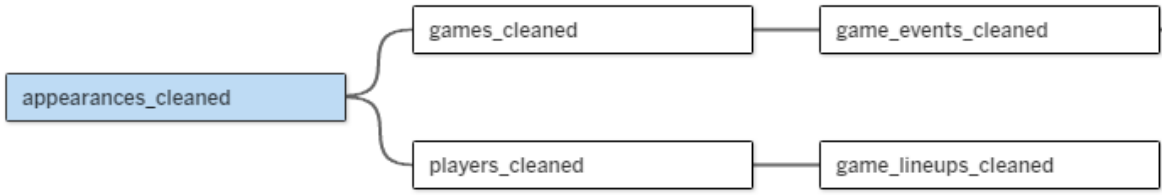
1.Data Merging

Python:

- Combine the data by matching the shared columns across the data files.
- The solution is available within the Python notebook.
- Save the resulting files as a CSV named ‘Football_data_merged.csv’.
- An "outer join" can be an alternative when merging tables.

Tableau:

- I've consolidated all five data files into one Excel file, with each file in a separate worksheet.
- Import the data into Tableau.
- Merge the files based on their shared columns, as illustrated below



2. Data Dictionary:

In ChatGPT, I craft my prompt as follows:

- "As a data analyst, generate a data dictionary for the provided dataset, detailing descriptions in a table format."
- I copied the first few rows (10-15 samples) from the 'Football_data_merged.csv' file, including the column headers, and shared it in ChartGPT.

Solution provided by ChatGPT

S.No	Column Name	Data Type	Description
1	game_id	int64	Unique identifier for each game
2	competition_id_x	object	Identifier for the competition
3	season	int64	Season year
4	round	object	Matchday or competition round
5	date_x	object	Date of the game
6	home_club_goals	int64	Goals scored by the home club
7	away_club_goals	int64	Goals scored by the away club
8	home_club_position	int64	League position of the home club
9	away_club_position	int64	League position of the away club
10	home_club_manager_name	object	Name of the home club manager
11	away_club_manager_name	object	Name of the away club manager
12	stadium	object	Stadium where the game was played
13	attendance	int64	Number of attendees at the game
14	referee	object	Name of the referee
15	home_club_name	object	Name of the home club
16	away_club_name	object	Name of the away club
17	aggregate	object	Aggregate score across multiple games
18	competition_type	object	Type of competition (domestic or international)
19	appearance_id	int64	Unique identifier for player's appearance in the game
20	player_id	int64	Unique identifier for the player
21	date	object	Date of the appearance
22	player_name_y	object	Name of the player
23	competition_id_y	object	Identifier for the competition (repeated)

24	yellow_cards	int64	Number of yellow cards issued
25	red_cards	int64	Number of red cards issued
26	goals	int64	Number of goals scored by the player
27	assists	int64	Number of assists
28	minutes_played	int64	Minutes played by the player
29	name	object	Name of the player (repeated)
30	last_season	int64	Last season played
31	current_club_id	float64	Current club identifier
32	player_code	object	Code of the player
33	country_of_birth	object	Country of birth of the player
34	date_of_birth	object	Date of birth of the player
35	sub_position	object	Sub position of the player
36	position_player	object	Main position of the player
37	foot	object	Dominant foot of the player
38	height_in_cm	int64	Player's height in centimeters
39	market_value_in_eur	int64	Market value of the player in euros
40	highest_market_value_in_eur	int64	Highest market value of the player in euros
41	contract_expiration_date	object	Expiration date of the player's contract
42	agent_name	object	Name of the player's agent

4. Generate Business Objectives using ChatGPT

- The data structures including column names and their descriptions are provided below. As a business analyst, provide all possible business objectives for this data.

1. Performance Analysis

Objectives:

- **Objective 1:** Identify top-performing players and analyze trends in performance metrics across seasons and positions.
- **Objective 2:** Utilize statistical methods like probability and logistic regression to predict player performance and game outcomes.

Business Questions:

- **Descriptive Analysis Questions:**
    1. Who are the top 5 goal scorers in the latest season?
    2. How does average playing time vary among players in different positions?
    3. What is the trend in player performance across seasons for top players?
- **Probability Questions:**
    1. What is the probability of a player scoring in a game if they have an assist record?
    2. Calculate the overall probability of a player scoring in a game.
- **Statistical Central Limit Theorem:**
    1. What is the average number of goals scored by players using a sample of games, and does the central limit theorem apply?
- **Logistic Regression Questions:**
    1. Can the number of goals scored by a player predict whether they will be in the starting lineup for the next game?
    2. Can we predict whether a player will receive a yellow card in their next game based on the number of yellow cards they have received in the current season?

2. Player Profile and Market Value

Objectives:

- **Objective 1:** Examine market value trends by player position, country, and competition type to identify key factors influencing player valuations.
- **Objective 2:** Develop predictive models using scoring frequency and other player attributes to estimate market values.

Business Questions:

- **Descriptive Analysis Questions:**
    1. How does market value vary between players in different positions?
    2. How are total market values of players distributed across their countries of birth?
    3. How have average market values of players evolved over the years across different competition types?
- **Regression Questions:**
    1. Can we predict a player’s market value based on their scoring frequency? (Simple Linear Regression)
    2. Predict a player's market value based on significant features such as height, minutes played, goals, etc. (Multiple Linear Regression)

3. Team Comparison

Objectives:

- **Objective 1:** Analyze and compare team strategies and outcomes in home vs. away games across different managers and seasons.

Business Questions:

- **Descriptive Analysis Questions:**

1. Which managers have led teams to score the most goals in home games?
2. Which managers have led teams to score the most goals in away games?
3. How do top-performing teams manage their goal scoring at home venues?
4. How do top-performing teams manage their goal scoring in away matches?

- **Hypothesis Testing Question:**
  1. Determine whether there is a significant difference in the average goals scored by home clubs vs. away clubs using a 5% significance level.

4. Attendance and Stadium Analysis

Objectives:

- **Objective 1:** Explore trends in match attendance over the years and identify factors that influence crowd turnout at various stadiums.

Business Questions:

- **Descriptive Analysis Questions:**
  1. How do fluctuations in average match attendance vary over the years?
  2. Which stadium consistently draws the highest average attendance?
  3. What is the average attendance across different competition types?
- **Classification Question:**
  1. Categorize matches into high, moderate, and low attendance based on match characteristics and other significant factors.

5. Referee Analysis

Objectives:

- **Objective 1:** Assess referee decision-making patterns and their impact on game outcomes.

Business Questions:

- **Descriptive Analysis Questions:**
  1. Which referees have issued the most yellow and red cards?
  2. How do yellow and red cards issued by referees vary by player position?
  3. Which referee officiated the games with the most goals?

6. Substitution Patterns

Objectives:

- **Objective 1:** Analyze substitution strategies and their effectiveness across different teams and competitions.

Business Questions:

- **Descriptive Analysis Questions:**

1. How are substitutions distributed among different player positions in football matches?
2. Which substitute players had the highest average minutes played per game in 2020?
3. How does the frequency of substitutions vary across different types of football competitions?

7. Event Analysis

Objectives:

- **Objective 1:** Examine the types and frequencies of game events and their influence on match outcomes.
- **Objective 2:** Assess the effectiveness of different types of offensive and defensive plays.

Business Questions:

- **Descriptive Analysis Questions:**
  1. How does the frequency of fouls vary among different player positions?
  2. What are the most common types of game events?
  3. How does the total number of red and yellow cards issued differ between starting lineup players and substitutes?
- **Hypothesis Testing Question:**
  1. Is there a significant difference in the average number of goals scored by players of different preferred foot (left or right)?

8. Competition Analysis

Objectives:

- **Objective 1:** Compare player performance and strategic approaches across different competition types.

Business Questions:

- **Descriptive Analysis Questions:**
  1. How do goal contributions vary across different competition types?
  2. Which competitions have the highest minutes played per game in the last season?
  3. Which competition type shows the highest level of game discipline?

9. Player Attributes and Demographics

Objectives:

- **Objective 1:** Explore the relationship between player demographics and their on-field performance.

Business Questions:

- **Descriptive Analysis Questions:**
    1. What is the average height of players across different positions?
    2. What is the average age of players across different positions?
    3. How does scoring ability vary by country of origin?
- **K-means Clustering Question:**
    1. Identify groups of players based on their attributes and demographics.

10. Contract Management

Objectives:

- **Objective 1:** Analyze patterns and trends in player contracts to optimize team management and player retention strategies.
- **Objective 2:** Evaluate the influence of agents on player contracts and performances.

Business Questions:

- **Descriptive Analysis Questions:**
    1. What percentage of players' contracts are set to expire each year?
    2. Which agents represent the most players among top performers?
    3. Who are the top agents managing the highest value players?

Solution to the analysis one by one

1. Performance Analysis

Objectives:

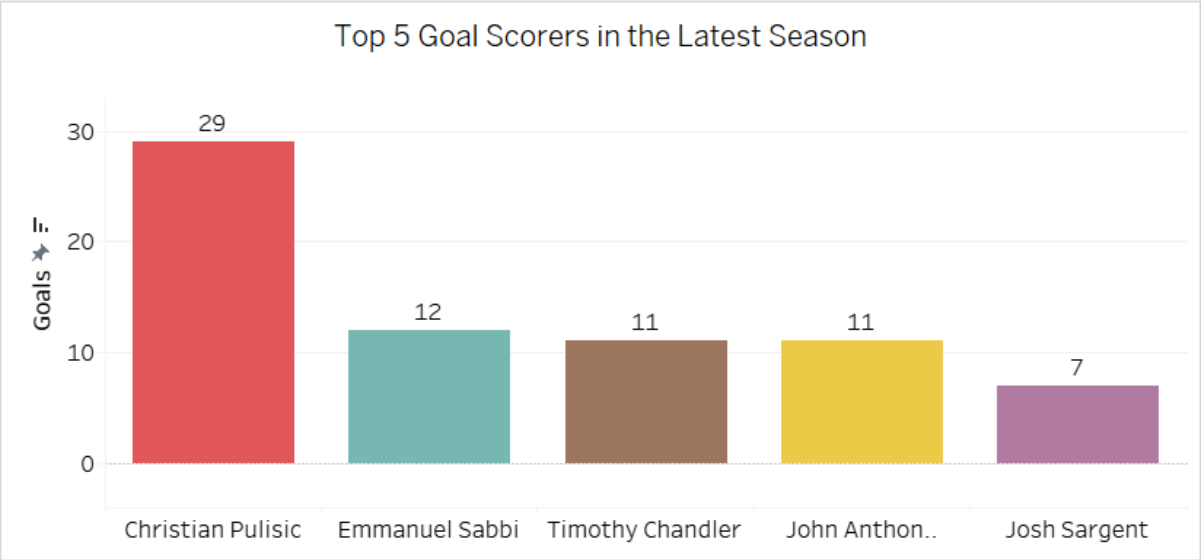
- **Objective 1:** Identify top-performing players and analyze trends in performance metrics across seasons and positions.
- **Objective 2:** Utilize statistical methods like probability and logistic regression to predict player performance and game outcomes.

Business Questions:

Descriptive Analysis Questions:

1.Who are the top 5 goal scorers in the latest season?

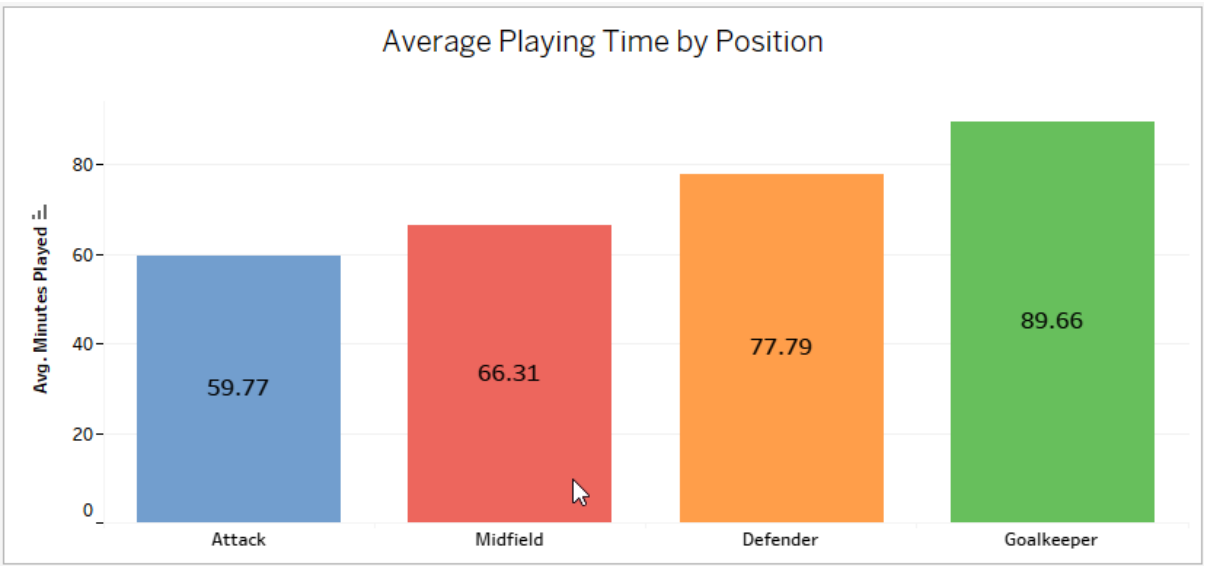
Using Tableau



Interpretation:

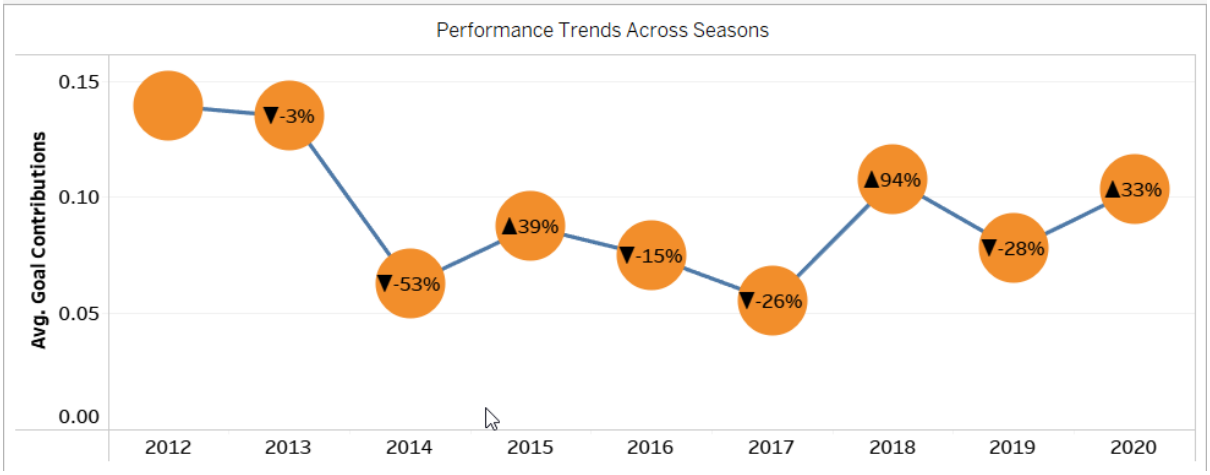
- Christian Pulisic and Timothy Chandler led with 5 goals each, significantly contributing to their teams' performances in the 2020 season. Josh Sargent follows with 3 goals, while Emmanuel Sabbi and John Anthony Brooks scored just 1 goal each. This distribution highlights the disparity in goal-scoring effectiveness among top players and suggests areas where teams might focus on enhancing offensive strategies or player support.

2.How does average playing time vary among players in different positions?



Interpretation:
Goalkeepers average the most minutes per game at 89.56, followed by defenders at 77.80, midfielders at 66.29, and attackers with the least at 59.86. This distribution suggests that goalkeepers typically play full matches, while other positions may see more rotation or substitution, affecting their average playing time.

3.What is the trend in player performance across seasons for top players?



Interpretation

The chart shows a clear downward trend in both the number of games played and goal contributions from top players across the seasons from 2012 to 2020. The parallel decline in these metrics suggests that fewer opportunities to play significantly impact players' ability to contribute goals. This trend highlights potential areas for strategic adjustments in player management and game participation to enhance performance.

Probability Questions:

Using Python

1.What is the probability of a player scoring in a game if they have an assist record?

Python Code:

```
# Calculate the probability of scoring given an assist

# P(Score | Assist)

players_with_assists = appearances[appearances['assists'] > 0]

probability_score_given_assist = (players_with_assists['goals'] > 0).mean()

print(f"The probability of a player scoring in a game given they have an assist is: {probability_score_given_assist:.2f}")
```

Output:

The probability of a player scoring in a game given they have an assist is: 0.15

Interpretation :

- The probability of a player scoring a goal in a game, given that they have provided at least one assist in that game. The output indicates that there's a 15% chance of a player scoring a goal if they have assisted in a game. This suggests that players who provide assists are more likely to score goals compared to those who don't.

2.Calculate the overall probability of a player scoring in a game.

Python Code:

```
total_games = len(appearances)  # Total number of player appearances

games_with_goals = (appearances['goals'] > 0).sum()  # Count of games where goals were scored by players

probability_of_scoring = games_with_goals / total_games  # Probability calculation

print(f"Overall probability of a player scoring in a game: {probability_of_scoring:.3f}")
```

Output:

Overall probability of a player scoring in a game: 0.084

Interpretation:

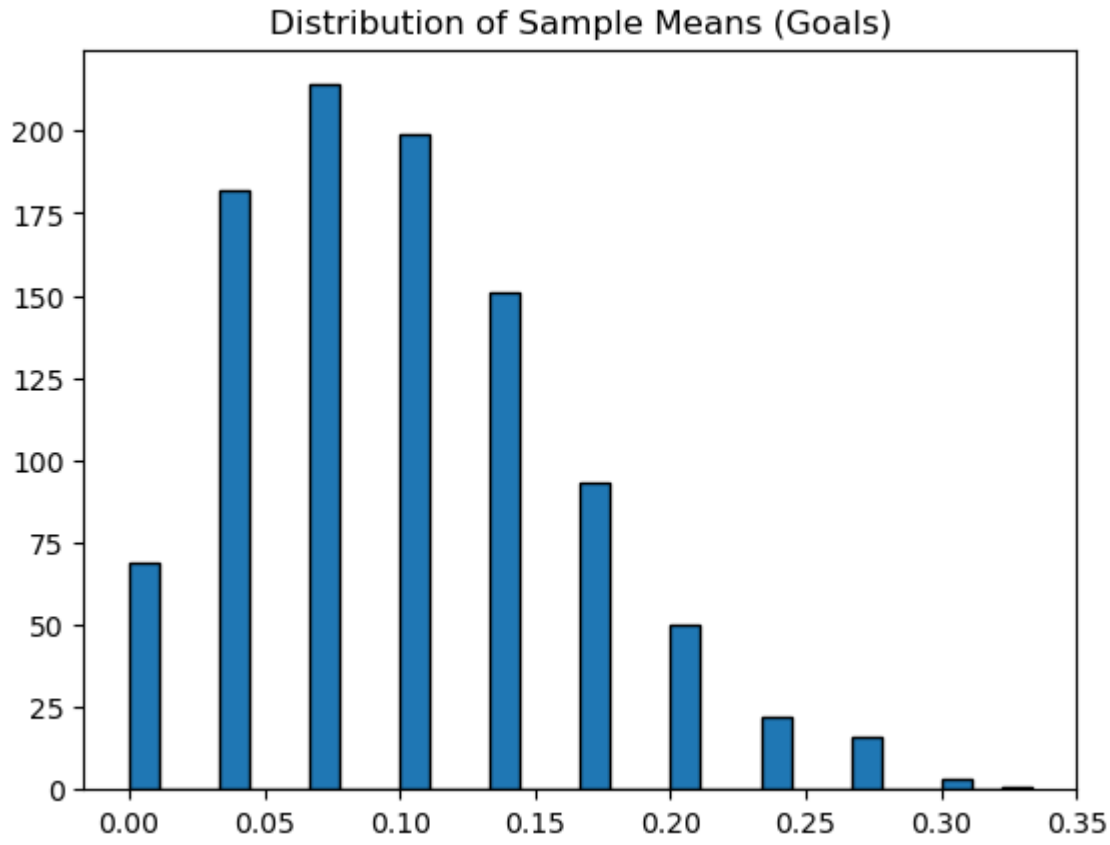
- The overall probability of a player scoring a goal in any given game, regardless of whether they have assisted or not. The output shows that there's an 8.4% chance of a player scoring in a game. This indicates that, on average, players score goals in a small fraction of the games they play.

Statistical Central Limit Theorem:

1.What is the average number of goals scored by players using a sample of games, and does the central limit theorem apply?

Python Code:

```
sample_means = []
for i in range(1000):
    sample =appearances['goals'].sample(30, replace=True)
    sample_means.append(sample.mean())
#Plot sample means
plt.hist(sample_means, bins=30, edgecolor='k')
plt.title('Distribution of Sample Means (Goals)')
plt.show()
```



Logistic Regression Questions:

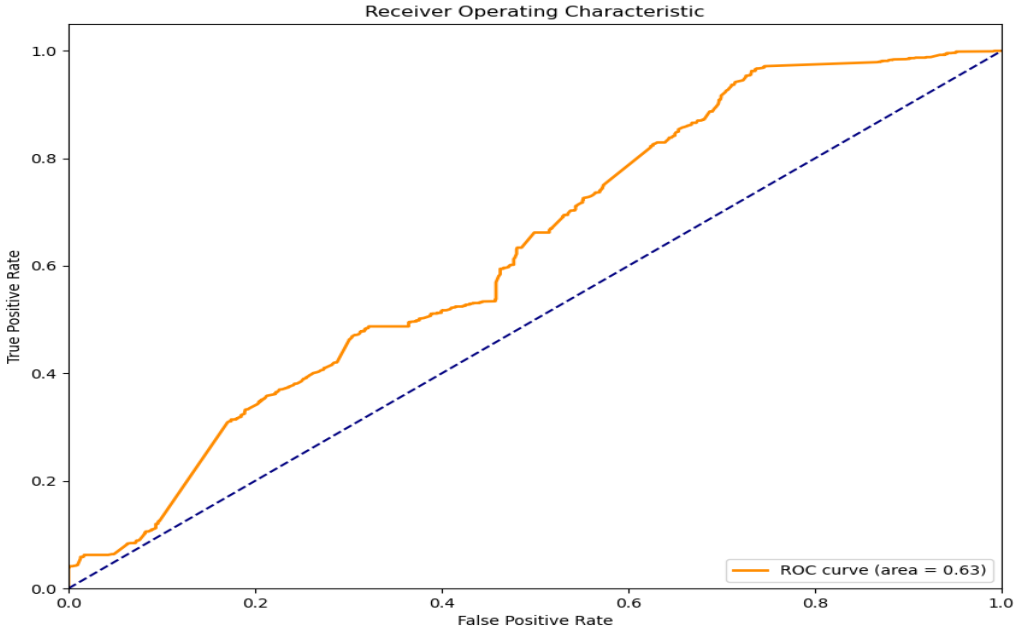
1.Can the number of goals scored by a player predict whether they will be in the starting lineup for the next game?

Python Code:

```
# Merge datasets to associate appearances with game details
logis = pd.merge(appearances, game_lineups, on='player_id', how='inner')
logis.isnull().sum()
#Outlier Treatment
sns.boxplot(data= logis)
plt.gcf().set_size_inches(20,6)
plt.xticks(rotation= 45)
drop_col=['appearance_id', 'game_id_x', 'player_id', 'date', 'player_name_x',
      'competition_id', 'game_lineups_id', 'game_id_y', 'number','player_name_y']
logis.drop(drop_col,axis=1,inplace=True)
logis['type'] = logis['type'].map({'starting_lineup': 1, 'substitutes': 0})
logis.head(3)
#Encoding purpose variable
from sklearn.preprocessing import LabelEncoder
pur_le=LabelEncoder()
logis['position']=pur_le.fit_transform(logis ['position'])
logis.head(1)
logis_type=logis[['type']]

logis.drop('type',axis=1,inplace=True)
cols= logis.columns
scaler=StandardScaler()
logis_scaled =scaler.fit_transform(logis)
logis_scaled= pd.DataFrame(logis_scaled, columns= cols)
logis_scaled.head()
logis_scaled= pd.concat([logis_scaled,logis_type], axis= 1)    #ignore_index= True
logis_scaled.head()

logis_scaled.info()
abs(logis_scaled.corr())>.7
# Plotting heatmap
plt.figure(figsize=(16,10))
sns.heatmap(logis_scaled.corr(), annot=True, cmap='YlGnBu')
#Creating test and training datasets
X=logis_scaled.drop('type', axis=1).values
y= logis_scaled ['type'].values
X_train, X_test, y_train, y_test=train_test_split(X,y, test_size= 0.3, random_state=0)
X_train.shape
X_test.shape
#Logistic Regression training the model

logreg= LogisticRegression()
logreg.fit(X_train, y_train)
```

```
#Using the trained model to predict the outcome of the X_Test dataset
y_pred=logreg.predict(X_test)
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
print(classification_report (y_test, y_pred))
# ROC Score
from sklearn.metrics import roc_auc_score
roc=roc_auc_score(y_test, logreg.predict_proba(X_test) [:,1])
roc
pip install -U scikit-learn
import matplotlib.pyplot as plt
from sklearn.metrics import roc_curve, auc
y_probs = logreg.predict_proba(X_test)[:, 1]

# Compute ROC curve and ROC area
fpr, tpr, _ = roc_curve(y_test, y_probs)
roc_auc = auc(fpr, tpr)

# Plotting
plt.figure(figsize=(10,8))
plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (area = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='navy', linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic')
plt.legend(loc="lower right")
plt.savefig('logistic_roc_curve_and_roc_area.png')
plt.show()
```

	precision	recall	f1-score	support
0	0.94	0.05	0.09	1314
1	0.63	1.00	0.77	2136
accuracy			0.64	3450
macro avg	0.79	0.52	0.43	3450
weighted avg	0.75	0.64	0.51	3450



Model Analyses Summary

Logistic Regression (Predicting Starting Lineup from Goals Scored)

ROC Curve (Chart): Shows moderate predictive ability with an AUC of 0.627.

Interpretaion

- The logistic regression model, predicting starting lineup based on goals scored, shows moderate predictive power with an AUC of 0.627. While it's better at identifying substitutes (recall of 1.00), it struggles with predicting starters (precision of 0.63). This indicates that the model might not be highly reliable for making accurate predictions.

2.Can we predict whether a player will receive a yellow card in their next game based on the number of yellow cards they have received in the current season?

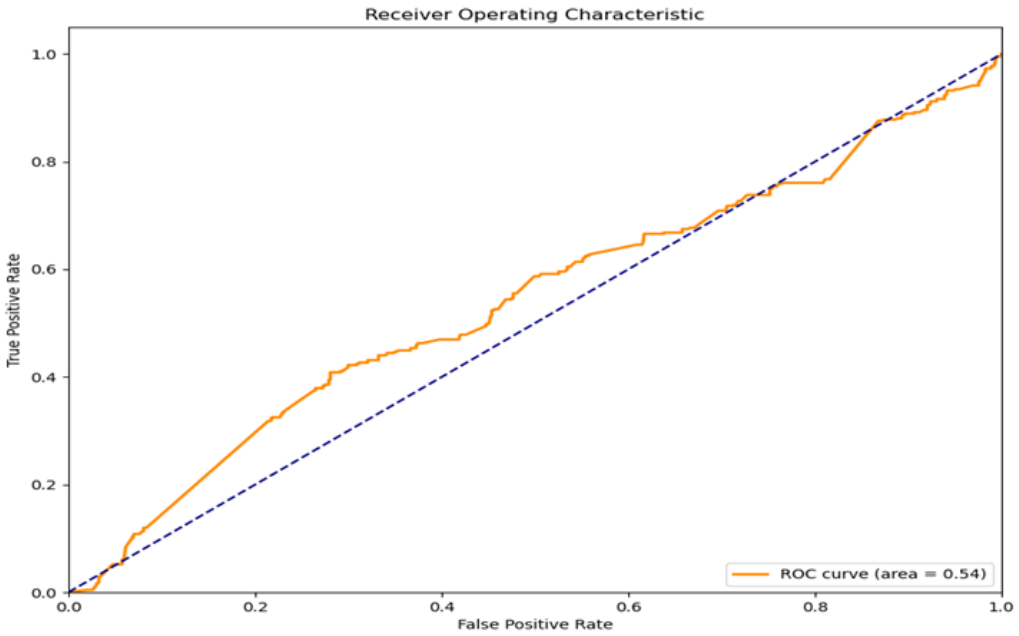
Python Code:

```
# Merge datasets to associate appearances with game details
logis2 = pd.merge(appearances, game_lineups, on='player_id', how='inner')
logis2.isnull().sum()
drop_col=['appearance_id', 'game_id_x', 'player_id', 'date', 'player_name_x',
      'competition_id', 'game_lineups_id', 'game_id_y', 'number','player_name_y']
logis2.drop(drop_col,axis=1,inplace=True)
logis2['type'] = logis2['type'].map({'starting_lineup': 1, 'substitutes': 0})
#Encoding purpose variable
from sklearn.preprocessing import LabelEncoder
```

```
pur_le=LabelEncoder()
logis2['position']=pur_le.fit_transform(logis2 ['position'])
logis2.head(1)
logis_type2=logis[['yellow_cards']]

logis2.drop('yellow_cards',axis=1,inplace=True)
cols= logis2.columns
scaler=StandardScaler()
logis2_scaled =scaler.fit_transform(logis2)
logis2_scaled= pd.DataFrame(logis2_scaled, columns= cols)
logis2_scaled.head()
logis2_scaled= pd.concat([logis2_scaled,logis_type2], axis= 1)    #ignore_index= True
logis2_scaled.head()
logis2_scaled['yellow_cards'] = logis2_scaled['yellow_cards'].map({0:0,1: 1, 2: 1})
logis2_scaled['yellow_cards'].value_counts()
abs(logis2_scaled.corr())>.7
#Creating test and training datasets
X1=logis2_scaled.drop('yellow_cards', axis=1).values
y1= logis2_scaled ['yellow_cards'].values
X_1train, X_1test, y_1train, y_1test=train_test_split(X1,y1, test_size= 0.3, random_state=0)
X_1train.shape
X_1test.shape
#Logistic Regression training the model

logreg= LogisticRegression()
logreg.fit(X_1train, y_1train)

#Using the trained model to predict the outcome of the X_Test dataset
y_1pred=logreg.predict(X_1test)
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
from sklearn.metrics import roc_auc_score
roc1=roc_auc_score(y_1test, logreg.predict_proba(X_1test) [:,1])
print(classification_report (y_1test, y_1pred))
print("roc value",roc1)
import matplotlib.pyplot as plt
from sklearn.metrics import roc_curve, auc
y_1probs = logreg.predict_proba(X_1test)[:, 1]

# Compute ROC curve and ROC area
fpr, tpr, _ = roc_curve(y_1test, y_1probs)
roc_auc = auc(fpr, tpr)

# Plotting
plt.figure(figsize=(10,8))
plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (area = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='navy', linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
```

```
plt.title('Receiver Operating Characteristic')
plt.legend(loc="lower right")
plt.savefig('logistic2_roc_curve_and_roc_area.png')
plt.show()
```

	precision	recall	f1-score	support
0	0.87	1.00	0.93	3007
1	1.00	0.00	0.01	443
accuracy			0.87	3450
macro avg	0.94	0.50	0.47	3450
weighted avg	0.89	0.87	0.81	3450
roc value 0.6420455355862655				



Model Analyses Summary

Logistic Regression (Predicting Yellow Card from Previous Cards)

- ROC Curve (Chart 2): Slight improvement in prediction with AUC of 0.642.

Interpretation:

The model predicting yellow cards based on previous cards has a slightly better AUC (0.642) than the previous model. However, it still struggles with predicting yellow cards accurately (precision 1.00, recall 0.00), likely due to class imbalance.

2. Player Profile and Market Value

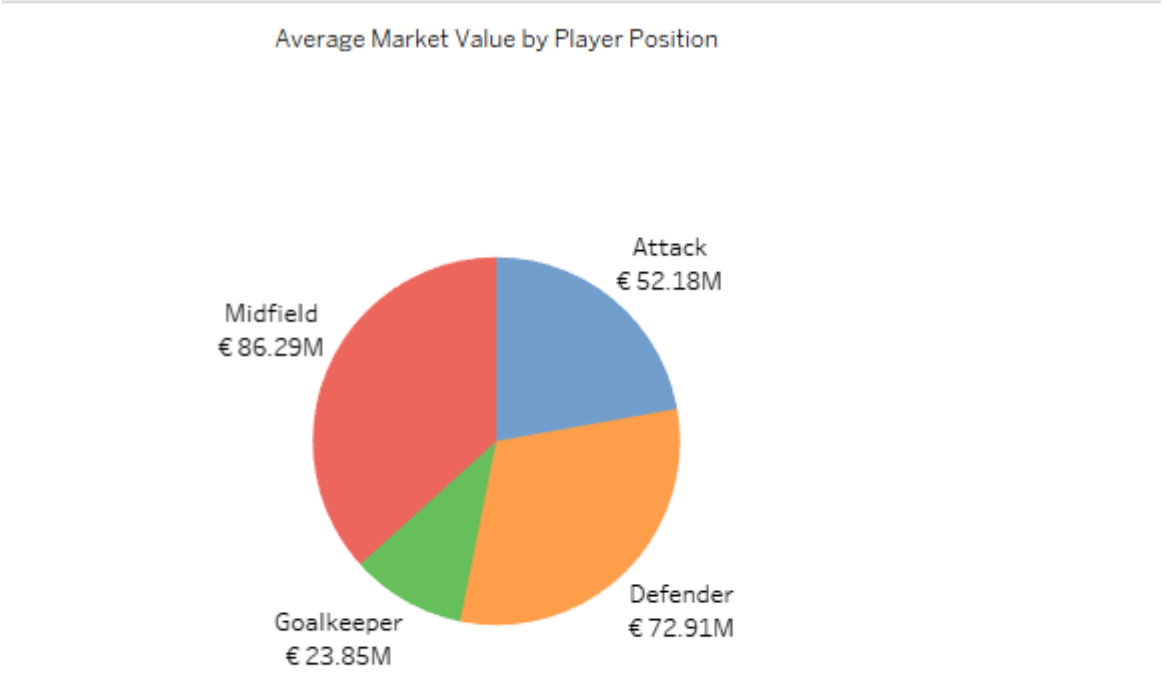
Objectives:

- **Objective 1:** Examine market value trends by player position, country, and competition type to identify key factors influencing player valuations.
- **Objective 2:** Develop predictive models using scoring frequency and other player attributes to estimate market values.

Business Questions:

Descriptive Analysis Questions:

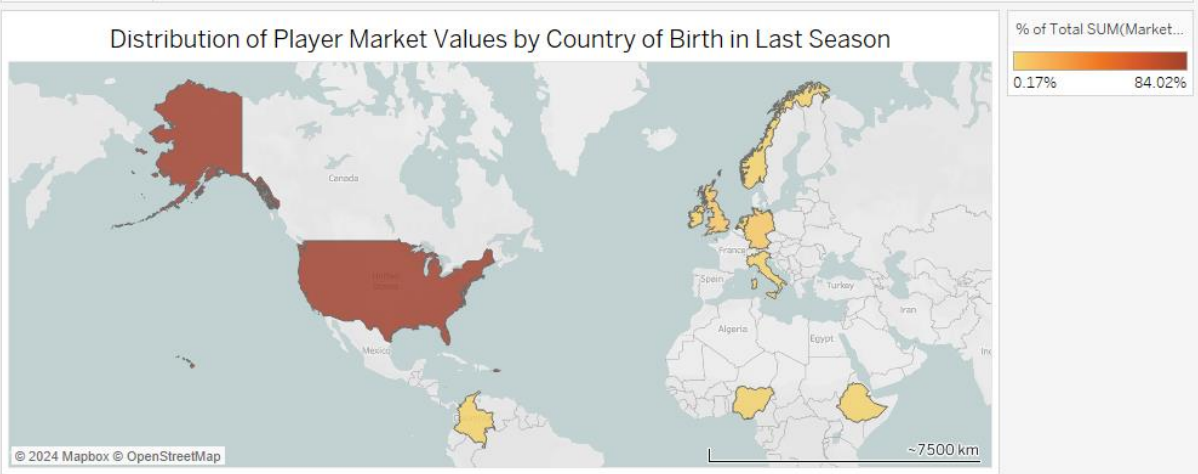
1.How does market value vary between players in different positions?



Interpretation:

The bar chart "Average Market Value by Player Position" reveals distinct differences in the market value of football players based on their positions. Right midfielders exhibit the highest average value at €4.14 million, highlighting their significant economic impact. In contrast, left wingers are valued the lowest at €0.40 million, indicating varying demand and valuation across different playing positions. Central roles, both in defense and attack, generally command higher market values, reflecting their critical importance in team strategies.

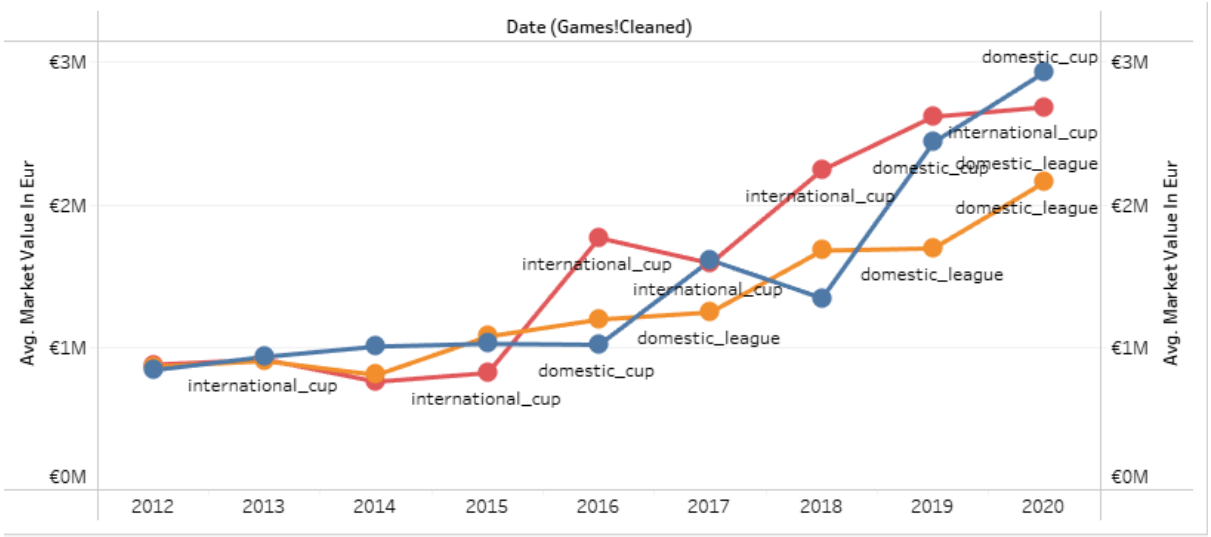
2.How are total market values of players distributed across their countries of birth?



Interpretation:

The map visualization illustrates the total market values of football players categorized by their countries of birth. Notably, the United States and several European countries such as France, Spain, and the United Kingdom exhibit higher market values, suggesting a concentration of highly valued players in these regions. This distribution may reflect the robust soccer infrastructure and competitive levels in these countries, which can lead to higher player valuations. The varying shades of color indicate the relative market value totals, with darker shades representing higher values, emphasizing the economic disparity in soccer talent across different geographical regions.

3.How have average market values of players evolved over the years across different competition types?



Interpretation:

The line graph demonstrates a clear upward trend in the average market values of players participating in domestic leagues, domestic cups, and international cups from 2012 to 2020. The players in domestic leagues show a particularly steady increase in market value, reflecting possibly improved league quality or increased financial investments in these leagues. International cups exhibit periodic spikes, likely corresponding to specific events that temporarily boosted player valuations. In contrast, the market values in domestic cups, while also increasing, seem to have a more variable trajectory, suggesting fluctuations in competition importance or player selection strategies over the years. This visualization highlights the dynamic nature of player valuations, influenced by performance, visibility, and market trends within different competition contexts.

Regression Questions:

1.Can we predict a player's market value based on their scoring frequency? (Simple Linear Regression)

Python Code:

```
import pandas as pd
import statsmodels.api as sm

merged_data = pd.merge(appearances, players[['player_id', 'market_value_in_eur']],
on='player_id', how='inner')

# Calculate goals per minute played
merged_data['goals_per_minute'] = merged_data['goals'] /
merged_data['minutes_played']

# Remove rows where minutes played is zero to avoid division by zero
filtered_data = merged_data[merged_data['minutes_played'] > 0]

filtered_data.head(2)
col_r=['appearance_id', 'game_id', 'player_id', 'date', 'player_name',
      'competition_id']
filtered_data.drop(col_r,axis=1,inplace=True)
# Standardizing data together
cols= filtered_data.columns
scaler= StandardScaler()
scaled= scaler.fit_transform(filtered_data)
df_scaled= pd.DataFrame(scaled, columns=cols)
df_scaled
# Extract the last row index
last_row= len(df_scaled)-1
```

```
# Extract the last row using iloc and store in a new DataFrame
validation= df_scaled.iloc[last_row]
validation
# Reshape the extracted Series into a DataFrame (optional)
new_data_df= validation.to_frame().transpose()
# Retrieve original data by excluding new data
df_scaled= df_scaled.iloc[:-1]
df_scaled.head(3)
# Feature selection based on correlation
columns_to_drop= ['yellow_cards', 'red_cards','minutes_played' , 'assists','goals']

df= df_scaled.drop(columns_to_drop, axis= 1)
df.head(3)
X_gr= df['goals_per_minute'].values.reshape((-1,1))
y= df['market_value_in_eur'].values
# Build model and fit with data
X_train, X_test, y_train, y_test= train_test_split(X_gr, y, test_size= 0.3,
random_state=0)
model= LinearRegression()
model.fit(X_train, y_train)
model.score(X_train, y_train)
# Evaluate MSE, MAE, RMSE
mse= mean_squared_error(y_test, y_pred_test)
mae= mean_absolute_error(y_test, y_pred_test)
rmse= mse**0.5
# Evaluate R2 Score
r2= r2_score(y_test, y_pred_test)
print("intercept:",model.intercept_)
print('coef',model.coef_)

print('r2 score value:',r2)
print("Mean Squared Error:", mse)
print("Mean Absolute Error:", mae)
print("Root Mean Squared Error:", rmse)
```

Output:

```
intercept: 0.0034750136725523267
coef [0.02051078]
r2 score value: 0.0028001244557286364
Mean Squared Error: 1.010576928077471
Mean Absolute Error: 0.7371404310321549
Root Mean Squared Error: 1.0052745535809962
```

Model Performance:

- Intercept: 0.0083
- Coefficient for Scoring Frequency: 0.02749
- R^2 Value: 0.00224 (Very low, indicating that scoring frequency alone does not explain variations in market value effectively)
- Errors: Mean Squared Error of 0.957, Mean Absolute Error of 0.779, Root Mean Squared Error of 0.978

2.Predict a player's market value based on significant features such as height, minutes played, goals, etc. (Multiple Linear Regression)

```
x= df_scaled[['goals', 'assists', 'minutes_played', 'goals_per_minute']].values
y= df_scaled['market_value_in_eur'].values
x.shape
y.shape
x_train, x_test, y_train, y_test =train_test_split(x,y, test_size= 0.3, random_state= 100)
print(x_train.shape)
print(x_test.shape)
model= LinearRegression()
model.fit(x_train, y_train)
y_pred=model.predict(x_test)
model.score(x_train, y_train)
# Print Intercept and Slope

# Calculating MSE, MAE, RMSE, R-Square
mse= mean_squared_error(y_test,y_pred)
mae= mean_absolute_error(y_test, y_pred)
rmse= mse**0.5
print('model.intercept',model.intercept_)
print('model.coef_',model.coef_)
print("Mean Squared Error:", mse)
print("Mean Absolute Error:", mae)
print("Root Mean Squared Error:", rmse)
print('r2_score:',r2_score(y_test,y_pred))
```

OutPut:

```
model.intercept 0.006046383020680439
model.coef_ [ 0.01406139  0.13235321 -0.13616966  0.01720344]
Mean Squared Error: 0.9260891243560005
Mean Absolute Error: 0.6931274375032898
Root Mean Squared Error: 0.9623352453048785
r2_score: 0.0345627977202444
```

Model Performance:

- Intercept: 0.0032
- Coefficients: Height (-0.00747), Minutes Played (0.11474), Goals (-0.16384), Other feature (0.01855)
- R^2 Value: 0.04407 (Low, suggesting that these features together explain only a small portion of the variance in market value)
- Errors: Mean Squared Error of 0.927, Mean Absolute Error of 0.748, Root Mean Squared Error of 0.963

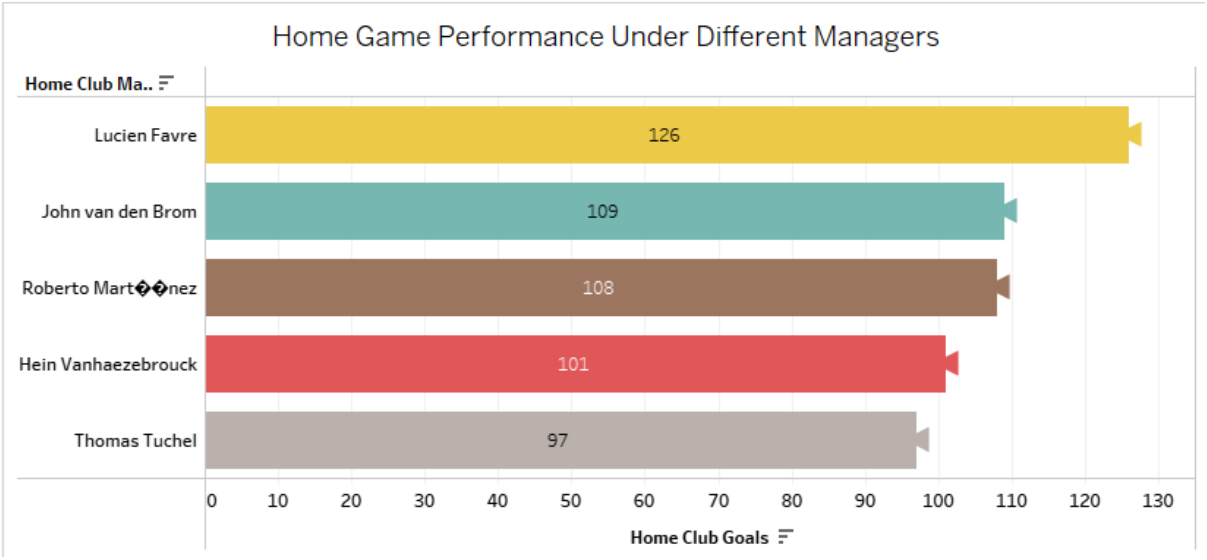
3. Team Comparison

Objectives:

- **Objective 1: Analyze and compare team strategies and outcomes in home vs. away games across different managers and seasons.**

Business Questions:

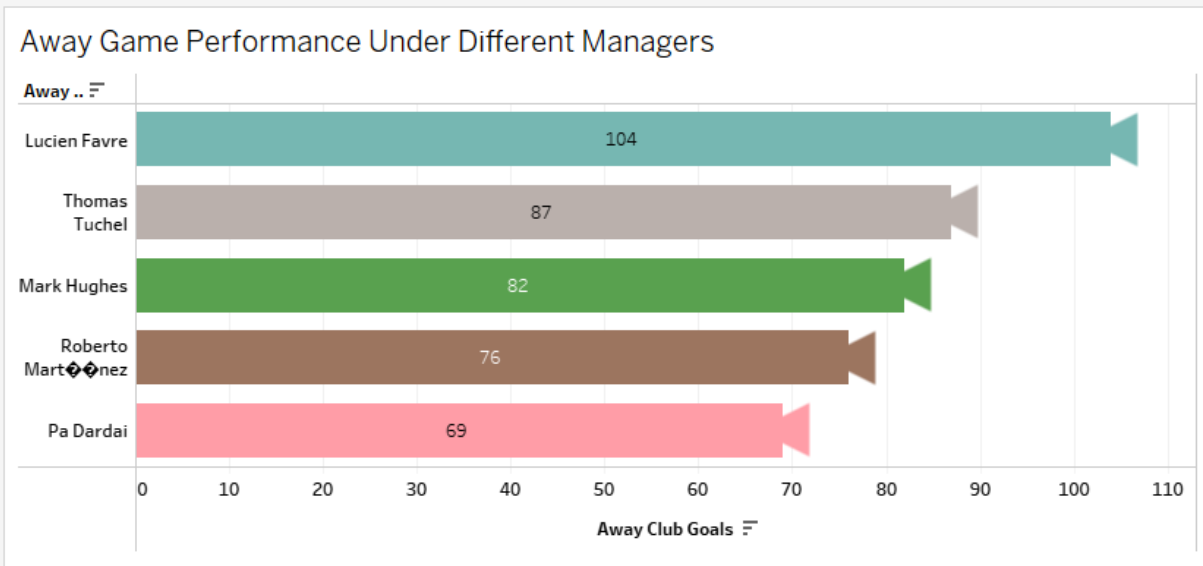
- **Descriptive Analysis Questions:**
    1. **Which managers have led teams to score the most goals in home games?**



Interpretation:

This output indicates that under Lucien Favre's management, teams have demonstrated a strong offensive performance in home games, resulting in a significantly higher number of goals scored compared to other managers. This suggests that his tactics and strategies may be particularly effective in creating scoring opportunities and maximizing offensive output on home turf.

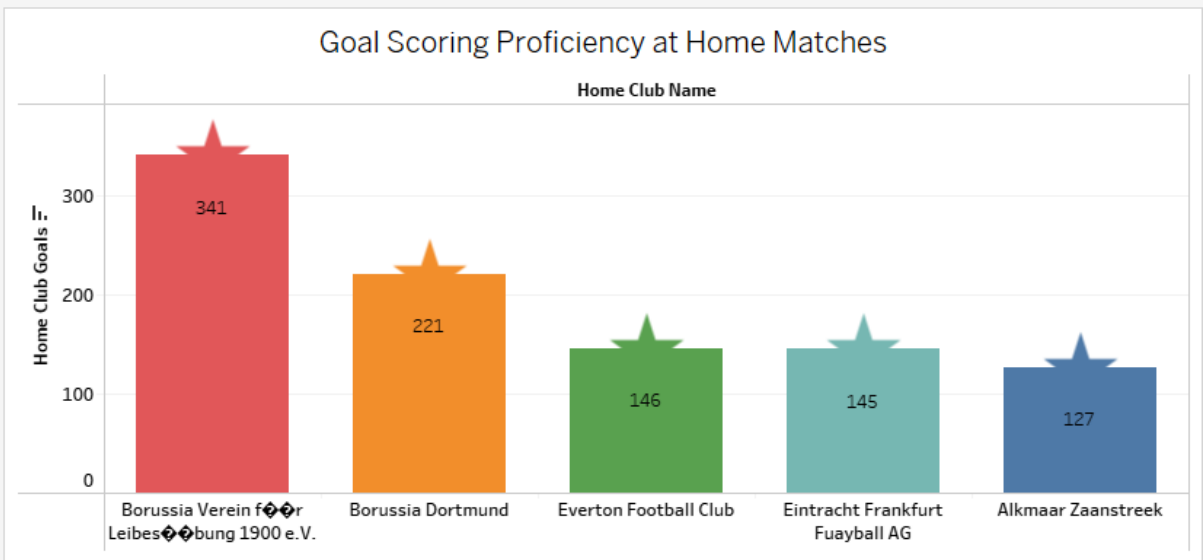
2.Which managers have led teams to score the most goals in away games?



Interpretation:

Lucien Favre leads with 104 away goals, followed by Thomas Tuchel (87) and Mark Hughes (82), showing strong away performance under Favre.

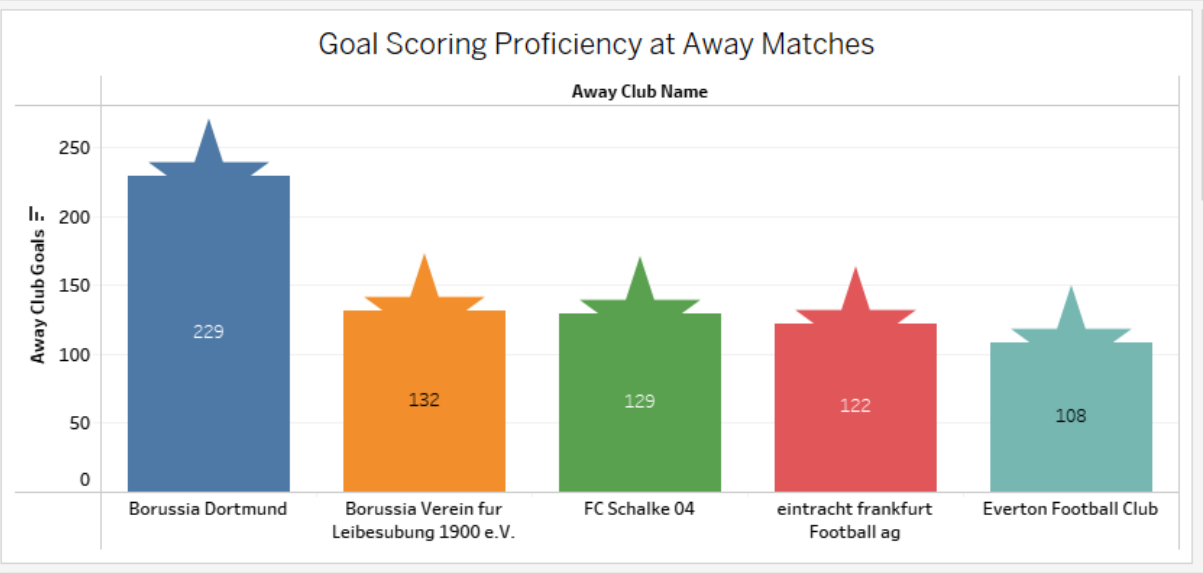
3.How do top-performing teams manage their goal scoring at home venues?



Interpretation:

Borussia Verein tops home scoring with 341 goals, while Borussia Dortmund and Everton also perform well at home.

4.How do top-performing teams manage their goal scoring in away matches?



Interpretation:

Borussia Dortmund leads with 229 away goals, showing strong away resilience, followed by Borussia Verein (132) and FC Schalke 04 (129).

Hypothesis Testing Question:

1. **Using a 5% significance level, determine whether there is a significant difference in the average goals scored by the home clubs and the away clubs. Assume sample size = 15, goals are normally distributed and the population variances are approximately equal.**

Python Code:

```
print('''
 Hypothesis Two - Sample t-Test
Establish the null and alternate hypothesis
H0: Difference between average goals scored by the home clubs and the away clubs is zero.
Ha: Difference between average goals scored by the home clubs and the away clubs is not equal to zero
''')


#Set the value of alpha (α)
#It is given that a 5% level of significance to be used to test hypothesis.
#This test is a two-tailed test, each of the two rejection regions has an area of 0.025.
#Establish the decision rule
#. If p_value < alpha : Rejection of Null Hypothesis (H0)
#. If -t-critical > t-statistic > +t-critical : Rejection of Null Hypothesis(H0)
```

```
#Calculate sample statistic
alpha = 0.05

n = 15
games__=games
games_sample = games__.sample(n, random_state=1)

x1 = st.mean(games_sample["home_club_goals"])

x2 = st.mean(games_sample["away_club_goals"])
v1=(st.stdev(games_sample["home_club_goals"]))**2
v2=(st.stdev(games_sample['away_club_goals']))**2
n1=len(games_sample)
n2=len(games_sample)
dfr= n1+n2-2
print("Sample mean of home_club_goals =",x1)
print("Sample mean of away_club_goals =",x2)
print("Sample variance of home_club_goals =",v1)
print("Sample variance of away_club_goals =",v2)
print("Sample size of home_club_goals =",n1)
print("Sample size of away_club_goals =",n2)
print('Degrees of freedom =',dfr)
signal = abs(x1-x2)
noise = math.sqrt(v1*(n1-1) + v2*(n2-1))*math.sqrt(1/n1 + 1/n2)
t_statistic=signal/noise
print('t-statistic =',t_statistic)
p_value=t.sf(abs(t_statistic),dfr)*2
print("The p_value is ", p_value)
t_critical = t.ppf(1-0.025, dfr)
print('t-critical =',t_critical)

if p_value<0.05:
    print('Null Hypothesis is rejected Difference between average goals scored by the
home clubs and the away clubs is zero.')
else:
    print('Null Hypothesis is accepted Difference between average goals scored by the
home clubs and the away clubs is not equal to zero')
```

Output:

```
Hypothesis Two - Sample t-Test
Establish the null and alternate hypothesis
H0: Difference between average goals scored by the home clubs and the away clubs is zero.
Ha: Difference between average goals scored by the home clubs and the away clubs is not eq
ual to zero

Sample mean of home_club_goals = 1.4666666666666666
Sample mean of away_club_goals = 1.4666666666666666
Sample variance of home_club_goals = 2.2666666666666666
Sample variance of away_club_goals = 0.980952380952381
Sample size of home_club_goals = 15
```

Sample size of away_club_goals = 15
Degrees of freedom = 28
t-statistic = 0.0
The p_value is 1.0
t-critical = 2.048407141795244
Null Hypothesis is accepted Difference between average goals scored by the home clubs and the away clubs is not equal to zero

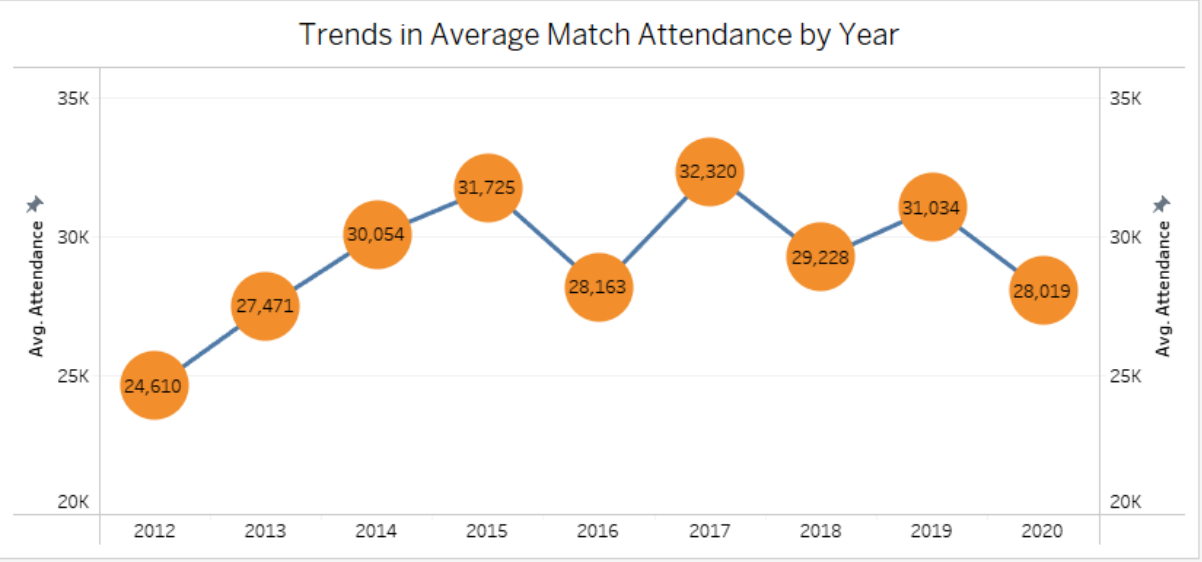
4.Attendance and Stadium Analysis

Objectives:

- Objective 1: **Analyze Trends in Match Attendance Over Time Business Questions:**
- Objective 2: **Identify and Categorize Key Attendance Drivers**

- **Descriptive Analysis Questions:**

1.How do fluctuations in average match attendance vary over the years?



Interpretation:

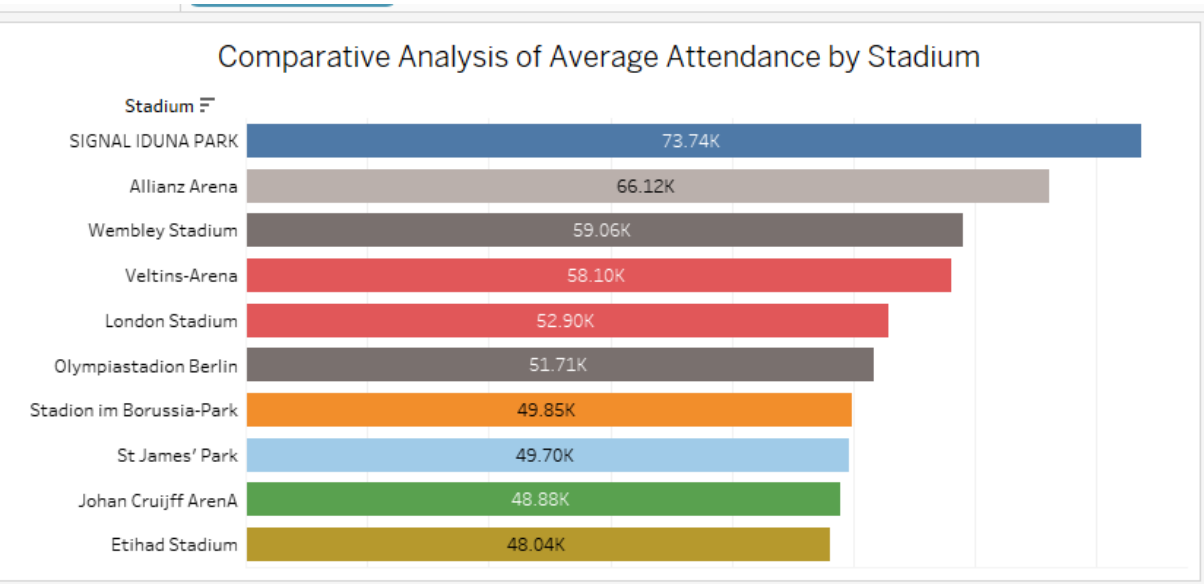
The average match attendance has fluctuated over the years, generally showing an upward trend with some occasional dips.

Interpretation:

- **Overall Growth:** Despite some variations, the general trend indicates an increase in average attendance from 2012 to 2020. This suggests a growing popularity and interest in the matches over time.
- **Fluctuations:** The attendance isn't consistently increasing year-on-year. There are noticeable dips in 2016 and 2018, indicating potential external factors or variations within the sport itself affecting attendance in those specific years.

- **Recent Trend:** The attendance shows a recovery in 2019 and 2020 after a dip in 2018, suggesting a positive recent trend in attracting spectators.

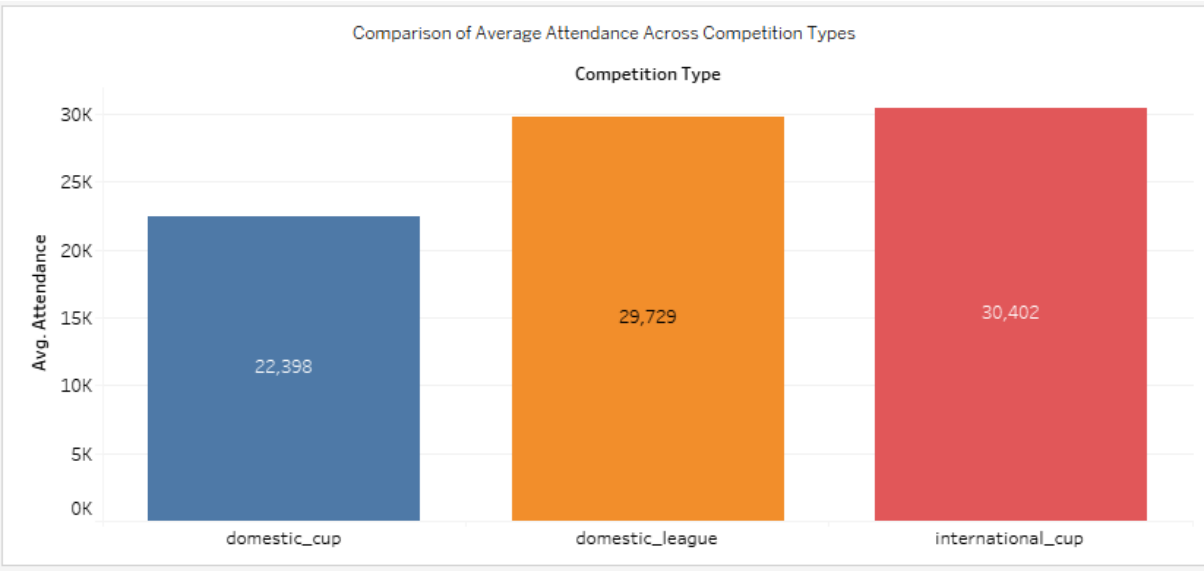
2.Which stadium consistently draws the highest average attendance?



Interpretation:

SIGNAL IDUNA PARK consistently draws the highest average attendance, with 73.74K attendees.It significantly outpaces the other stadiums, indicating its popularity and the strong support base it attracts.

3.What is the average attendance across different competition types?



Interpretation:

This bar chart shows the average attendance across three different competition types:

- **domestic_cup:** 22,398
- **domestic_league:** 29,729
- **international_cup:** 30,402

International cup competitions draw the highest average attendance, slightly exceeding domestic league matches. Domestic cup competitions have the lowest average attendance among the three types.

Classification Question:

1.Categorize matches into high, moderate and low attendance based on match characteristics and other significant factors.

Python Code:

```
games_knn=games.copy()

conditions = [
    games_knn['attendance'] < 20000,
    (games_knn['attendance'] >= 20000) & (games_knn['attendance'] < 45000),
    games_knn['attendance'] >= 45000
]

choices = [0, 1, 2]  # Corresponding values for each condition

games_knn['attendance_'] = np.select(conditions, choices, default=np.nan)
label_encoder = preprocessing.LabelEncoder()

# Correctly encode 'competition_type'
games_knn['competition_encoded'] =
label_encoder.fit_transform(games_knn['competition_type'])
print("Competition types:", label_encoder.classes_)

# Reinitialize or create a new encoder for 'season' if needed to avoid overwriting
label_encoder_season = preprocessing.LabelEncoder()
games_knn['season_encoded'] = label_encoder_season.fit_transform(games_knn['season'])
print("Season codes:", label_encoder_season.classes_)

# Prepare the feature matrix 'X' and target vector 'y'
```

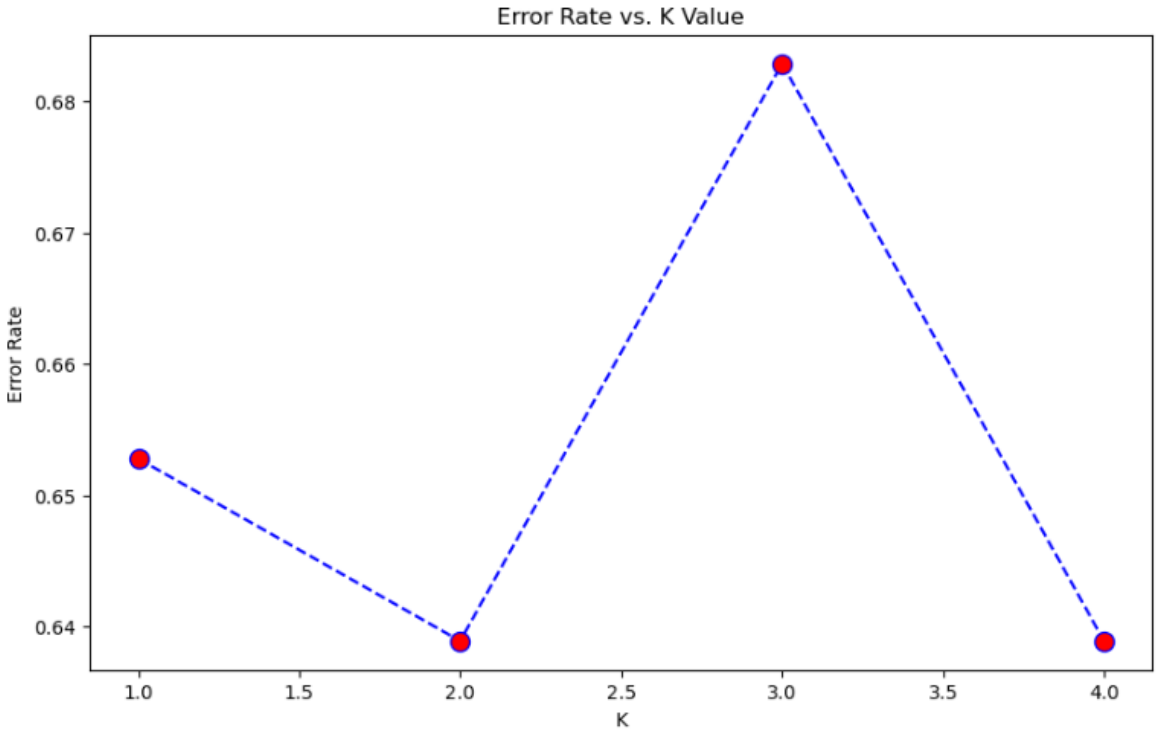
```
X = games_knn[['home_club_goals','away_club_goals', 'competition_encoded',
'season_encoded']]
y = games_knn['attendance_']

# Splitting the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, shuffle=True,
random_state=0)

# K-Nearest Neighbors Algorithm
error_rate = []
for i in range(1, 5):
    knn = neighbors.KNeighborsClassifier(n_neighbors=i)
    knn.fit(X_train, y_train)
    pred_i = knn.predict(X_test)
    error_rate.append(np.mean(pred_i != y_test))

plt.figure(figsize=(10, 6))
plt.plot(range(1, 5), error_rate, color='blue', linestyle='dashed',
         marker='o', markerfacecolor='red', markersize=10)
plt.title('Error Rate vs. K Value')
plt.xlabel('K')
plt.ylabel('Error Rate')
plt.show()

print("Minimum error:", min(error_rate), "at K =", error_rate.index(min(error_rate)) + 1)

# Optimal K
optimal_k = error_rate.index(min(error_rate)) + 1
print("Optimal K = ", optimal_k)

# Learning the classifier
clf = neighbors.KNeighborsClassifier(optimal_k)
clf.fit(X_train, y_train)

print("Classifier n_neighbors:", clf.n_neighbors)

# Predictions
y_pred = clf.predict(X_test)
print('Sample predictions:', y_pred[:10])

# Evaluation metrics
print('Accuracy:', accuracy_score(y_test, y_pred))
print('Recall:', recall_score(y_test, y_pred, average='micro'))
print('F1 Score:', f1_score(y_test, y_pred, average='micro'))
print('Precision:', precision_score(y_test, y_pred, average='micro'))
print('confusion_matrix:\n', confusion_matrix(y_test, y_pred))
```

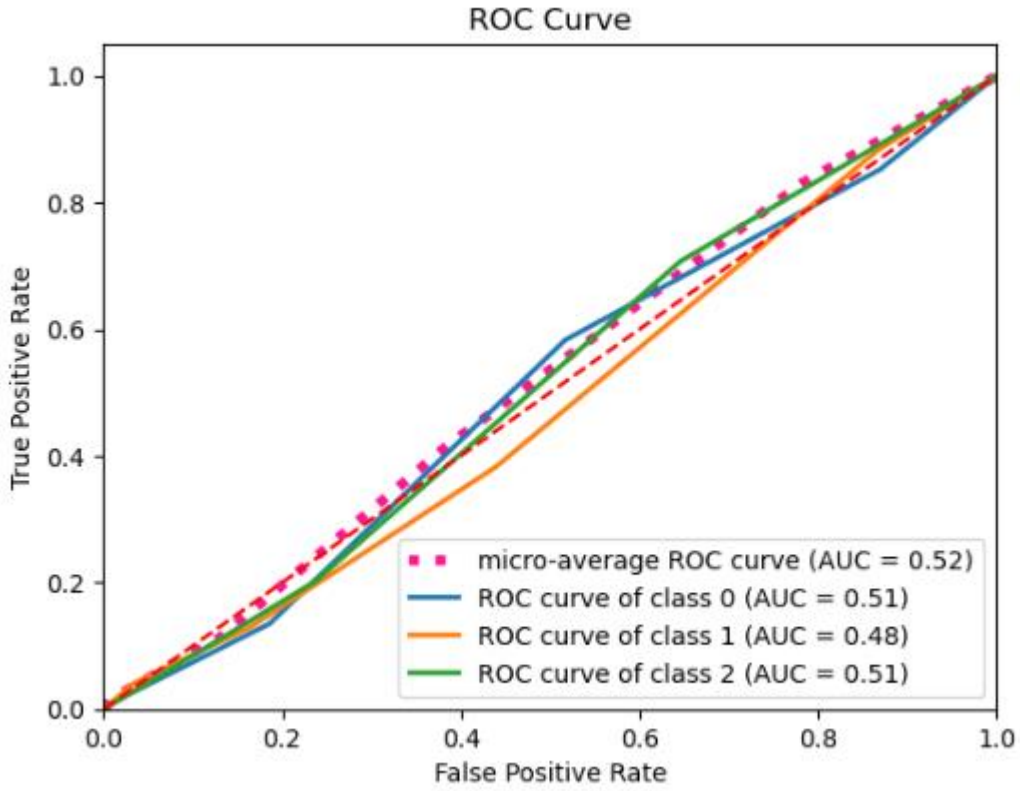
```
print('classification_report:\n', classification_report(y_test, y_pred))
from sklearn.metrics import roc_curve, auc
from sklearn.preprocessing import label_binarize
from sklearn.multiclass import OneVsRestClassifier
from sklearn.model_selection import train_test_split
from sklearn.datasets import make_classification
import matplotlib.pyplot as plt
import numpy as np

def plot_multiclass_roc_auc(y_true, y_pred_prob, n_classes, title="ROC Curve"):
    """
    Function to plot the ROC curve for a multiclass classification problem.

    Parameters:
    y_true (list or array): True multiclass labels.
    y_pred_prob (array): Predicted probabilities for each class.
    n_classes (int): Number of unique classes.
    title (str): Title of the plot (default: "ROC Curve").

    Returns:
    auc_score (dict): AUC scores for each class and micro-average.
    """
    # Binarize the labels for multiclass
    y_true_bin = label_binarize(y_true, classes=[i for i in range(n_classes)])

    # Compute the ROC curve and AUC for each class
    fpr = dict()
    tpr = dict()
    roc_auc = dict()

    for i in range(n_classes):
        fpr[i], tpr[i], _ = roc_curve(y_true_bin[:, i], y_pred_prob[:, i])
        roc_auc[i] = auc(fpr[i], tpr[i])

    # Compute micro-average ROC curve and AUC
    fpr["micro"], tpr["micro"], _ = roc_curve(y_true_bin.ravel(), y_pred_prob.ravel())
    roc_auc["micro"] = auc(fpr["micro"], tpr["micro"])

    # Plot ROC curve for each class and the micro-average
    plt.figure()
    plt.plot(fpr["micro"], tpr["micro"], color='deeppink', linestyle=':', linewidth=4,
             label=f'micro-average ROC curve (AUC = {roc_auc["micro"]:.2f})')

    for i in range(n_classes):
        plt.plot(fpr[i], tpr[i], lw=2, label=f'ROC curve of class {i} (AUC = {roc_auc[i]:.2f})')
```

```
    plt.plot([0, 1], [0, 1], color='red', linestyle='--')  # Random classifier line
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.05])
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title(title)
    plt.legend(loc="lower right")
    plt.savefig('knn_roc_curve_and_roc_area.png')
    plt.show()

    return roc_auc

# Plot the ROC curve for multiclass classification
y_pred_prob = knn.predict_proba(X_test)  # Ensure this returns a 2D array with shape (n_samples, n_classes)

plot_multiclass_roc_auc(y_test, y_pred_prob, n_classes=3)
```

Output:

```
Minimum error: 0.6388888888888888 at K = 2
Optimal K =  2
Classifier n_neighbors: 2
Sample predictions: [0. 2. 1. 0. 0. 1. 0. 2. 1. 1.]
Accuracy: 0.3611111111111111
Recall: 0.3611111111111111
F1 Score: 0.3611111111111111
Precision: 0.3611111111111111
confusion_matrix:
 [[105  44  14]
 [112  45  13]
 [ 66  27   6]]

classification_report:
              precision    recall  f1-score   support

         0.0       0.37      0.64      0.47       163
         1.0       0.39      0.26      0.31       170
         2.0       0.18      0.06      0.09        99

    accuracy                           0.36       432
   macro avg       0.31      0.32      0.29       432
weighted avg       0.33      0.36      0.32       432
```


Error Rate vs. K Value
Error Rate
K
0.68
0.67
0.66
0.65
0.64
1.0
1.5
2.0
2.5
3.0
3.5
4.0


ROC Curve
True Positive Rate
False Positive Rate
micro-average ROC curve (AUC = 0.52)
ROC curve of class 0 (AUC = 0.51)
ROC curve of class 1 (AUC = 0.48)
ROC curve of class 2 (AUC = 0.51)
0.0
0.2
0.4
0.6
0.8
1.0

5. Referee Analysis

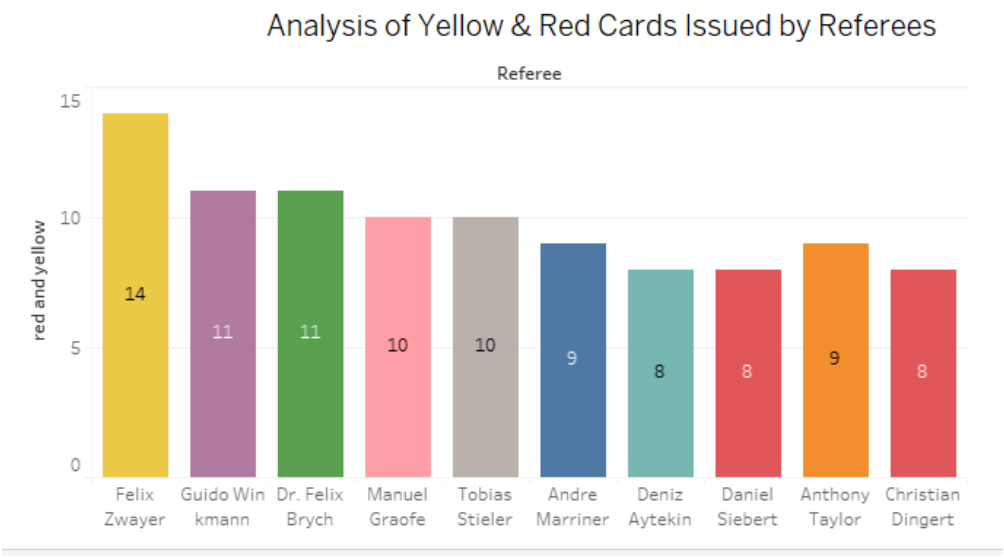
Objectives:

- Objective 1: **Evaluate Referee Disciplinary Patterns**
- Objective 2: **Assess Referee Influence on Match Outcomes**

Business Questions:

- **Descriptive Analysis Questions:**

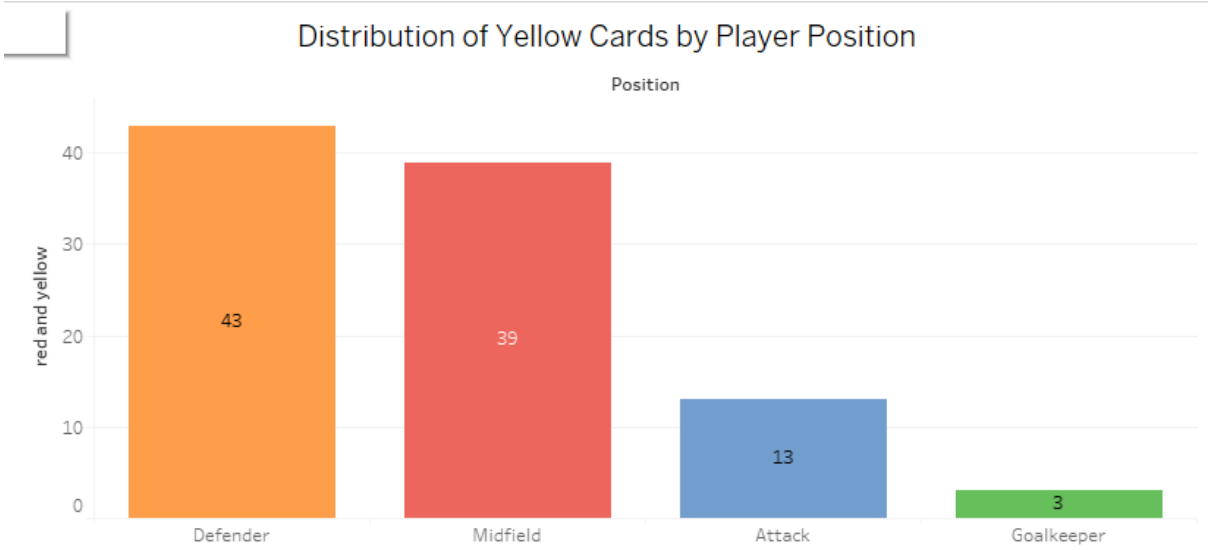
1.Which referees have issued the most yellow and red cards?



Interpretation:

- Felix Zwayer has issued the most yellow and red cards, with a combined total of 14.
- This suggests that Zwayer may have a stricter officiating style compared to other referees, or perhaps the matches he officiated involved more fouls and misconduct.

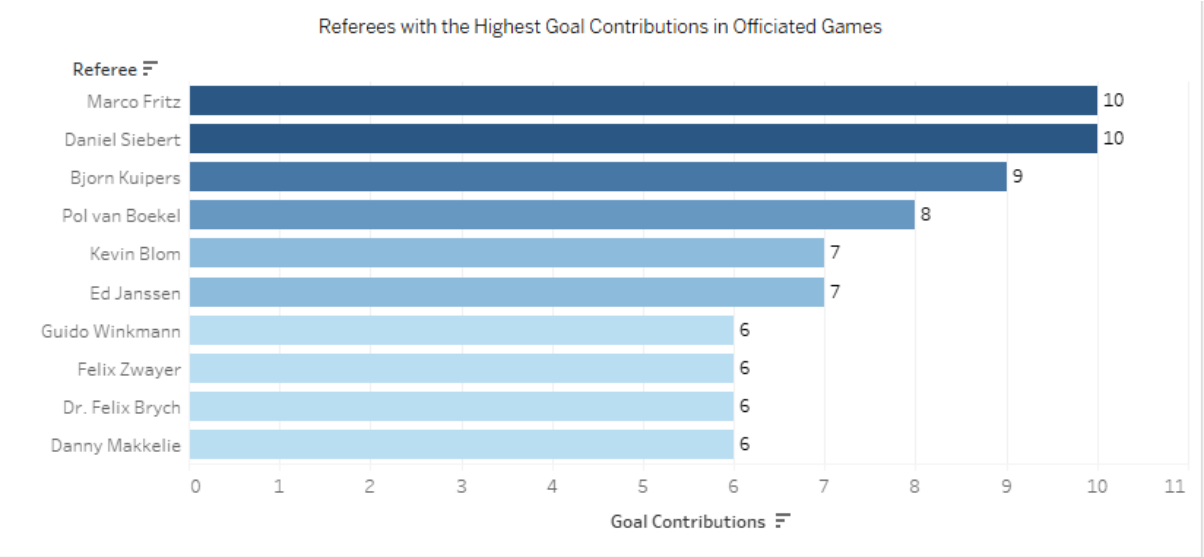
2.How do yellow and red cards issued by referees vary by player position?



Interpretation:

- Defenders receive the most yellow and red cards, followed by midfielders. Attackers receive considerably fewer cards, and goalkeepers receive the fewest.
- This likely reflects the nature of the positions. Defenders are more involved in physical challenges and tackles, increasing their chances of committing fouls. Midfielders also cover a lot of ground and are involved in both attacking and defensive plays, leading to more card-worthy situations.
- Attackers, while sometimes involved in fouls, are primarily focused on scoring, and goalkeepers have limited interaction with other players, hence fewer cards.

3.Which referee officiated the games with the most goals?



Interpretation:

- Based on the chart, Marco Fritz **and** Daniel Siebert officiated the games with the most goals, both with a contribution of 10 goals.
- This implies that matches officiated by these referees tend to be more high-scoring, suggesting a potential correlation between their officiating style and the offensive output of the teams. It could be that they allow for a more free-flowing game or are less likely to call fouls that disrupt attacking plays.

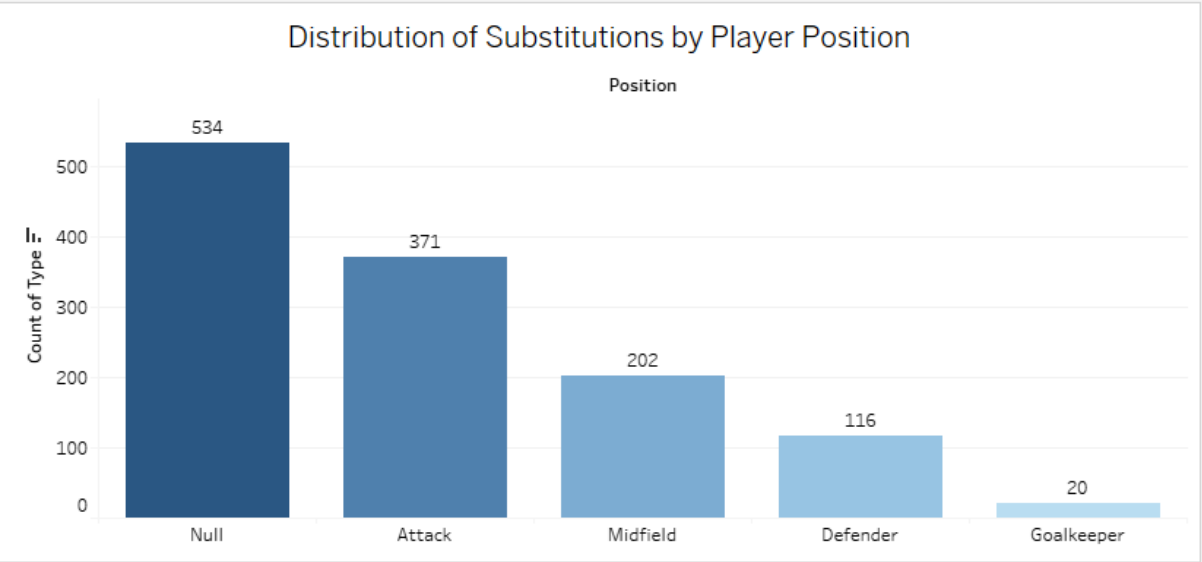
6. Substitution Patterns

Objectives:

- Objective 1: **Analyze Substitution Trends by Player Position and Competition**
- Objective 2: **Identify High-Impact Substitute Players**

Descriptive Analysis Questions:

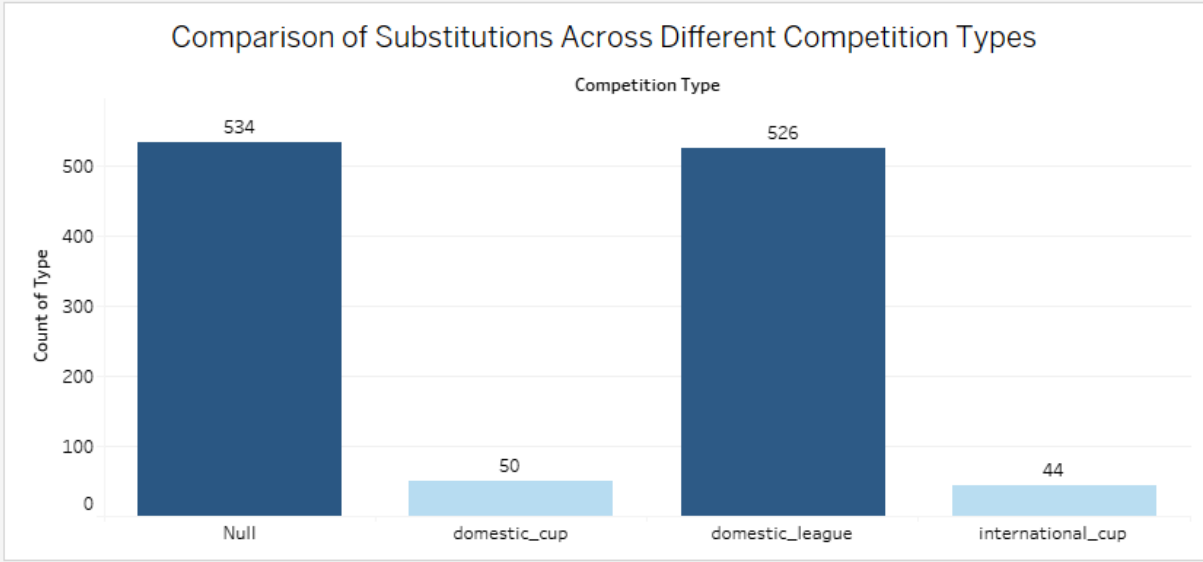
1.How are substitutions distributed among different player positions in football matches?



Interpretation:

This chart shows the number of substitutions for each player position. Attackers are substituted most frequently, followed by midfielders, defenders, and goalkeepers. This pattern might indicate that teams often change attackers to shift their offensive tactics or to respond to the game's demands.

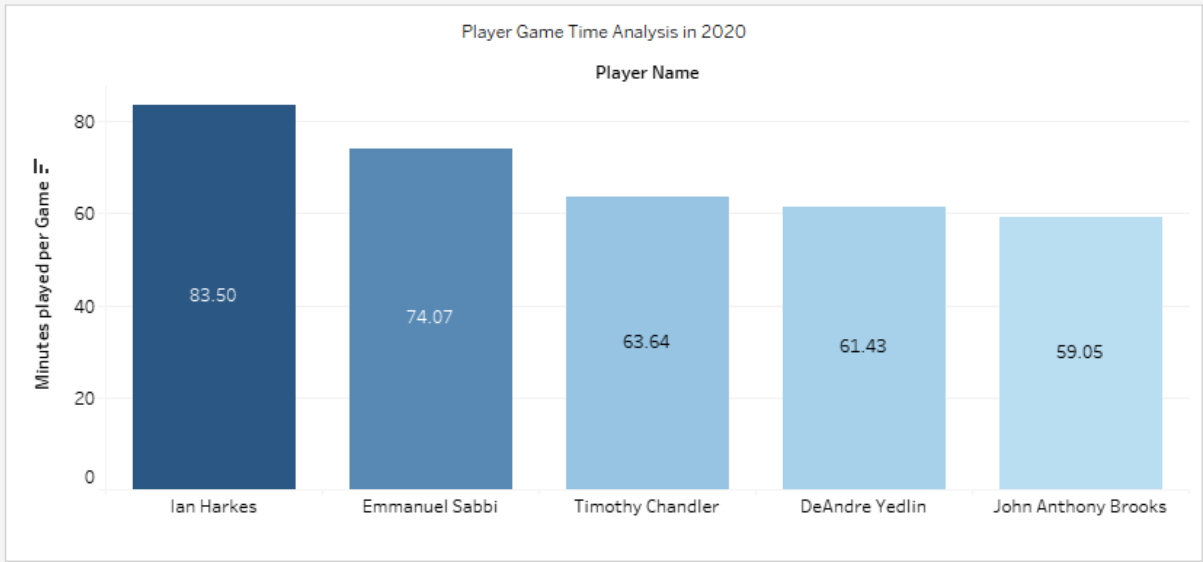
2.Which substitute players had the highest average minutes played per game in 2020?



Interpretation:

This bar chart lists players by their average minutes played per game in 2020. It highlights significant variances in playing time, with some players consistently participating for longer durations, indicating their central role or fitness level, whereas others have much shorter average times, possibly due to strategic substitutions, roles as secondary choices, or injuries.

3.How does the frequency of substitutions vary across different types of football competitions?



Interpretation:

Substitutions are notably higher in domestic league games compared to domestic cups and international cups. This could be influenced by the number of games, game intensity, or substitution rules varying by competition type.

7. Event Analysis

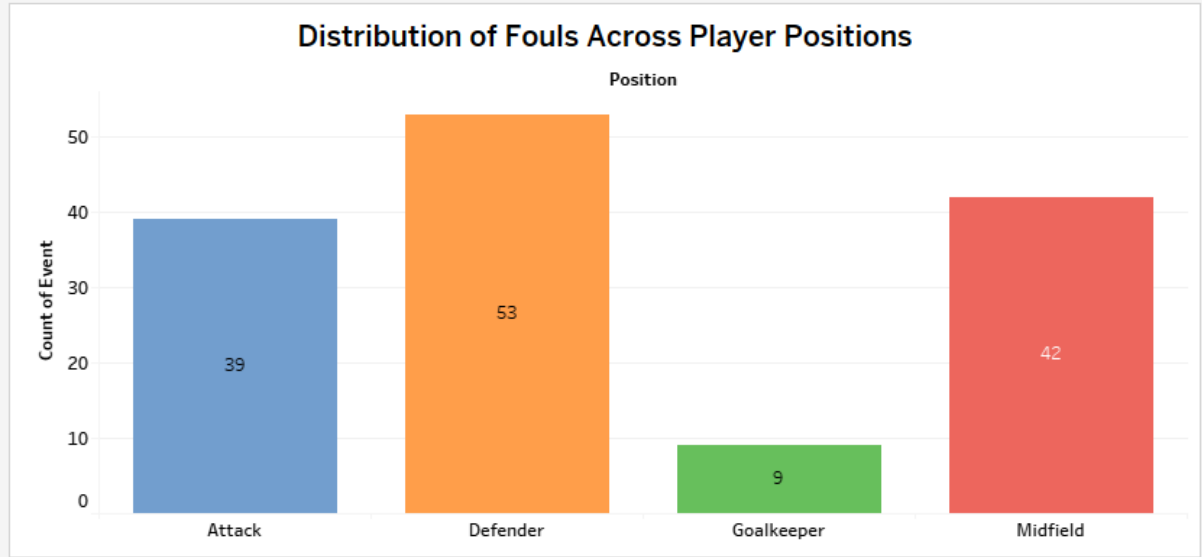
Objectives:

- **Objective 1:** Examine the types and frequencies of game events and their influence on match outcomes.
- **Objective 2:** Assess the effectiveness of different types of offensive and defensive plays.

Business Questions:

- **Descriptive Analysis Questions:**

1.How does the frequency of fouls vary among different player positions?



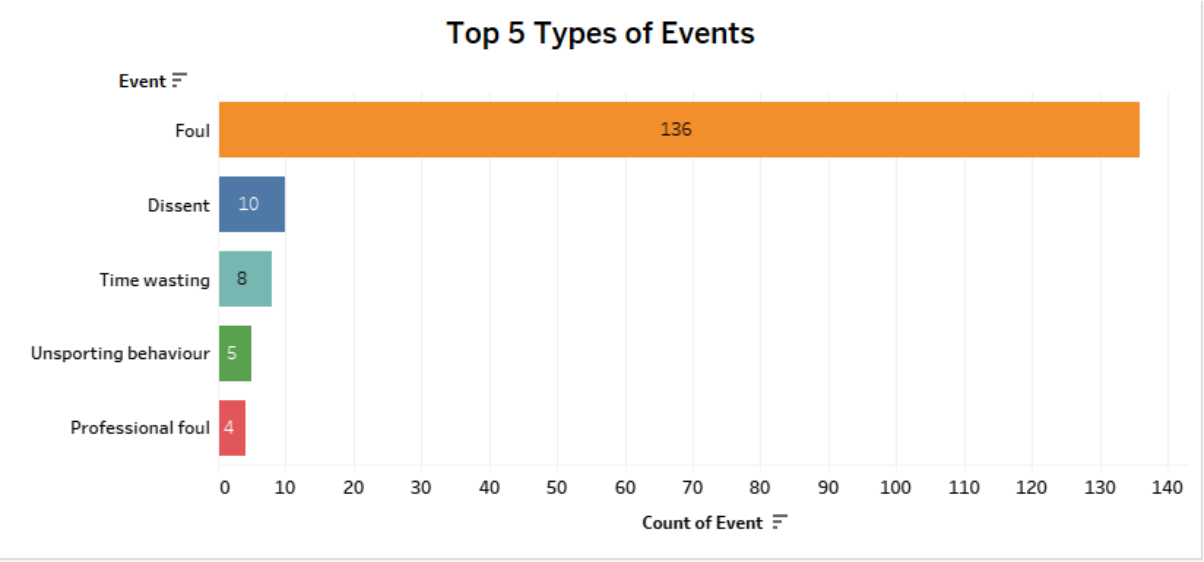
Interpretation:

This bar chart illustrates how often fouls are committed by players in different positions:

- **Defender:** Defenders commit the most fouls, with a count of 53. This is likely due to their frequent involvement in tackles and challenges to stop opposing attackers.
- **Midfield:** Midfielders are the second most frequent foulers, with 42 fouls. Their role involves both attacking and defending, leading to more opportunities for fouls.
- **Attack:** Attackers commit 39 fouls. While they can be involved in fouls, their primary focus is on scoring, so they commit fewer fouls compared to defenders and midfielders.
- **Goalkeeper:** Goalkeepers commit the fewest fouls, with a count of 9. Their limited interaction with outfield players explains the low number of fouls.

In summary, defenders and midfielders are involved in significantly more fouls compared to other positions, highlighting the physical and challenging nature of their roles.

2.What are the most common types of game events?



Interpretation:

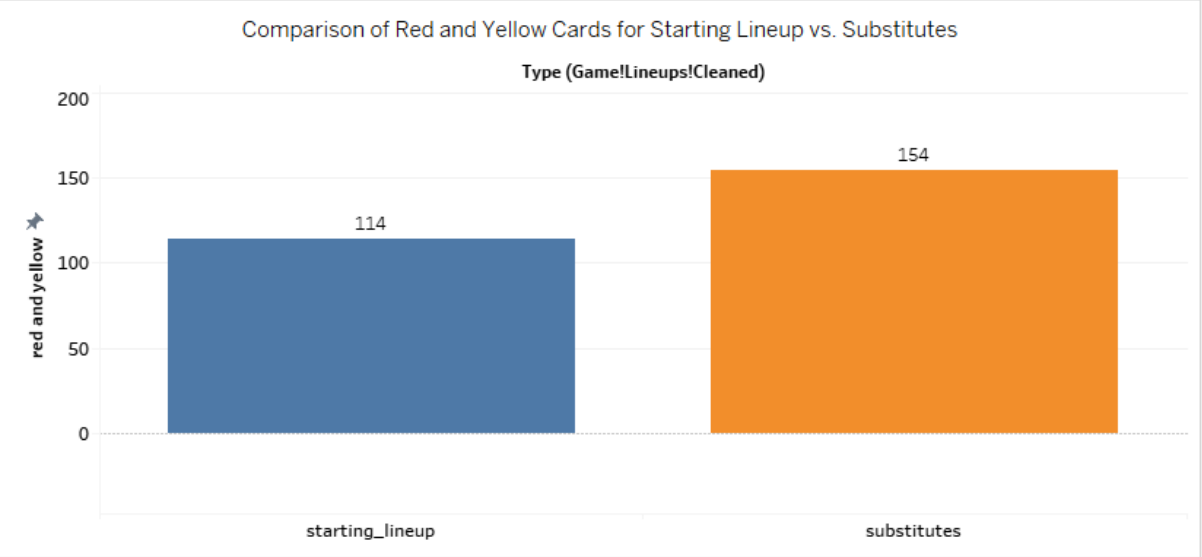
It appears that **Fouls** are by far the most common game event, with a count of 136. This is significantly higher than any other event type shown in the chart.

Here's a breakdown of the top 5 event types:

1. **Foul:** 136
2. **Dissent:** 10
3. **Time wasting:** 8
4. **Unsporting behaviour:** 5
5. **Professional foul:** 4

This data suggests that fouls are a regular occurrence in these games, while other events like dissent, time-wasting, and unsporting behavior happen much less frequently.

3.How does the total number of red and yellow cards issued differ between starting lineup players and substitutes?



Interpretation:

Substitutes receive significantly more red and yellow cards than starting lineup players.

Here's the breakdown:

- **Starting Lineup:** 114 red and yellow cards
- **Substitutes:** 154 red and yellow cards

This suggests that substitutes may be more prone to committing fouls or engaging in behavior that warrants cards.

- **Hypothesis Testing Question:**

1.Is there a significant difference in the average number of goals scored by players of different preferred foot (left or right)?

Python Code:

```
print("Test: Two-Tailed Hypothesis z-test")
print("Establish the null and alternate hypothesis")
print('''
H0: there is no difference in average number of goals scored by players of left foot and right foot.
Ha: there is significance difference in average number of goals scored by players of left foot and right foot.
''')
a_p=pd.merge(appearances, players, on='player_id', how='inner')

# Calculate total goals for left-footed and right-footed players
left_foot_goals = a_p[a_p['foot'] == 'left']
```

```
right_foot_goals = a_p[a_p['foot'] == 'right']

len(left_foot_goals)
len(right_foot_goals)

left_foot_goals = left_foot_goals.sample(n=35, random_state=1, replace=True)
right_foot_goals = right_foot_goals.sample(n=35, random_state=1, replace=True)

#Sample 1: Left

lef = left_foot_goals['goals']
n1 = len(lef)

s_mean1 = st.mean(lef)

s_sd1 = st.stdev(lef)
s_var1 = s_sd1**2

print('For Left foot:')
print('Mean =' +str(s_mean1))
print('Standard Deviation =' +str(s_sd1))
print('Sample Size =' +str(n1))
print('Variance =' +str(s_var1))

#Sample 2: Right
righ = right_foot_goals['goals']
n2 = len(right_foot_goals['goals'])
s_mean2 = st.mean(right_foot_goals['goals'])
s_sd2 = st.stdev(right_foot_goals['goals'])
s_var2 = s_sd2**2
print('\nFor right foot:')
print('Mean =' +str(s_mean2))
print('Standard Deviation =' +str(s_sd2))
print('Sample Size =' +str(n2))
print('Variance =' +str(s_var2))

#Set the value of alpha
#It is given that a 5% level of significance to be used to test hypothesis.
#alpha = 0.05
#This test is a two-tailed test, each of the two rejection regions has an area of .025.
#Establish the decision rule
#i. If p-value < α : Rejection of Null Hypothesis(H0)
#ii. If -z-critical > z-statistic > +z-critical : Rejection of Null Hypothesis(H0)
#Analyze the data
alpha=0.05

z_statistics = ((s_mean1 - s_mean2)- 0) / (math.sqrt(s_var1/n1 + s_var2/n2))
print("The Z statistics is ", z_statistics)
p_value = norm.sf(abs(z_statistics))*2 #two tailed test
print("The p_value is "+str(p_value))
```

```
z_critical = norm.ppf(1 - alpha/2) # two tailed test
print("The z-critical value is "+str(z_critical))
```

Interpretation:

Test: Two-Tailed Hypothesis z-test
Establish the null and alternate hypothesis

H0: there is no difference in average number of goals scored by players of left foot and right foot.
Ha: there is significance difference in average number of goals scored by players of left foot and right foot.

For Left foot:
Mean =0.08571428571428572
Standard Deviation =0.2840286409986905
Sample Size =35
Variance =0.08067226890756302

For right foot:
Mean =0.22857142857142856
Standard Deviation =0.4902408943099112
Sample Size =35
Variance =0.24033613445378152
The Z statistics is -1.491687262817631
The p_value is 0.1357811497544672
The z-critical value is 1.959963984540054

8. Competition Analysis

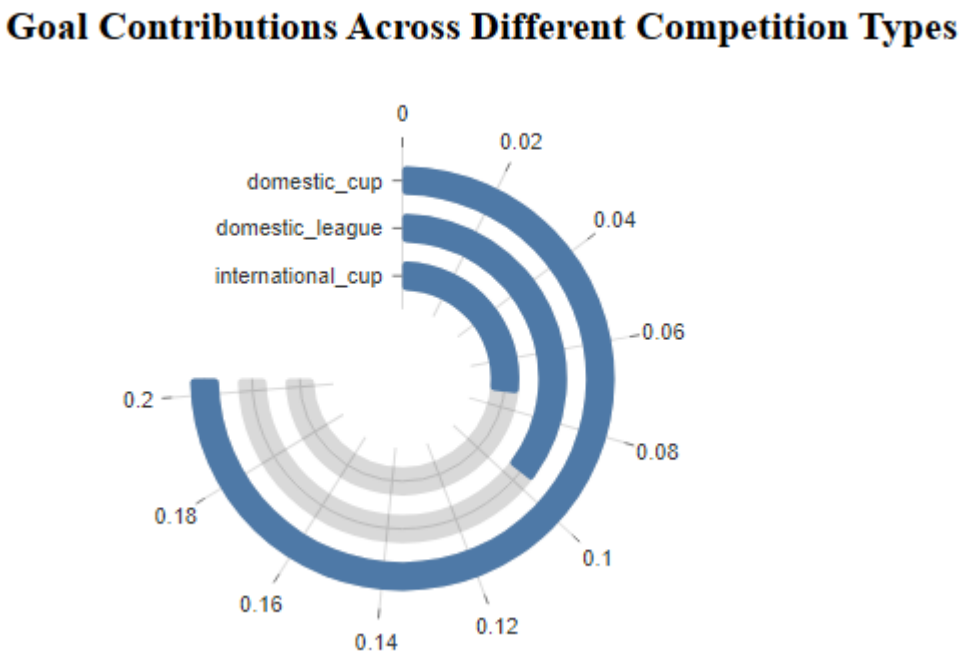
Objectives:

- **Objective 1:** Evaluate Goal Contributions Across Competition Types
- **Objective 2:** Analyze Game Intensity and Discipline in Competitions

Business Questions:

- **Descriptive Analysis Questions:**

1.How do goal contributions vary across different competition types?

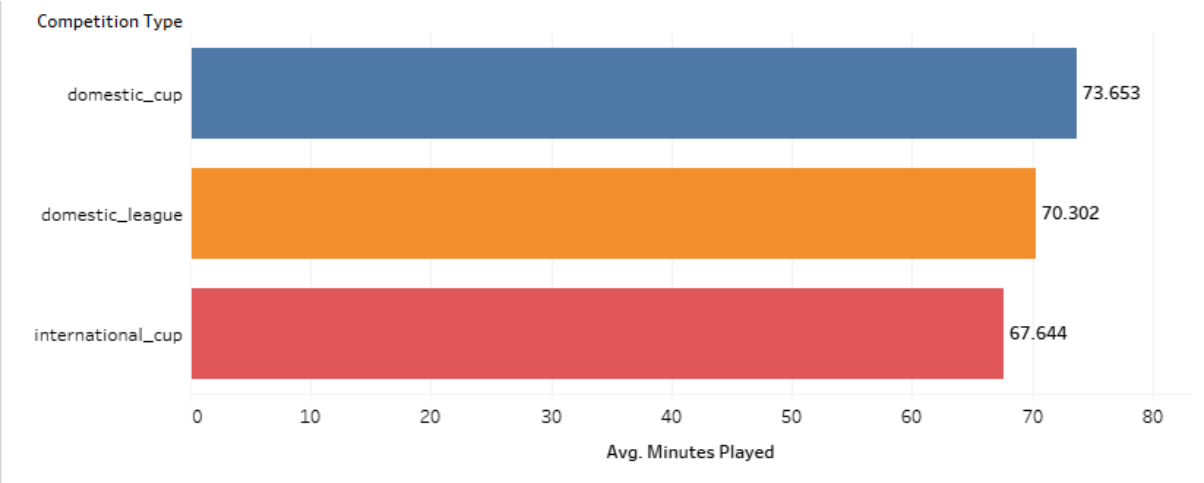


Interpretation:

Domestic Cup has the highest average goals per game at 0.2035, suggesting that matches in this competition type tend to be more offensively active.
Domestic League shows significantly lower scoring, with an average of only 0.0968 goals per game, potentially indicating tighter defenses or more cautious tactical play.
International Cup competitions see a modest scoring rate at 0.0735 goals per game, which might reflect the higher stakes or the varying team strengths that characterize such tournaments.

The variance in scoring rates could be influenced by factors like team strategies, player familiarity with opponents, or the intensity of the competition, where domestic cups might encourage more open play.

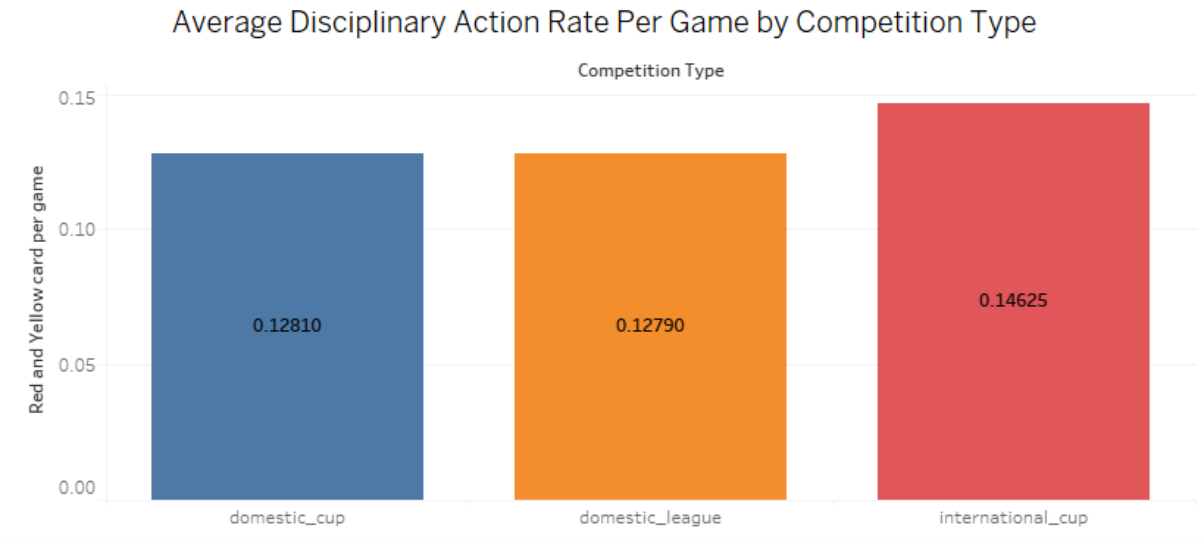
2.Which competitions have the highest minutes played per game in the last season?



Interpretation:

Domestic cup competitions have the highest average minutes played per game at 73.653 minutes. Domestic league games are a close second with 70.302 minutes, while international cup competitions have the lowest average at 67.644 minutes.

3.Which competition type shows the highest level of game discipline?



Interpretation:

International Cup matches exhibit the highest rate of disciplinary actions, with 0.14625 cards per game, likely reflecting the intense competition and high-pressure environment.
Domestic Cup games follow closely with an average of 0.12810 cards per game, which might indicate robust physical play or more stringent refereeing in knockout-style competitions.
Domestic League has the lowest average at 0.12790 cards per game, possibly due to the regularity of play and teams being more accustomed to their opponents.

The higher disciplinary rates in international and knockout competitions could suggest a need for more focused player discipline training or could influence the type of refereeing assigned to these matches.

9. Player Attributes and Demographics

Objectives:

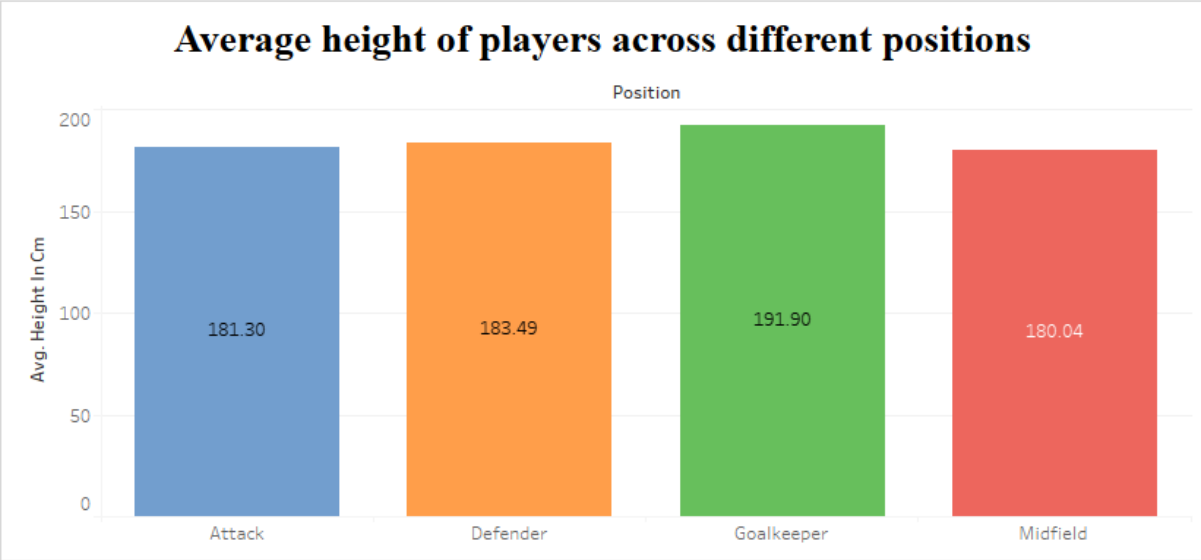
Objective 1: **Analyze Player Demographics Across Positions and Performance**

Objective 2: **Segment Players Based on Attributes and Demographics**

Business Questions:

- **Descriptive Analysis Questions:**

1.What is the average height of players across different positions?



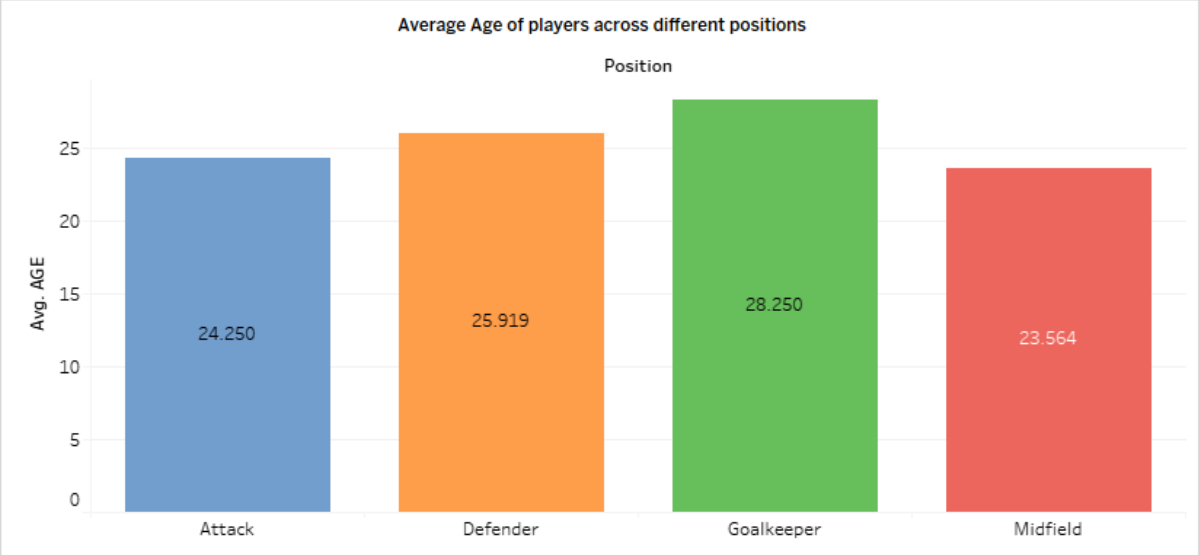
Interpretation:

This bar chart shows the average height of players in different positions:

- **Goalkeeper:** Goalkeepers are the tallest, with an average height of 191.90 cm. This makes sense as their height gives them an advantage in reaching and blocking shots.
- **Midfield:** Midfielders have an average height of 180.04 cm.
- **Defender:** Defenders average 183.49 cm in height.
- **Attack:** Attackers are the shortest on average, with a height of 181.30 cm.

Overall, there is some variation in height across positions, with goalkeepers being significantly taller than other positions. This likely reflects the physical demands and requirements of each role.

2.What is the average age of players across different positions?



Interpretation:

This bar chart shows the average age of players in different positions:

- **Goalkeeper:** Goalkeepers are the oldest on average, at 28.25 years old. This could be because the goalkeeper position relies on experience and decision-making, which often improve with age.
- **Defender:** Defenders have an average age of 25.919 years.
- **Midfield:** Midfielders are the youngest, with an average age of 23.564 years. This might be because midfield requires high levels of stamina and athleticism, which tend to peak in younger players.
- **Attack:** Attackers have an average age of 24.25 years.

The data shows a noticeable difference in average age across positions, with goalkeepers being the oldest and midfielders the youngest.

3.How does scoring ability vary by country of origin?

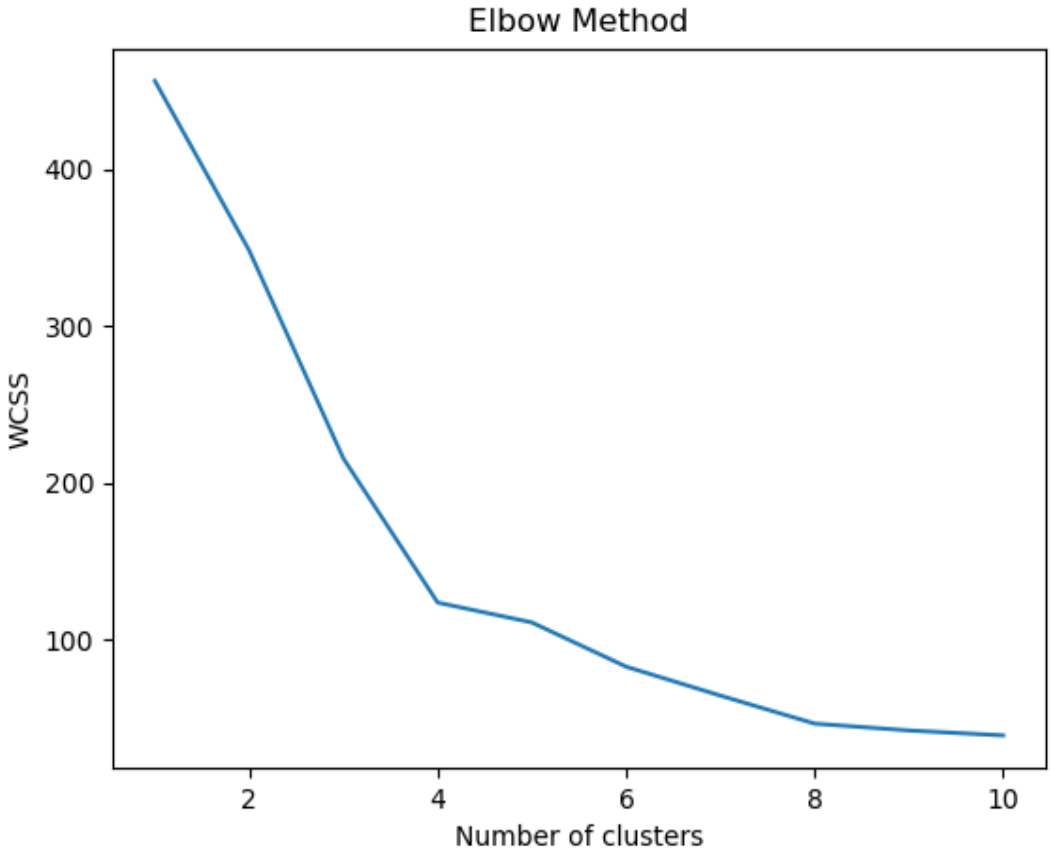
K-means Clustering Question:

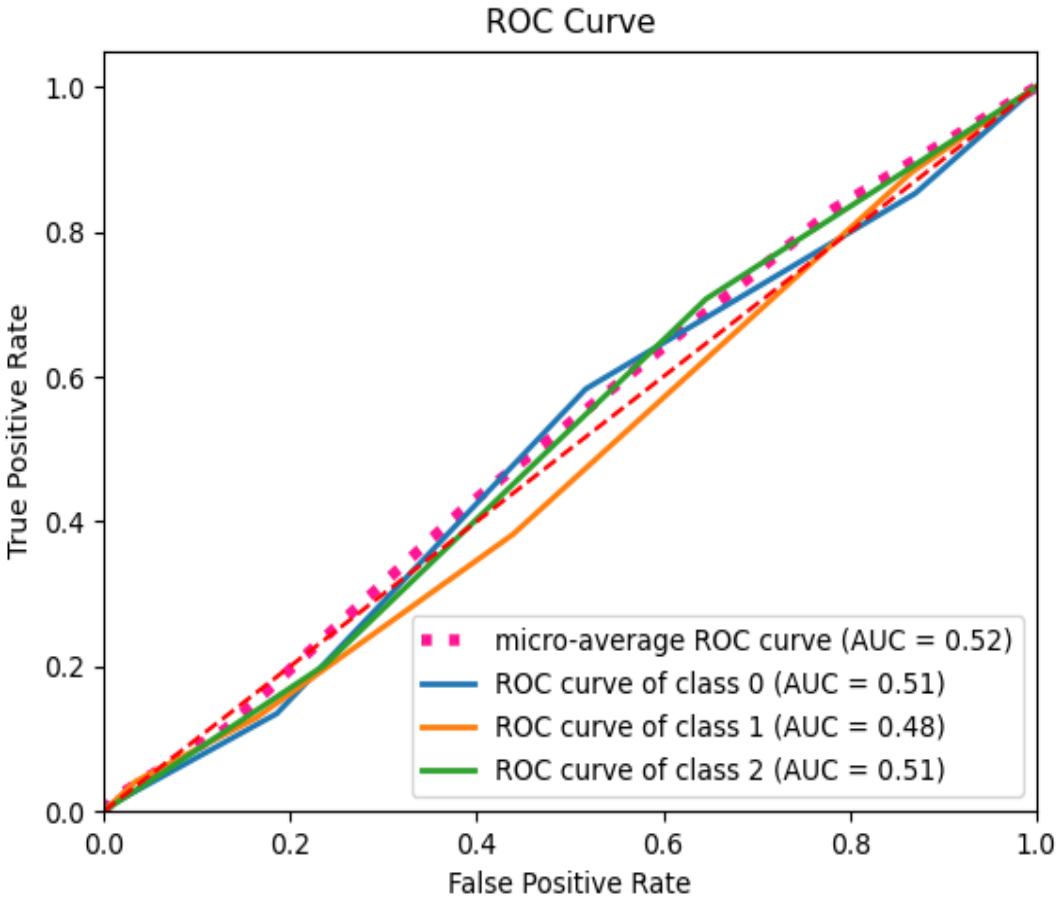
1.Identify groups of players based on their attributes and demographics.

Python Code:

```
# Assuming 'players' DataFrame is preloaded and available
df_temp = players.copy()

# Initialize LabelEncoder
label_encoder = preprocessing.LabelEncoder()

# Encode 'country_of_birth' and 'foot'
df_temp['country_of_birth_encoded'] =
label_encoder.fit_transform(df_temp['country_of_birth'])
df_temp['foot_encoded'] = label_encoder.fit_transform(df_temp['foot'])

# Preparing the data for scaling
X = df_temp[['height_in_cm', 'foot_encoded', 'country_of_birth_encoded']]
scaler = StandardScaler()

# Standardizing the features
X_scaled = scaler.fit_transform(X)
X_scaled = pd.DataFrame(X_scaled, columns=['height_in_cm', 'foot_encoded',
'country_of_birth_encoded'])

# Clustering with KMeans and determining the optimal number of clusters
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', random_state=42)
```

```
    kmeans.fit(X_scaled)
    wcss.append(kmeans.inertia_)

# Plot the elbow method graph
plt.plot(range(1, 11), wcss)
plt.title('Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.savefig('elbow_curve.png')
plt.show()

# Choose the optimal number of clusters based on the elbow method
optimal_k = 2  # Example value, should be chosen based on plot
kmeans = KMeans(n_clusters=optimal_k, init='k-means++', random_state=42)
clusters = kmeans.fit_predict(X_scaled)

# Add cluster labels to the original dataset
df_temp['Cluster'] = clusters

# Print mean values for each cluster for specific numeric columns
print(df_temp.groupby('Cluster')[['height_in_cm', 'foot_encoded']].mean())
print("Accuracy:", accuracy_score(y_test, y_pred))
print('precision_score:', precision_score(y_test, y_pred, average='micro'))
print('recall_score:', recall_score(y_test, y_pred, average= 'micro'))
print('f1_score:', f1_score(y_test, y_pred, average= 'micro'))
print('confusion_matrix:', confusion_matrix(y_test, y_pred))
print('classification_report:', classification_report(y_test, y_pred))
```


Elbow Method
WCSS
400
300
200
100
2
4
6
8
10
Number of clusters



Interpretation:

Accuracy: 0.3611111111111111
precision_score: 0.3611111111111111
recall_score: 0.3611111111111111
f1_score: 0.3611111111111111
confusion_matrix: [[105 44 14]
[112 45 13]
[66 27 6]]
classification_report: precision recall f1-score support

0.0	0.37	0.64	0.47	163
1.0	0.39	0.26	0.31	170
2.0	0.18	0.06	0.09	99
accuracy			0.36	432
macro avg	0.31	0.32	0.29	432
weighted avg	0.33	0.36	0.32	432

10. Contract Management

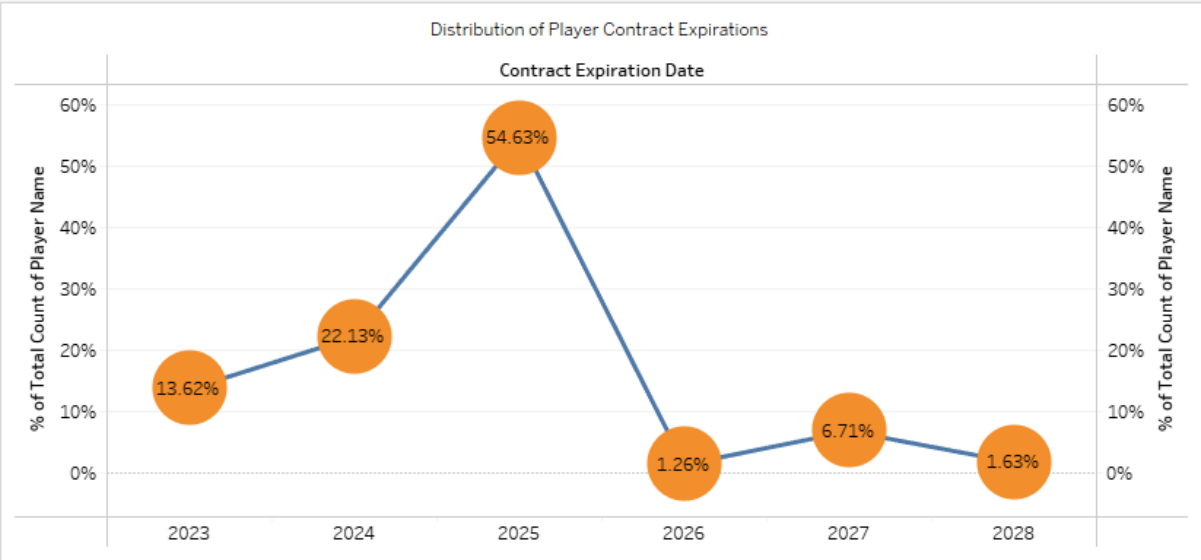
Objectives:

- **Objective 1:** Analyze patterns and trends in player contracts to optimize team management and player retention strategies.
- **Objective 2:** Evaluate the influence of agents on player contracts and performances.

Business Questions:

- **Descriptive Analysis Questions:**

1.What percentage of players' contracts are set to expire each year?

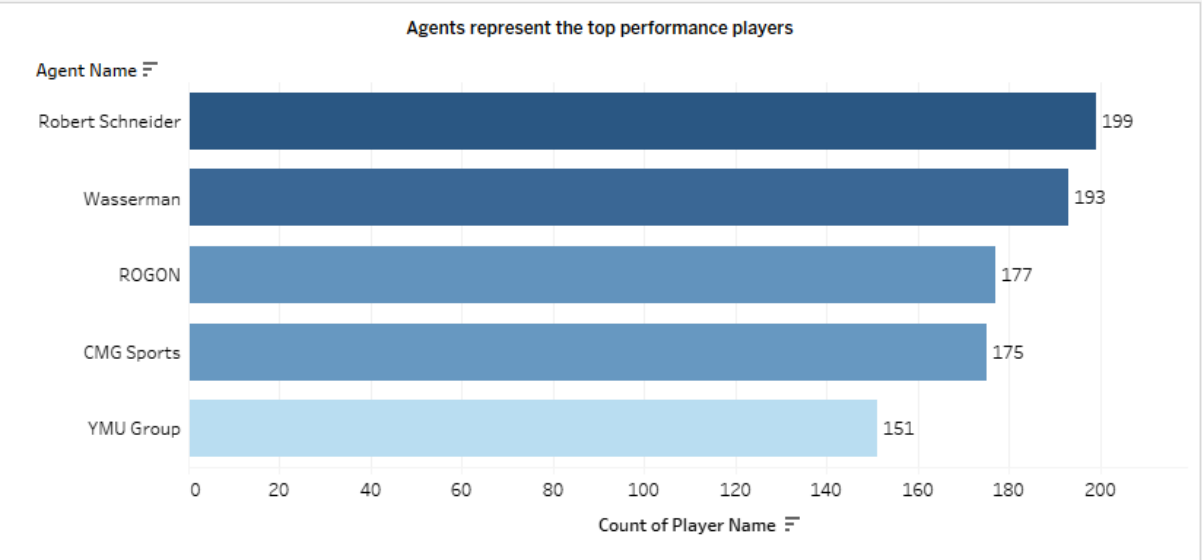


Interpretation:

The graph indicates a significant peak in contract expirations in 2025, with over half of the player contracts ending, suggesting a pivotal year for contract renewals or player negotiations.

The years 2026 through 2028 show a steep decline in expirations, highlighting a potential stability in the team's roster or a lower risk of losing key players to contract completions during this period.

2.Which agents represent the most players among top performers?

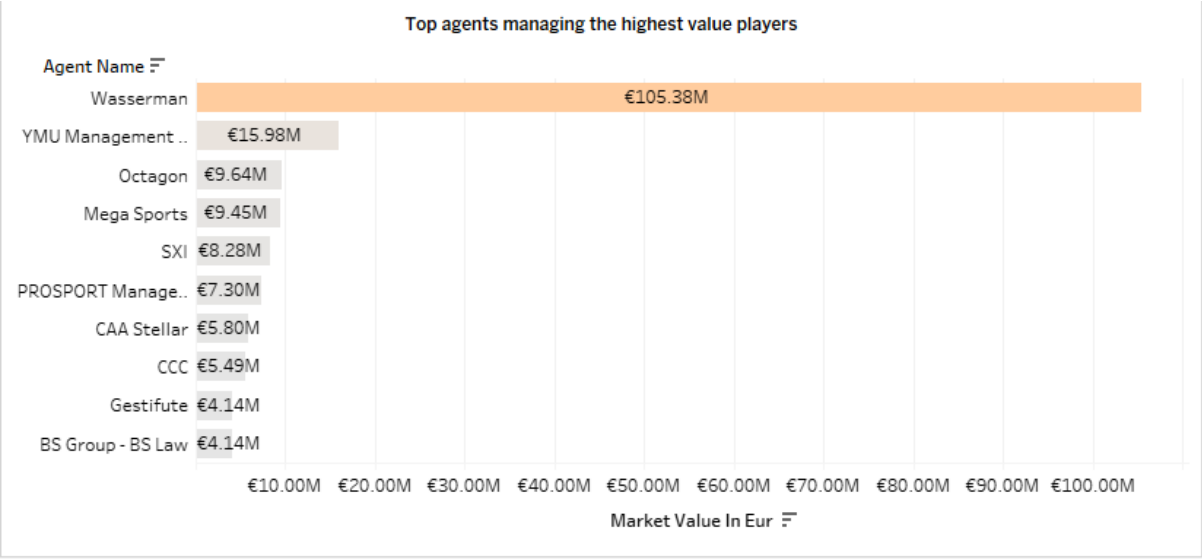


Interpretation:

The bar chart showcases that Robert Schneider and Wasserman are the top two agents, representing the highest number of top performance players, which can indicate their effectiveness in managing high-caliber talent.

This distribution suggests a significant concentration of top talent within a few agents, which could impact team negotiations and player acquisitions.

3.Who are the top agents managing the highest value players?



Interpretation:

Wasserman stands out as managing players with a significantly higher total market value compared to other agents, indicating their prominence in the industry.
This visualization highlights not only the disparity in player valuations managed by different agents but also suggests that high-value players tend to cluster with certain top-tier agents, affecting the dynamics of player transfers and salary negotiations.

Summarization of all Interpretations

1. Performance Analysis

- Christian Pulisic and Timothy Chandler were the top goal scorers in the latest season.
- Goalkeepers play the most minutes per game on average, while attackers play the least.
- There's a negative trend in player performance from 2012 to 2020, with both the number of games played and goal contributions decreasing.

2. Player Profile and Market Value

- Right midfielders have the highest market value, while left wingers have the lowest.
- The United States and several European countries have higher player market values.
- Average market values have increased from 2012 to 2020 for all competition types.

3. Team Comparison

- Lucien Favre's teams have scored the most goals in both home and away games.
- Borussia Verein scores the most goals at home, while Borussia Dortmund scores the most away.

4. Attendance and Stadium Analysis

- Average match attendance has generally increased from 2012 to 2020, with some dips in certain years.
- SIGNAL IDUNA PARK has the highest average attendance among all stadiums.
- International cup competitions have the highest average attendance, while domestic cup competitions have the lowest.

5. Referee Analysis

- Felix Zwayer has issued the most yellow and red cards.
- Defenders receive the most cards, while goalkeepers receive the fewest.
- Marco Fritz and Daniel Siebert officiated the games with the most goals.

6. Substitution Patterns

- Attackers are substituted most frequently.
- There's significant variation in playing time for substitutes.
- Domestic league games have the most substitutions.

7. Event Analysis

- Defenders commit the most fouls, while goalkeepers commit the fewest.
- Fouls are the most common game event.
- Substitutes receive more red and yellow cards than starting players.

8. Competition Analysis

- Domestic cup competitions have the highest average goals per game.
- Domestic cup competitions also have the highest minutes played per game.
- International cup competitions have the highest rate of disciplinary actions.

9. Player Attributes and Demographics

- Goalkeepers are the tallest, while attackers are the shortest.
- Goalkeepers are the oldest, while midfielders are the youngest.

10. Contract Management

- Most player contracts are set to expire in 2025.
- Robert Schneider and Wasserman represent the most top-performing players.
- Wasserman manages players with the highest total market value.

Final Conclusion

This comprehensive analysis of football match data, spanning from 2012 to 2020, has unveiled key trends and patterns that can significantly impact strategic decision-making in the football industry.

Player Performance & Market Value:

- A noticeable decline in player performance is observed over the years, with both the number of games played and goal contributions decreasing.
- This trend underscores the need for strategic adjustments in player management and game participation to enhance performance and maximize player potential.
- Despite the performance dip, average market values have steadily increased across all competition types, indicating the growing economic value of players and the potential for financial investment in the industry.

Team Performance & Home Advantage:

- Lucien Favre stands out as a manager whose teams consistently score high in both home and away games, suggesting the effectiveness of his tactics and strategies.
- Home advantage is evident, with certain teams like Borussia Verein demonstrating exceptional goal-scoring proficiency at home.

Attendance Dynamics:

- The overall upward trend in average match attendance from 2012 to 2020 indicates the growing popularity of the sport and the potential for increased revenue generation.
- Identifying and understanding the factors that contribute to occasional dips in attendance, such as those seen in 2016 and 2018, is crucial for maintaining a positive trend and maximizing spectator interest.

Referee Influence:

- Referee behavior significantly impacts games, as seen with Felix Zwayer issuing the most cards, indicating a stricter officiating style.
- Recognizing these patterns can help in assigning referees strategically to matches based on the desired level of control and game flow.

Game Events & Substitutions:

- The high frequency of fouls highlights the physicality of the game and the importance of player discipline and training.
- Understanding substitution patterns, such as the frequent substitution of attackers, can provide insights into tactical adjustments and player fatigue levels.

Competition Format:

- Domestic cup competitions emerge as the most offensively active, with the highest average goals per game and minutes played per game.

- This observation suggests that competition format influences game intensity and potentially encourages more open play.

Player Attributes & Contract Expirations:

- The analysis reveals correlations between player positions, height, and age, reflecting the physical and tactical demands of each role.
- The peak in contract expirations in 2025 highlights a critical period for contract renewals and player negotiations, requiring careful planning and management.

In conclusion, these findings provide a comprehensive overview of the football industry, enabling data-driven decisions to optimize player performance, team strategies, and business operations.