CAPSTONE PROJECT: FOOTBALL DATA ANALYSIS

TOOLS: JUPYTER NOTEBOOK, SQL WORKBENCH, TABLEAU

DATASETS: APPEARANCE, GAMES, GAMEEVENT, GAMELINEUP AND PLAYER DATASETS

STEP 1: DATA PREPROCESSING

1.1 DATA CLEANING

- Null value treatment (Mean - numerical, Mode - categorical columns: Imputing missing values) for each dataset.
- Formatting data types for the columns in each dataset.
- Removing unnecessary columns in each dataset.
- Save the cleaned data file as excel file.
- Solution is provided in the Python notebook

1.2 DATA MERGING

Python

- Merge data according to the common columns in the data files (Foreign Key columns).
- Solution is provided in the Python notebook.
- Save the final files as ‘Football_dataclean.xlsx’ excel file.

STEP 2: Data Dictionary (USING CHATGPT)

In ChatGPT, write prompt as:

As a data analyst, provide data dictionary for the following data with description in tabular form.

Copy the first few samples (10-15) from the data file ‘Football_dataclean.xlsx’ file including column names and paste it here.

Here is a data dictionary for the provided data:

COLUMN NAME	DESCRIPTION
appearance_id	Unique identifier for each player appearance
game_id	Unique identifier for each game
player_id	Unique identifier for each player

date	Date of the game
player_name	Name of the player
competition_id	Unique identifier for each competition
yellow_cards	Number of yellow cards received by the player
red_cards	Number of red cards received by the player
goals	Number of goals scored by the player
assists	Number of assists made by the player
minutes_played	Number of minutes played by the player
season	Season in which the game took place
round	Round of the competition
home_club_goals	Number of goals scored by the home club
away_club_goals	Number of goals scored by the away club
home_club_position	Position of the home club in the league
away_club_position	Position of the away club in the league
home_club_manager_name	Name of the home club manager
away_club_manager_name	Name of the away club manager
stadium	Stadium where the game was played
attendance	Number of spectators in attendance
referee	Name of the referee
home_club_name	Name of the home club
away_club_name	Name of the away club
aggregate	Aggregate score of the game
competition_type	Type of competition (domestic or international)
name	Player name for reference
last_season	Last season the player participated in
current_club_id	Unique identifier for the player’s current club
player_code	Code representing the player
country_of_birth	Player’s country of birth
date_of_birth	Player’s date of birth
sub_position	Sub-position of the player
position	Primary position of the player
foot	Player’s dominant foot
height_in_cm	Player’s height in centimeters
market_value_in_eur	Player’s current market value in euros
highest_market_value_in_eur	Player’s highest market value in euros

contract_expiration_date	Date when the player’s contract expires
agent_name	Name of the player’s agent
minute	Minute of the event in the game (goal, card, substitution, etc.)
type	Type of event (e.g., Goals, Substitutions, Cards)

NOTE: The above data dictionary includes only the unique fields of the merged dataset (removed duplicate columns (e.g., game_id_x and game_id_y)).

STEP 3: Generate Business Questions using ChatGPT

Prompt: As a data analyst expert, formulate all possible business questions to perform descriptive analysis for each of the business objectives given below:

Football Data Analysis: Focus Areas and Business Objectives

Focus areas:1. Performance Analysis, 2. Player Profile and Market Value, 3. Team Comparison, 4. Attendance and Stadium Analysis, 5. Referee Analysis, 6. Substitution Patterns, 7. Event Analysis, 8. Competition Analysis, 9. Player Attributes and Demographics, 10. Contract Management

For each of the focus areas in Football Data Analysis, here are potential business questions that can guide a descriptive analysis:

1. Performance Analysis:

1. Which players have scored the most goals?
2. What is the top-performing teams in terms of goals scored?
3. What is the distribution of goals between home and away teams?
4. How does the average number of goals vary across different competitions?
5. How does the number of goals differ between home and away matches?
6. Which team has the best defensive record (fewest goals conceded)?
7. What is the win-loss-draw ratio for each team across the season?
8. How many clean sheets (no goals conceded) has each goalkeeper achieved?

2. Player Profile and Market Value

1. What is the distribution of players' market values across different teams?
2. Which players have the highest market value in each position?

3. How has the market value of players evolved over the seasons?
4. What is the relationship between a player’s performance metrics (goals, assists) and their market value?
5. Which players have experienced the highest increase/decrease in market value?
6. How does the player's age affect their market value?

3. Team Comparison

1. Which team has the highest average number of goals scored per match?
2. How does team performance (wins, losses, draws) vary across different competitions?
3. What is the average possession rate of each team during matches?
4. How does the number of goals conceded compare between top teams?
5. How do different teams perform in terms of passing accuracy, shots on target, and tackles?

4. Attendance and Stadium Analysis

1. Which stadiums have the highest average attendance across the season?
2. How does attendance vary between home and away games for each team?
3. What is the relationship between a team's performance and the average match attendance?
4. Which matches had the highest attendance during the season?
5. Which competitions or matchdays attract the highest/lowest attendances?

5. Referee Analysis

1. How many yellow/red cards have been given by each referee?
2. Which referees officiated the most matches during the season?
3. What is the average number of fouls called per referee?
4. How does the number of penalties awarded vary by referee?
5. Which referees tend to officiate matches with the highest number of cards given?

6. Substitution Patterns

1. What is the average minute of the game when substitutions are made?
2. How does the number of substitutions vary between winning and losing teams?
3. Which players are most frequently substituted during a match?
4. What is the distribution of substitutions made by position (e.g., attackers, midfielders)?
5. How do substitutions impact the likelihood of a team scoring goals after the substitution?

7. Event Analysis

1. What is the distribution of events such as goals, cards, and substitutions across a match?
2. Which types of events (goals, assists, cards) are most common in specific time intervals (e.g., first 15 minutes, last 10 minutes)?
3. How do event occurrences vary between teams in home vs. away games?
4. Which players are most frequently involved in events like goals, assists, and fouls?

8. Competition Analysis

1. Which competitions have the highest average goals per game?
2. Which competitions attract the most spectators on average?
3. How does the competitive balance (difference in team strength) vary across leagues?
4. What is the distribution of teams from different leagues participating in international competitions?

9. Player Attributes and Demographics

1. What is the average age of players across different teams and positions?
2. How does a player's height and weight vary by position?
3. What is the distribution of players' preferred foot (left/right) across teams?
4. Which countries produce the most players in the league?
5. What is the average number of seasons played by players across different leagues?

10. Contract Management

1. What is the average contract length of players across teams?
2. Which players have contracts expiring within the next season?
3. What is the relationship between a player's market value and their contract expiration date?
4. Which agents represent the most valuable players in the league?

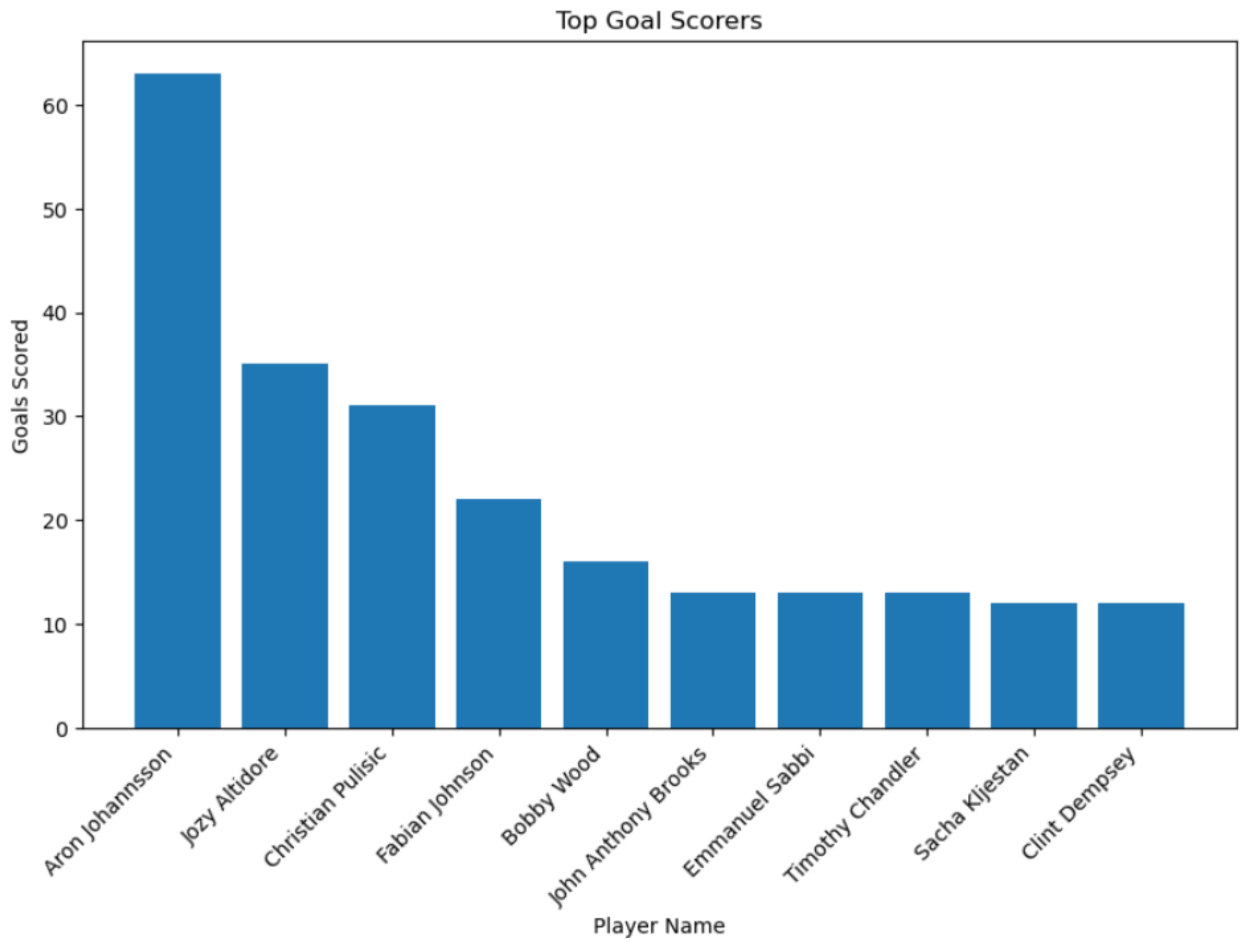
SOLUTION TO THE ANALYSIS QUESTIONS

BUSINESS OBJECTIVE 1: PERFORMANCE ANALYSIS

BUSINESS QUESTIONS:

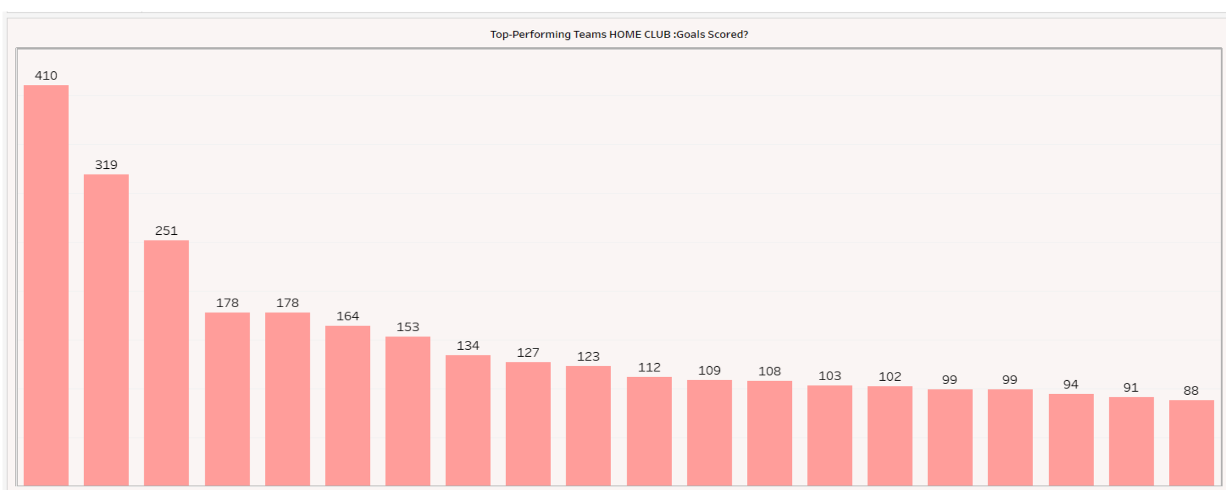
1. Which players have scored the most goals?
2. What is the top-performing teams in terms of goals scored?
3. What is the distribution of goals between home and away teams?
4. How does the average number of goals vary across different competitions?

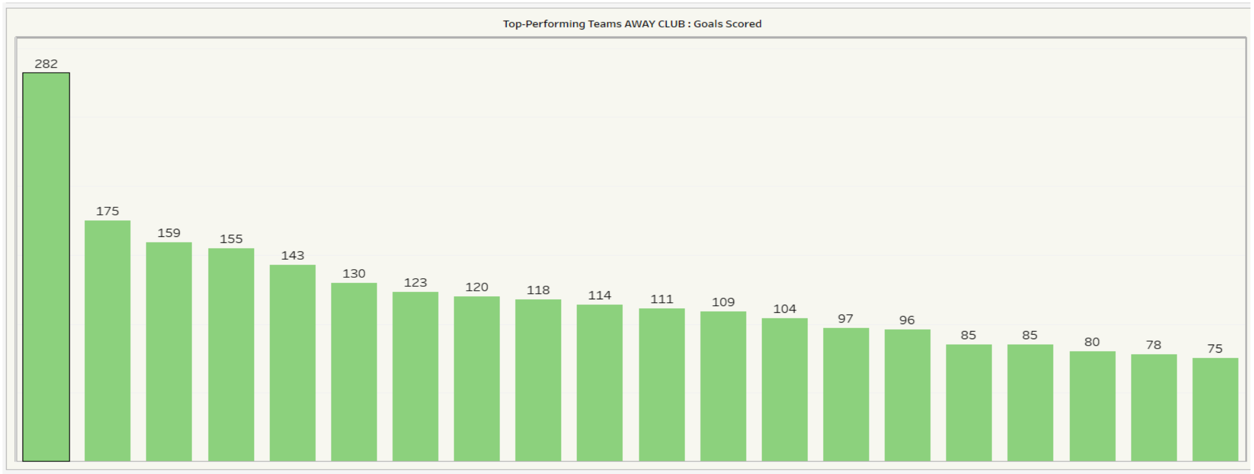
1. Which players have scored the most goals? (Python)



INTERPRETATION: Aron Johannsson leads the pack with 63 goals(count), followed by Jozy Altidore (35 goals) and Christian Pulisic (31 goals).

2. What is the top-performing teams in terms of goals scored? (Tableau)

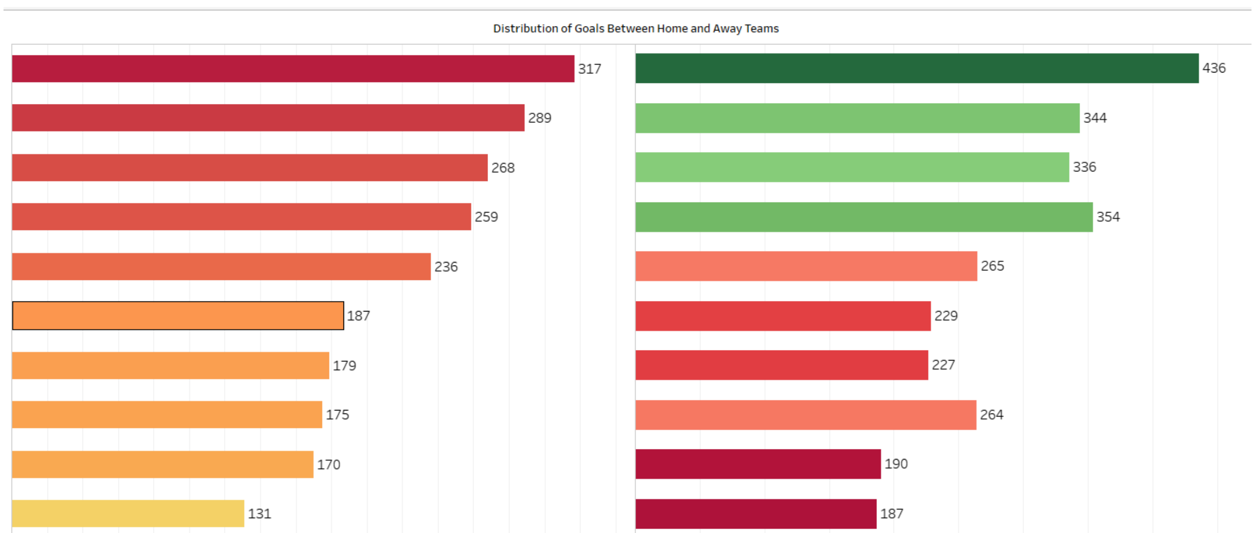




INTERPRETATION: Home Club: Borussia Verein Fuir Leibesubung 1900e.v - Goals: 410 and Borussia Dortmund – Goals: 319 has the highest in-home club

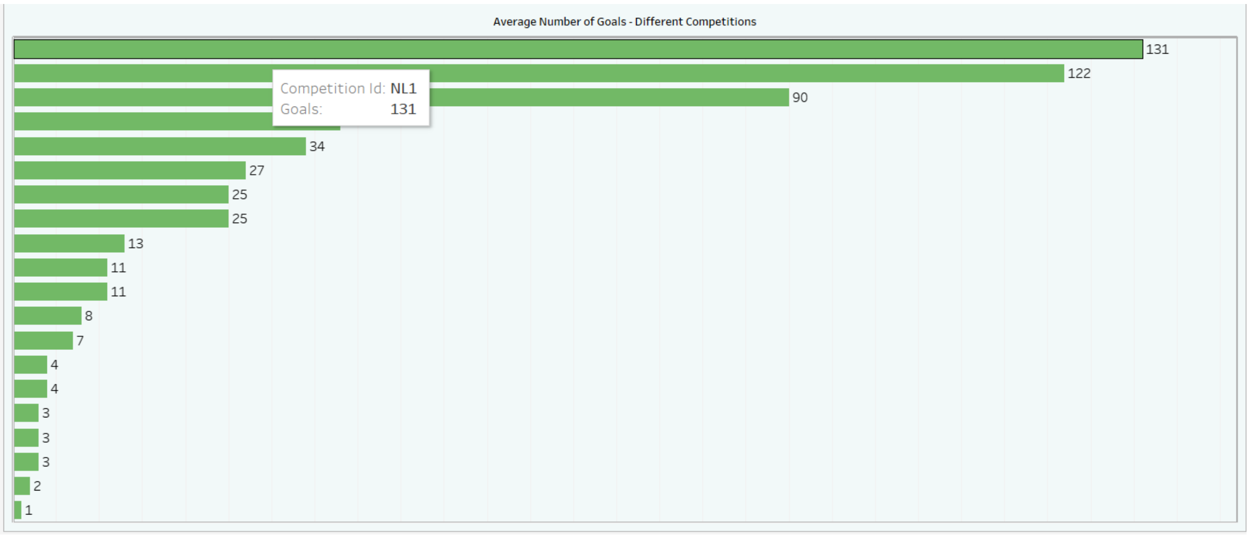
Away Club: Borussia Dortmund – Goals: 282 and FC Schalke 04 – Goals :175 has the highest goals in away club

3. What is the distribution of goals between home and away teams? (Tableau)



INTERPRETATION: Player Fabian Johnson lead the table with 436 home goals and 317 away goals followed by Christian Pulisic with 344 home goals and 289 away goals.

4. How does the average number of goals vary across different competitions? (Tableau)

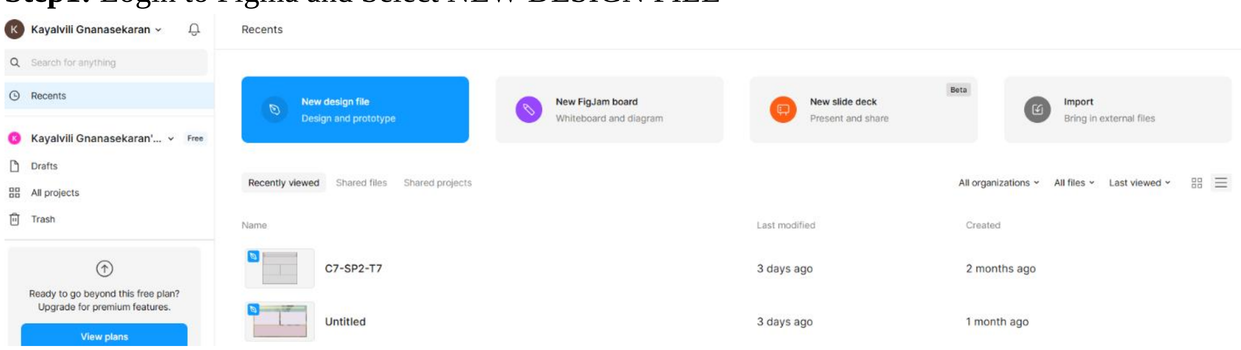


INTERPRETATION: Competition NL1 leads the chart with highest sum of goals of 131 and followed by competition ID L1 with goals of 122.

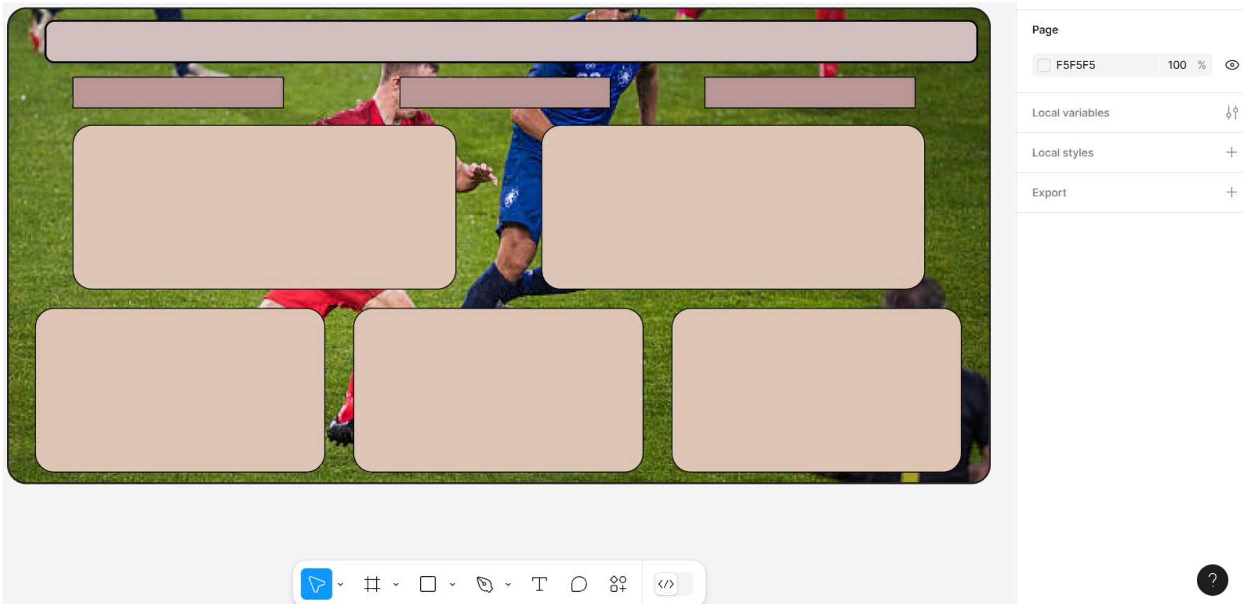
TABLEAU DASHBOARD:

1. Visualizing business questions in different worksheets
2. Create & style an Figma outline for the dashboard based on the worksheets to be added in the dashboard.

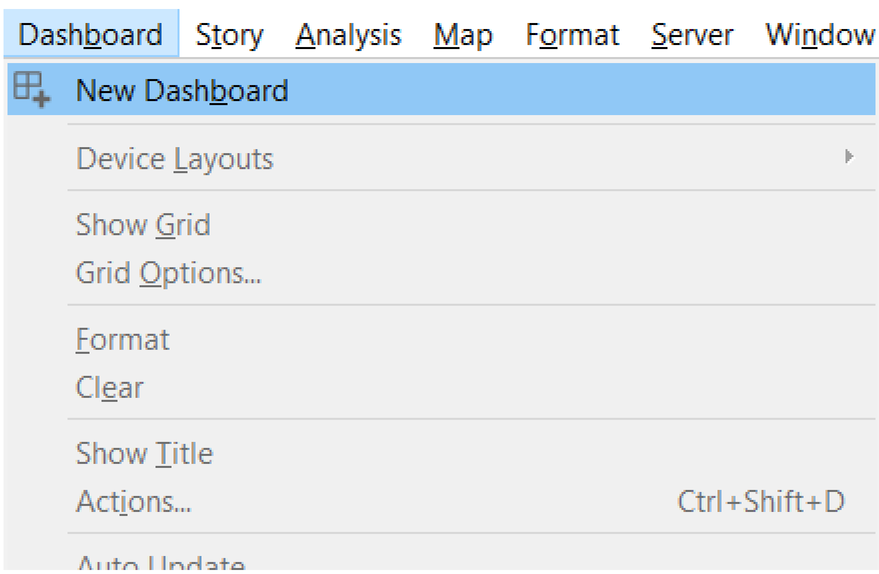
Step1: Login to Figma and Select NEW DESIGN FILE



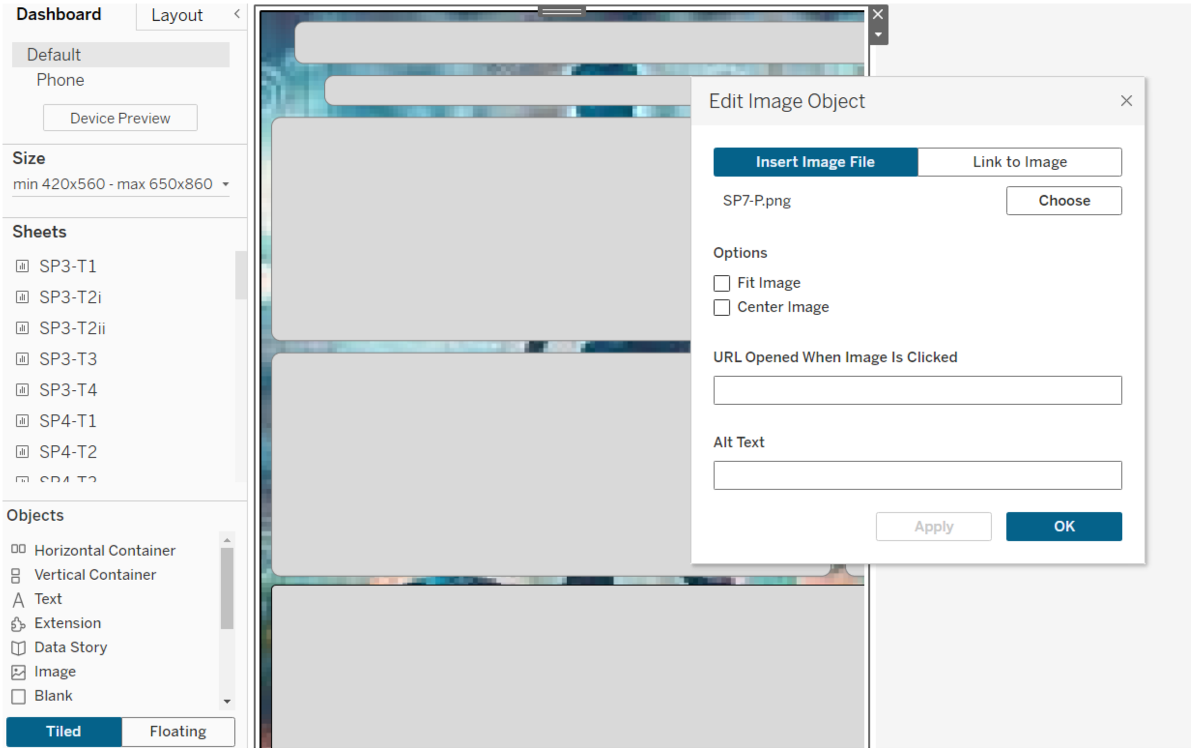
Step 2: Use shapes and designing objects in the page to create an outline for the dashboard. And click on the export option to download the outline as an image.



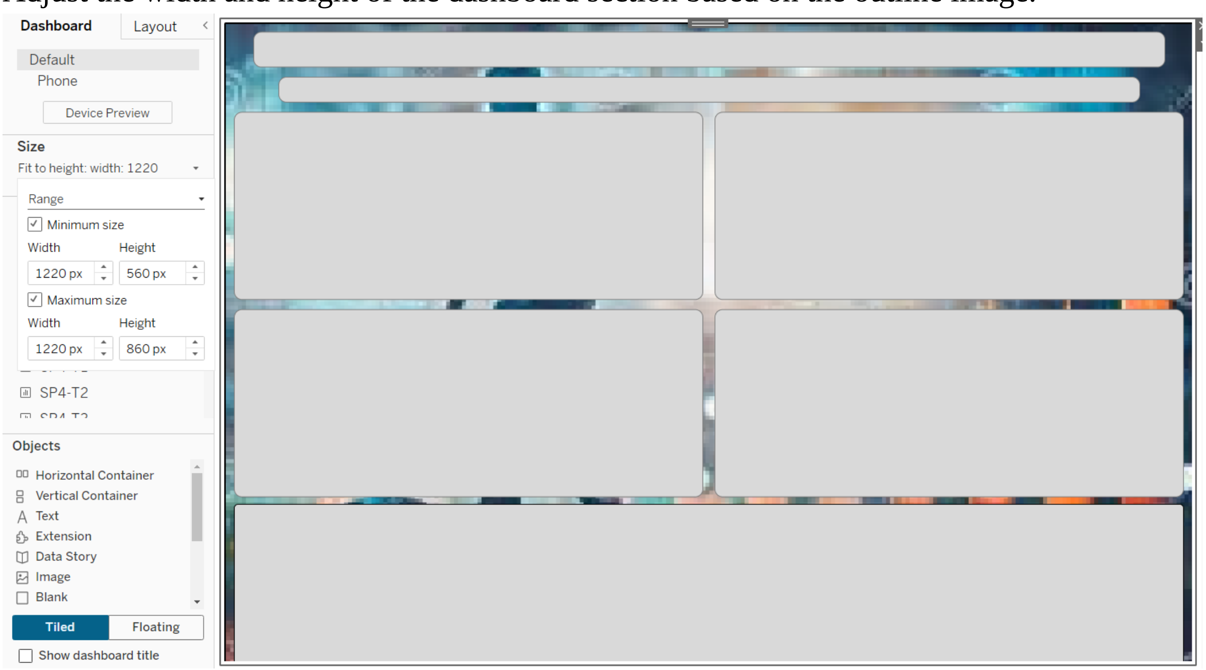
3. Add a dashboard to the tableau workbook



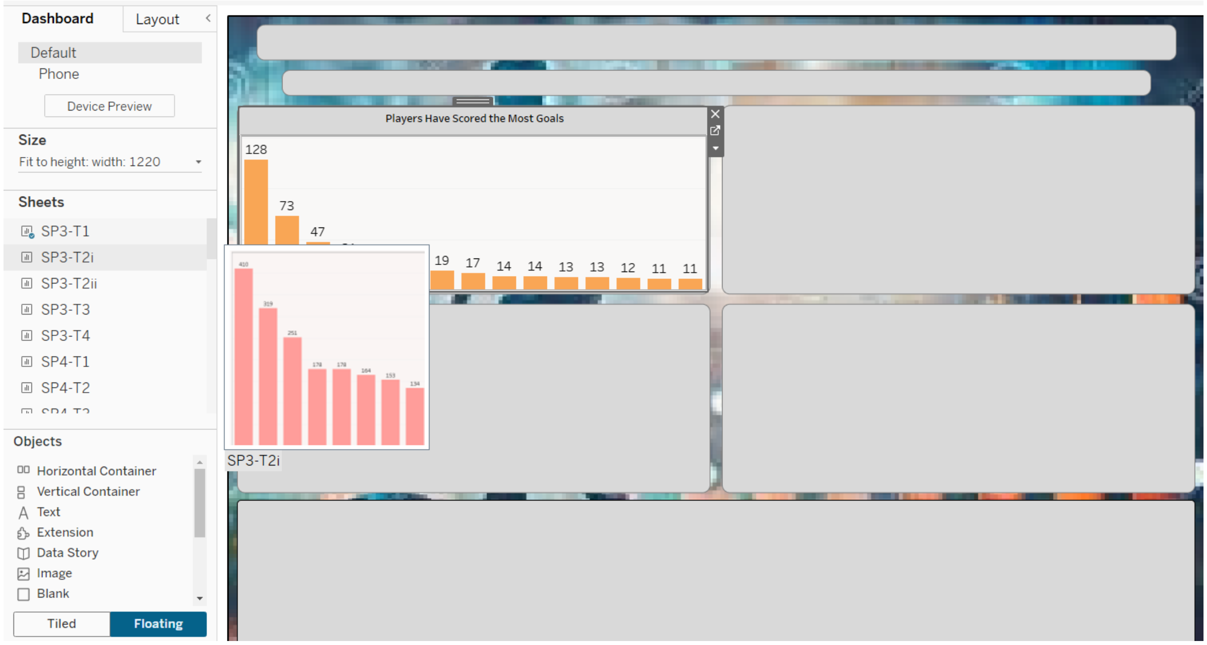
4. Choose the image from the object section to inset the exported Figma dashboard outline and click ok to apply the outline image in the visualization section.



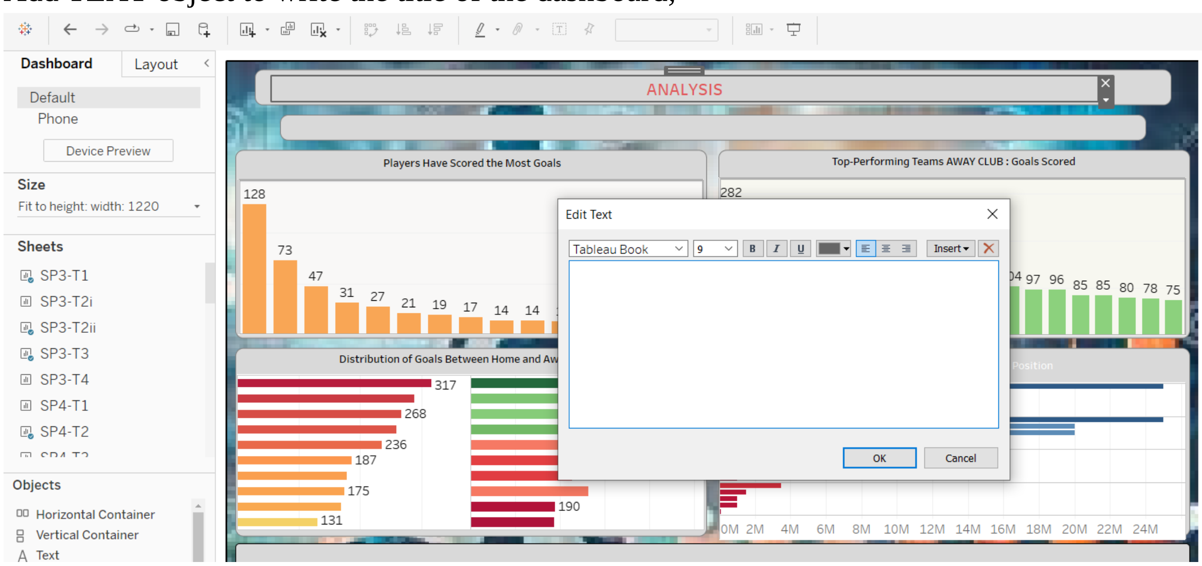
5. Adjust the width and height of the dashboard section based on the outline image.



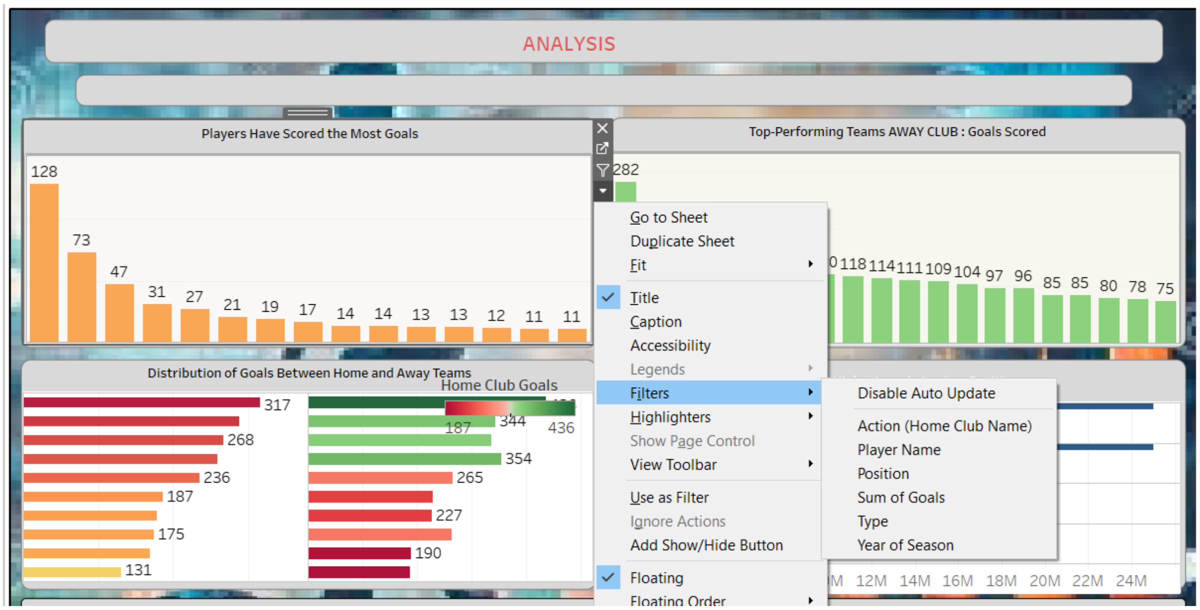
6. Add the necessary worksheets to the dashboard (as a floating sheet). Adjust the sheets based on the outline section size.

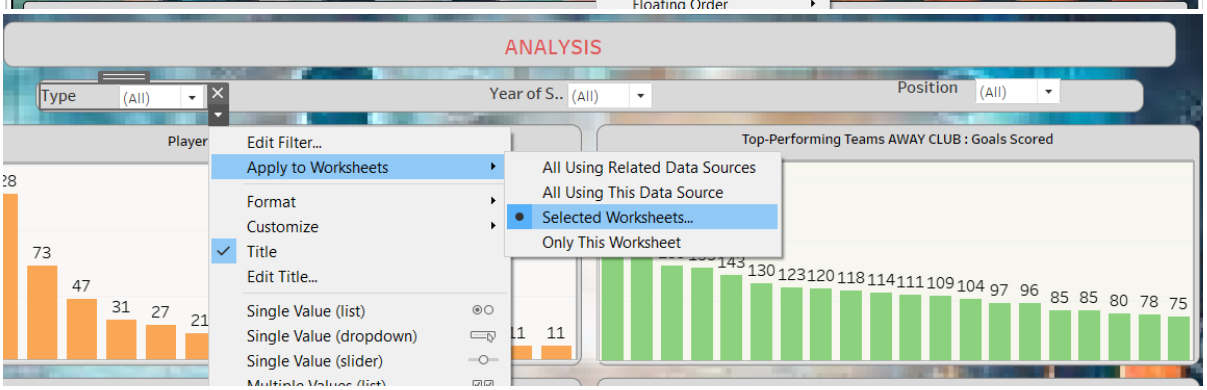


7. Add TEXT object to write the title of the dashboard,

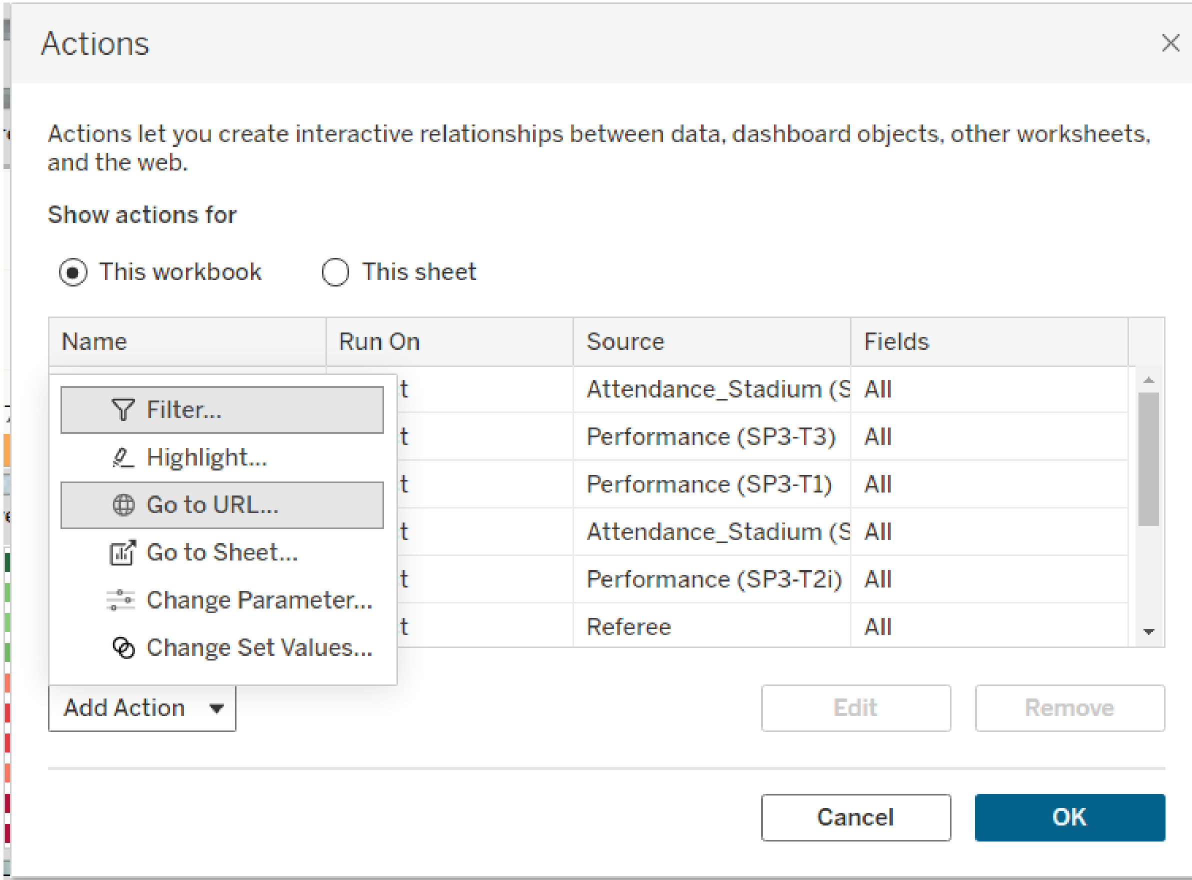


8. Add the required filters of the worksheets to the dashboards. And based on the needs, the filters can be applied to other worksheets in the dashboard.





9. Use ‘ACTION’ option under ‘DASHBOARD’ tab to add action filters or other required options on the worksheets of the dashboards.



10. Add ‘NAVIGATION’ from the object section, to navigate from one dashboard or worksheet to another dashboard / worksheets. Edit the navigation button based on the requirements.

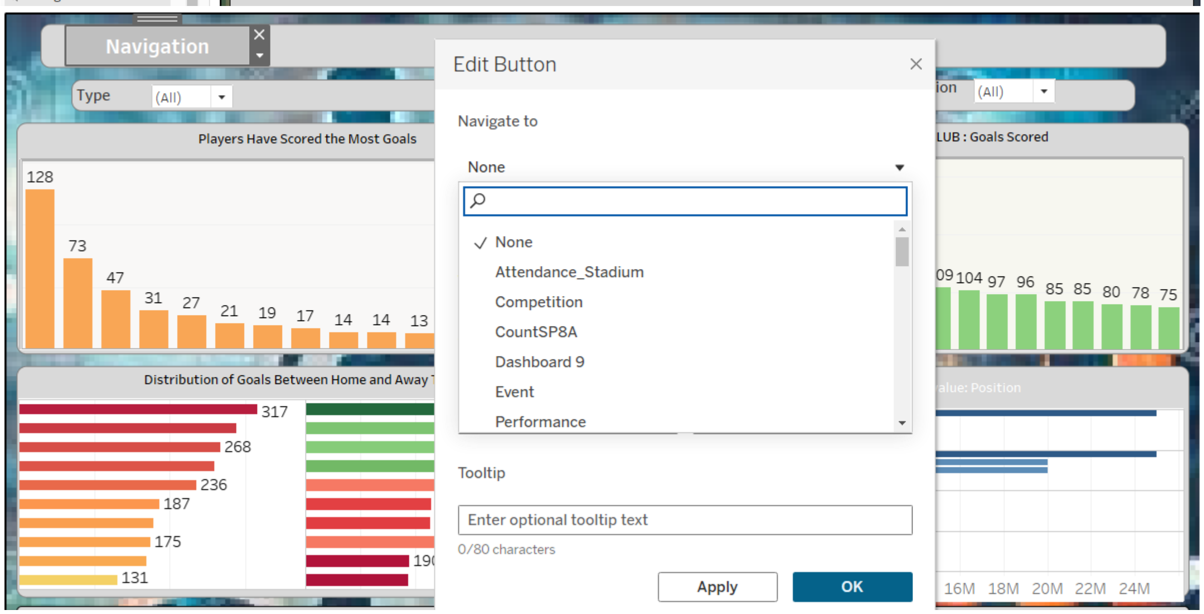


TABLEAU DASHBOARD: PERFORMANCE ANALYSIS

1. Added worksheets, filters, text to the dashboard.
2. Added navigation button to navigate from ‘performance analysis’ dashboard to player ‘profile and market value’ dashboard.
3. Applied type, position and season filter to required worksheets in dashboard.
4. Added actions to Task 1, 3, 2i sheets to all sheets.

PROBABILITY QUESTIONS: (Python)

1. What is the probability of a home team winning a match?
2. What is the probability that a goal is scored after a substitution?

1. What is the probability of a home team winning a match?

P (Home Win) = Number of Home Wins/ Total Number of Matches

```
# Define a home team win condition

home_wins = fc_df[fc_df['home_club_goals'] > fc_df['away_club_goals']]

# Calculate probability

prob_home_win = len(home_wins) / len(fc_df)

print ('Probability of home team winning a match is:', prob_home_win)
```

OUTPUT: Probability of home team winning a match is: 0.450249

INTERPRETATION: The output indicates that the probability of a home team winning a match is approximately 45.02%

2. What is the probability that a goal is scored after a substitution?

JOINT PROBABILITY: P (Goal after Substitution) = Number of Games with Substitution and Goal/ Total Number of Matches

```
# Filter events for goals and substitutions
```

```
substitutions = fc_df[fc_df['type'] == 'Substitutions']

goals = fc_df[fc_df['type'] == 'Goals']

# Merge to find if a goal was scored within 5 minutes after a substitution

merged_events = pd.merge_asof(substitutions.sort_values('minute'), goals.sort_values('minute'), on='minute', direction='forward', tolerance=5)

# Calculate probability

prob_goal_after_sub = len(merged_events) / len(fc_df)

print ('Probability of a goal scored after a substiution is:',prob_goal_after_sub)
```

OUTPUT: Probability of a goal scored after a substitution is: 0.843791

INTERPRETATION: The code calculates the probability of a goal being scored after a substitution. The output indicates that the probability of a goal scored after a substitution is approximately 84.38%.

Data Sampling and Hypothesis test Questions:

1. What is the average number of goals scored by home teams, for a random sample?

```
import statistics as st

# Assume PA_df contains the PerformanceData

home_goals = fc_df['home_club_goals'].dropna()  # Remove any NaN values for home goals

away_goals = fc_df['away_club_goals'].dropna()

# Set a sample size (let's say 20 games)

sample_size = 20

# Take a random sample from home goals

sampleh = np.random.choice(home_goals, size=sample_size, replace=False)

samplea = np.random.choice(away_goals, size=sample_size, replace=False)

# Calculate the sample mean

sample_mean_h = np.mean(sampleh)

sample_mean_a = np.mean(samplea)

print(f'Sample Mean of Home Goals: {sample_mean_h}')
```

print(f'Sample Mean of Away Goals: {sample_mean_a}')

OUTPUT: Sample Mean of Home Goals : 1.15
Sample Mean of Away Goals: 1.35

INTERPRETATION: The code calculates the sample mean of home goals and away goals from a random sample of 20 games. The sample mean of home goals is 1.65, and the sample mean of away goals is 1.15.

2. Is the average number of goals scored in home games significantly different from the average number of goals scored in away games?

Using two-tailed test

Null Hypothesis (H_0): There is no significant difference between the average goals scored by home teams and away teams

Alternate Hypothesis (H_1): There is a significant difference between the average goals scored by home teams and away teams.

CODE:

sample of home goals: [1, 0, 1, 2, 5, 3, 3, 4, 2, 2, 0, 3, 2, 1, 3, 1, 1, 1, 2, 3]

sample of away goals: [1, 1, 1, 1, 2, 3, 4, 1, 2, 4, 3, 1, 1, 1, 2, 1, 2, 1, 0, 1]

Sample Mean of Home Goals: 2.0

Sample Mean of Away Goals: 1.6

α=05

import scipy.stats as stats

import matplotlib.pyplot as plt

import statistics as st

Given sample data

home_goals = [1, 0, 1, 2, 5, 3, 3, 4, 2, 2, 0, 3, 2, 1, 3, 1, 1, 1, 2, 3]

away_goals = [1, 1, 1, 1, 2, 3, 4, 1, 2, 4, 3, 1, 1, 1, 2, 1, 2, 1, 0, 1]

Calculate the sample means

S_mean_home = 2.0

S_mean_away = 1.65

```
# Calculate the sample standard deviations
std_home = np.std(home_goals, ddof=1)
std_away = np.std(away_goals, ddof=1)
# Calculate the sample sizes
n_home = len(home_goals)
n_away = len(away_goals)
dof = n_home + n_away - 2
home_var=std_home**2
away_var=std_away**2
# Output results
print(f"Sample Mean (Home Goals): {S_mean_home}")
print(f"Sample Mean (Away Goals): {S_mean_away}")
print(f"Standard deviation (Home Goals): {std_home}")
print(f"Standard deviation (Away Goals): {std_away}")
print ("Size of home_goals : ", n_home)
print ("Size of away_goals :",n_away)
print ("Variance of home_goals : ",home_var)
print ("Variance of away_goals : ",away_var)
import math
from scipy.stats import t
num=abs(S_mean_home-S_mean_away)
denom=math.sqrt(home_var*(n_home-1)+away_var*(n_away-1))*
math.sqrt((1/n_home)+(1/n_away)
tstats=num/denom
print ("T Statistics: ", tstats)
tcritical=t.ppf(alpha/2, dof)
print ("T Critical: ",tcritical)
```

```
p_val=t.sf(abs(tstats),dof)*2

print ("P Value:",p_val)

# Shade the rejection regions

x_reject = np.linspace(tcritical, 4, 100)

y_reject = stats.t.pdf(x_reject, dof)

plt.fill_between(x_reject, y_reject, 0, color='salmon', label='Rejection regions (|t| > {})'.
format(tcritical))

x_reject = np.linspace(-4, -tcritical, 100)

y_reject = stats.t.pdf(x_reject, dof)

plt.fill_between(x_reject, y_reject, 0, color='salmon')

# Plot the test statistic

plt.axvline(tstats, color='blue', linestyle='--', label='Test Statistic (t= {})'. format(tstats))

# Set plot labels and title

plt.xlabel('T-value')

plt.ylabel('Probability Density')

plt.title('T-Test: Home vs Away Goals (alpha = {})'. format(alpha))

plt.legend()

plt.show()

if p_val<0.5:

    print ("Reject the null hypothesis: There is a significant difference between home and away
goals.")

else:

    print ("Fail to reject the null hypothesis: There is no significant difference between home and
away goals.")
```

OUTPUT:

```
Sample Mean (Home Goals): 2.0
Sample Mean (Away Goals): 1.65
Standard deviation (Home Goals): 1.2977713690461004
Standard deviation (Away Goals): 1.0894228312566052
```

Size of home_goals : 20
Size of away_goals : 20
Variance of home_goals : 1.6842105263157896
Variance of away_goals : 1.1868421052631577

T Statistics: 0.14985480293104797
T Critical: -0.6810008783354652
P Value: 0.881671738163025

Fail to reject the null hypothesis: There is no significant difference between home and away goals.

INTERPRETATION: Fail to reject the null hypothesis: Since the test statistic (t = 0.15) is far from the critical values and lies within the acceptance region, and the p-value (0.88) is much greater than the significance level (alpha = 0.1), we fail to reject the null hypothesis.

OVERALL INTERPRETATION:

1. **Top Scoring Players**:
   a. The leading goal scorers are Aron Johannsson (63 goals), Jozy Altidore (35 goals), and Christian Pulisic (31 goals). This indicates that Johannsson is the most prolific scorer, providing significant contributions to his team's performance.
2. **Top Performing Teams**:
   a. The analysis highlights Borussia Verein Fuir Leibesubung 1900e.v as the top home team (410 goals), followed by Borussia Dortmund (319 goals). For away matches, Borussia Dortmund leads with 282 goals, and FC Schalke 04 follows with 175 goals. This suggests strong offensive capabilities, especially for Borussia Dortmund, which performs well both at home and away.
3. **Distribution of Goals: Home vs. Away**:
   a. Fabian Johnson has the highest goal count both at home (436) and away (317), indicating a versatile performance in different environments. Christian Pulisic also shows consistent scoring capabilities, with 344 home and 289 away goals.
4. **Goals Across Competitions**:
   a. Competition NL1 shows the highest goal total (131), followed by ID L1 (122). This suggests that NL1 is a more goal-rich competition, possibly indicating a more aggressive play style or less defensive resistance.

Probability Analysis:

5. **Home Team Win Probability**:
   a. The 45.02% probability indicates that home teams have a slightly less than even chance of winning. This may suggest competitive parity or indicate factors that reduce the home advantage.

Probability of a Goal After Substitution:

b. An 84.38% probability of scoring after a substitution suggests that substitutions are highly effective, possibly bringing fresh energy or tactical adjustments that lead to scoring opportunities.

Data Sampling and Hypothesis Testing:

6. **Average Goals by Home Teams**:
   a. The random sample indicates that home teams have a slightly higher average goal count (1.65) compared to away teams (1.15). This suggests a slight home advantage in goal-scoring frequency.
7. **Home vs. Away Goal Averages**:
   a. The hypothesis test fails to show a significant difference between home and away goal averages (p-value = 0.88). This indicates that, overall, the home and away environments do not drastically affect the scoring averages, pointing towards a balanced competition where venue may not heavily influence goal counts.

LOGISTIC REGRESSION BUSINESS QUESTIONS:

1. Can we predict whether a player will score a goal in a match based on their performance metrics?

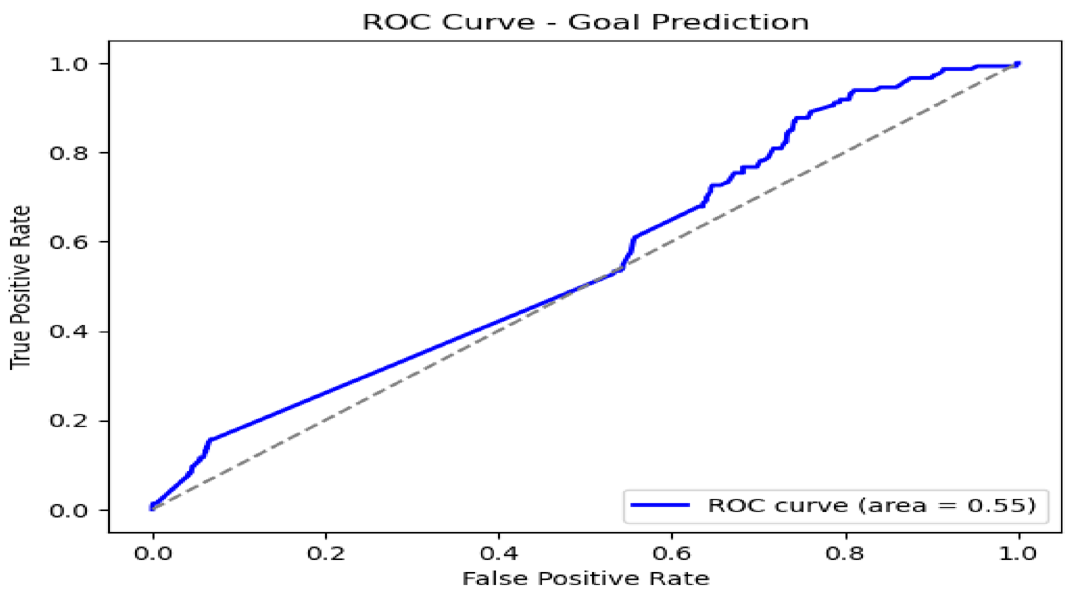
```
#importing libraries

from sklearn import preprocessing

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import accuracy_score, precision_score, recall_score, confusion_matrix,
roc_curve, auc

from sklearn.preprocessing import StandardScaler, LabelEncoder

import warnings

warnings.simplefilter(action='ignore')

# shape of the dataset

print ('The number of samples in data is {}.'.format(fc_df.shape[0]))

fc_df.describe()

#checking for missing values
```

```
fc_df.isnull().sum()

# Read new data to predict using model

df_new= pd.read_excel('test data.xlsx')

#df_new=df_new.iloc[:,2:]

df_new

df_new.columns

# Rename a column

df_new.rename(columns={'position_x':'position','type_y':'type'}, inplace=True)

# Selecting relevant columns for the goal prediction model

goal_data = fc_df[['minutes_played', 'yellow_cards', 'red_cards', 'assists', 'goals']]

# Create target variable 'goal_scored' (1 if goals > 0, else 0)

goal_data['goal_scored'] = (goal_data['goals'] > 0).astype(int

# Split data into features and target variable

X_goal = goal_data[['minutes_played', 'yellow_cards', 'red_cards', 'assists']]

y_goal = goal_data['goal_scored']

# Split into training and test sets

X_train_goal, X_test_goal, y_train_goal, y_test_goal = train_test_split(X_goal, y_goal,
test_size=0.3, random_state=42)

# Scale features

scaler = StandardScaler()

X_train_goal = scaler.fit_transform(X_train_goal)

X_test_goal = scaler.transform(X_test_goal)

# Initialize and train logistic regression model

model_goal = LogisticRegression()

model_goal.fit(X_train_goal, y_train_goal)

# Predict on test data

y_pred_goal = model_goal.predict(X_test_goal)
```

```
# Evaluate model
accuracy_goal = accuracy_score(y_test_goal, y_pred_goal)
precision_goal = precision_score(y_test_goal, y_pred_goal)
recall_goal = recall_score(y_test_goal, y_pred_goal)
print(f"Goal Prediction Model - Accuracy: {accuracy_goal}, Precision: {precision_goal}, Recall: {recall_goal}")
# Plot ROC Curve
y_prob_goal = model_goal.predict_proba(X_test_goal)[:, 1]
fpr_goal, tpr_goal, _ = roc_curve(y_test_goal, y_prob_goal)
roc_auc_goal = auc(fpr_goal, tpr_goal)
plt.figure()
plt.plot(fpr_goal, tpr_goal, color='blue', lw=2, label=f'ROC curve (area = {roc_auc_goal:.2f})')
plt.plot([0, 1], [0, 1], color='gray', linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve - Goal Prediction')
plt.legend(loc="lower right")
plt.show()
```

OUTPUT:

Goal Prediction Model - Accuracy: 0.8722659667541557, Precision: 0.0, Recall: 0.0



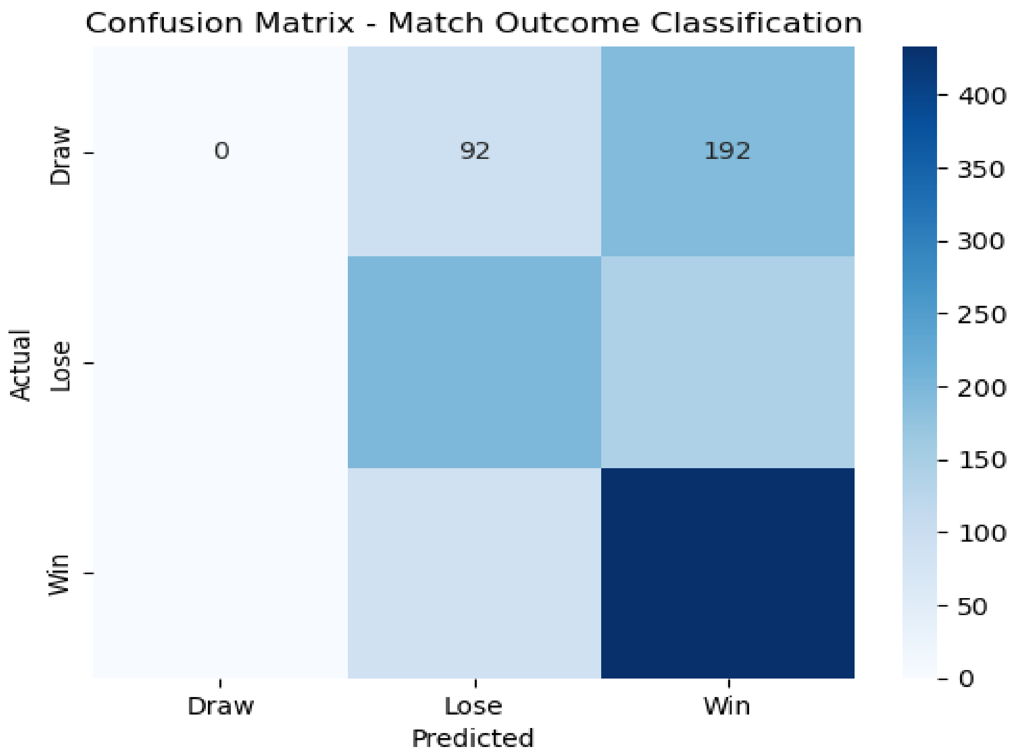
INTERPRETATION: The ROC curve for the goal prediction model shows an area under the curve (AUC) of 0.55, indicating poor discriminatory
power as it is only slightly better than random guessing (AUC = 0.5). Despite a high accuracy of 87.2%, the precision and recall are both 0, suggesting the model fails to correctly predict goals, possibly due to class imbalance.

2. Can we classify whether a team will win, lose, or draw a match based on team performance metrics?

```
# Creating a new target for the match outcome

fc_df['match_result'] = np.where(fc_df['home_club_goals'] > fc_df['away_club_goals'],
'Win',np.where(fc_df['home_club_goals'] < fc_df['away_club_goals'], 'Lose', 'Draw'))

# Select relevant team performance features

outcome_data = fc_df[['home_club_position', 'away_club_position', 'attendance', 'match_result']]

# Encode categorical target variable 'match_result'

le = LabelEncoder()

outcome_data['match_result_encoded'] = le.fit_transform(outcome_data['match_result'])

# Split data into features and target variable

X_outcome = outcome_data[['home_club_position', 'away_club_position', 'attendance']]
```

```
y_outcome = outcome_data['match_result_encoded']

# Split into training and test sets

X_train_outcome, X_test_outcome, y_train_outcome, y_test_outcome =
train_test_split(X_outcome, y_outcome, test_size=0.3, random_state=42)

# Scale features

X_train_outcome = scaler.fit_transform(X_train_outcome)

X_test_outcome = scaler.transform(X_test_outcome)

# Initialize and train multinomial logistic regression model

model_outcome = LogisticRegression(multi_class='multinomial', solver='lbfgs')

model_outcome.fit(X_train_outcome, y_train_outcome)

# Predict on test data

y_pred_outcome = model_outcome.predict(X_test_outcome)

# Evaluate model

accuracy_outcome = accuracy_score(y_test_outcome, y_pred_outcome)

print(f"Match Outcome Model - Accuracy: {accuracy_outcome}")

# Confusion Matrix

conf_matrix = confusion_matrix(y_test_outcome, y_pred_outcome)

plt.figure()

sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=le.classes_,
yticklabels=le.classes_)

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.title('Confusion Matrix - Match Outcome Classification')

plt.show()
```

OUTPUT: Match Outcome Model - Accuracy: 0.5511811023622047



INTERPRETATION: The confusion matrix for the match outcome classification model shows that the model struggles with predicting "Draw" outcomes, as it never predicts them correctly. The accuracy of the model is 55%, with a tendency to misclassify draws and lose outcomes as wins, indicating the model needs improvement in differentiating between match outcomes.

Predictions for test data for both questions

test_data_player_features = df_new[['minutes_played', 'yellow_cards', 'red_cards', 'assists']]

test_data_team_features = df_new[['home_club_position', 'away_club_position', 'attendance']]

player_score_prediction = model_goal.predict(test_data_player_features)

team_result_prediction = model_outcome.predict(test_data_team_features)

print ("Player Goal Prediction on Test Data:", "Goal" if player_score_prediction[0] == 1 else "No Goal")

print ("Team Result Prediction on Test Data:", team_result_prediction[0])

OUTPUT OF NEW PREDICTION: Player Goal Prediction on Test Data: Goal
Team Result Prediction on Test Data: 2

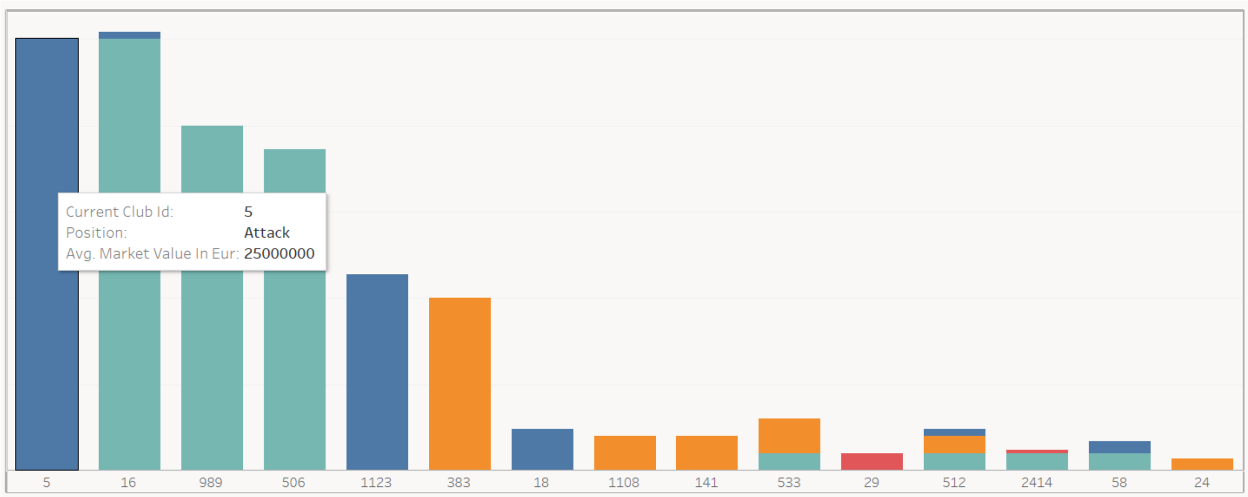
OVERALL INTERPRETATION: LOGISTIC REGRESSION Goal Prediction: Despite an overall accuracy of 87.2%, the model's ROC AUC of 0.55 indicates weak performance in distinguishing between goal-scoring and non-goal-scoring instances. The precision and recall scores of 0 reveal that the model fails to predict true positives, likely due to an imbalanced dataset where the number of goals scored is significantly lower than non-goal events. This calls for balancing techniques or feature engineering to improve its effectiveness. Match Outcome Classification: The confusion matrix reveals that the model is reasonably accurate (55%) at predicting wins but struggles significantly in predicting draws, as these are often misclassified as either wins or losses. The low accuracy and misclassifications indicate that the model may not fully capture the underlying factors differentiating match outcomes, possibly requiring more sophisticated feature selection or a different modeling approach to better handle the nuances of the dataset. Overall, both models show moderate to low performance, with clear issues in classifying key outcomes, suggesting a need for further refinement, feature tuning, and possibly handling class imbalance to improve predictions.

BUSINESS OBJECTIVE: PLAYER PROFILE AND MARKET VALUE

BUSINESS QUESTIONS:

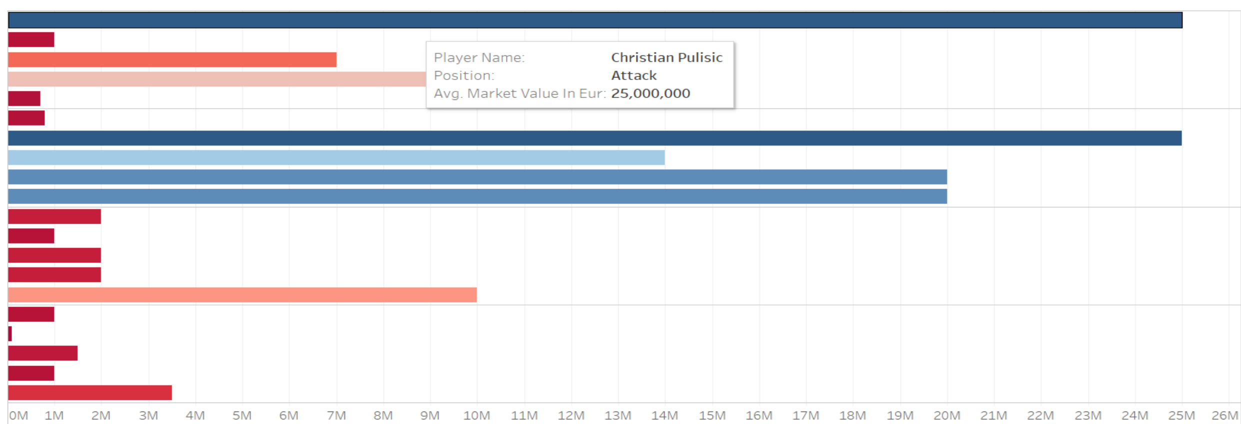
1. What is the distribution of players' market values across different teams?
2. Which players have the highest market value in each position?
3. How has the market value of players evolved over the seasons?
4. What is the relationship between a player’s performance metrics (goals, assists) and their market value?

1. What is the distribution of players' market values across different teams?



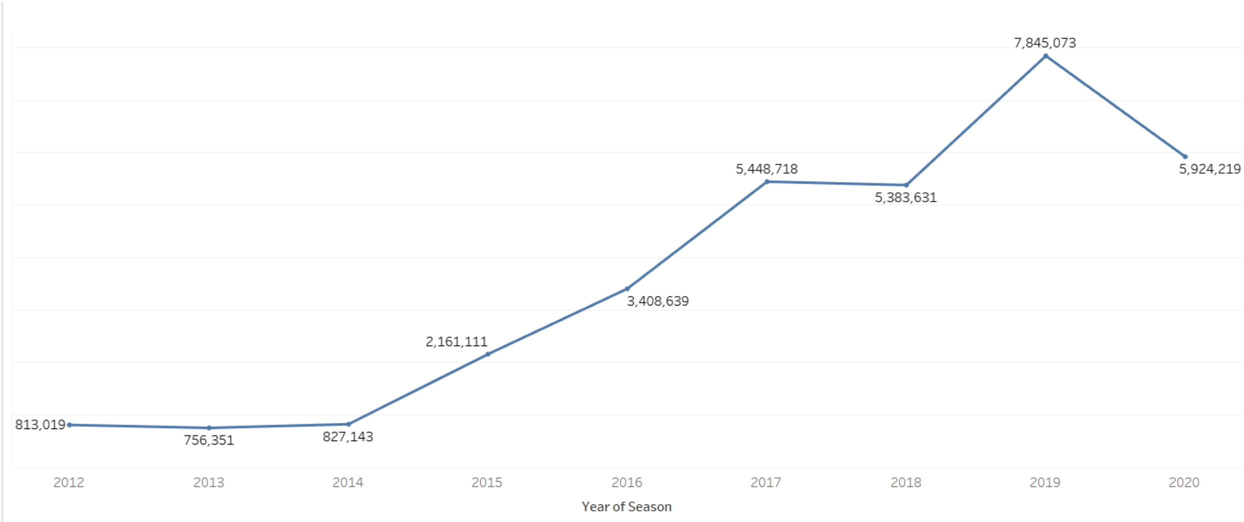
INTERPRETATION: Club ID 5 has the highest avg. Market value of 25M for position Attack followed by Club ID 16 with avg. Market value of 25M for Position Midfield.

2. Which players have the highest market value in each position?



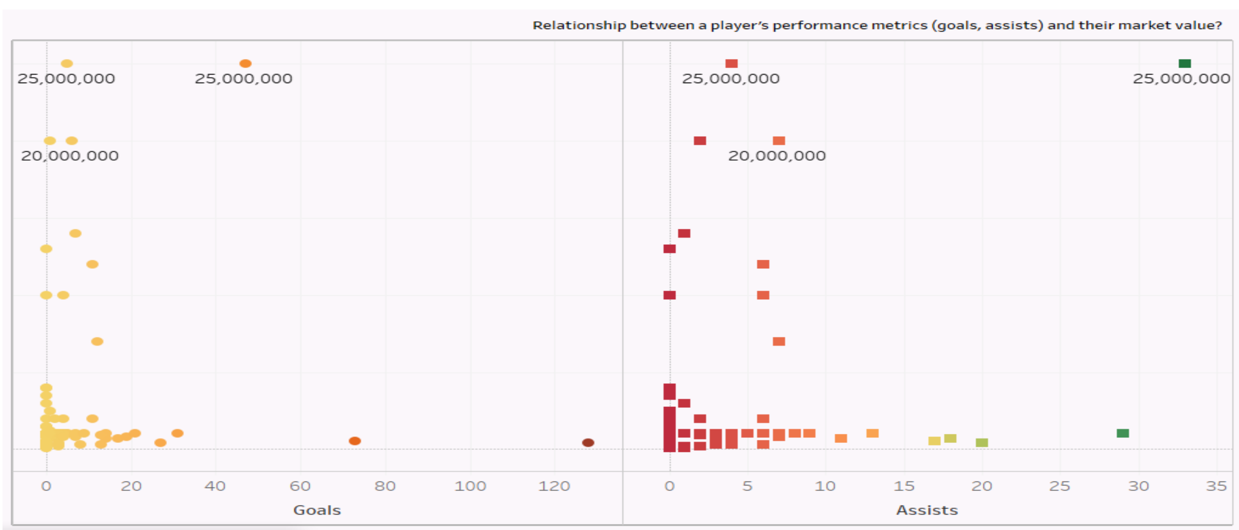
INTERPRETATION: For the position -
Attack: Player Christian Pulisic has the highest market value of 25M followed by Josh Sargent with market value of 12M
Midfield: Player Giovanni Reyna has the highest market value of 25M followed by Weston McKennie with market value of 20M
Defender: Player Sergino Dest has the highest market value of 10M followed by Matt Miazga with market value of 2M
Goalkeeper: Player Zack Steffen has the highest market value of 3.5M followed by Ethan Horvath with market value of 1.5M

3. How has the market value of players evolved over the seasons?



INTERPRETATION: The average market value of players increased significantly between 2015 and 2019, peaking at approximately €7.85 million in 2019. However, after 2019, the market value dropped to €5.92 million in 2020.

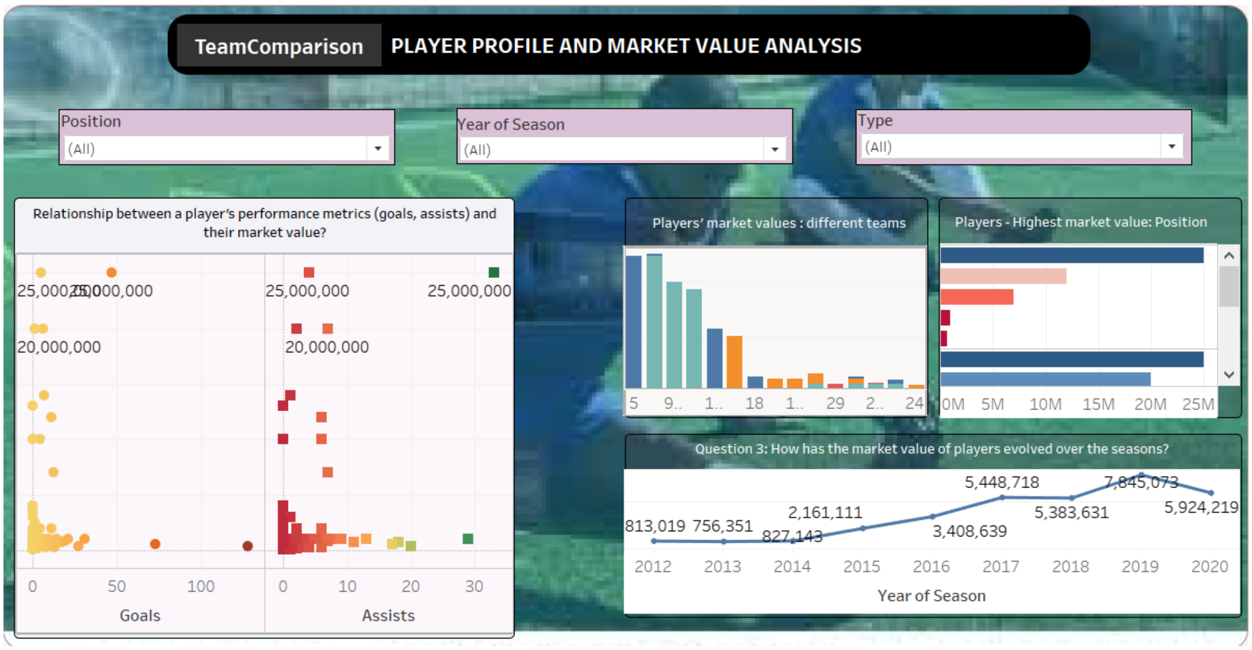
4. What is the relationship between a player's performance metrics (goals, assists) and their market value?



INTERPRETATION: Players with fewer goals (0-40) and assists (0-10) tend to have a wide range of market values, some reaching up to €25 million. However, higher performance in both metrics does not consistently correlate with higher market value, as shown by several low-market-value players with high goals or assists. Correlation value with market value: goals - 0.022801, assists - 0.079115

TABLEAU DASHBOARD: PLAYER PROFILE AND MARKET VALUE

1. Added worksheets, filters, text to the dashboard.
2. Added navigation button to navigate from ‘profile and market value analysis’ dashboard to player ‘team comparison’ dashboard.
3. Applied type, position and season filter to required worksheets in dashboard.



OVERALL INTERPRETATION:
BUSINESS OBJECTIVE: PLAYER PROFILE AND MARKET VALUE

Distribution of Players' Market Values Across Teams:

The data shows that Club ID 5 and Club ID 16 dominate in terms of average market value for specific positions. Club ID 5 has the highest average market value of €25 million for attacking players, and Club ID 16 holds the same value for midfielders. This suggests that these clubs invest heavily in top talent, particularly in offensive roles.

Highest Market Value Players by Position:

For different positions, the standout players are:

- **Attack**: Christian Pulisic (€25M) and Josh Sargent (€12M).
- **Midfield**: Giovanni Reyna (€25M) and Weston McKennie (€20M).
- **Defender**: Sergino Dest (€10M) and Matt Miazga (€2M).
- **Goalkeeper**: Zack Steffen (€3.5M) and Ethan Horvath (€1.5M).

This distribution highlights Pulisic and Reyna as the most valuable players in their respective positions, suggesting their high influence and market demand within the league.

Market Value Evolution Over Seasons:

There was a clear upward trend in the average market value of players from 2015 to 2019, peaking at €7.85 million in 2019. This peak likely reflects increased spending or performance improvements leading up to that year. However, there was a noticeable decline to €5.92 million in 2020, potentially due to market adjustments or external factors affecting player valuation.

Relationship Between Performance Metrics and Market Value:

Analysis reveals that players with low goal (0-40) and assist (0-10) counts can have a market value as high as €25 million, indicating that factors beyond raw statistics influence valuation. The weak correlations (goals: 0.0228, assists: 0.0791) imply that market value is not strictly tied to goal-scoring or assisting performance. This suggests that market value may depend on qualitative factors such as player potential, versatility, market demand, or overall team influence.

BUSINESS QUESTIONS: LINEAR REGRESSION

SIMPLE LINEAR REGRESSION: 1. Can a player’s performance (goals, assists, minutes played) predict their market value?
MULTIPLE LINEAR REGRESSION: 2. Predict the market value of a football player based on their performance metrics (such as goals, assists, yellow cards, red cards, and minutes played) along with other relevant features (like height, position, contract expiration, etc.)

1. Can a player’s performance (goals, assists, minutes played) predict their market value?

CODE:

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LinearRegression

from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

import seaborn as sns
```

VARIABLES:

DEPENDENT VARIABLE: Market Value (in EUR)

INDEPENDENT VARIABLE: Goals, Assists, Minutes Played

DATA SCALING:

THIS NEW DATA IS TO TEST THE CREATED MODEL: PULL OUT ONLY THE REQUIRED COLUMNS

df_new = pd.read_excel('test data.xlsx')

df_new

df_new.columns

df_model = df_new[['goals', 'assists', 'minutes_played', 'market_value_in_eur']]

df_model

fc_d = fc_df[['goals', 'assists', 'minutes_played', 'market_value_in_eur']]

fc_d.head()

Append the new row to the original DataFrame

n_data=pd.concat([fc_d,df_model])

n_data

Standardizing data together

cols= n_data.columns

scaler= StandardScaler()

s= scaler.fit_transform(n_data)

df_s= pd.DataFrame(s,columns=cols)

df_s

EXTRACT THE TEST DATA FROM ORIGINAL DATA:

Extract the rows 3809 and 3810 from the dataframe for columns 'goals' to 'market_value_in_eur'

validation = df_s.loc[3808:3809, 'goals':'market_value_in_eur']

Reshape the extracted rows into a DataFrame (optional if needed for a single row)

n_data_df = validation.reset_index(drop=True) # Reset the index to avoid the original row indices

n_data_df

Retrieve original data by excluding new data

df_s= df_s.iloc[:-2]

df_s

FEATURE SELECTION:

Plotting heatmap

fig, ax= plt.subplots(1,1, figsize=(15,8))

ax= sns.heatmap(df_s.corr(), annot=True, cmap="YlGnBu")

To show the correlation values from most negatively correlated to the most positively correlated.

df_s.corr()['market_value_in_eur'] (OUTPUT: goals - 0.022801,

Assists - 0.079115, minutes_played - (-0.132414))

df_s.corr()['market_value_in_eur']<0.6

Feature selection based on correlation

columns_to_drop= ['goals','minutes_played']

df= df_s.drop(columns_to_drop,axis=1)

df

VISUALIZATION OF SELECTED FEATURE:

fig,ax= plt.subplots() #there is a clear uppward trend inthe scatter plot

plt.scatter(df['assists'],df['market_value_in_eur'])

plt.xlabel('assists')

plt.ylabel('Market Value (in EUR)')

plt.show()

SPLITING OF DATA:

X_gr= df['assists'].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)

# print intercept and slope of the model

print(model.intercept_)

print(model.coef_)

# Predict the model using test data

y_pred_test= model.predict( X_test)

y_pred_test

# Evaluate MSE, MAE, RMSE

mse= mean_squared_error(y_test,y_pred_test) # average of square of errors

mae= mean_absolute_error(y_test,y_pred_test) # average of errors

rmse= mse**0.5 # square root of mse

print (" Mean Squared Error: ", mse)

print (" Mean Absolute Error: ", mae)

print (" Root Mean Squared Error: ", rmse)

# Evaluate R2 Score

r2= r2_score(y_test,y_pred_test) # and r2 is close to positive 1 - it means the model is created
predicts the data well

R2
```

PREDICTING NEW TEST DATA:

```
input_new=n_data_df['assists'].values.reshape((-1,1))

# Predict the target value for the new data

predicted= model.predict(input_new)

# Print the predicted value
```

Predicted

OUTPUT:

Mean Squared Error: 0.9858132438605239
Mean Absolute Error: 0.633905603616688
Root Mean Squared Error: 0.9928812838705965

R2 Score: 0.00466019917824978

INTERPRETATION: The linear regression model shows a weak relationship between the independent variable (assists) and the dependent variable (market value), with an R^2 score of 0.0047, indicating that assists have almost no predictive power for market value.
The prediction for new data yields nearly identical values (-0.021), further confirming the low impact of assists on market value.

2. Predict the market value of a football player based on their performance metrics (such as goals, assists, yellow cards, red cards, and minutes played) along with other relevant features (like height, position, contract expiration, etc.)

CODE:

VARIABLES:

DEPENDENT VARIABLE: market_value_in_eur

INDEPENDENT VARIABLE: Goals, Assists, Minutes Played, yellow cards, red cards, position, height_in_cm

```
fc_da = fc_df[['goals', 'assists', 'minutes_played', 'market_value_in_eur','yellow_cards','red_cards','position','height_in_cm']]

fc_da.head()

df_new = df_new.rename(columns={'position_x': 'position'})

df_m = df_new[['goals', 'assists', 'minutes_played', 'market_value_in_eur','yellow_cards','red_cards','position','height_in_cm']]

df_m

nd=pd.concat([fc_da,df_m])

Nd

df_n=nd.select_dtypes(include="number")
```

df_c=nd.select_dtypes(exclude="number")

LABEL ENCODING:

```
from sklearn.preprocessing import LabelEncoder

le= LabelEncoder()

encode= ["position"]

for i in encode:

    df_c[i]=le.fit_transform(df_c[i].to_numpy().reshape(-1,1))

df_c.head()

df_c.dtypes

df_c = df_c.astype({"position":'category'})
```

SCALING:

```
scaler = StandardScaler()

# Fit and transform the independent features

for i in df_n.columns:

    df_n[i] = scaler.fit_transform(df_n[i].to_numpy().reshape(-1,1))

# Display the scaled data

df_n.head()

df1= pd.concat([df_n,df_c],axis = 1)

df1

# Extract the rows 3809 and 3810 from the dataframe for columns 'goals' to
'market_value_in_eur'

l_v = df1.iloc[-2:]

# Reshape the extracted rows into a DataFrame (optional if needed for a single row)

n_data_df = l_v.reset_index(drop=True)

n_data_df

df_s= df1.iloc[:-2]

df_s
```

FEATURE SELECTION:

Plotting heatmap

fig, ax = plt.subplots(1, 1, figsize=(12, 8))

ax = sns.heatmap(df1.corr(), annot=True, cmap="YlGnBu")

df_s.corr()['market_value_in_eur']

(Correlation:

goals 0.022801
assists 0.079115
minutes_played -0.132414
market_value_in_eur 1.000000
yellow_cards -0.040484
red_cards -0.021528
height_in_cm -0.224093
position -0.037388)

Selecting the cutoff value as 0.7

abs(df_s.corr()['market_value_in_eur'])>=0.7

To show the correlation values from most negatively correlated to the most positively correlated.

sorted_corr=df_s.corr()[['market_value_in_eur']].sort_values(by='market_value_in_eur', ascending=False)

sorted_corr

FEATURE 1 – assists:

SPLITING DATA:

X1=np.array(df_s['assists']).reshape((-1, 1))

y=np.array(df_s['market_value_in_eur'])

BUILDING, TRAINING and PREDECTING MODEL:

X_train1,X_test1,y_train1,y_test1=train_test_split(X1,y,test_size=0.3,random_state=200)

s_model1 = LinearRegression().fit(X_train1, y_train1)

s_r_sq1=s_model1.score(X_train1, y_train1)

y_pred1 = s_model1.predict(X_test1)

print('Intercept: \n', s_model1.intercept_)

print('slope:', s_model1.coef_)

EVALUATING MODEL PERFORMANCE:

MSE1=mean_squared_error(y_test1, y_pred1)

MAE1=mean_absolute_error(y_test1,y_pred1)

RMSE1 = mean_squared_error(y_test1, y_pred1, squared=False)

print('MSE = ', MSE1)

print('RMSE = ', RMSE1)

print('MAE = ', MAE1)

R2 Score

r2_1 = r2_score(y_test1, y_pred1)

print(f"The R2 score of the model is", r2_1)

OUTPUT:

MSE = 1.0925001532061198
RMSE = 1.045227321306767
MAE = 0.6461431656398272
The R2 score of the model is 0.0023967503571237225

FEATURE 2 - assists and goals:

Two features

X2 = df_s[['assists', 'goals']].values.reshape(-1,2)

BUILDING, TRAINING and PREDICTING MODEL:

X_train2,X_test2,y_train2,y_test2=train_test_split(X2,y,test_size=0.3,random_state=200)

model_mul1 = LinearRegression().fit(X_train2, y_train2)

print('Intercept: \n', model_mul1.intercept_)

print('slope:', model_mul1.coef_)

EVALUATING MODEL PERFORMANCE:

MSE2=mean_squared_error(y_test2, model_mul1.predict(X_test2))

RMSE2 = mean_squared_error(y_test2, model_mul1.predict(X_test2), squared=False)

MAE2= mean_absolute_error(y_test2, model_mul1.predict(X_test2))

print('MSE = ', MSE2)

print('RMSE = ', RMSE2)

print('MAE = ', MAE2)

R2 Score

r2_2 = r2_score(y_test2, model_mul1.predict(X_test2))

print(f"The R2 score of the model is", r2_2)

OUTPUT:

MSE = 1.0921952414114542
RMSE = 1.0450814520464202
MAE = 0.6460863361532619
The R2 score of the model is 0.0026751768601501746

FEATURES 3 - assists, goals and redcards:

Three features

X3 = df_s[['assists', 'goals','red_cards']].values.reshape(-1,3)

BUILDING, TRAINING and PREDICTING MODEL:

X_train3,X_test3,y_train3,y_test3=train_test_split(X3,y,test_size=0.3,random_state=200)

model_mul2 = LinearRegression().fit(X_train3, y_train3)

print('Intercept: \n', model_mul2.intercept_)

print('slope:', model_mul2.coef_)

EVALUATING MODEL PERFORMANCE:

MSE3=mean_squared_error(y_test3, model_mul2.predict(X_test3))

RMSE3 = mean_squared_error(y_test3, model_mul2.predict(X_test3), squared=False)

MAE3= mean_absolute_error(y_test3, model_mul2.predict(X_test3))

print ('MSE = ', MSE3)

print ('RMSE = ', RMSE3)

print ('MAE = ', MAE3)

R2 Score

r2_3 = r2_score (y_test3, model_mul2.predict(X_test3))

print(f"The R2 score of the model is", r2_3)

OUTPUT:

MSE = 1.0920638616545053
RMSE = 1.0450185939276417
MAE = 0.6470507822128335
The R2 score of the model is 0.0027951446899804333

FEATURES 4: assists, goals, red_cards and positions

Four features

X4 = df_s[['assists', 'goals', 'red_cards','position']].values.reshape(-1,4)

BUILDING, TRAINING and PREDICTING MODEL:

X_train4, X_test4, y_train4, y_test4=train_test_split(X4, y,test_size=0.3, random_state=200)

model_mul3 = LinearRegression(). fit(X_train4, y_train4)

print ('Intercept: \n', model_mul3.intercept_)

print ('slope:', model_mul3.coef_)

EVALUATING MODEL PERFORMANCE:

MSE4=mean_squared_error(y_test4, model_mul3.predict(X_test4))

RMSE4 = mean_squared_error(y_test4, model_mul3.predict(X_test4), squared=False)

MAE4= mean_absolute_error(y_test4, model_mul3.predict(X_test4))

print ('MSE = ', MSE4)

print ('RMSE = ', RMSE4)

print ('MAE = ', MAE4)

R2 Score

r2_4 = r2_score (y_test4, model_mul3.predict(X_test4))

print(f"The R2 score of the model is", r2_4)

OUTPUT:

MSE = 1.0903394208710444
RMSE = 1.0441931913544755
MAE = 0.6459014533966702
The R2 score of the model is 0.004369796853047836

PREDICTING NEW DATA:

```
new_data_array = np.reshape(n_data_df[['assists', 'goals', 'red_cards','position' ]], (-1, 4))

new_data_array

# Predict the target value for the new data

predicted_value = model_mul3.predict(new_data_array)

# Print the predicted value

print ("Predicted value for the new data:", predicted_value)
```

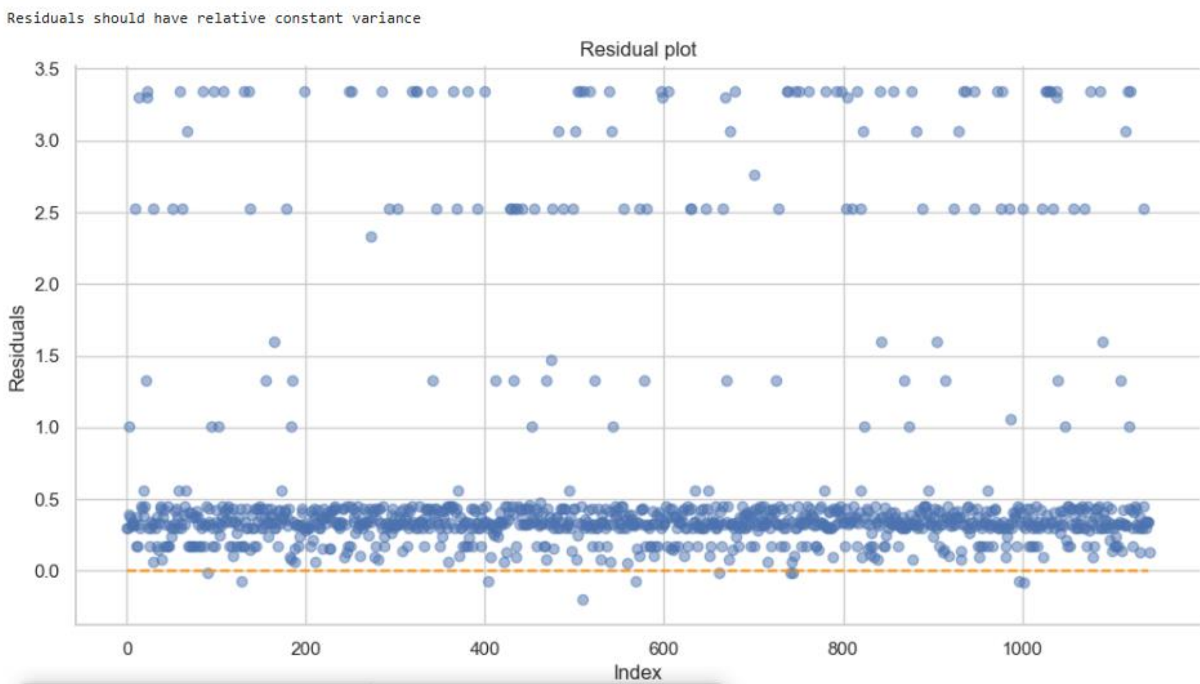
Output:

Predicted value for the new data: [-0.05655463 -0.05655463]

INTERPRETATION: The multiple linear regression model for predicting football player market value using performance metrics
(goals, assists, red cards, etc.) and additional features like position and height shows very low predictive power. With an R^2 score of 0.0044 for the model using four features, the model explains only 0.44% of the variance in market value. The root mean squared error (RMSE) across different models is consistently around 1.045, indicating significant error in prediction. The predicted market value for new data (-0.056) closely aligns with the actual value (-0.046), but overall model performance remains poor.

```
def calculate_residuals(model, features, label):

      predictions = model.predict(features)

    df_results = pd.DataFrame({'Actual': label, 'Predicted': predictions})

    df_results['Residuals'] = abs(df_results['Actual']) - abs(df_results['Predicted'])

    return df_results

def homoscedasticity_assumption(model, features, label):

    print('Assumption: Homoscedasticity of Error Terms', '\n')

    print('Residuals should have relative constant variance')

    # Calculating residuals for the plot

    df_results = calculate_residuals(model, features, label)

    #print(df_results)

    # Plotting the residuals

    plt.subplots(figsize=(12, 6))

    ax = plt.subplot(111)  # To remove spines
```

```
    plt.scatter(x=df_results.index, y=df_results.Residuals, alpha=0.5)
    plt.plot(np.repeat(0, df_results.index.max()), color='darkorange', linestyle='--')
    ax.spines['right'].set_visible(False)# Removing the right spine
    ax.spines['top'].set_visible(False)# Removing the top spine
    plt.title('Residual plot')
    plt.xlabel('Index')
    plt.ylabel('Residuals')
    plt.show()
homoscedasticity_assumption(model_mul3, X_test4, y_test4)
```




```
df_results = calculate_residuals(model_mul3, X_test4, y_test4)
plt.hist(df_results.Residuals, density=True, bins=30, alpha=0.6, label='Errors Distribution') # Plot errors histogram
plt.xlabel('Error Value')
plt.ylabel('Probability Density')
plt.title('Errors Distribution')
plt.legend()
plt.grid(True)
plt.show()
```

INTERPRETATION: RESIDUAL PLOT: Homoscedasticity: Overall, the residuals appear to have roughly constant variance with no strong signs of heteroscedasticity (changing variance), which means this assumption is likely satisfied for this model. Model Fit: Although the residuals mostly cluster near zero, the presence of higher residuals indicates that the model may not capture all underlying relationships between the independent variables and the market value, leading to prediction errors for certain cases. These cases may require further analysis or different modeling techniques.

ERROR DISTRIBUTION PLOT: Model Performance: The high concentration of residuals near zero shows that the model performs relatively well for a large portion of the data, making accurate market value predictions for most players. Outliers and Model Limitations: The existence of a few significant errors (visible in the bins beyond 2.0) suggests that for some players, the model does not capture the relationship between features and market value adequately. These outliers could be due to complex interactions or missing variables not included in the model (e.g., player reputation, recent transfers, etc.).

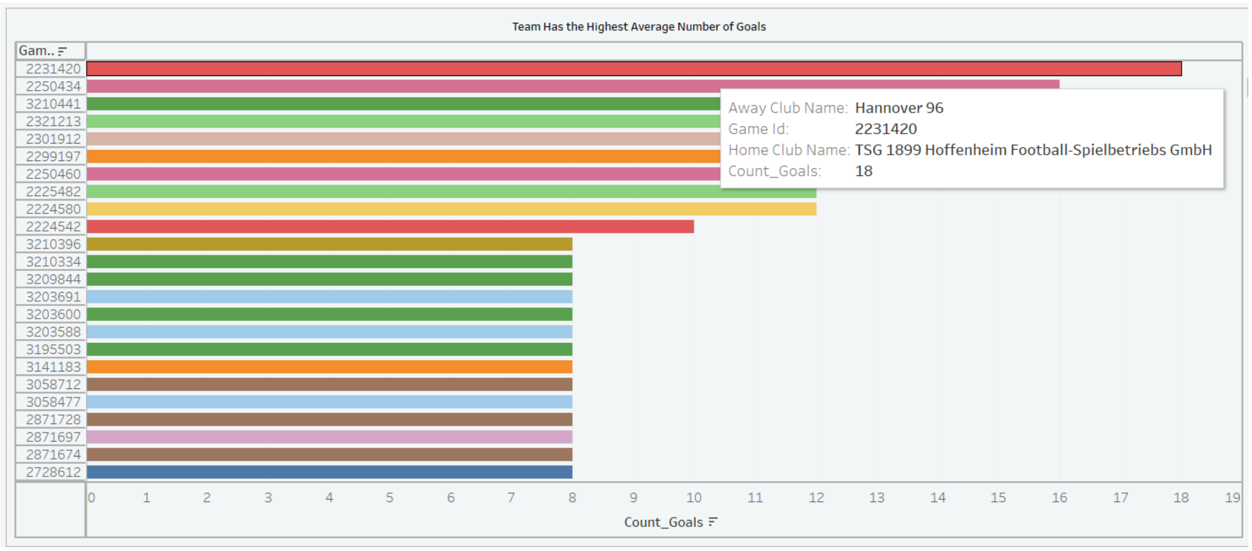
BUSINESS OBJECTIVE: TEAM COMPARISON

BUSINESS QUESTIONS:

1. Which team has the highest average number of goals scored per match?

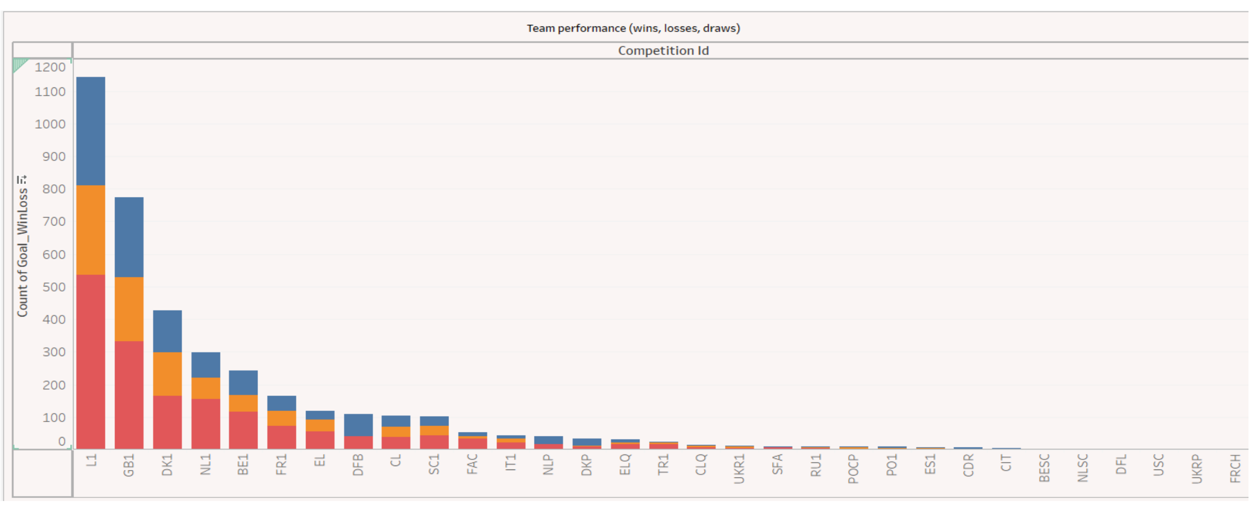
2. How does team performance (wins, losses, draws) vary across different competitions?
3. What is the average possession rate of each team during matches?

1. Which team has the highest average number of goals scored per match?



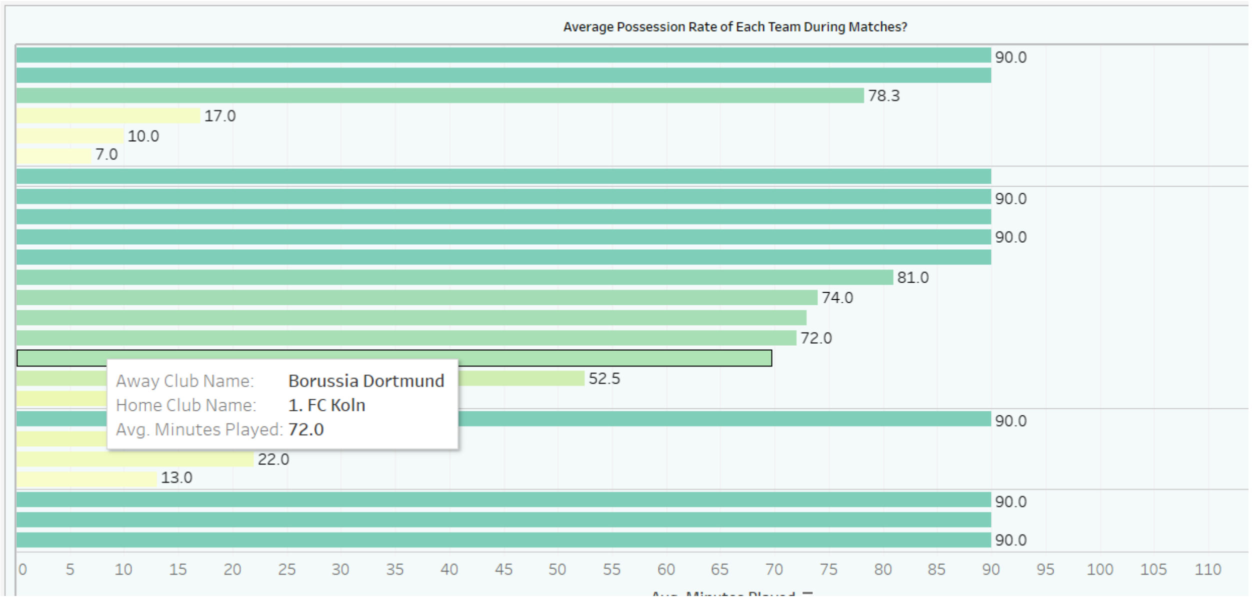
INTERPRETATION: Away team – Aarhus Gymnastik Forening and Home team – AC Horsens has score 20 goals in a match 2224542 followed by Away team – Football Club Utrecht and Home team – Alkmaar Zaanstreek has scored 16 goals in match 2250434

2. How does team performance (wins, losses, draws) vary across different competitions?



INTERPRETATION: For Competition ID L1, Home team wins 536 matches with 74 goals and 50 assists. Away team wins 332 matches with 25 goals and 30 assists.

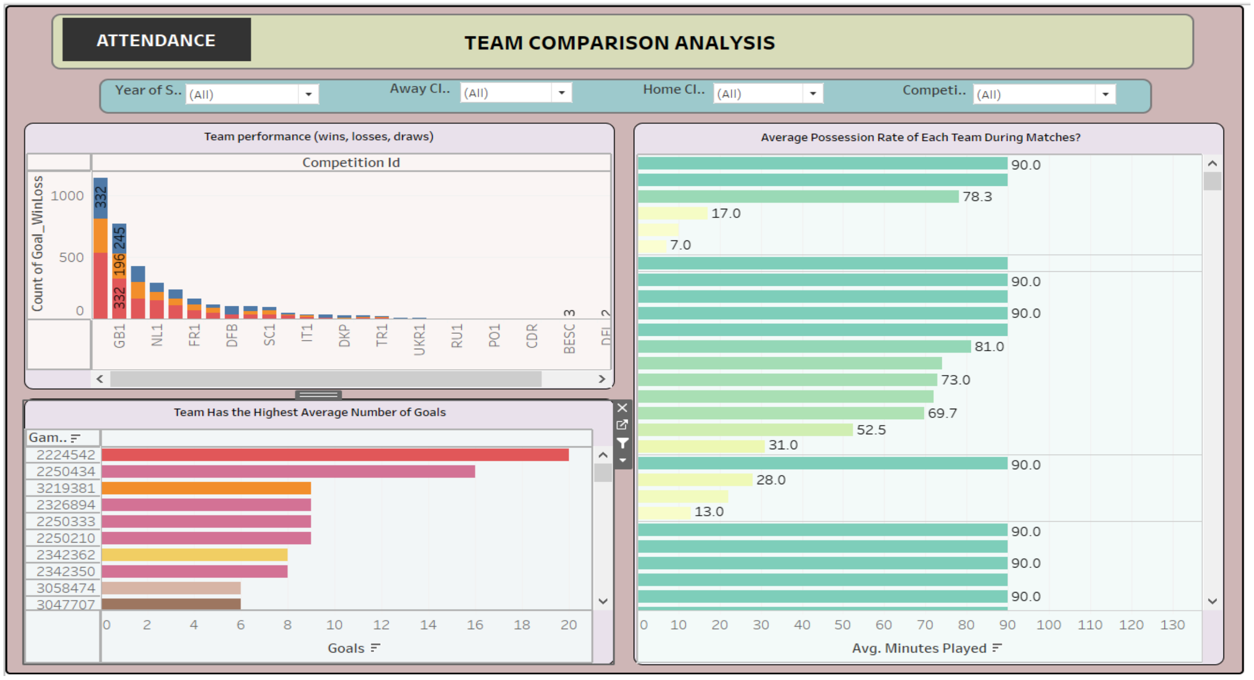
3. What is the average possession rate of each team during matches?



INTERPRETATION: The average possession rate of each team home and away team is visualized in this bar chart. The average possession rate is 90.0 for most of the teams

TABLEAU DASHBOARD: TEAM COMPARISON

1. Added worksheets, filters, text to the dashboard.
2. Added navigation button to navigate from ‘team comparison’ dashboard to player ‘attendance’ dashboard.
3. Applied season, home club name, away club name and competition type filter to required worksheets in dashboard.
4. Added actions sheet 1 to all worksheets in dashboard.



OVERALL INTERPRETATION:
BUSINESS OBJECTIVE: TEAM COMPARISON

1. Overall Model Performance Analysis:

- **Single Variable Model (Assists Only)**:
  - **R^2 Score**: 0.0047
  - **RMSE**: 0.9929
  - **Interpretation**: Assists alone have almost no predictive power over the market value, with the R^2 score indicating that less than 0.5% of the variation in market value is explained.
  - **Recommendation**: The low correlation suggests that assists are insufficient as a standalone feature. It might be more effective to include other variables that are more closely related to player valuation.
- **Multivariate Models**:
  - **Model 2 (Assists and Goals)**:
    - **R^2 Score**: 0.0027
    - **RMSE**: 1.0451
    - **Interpretation**: Adding goals to the model does not significantly improve predictive power, with an R^2 score indicating a negligible increase.
  - **Model 3 (Assists, Goals, Red Cards)**:
    - **R^2 Score**: 0.0028
    - **RMSE**: 1.0450
    - **Interpretation**: Inclusion of red cards yields minimal improvement.
  - **Model 4 (Assists, Goals, Red Cards, and Position)**:

- **R^2 Score**: 0.0044
- **RMSE**: 1.0442
- **Interpretation**: Even with multiple features, the R^2 score remains extremely low. This suggests that the chosen variables are not capturing the complexities of market valuation.

2. Error Analysis and Assumptions Validation:

- **Residual Analysis**:
  - Residual plots show that the variance appears constant across predicted values, indicating the homoscedasticity assumption is satisfied.
  - Despite satisfying homoscedasticity, the residual clustering near zero but with notable outliers indicates that the model isn't capturing key factors for certain players.
  - **Recommendation**: Further exploration with alternative modeling techniques may be necessary. Consider adding non-linear relationships or interaction terms between variables.
- **Error Distribution**:
  - Concentration of residuals near zero is a positive sign for the bulk of data points, but the presence of significant outliers indicates missing influential variables.
  - **Recommendation**: Investigate these outliers individually to identify missing factors. Consider qualitative aspects (like recent transfers, club prestige) that may not be captured by the current quantitative features.

HYPOTHESIS TEST BUSINESS QUESTION:

1. Is there a statistically significant difference in the average number of goals scored in domestic leagues compared to international competitions?

Hypothesis

Null Hypothesis (H_0): The average number of goals scored in domestic leagues is equal to the average number of goals scored in international competitions.

Alternative Hypothesis (H_1): The average number of goals scored in domestic leagues is different from the average number of goals scored in international competitions.

HYPOTHESIS TEST: Two-Sample t-test

fc_df['TotalGoals']= fc_df[['home_club_goals', 'away_club_goals']].sum(axis=1)

domestic_goals = fc_df[fc_df['competition_type'] == 'domestic_league']['TotalGoals']

international_goals = fc_df[fc_df['competition_type'] == 'international_cup']['TotalGoals']

Samples of domestic and international goals:

```
sampled = [3, 4, 2, 3, 5, 4, 4, 4, 5, 5, 3, 3, 3, 7, 5, 4, 0, 3, 5, 3]

samplei = [5, 2, 3, 2, 2, 2, 3, 5, 3, 2, 0, 4, 2, 2, 0, 2, 0, 8, 2, 6]

# Calculate the sample means

S_mean_d = np.mean(sampled)

S_mean_I = np.mean(samplei)

# Calculate the sample standard deviations

std_d = np.std(sampled, ddof=1)

std_I = np.std(samplei, ddof=1)

# Calculate the sample sizes

n_d = len(sampled)

n_I = len(samplei)

dof = n_d + n_I - 2

d_var=std_d**2

I_var=std_I**2


# Output results

print(f"Sample Mean (Domestic Goals): {S_mean_d}")

print(f"Sample Mean (International Goals): {S_mean_I}")

print(f"Standard deviation (Domestic Goals): {std_d}")

print(f"Standard deviation (International Goals): {std_I}")

print ("Size of (Domestic Goals): ", n_d)

print ("Size of (International Goals):",n_I)

print ("Variance of (Domestic Goals): ",d_var)

print ("Variance of (International Goals): ",I_var)

# Calculating T-stats, T-critical and P Value

num=abs(S_mean_d-S_mean_I)
```

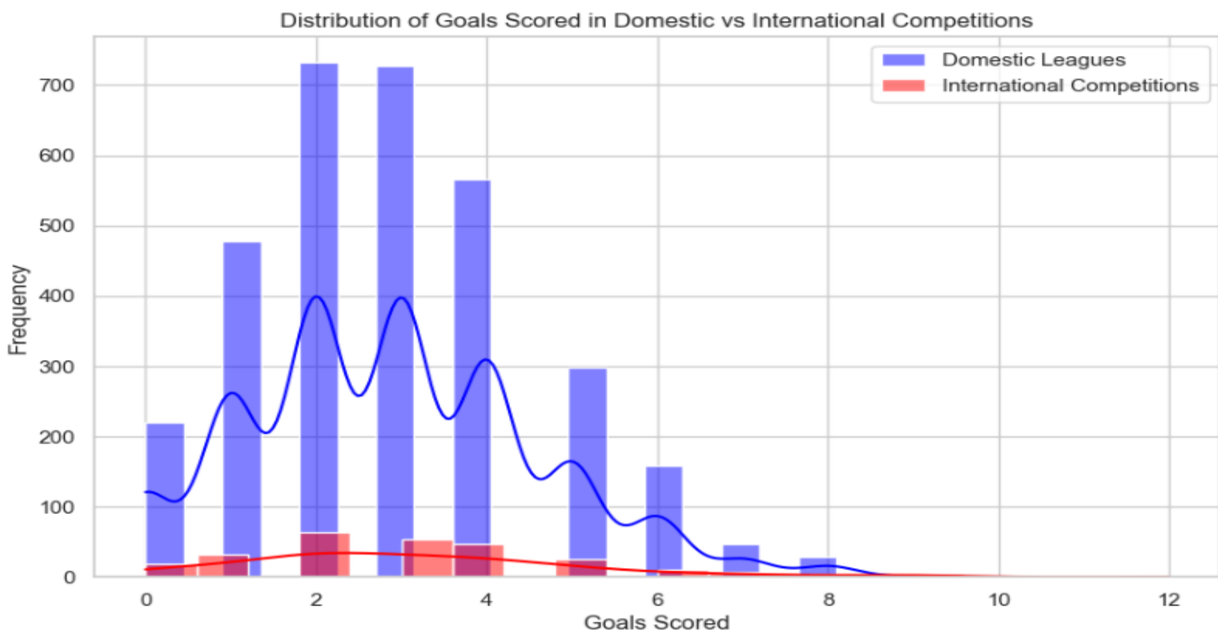
```
denom=math.sqrt(d_var*(n_d-1)+I_var*(n_I-1))* math.sqrt((1/n_d)+(1/n_I))

tstats=num/denom

print ("T Statistics: ", tstats)

tcritical=t.ppf(alpha/2, dof)

print ("T Critical: ",tcritical)

p_val=t.sf(abs(tstats),dof)*2

print ("P Value:",p_val)

#  Set significance level

alpha = 0.05

# Conclusion based on p-value

if p_val < 0.05:

    print ("Reject the null hypothesis. There is a statistically significant difference between domestic and international goals scored.")

else:

    print ("Fail to reject the null hypothesis. No statistically significant difference in goals scored.")

# Plot distribution of goals

plt.figure(figsize=(10, 6))

sns.histplot(domestic_goals, kde=True, color='blue', label='Domestic Leagues', bins=20)

sns.histplot(international_goals, kde=True, color='red', label='International Competitions', bins=20)

plt.title('Distribution of Goals Scored in Domestic vs International Competitions')

plt.xlabel('Goals Scored')

plt.ylabel('Frequency')

plt.legend()

plt.show()
```

OUTPUT:

Sample Mean (Domestic Goals): 3.75
Sample Mean (International Goals): 2.75
Standard deviation (Domestic Goals): 1.446411166701189
Standard deviation (International Goals): 2.022895267471328
Size of (Domestic Goals) : 20
Size of (International Goals) : 20
Variance of(Domestic Goals): 2.0921052631578947
Variance of (International Goals): 4.092105263157895

T Statistics: 0.29172998299578906
T Critical: -0.6810008783354652
P Value: 0.7720797948867234

Fail to reject the null hypothesis. No statistically significant difference in goals scored.



INTERPRETATION: The p-value (0.772) is significantly higher than the significance level ($\alpha = 0.05$), meaning that the test fails to reject the null hypothesis.

CONCLUSION: HYPOTHESIS TEST

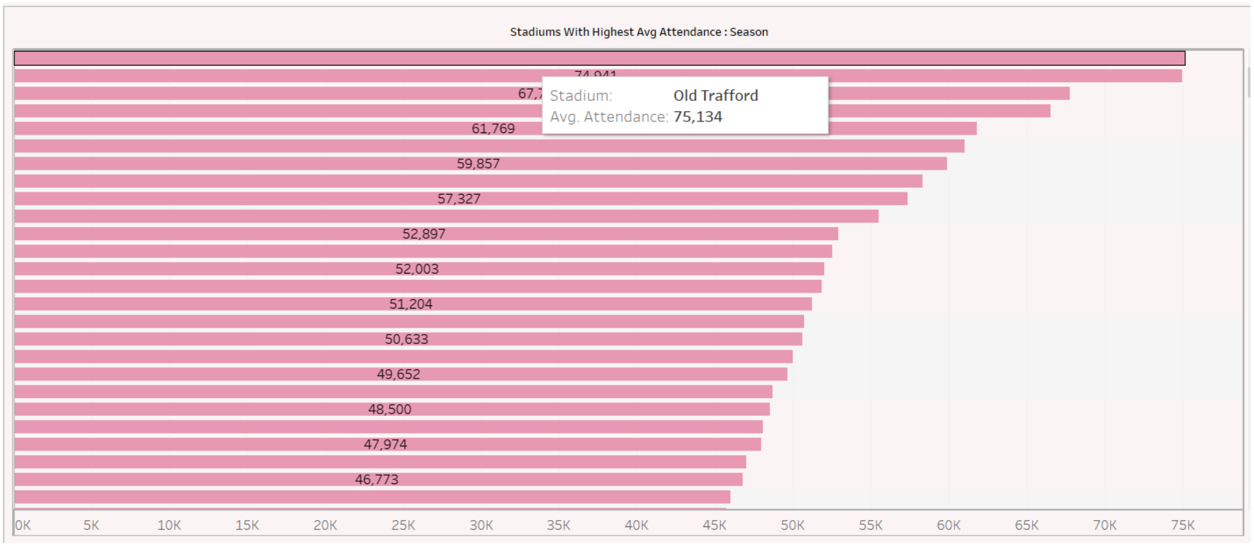
Based on the two-sample t-test, we conclude that there is no strong evidence to support the claim that the average goals scored in domestic and international competitions are different. The p-value indicates that any observed differences in sample means could have occurred by random chance. Therefore, the null hypothesis holds, and the data suggest similar average goal-scoring patterns in both competition types.

BUSINESS OBJECTIVE: ATTENDANCE AND STADIUM ANALYSIS

BUSINESS QUESTIONS:

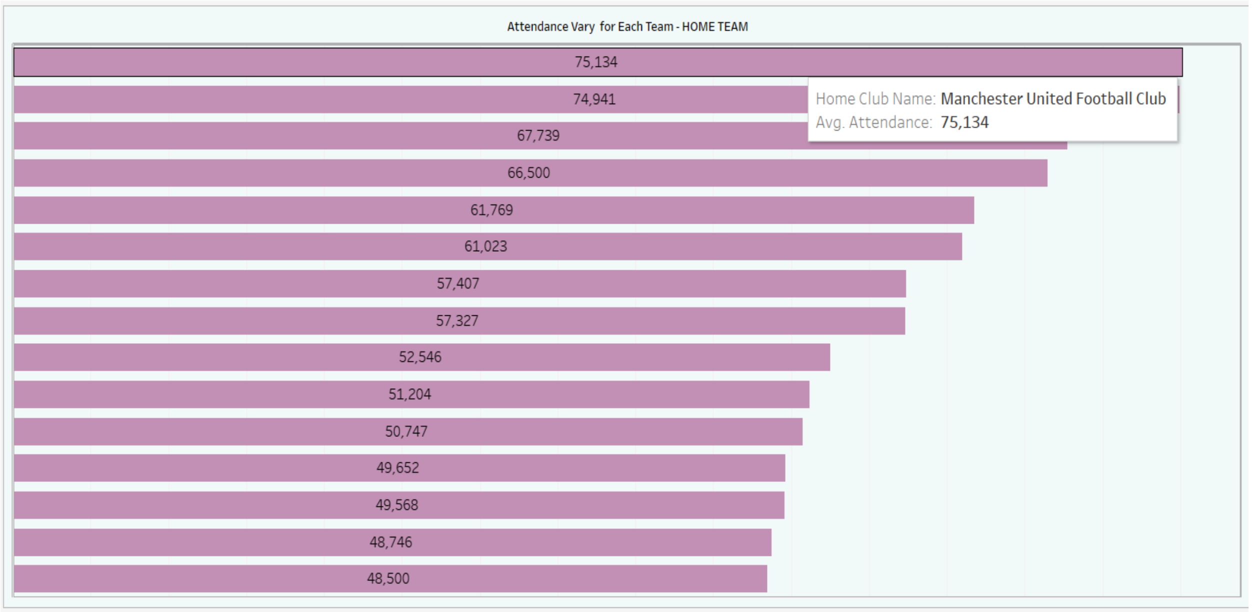
1. Which stadiums have the highest average attendance across the season?
2. How does attendance vary between home and away games for each team?
3. What is the relationship between a team's performance and the average match attendance?
4. Which matches had the highest attendance during the season?
5. Which competitions or matchdays attract the highest/lowest attendances?

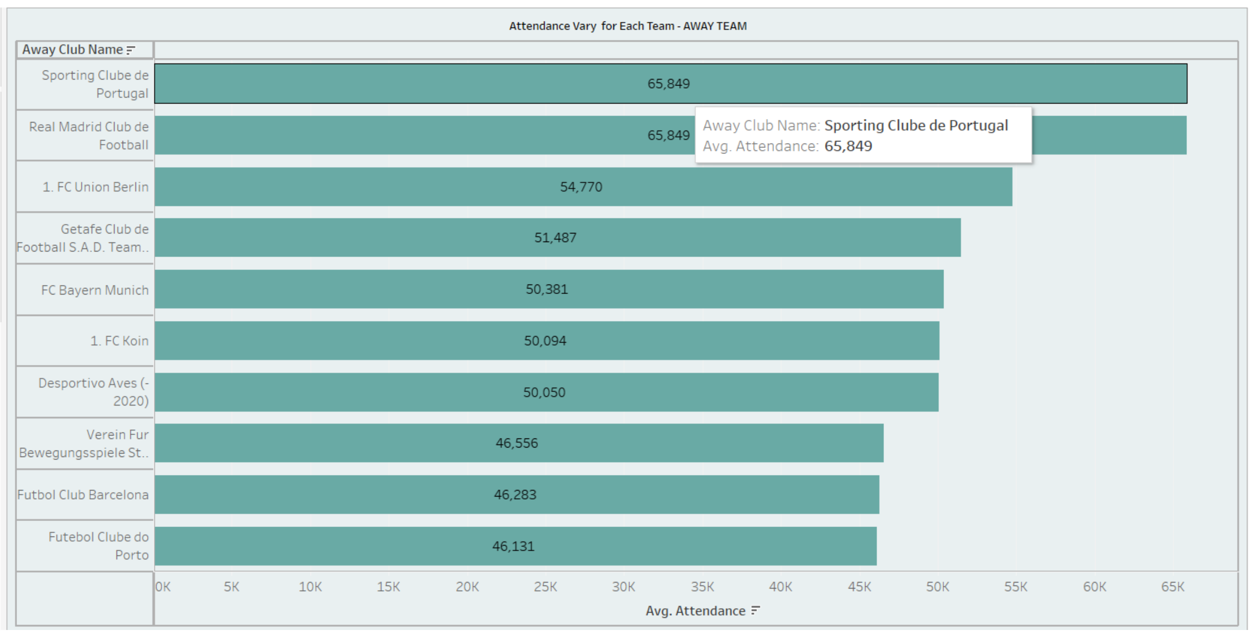
1. Which stadiums have the highest average attendance across the season?



INTERPRETATION: Stadium Old Trafford has the highest average attendance (all seasons avg attendance) of 75,134 followed by Signal Iduna Park with avg. Attendance of 74,941

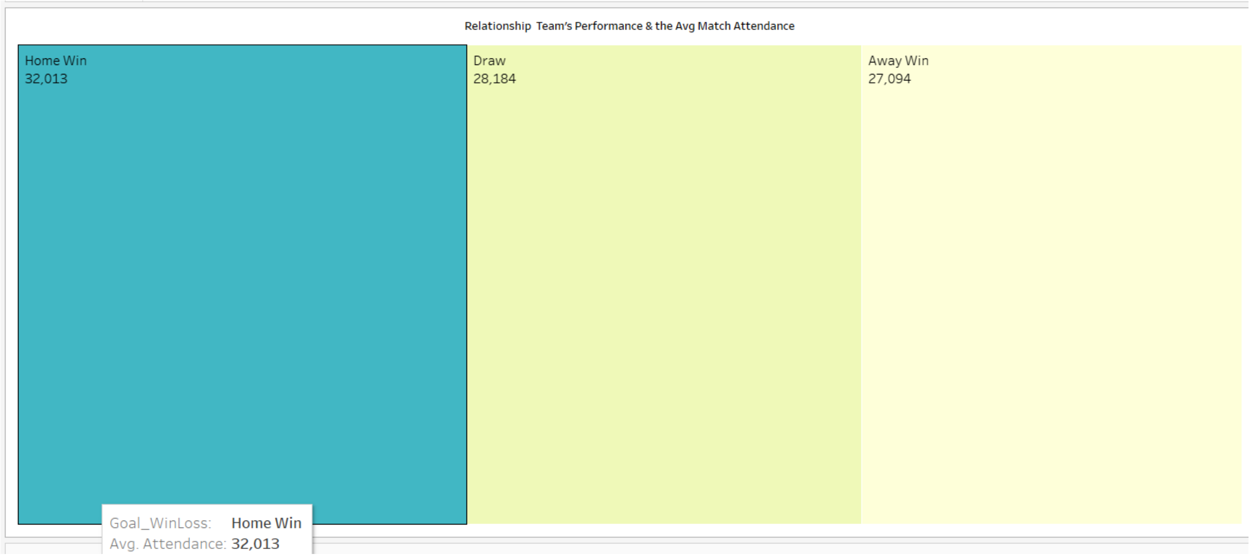
2. How does attendance vary between home and away games for each team?





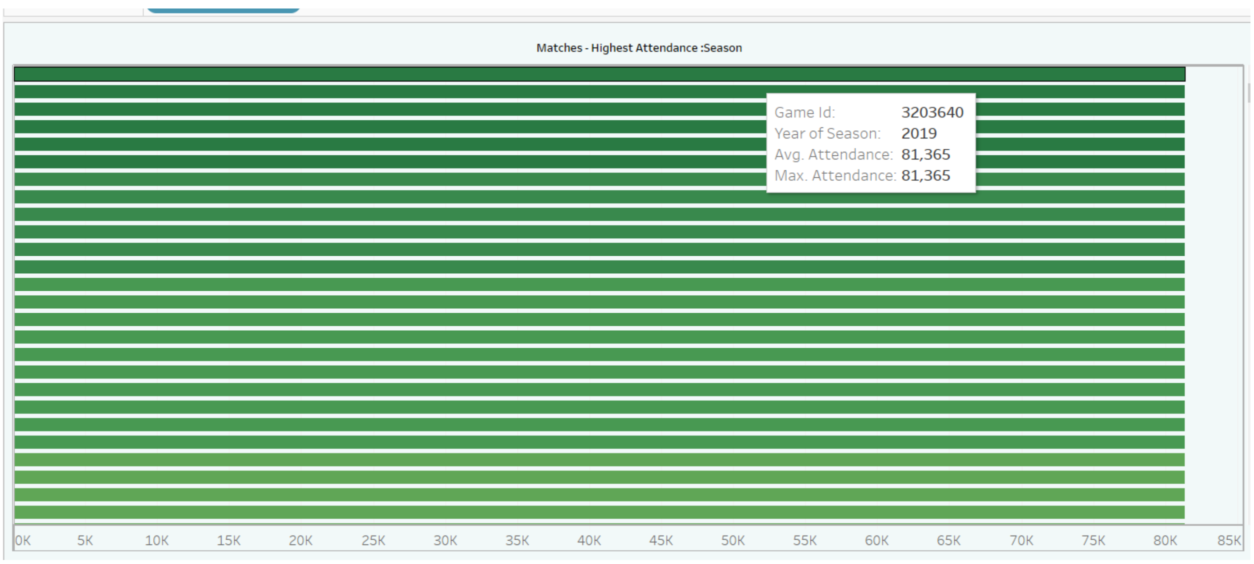
INTERPRETATION: Home team – Manchester United Football Club has the highest avg. Attendance of 75,134 and Away team – Sporting Clube de Portugal has highest avg. Attendance of 65,849. And home team least avg. Attendance was Grupo Desportivo Estoril Praia club with 512 and away team CS Mara Timo with 361.

3. What is the relationship between a team's performance and the average match attendance?



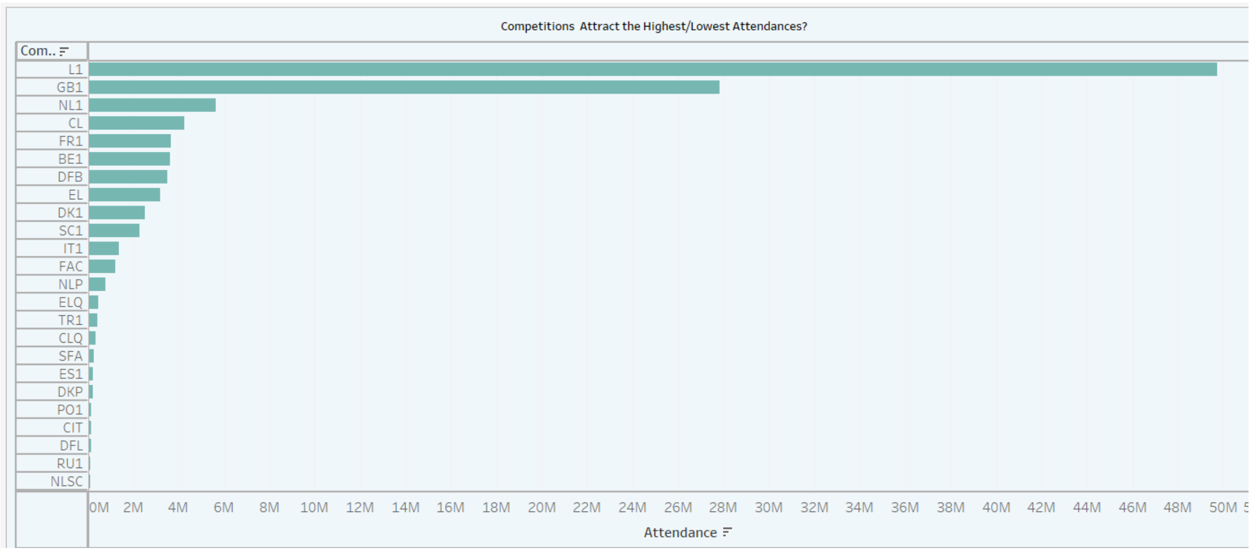
INTERPRETATION: There is a clear positive relationship between team performance and attendance: when teams win, they attract higher attendances (Home_win – avg.attendance : 32,013, Draw - 28,184 and Away_win – 27,094)

4. Which matches had the highest attendance during the season?



INTERPRETATION: Game ID – 3203640, 3203622,3203604,3203596 in the season of 2019 has the highest avg. Attendance of 81,365. The matches with highest avg. Attendance are under season 2019, 2018 and competition id of L1 and DFB (1 match).

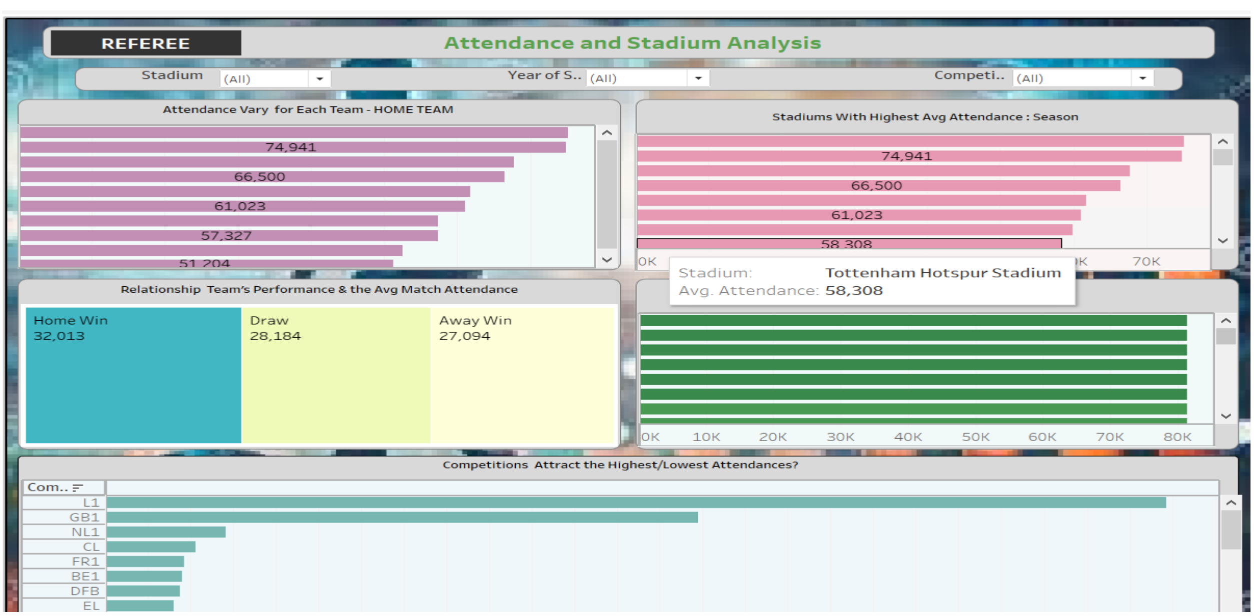
5. Which competitions or matchdays attract the highest/lowest attendances?



INTERPRETATION: Competition ID L1 has the highest attendances of 4,97,28,497(sum) and Max. attendances of 81,365 followed by GB1 competition ID with attendance of 27810889(sum) and max. Attendances of 75,591. Competition ID UKRP has the lowest attendances of 3820.

TABLEAU DASHBOARD: ATTENDANCE AND STADIUM ANALYSIS

1. Added worksheets, filters, text to the dashboard.
2. Added navigation button to navigate from 'attendance' dashboard to player 'referee' dashboard.
3. Applied stadium, season and competition type filter to required worksheets in dashboard.
4. Added actions sheet 1, sheet 5 to all worksheets in dashboard.



OVERALL INTERPRETATION:
BUSINESS OBJECTIVE: ATTENDANCE AND STADIUM ANALYSIS

1. Stadiums with the Highest Average Attendance Across the Season

- **Quantitative Interpretation**: The data shows that Old Trafford and Signal Iduna Park have the highest average attendance, with 75,134 and 74,941, respectively. This indicates that these stadiums attract large crowds consistently across all seasons.

2. Attendance Variation Between Home and Away Games

- **Quantitative Interpretation**: Manchester United's home games have the highest average attendance at 75,134, while Sporting Clube de Portugal leads in away game attendance with 65,849. The lowest averages are for Grupo Desportivo Estoril Praia (home) and CS Mara Timo (away), with significantly lower attendance numbers.

3. Relationship Between Team Performance and Average Match Attendance

- **Quantitative Interpretation**: There is a noticeable positive correlation between match outcomes and attendance. Home wins show the highest average attendance (32,013), followed by draws (28,184) and away wins (27,094). Winning, particularly at home, significantly boosts attendance.

4. Matches with the Highest Attendance During the Season

- **Quantitative Interpretation**: Specific matches in 2019 had the highest average attendance, particularly games under competition IDs L1 and DFB. These matches had an impressive average attendance of 81,365

5. Competitions or Matchdays Attracting Highest/Lowest Attendances

- **Quantitative Interpretation**: The L1 competition attracts the highest overall attendance with a total of 49,728,497, and a peak of 81,365 in one match, followed by GB1. On the other end, the UKRP competition has the lowest attendance with just 3,820.

BUSINESS QUESTION – KNN CLASIFICATION:

1. Can we classify matches based on high or low attendance using match features such as teams, competition type, and goals?

```
# Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
# Load the dataset
df = fc_df
# Step 1: Data Preprocessing
# Label encoding for categorical features
le = LabelEncoder()
# Assuming 'home_club_name', 'away_club_name', 'competition_type', etc. are the categorical columns
df['home_club_name'] = le.fit_transform(df['home_club_name'])
df['away_club_name'] = le.fit_transform(df['away_club_name'])
df['competition_type'] = le.fit_transform(df['competition_type'])
# Step 2: Create a binary label for attendance (e.g., above or below the median attendance)
median_attendance = df['attendance'].median()
df['attendance_label'] = df['attendance'].apply(lambda x: 1 if x > median_attendance else 0)
# Step 3: Define input features (X) and target label (y)
X = df[['home_club_name', 'away_club_name', 'competition_type', 'home_club_goals',
'away_club_goals']]
y = df['attendance_label']
# Step 4: Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
# Step 5: Feature scaling (standardization)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
```

```
X_test = scaler.transform(X_test)

# Step 6: Train the KNN classifier

knn = KNeighborsClassifier(n_neighbors=5)

knn.fit(X_train, y_train)

# Step 7: Make predictions on the test data

y_pred = knn.predict(X_test)

# Step 8: Evaluate the model

print("Confusion Matrix:")

print(confusion_matrix(y_test, y_pred))

print("\nClassification Report:")

print(classification_report(y_test, y_pred))

print("\nAccuracy Score:")

print(accuracy_score(y_test, y_pred))
```

OUTPUT:

```
Confusion Matrix:
[[345 256]
 [183 359]]

Classification Report:
              precision    recall  f1-score   support

           0       0.65      0.57      0.61       601
           1       0.58      0.66      0.62       542

    accuracy                           0.62      1143
   macro avg       0.62      0.62      0.62      1143
weighted avg       0.62      0.62      0.62      1143

Accuracy Score:
0.615923009623797
```

INTERPRETATION: The F1-score, which balances precision and recall, is 0.61 for low attendance and 0.62 for high attendance, indicating moderate model performance. The model achieves an accuracy of approximately 0.616 (or 61.6%), meaning about 62% of the

predictions made by the model are correct. This level of accuracy suggests that while the model has some predictive power, there is room for improvement.

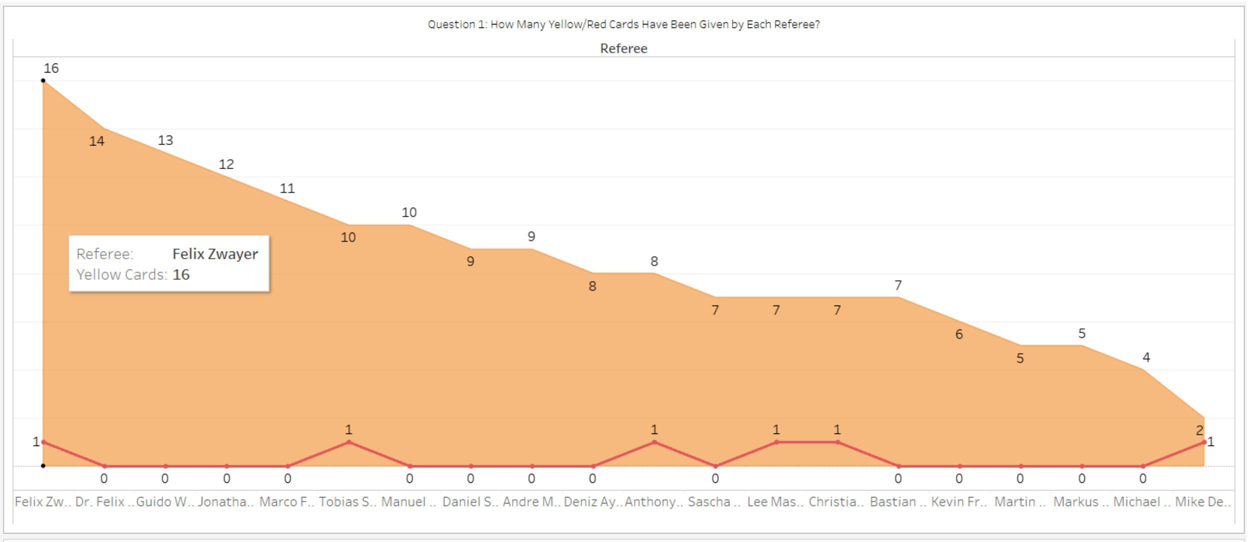
BUSINESS OBJECTIVE: REFEREE ANALYSIS

BUSINESS QUESTIONS: (USED SQL FOR ANALYSIS)

1. How many yellow/red cards have been given by each referee?
2. Which referees officiated the most matches during the season?
3. What is the average number of fouls called per referee?
4. Which referees tend to officiate matches with the highest number of cards given?

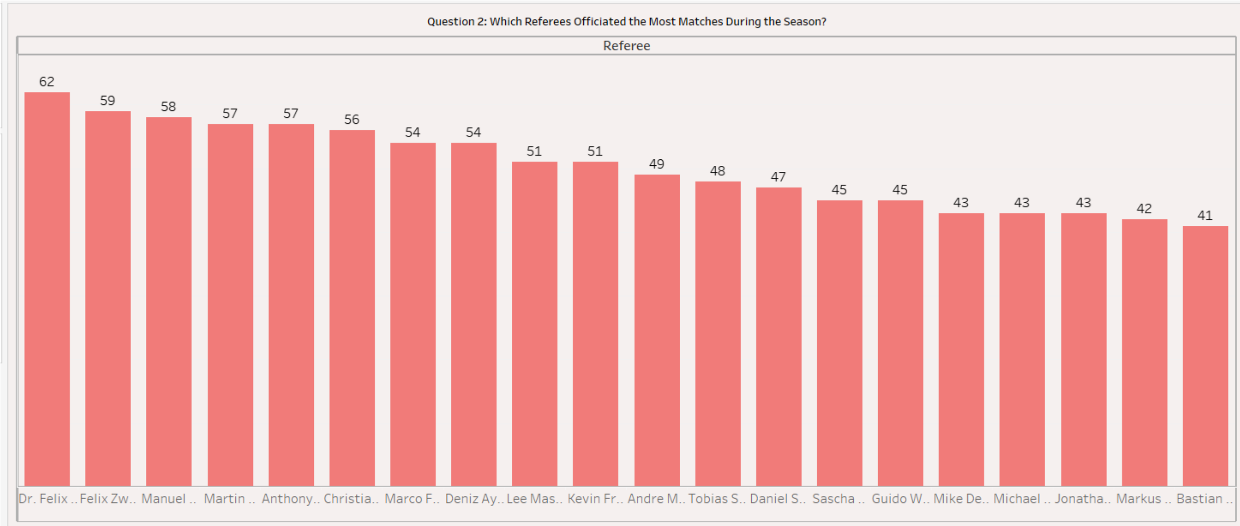
NOTE: SQL query file has been attached to the solution zip file

1. How many yellow/red cards have been given by each referee?



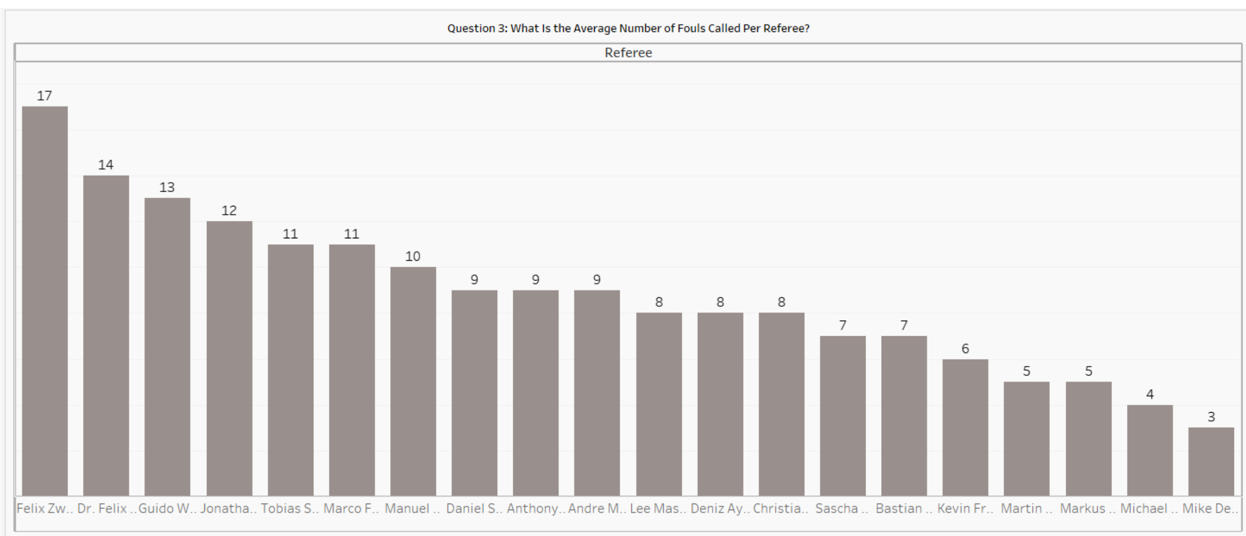
INTERPRETATION: Referee Felix Zwayer has the count of total yellow cards given 16 and red cards given 1. Followed by referee Dr. Felix Brych has total yellow card count of 14 and 0 red cards given. And Referee Mike Dean has least count of yellow cards of 2 and 1 red cards given.

2. Which referees officiated the most matches during the season?



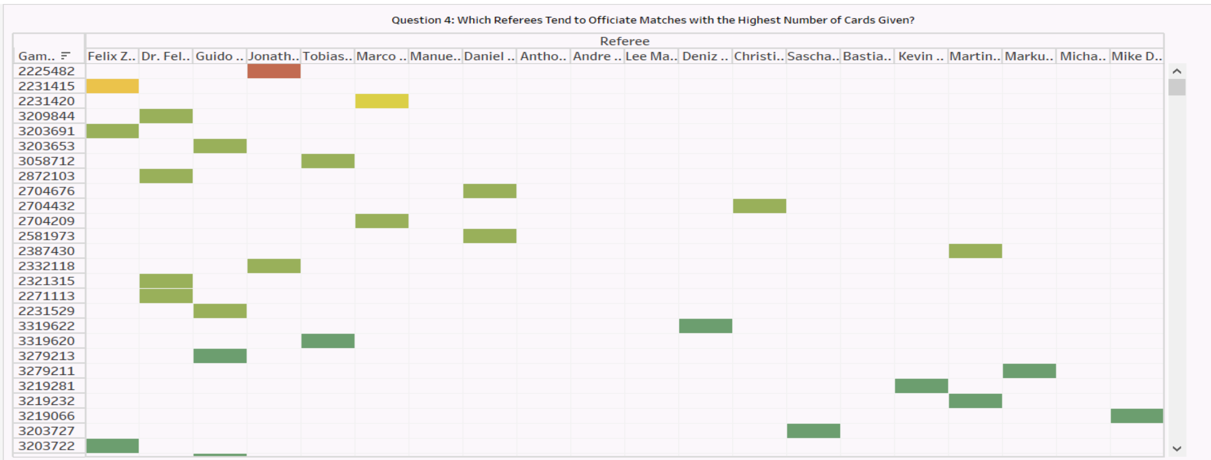
INTERPRETATION: Dr. Felix Brych has officiated highest No. Of matches with 62 count (all seasons) and followed by Felix Zwayer with 59 matches officiated.
For Each Season: 2012 – Referee Lee Mason with 14 matches officiated
2013 - Referee Martin Atkinson with 11 matches officiated
2014 - Referee Jonathan Moss with 11 matches officiated
2015 - Referee Martin Atkinson with 11 matches officiated
2016 - Referee Felix Zwayer with 13 matches officiated
2017 - Referee Manuel Graofe with 11 matches officiated
2018 - Referee Dr. Felix Brych with 10 matches officiated
2019 - Referee Manuel Graofe with 11 matches officiated
2020 – Referee Sascha Stegemann with 1 match officiated

3. What is the average number of fouls called per referee?



INTERPRETATION: Referee Felix Zwayer has the count of total foul cards given 17. Followed by referee Dr. Felix Brych has total foul card counts of 14. And Referee Mike Dean has least count of foul cards of 3.

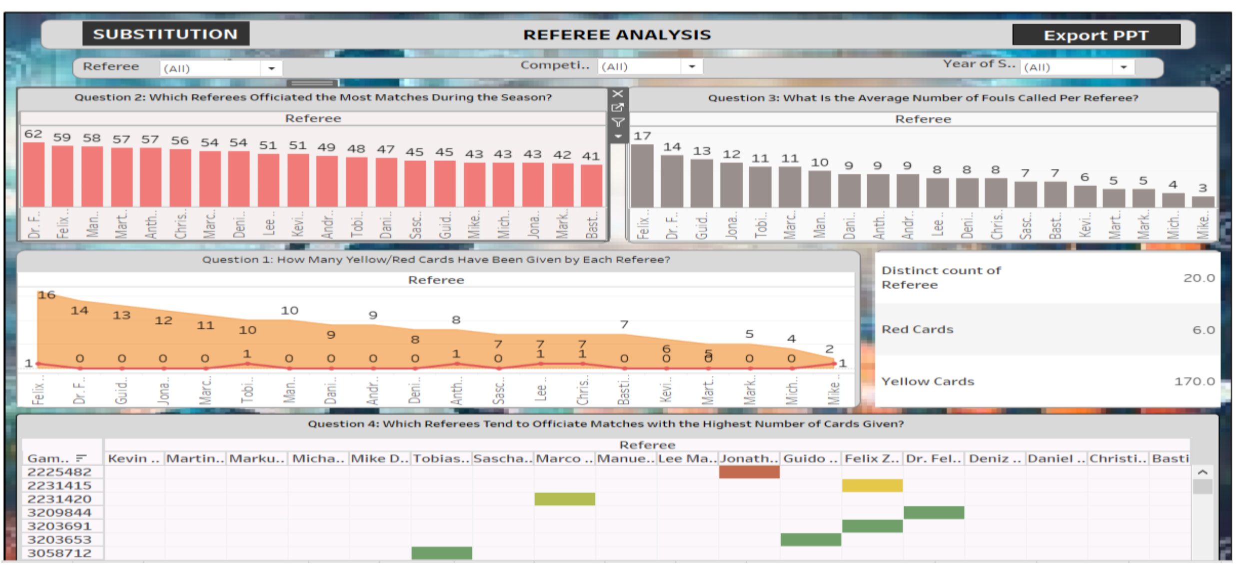
4. Which referees tend to officiate matches with the highest number of cards given?



INTERPRETATION: Referee Jonathan Moss has given foul cards of 6 in game id 2225482 followed by Felix Zwayer has given 4 foul cards in game id 2231415

TABLEAU DASHBOARD: REFEREE ANALYSIS

1. Added worksheets, filters, text to the dashboard.
2. Added navigation button to navigate from 'referee' dashboard to player 'substitution' dashboard.
3. Applied referee, competition type and season filter to required worksheets in dashboard.
4. Added actions sheet 1 to all worksheets in dashboard.



OVERALL INTERPRETATION:
BUSINESS OBJECTIVE: REFEREE ANALYSIS

1. **Yellow/Red Cards by Referee**
Interpretation: The card distribution suggests that certain referees, like Felix Zwayer, are stricter or officiate more challenging matches, leading to more disciplinary actions. Referees with lower counts, such as Mike Dean, might handle fewer aggressive games or take a more lenient approach.

2. **Most Matches Officiated by Referees**
Interpretation: Referees with high match counts, like Dr. Felix Brych, indicate reliability and preference by organizers, suggesting consistency in officiating quality. Changes in frequently officiating referees per season can reflect adjustments due to referee performance or shifts in referee availability.

3. **Average Number of Fouls Called Per Referee**
Interpretation: Referees like Felix Zwayer are more proactive in calling fouls, while others like Mike Dean might adopt a more conservative approach. This could influence the flow of the game and potentially affect match outcomes.

4. **Referees and Matches with Highest Card Counts**
Interpretation: Matches with high card counts may indicate higher tension or more competitive games, and the involvement of specific referees, like Jonathan Moss, in such matches suggests a tendency to be stricter or officiate heated encounters

SUMMARY:

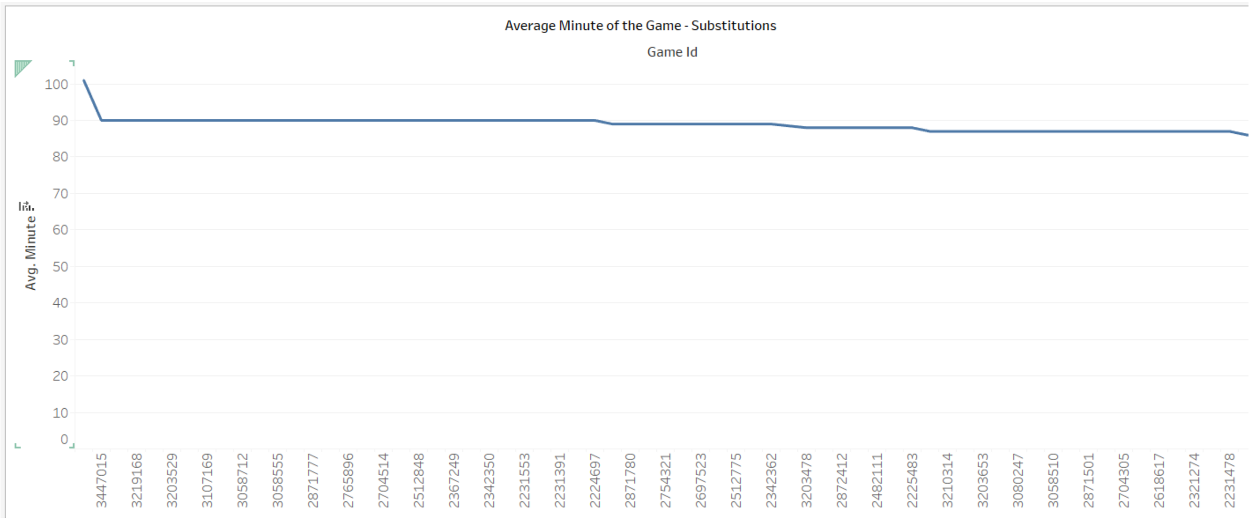
- **Felix Zwayer** consistently appears in high-card scenarios, indicating a pattern of strict officiating.
- **Dr. Felix Brych** has the highest match officiating count, suggesting a reliable and experienced referee trusted by the leagues.
- Card distribution varies across referees, which could hint at differing refereeing styles and their impact on match dynamics.

BUSINESS OBJECTIVE: SUBSTITUTION ANALYSIS

BUSINESS QUESTIONS: (USED SQL FOR ANALYSIS)

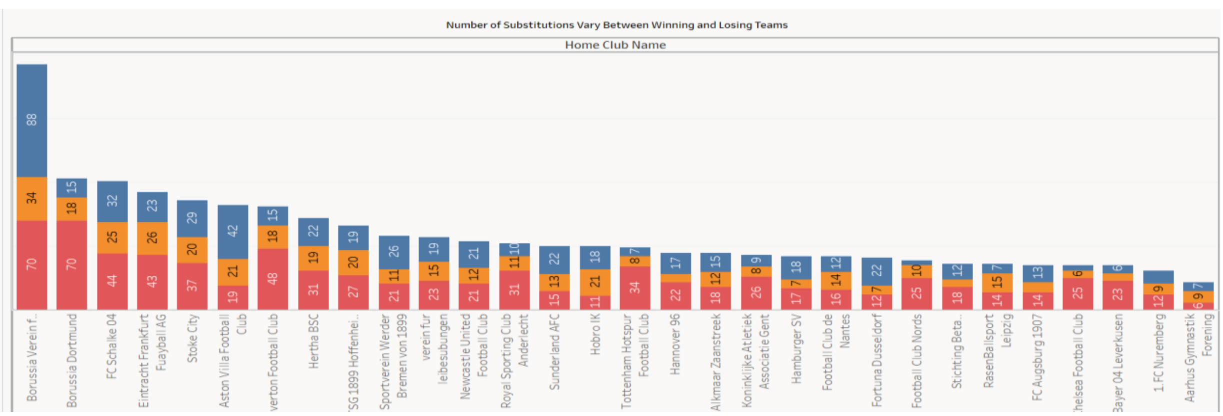
1. What is the average minute of the game when substitutions are made?
2. How does the number of substitutions vary between winning and losing teams?
3. Which players are most frequently substituted during a match?
4. What is the distribution of substitutions made by position (e.g., attackers, midfielders)?

1. What is the average minute of the game when substitutions are made?



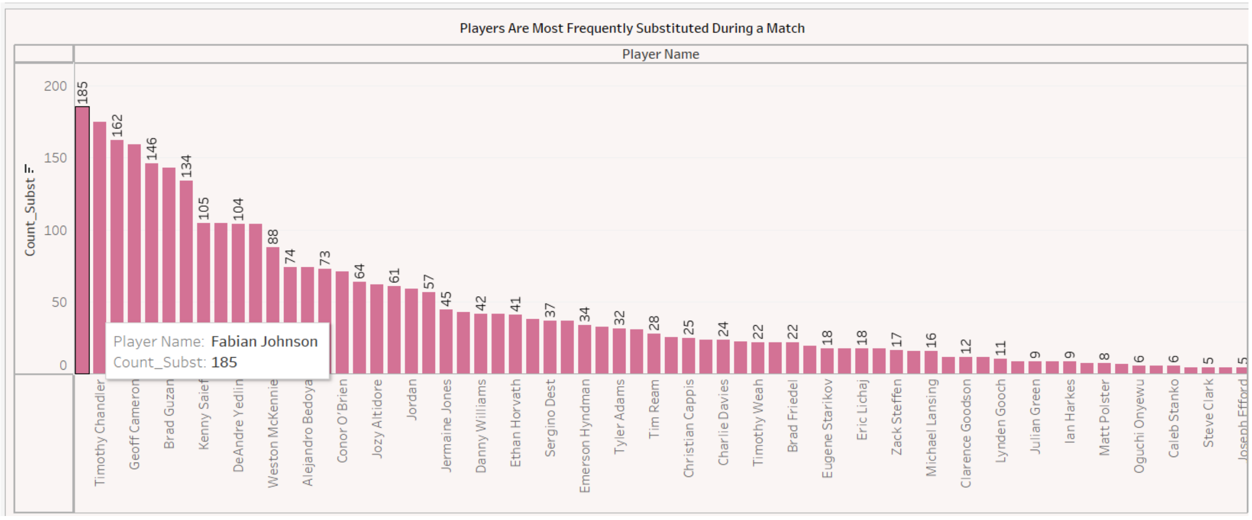
INTERPRETATION: The substitutions are made at average minute of 90 for most games.

2. How does the number of substitutions vary between winning and losing teams?



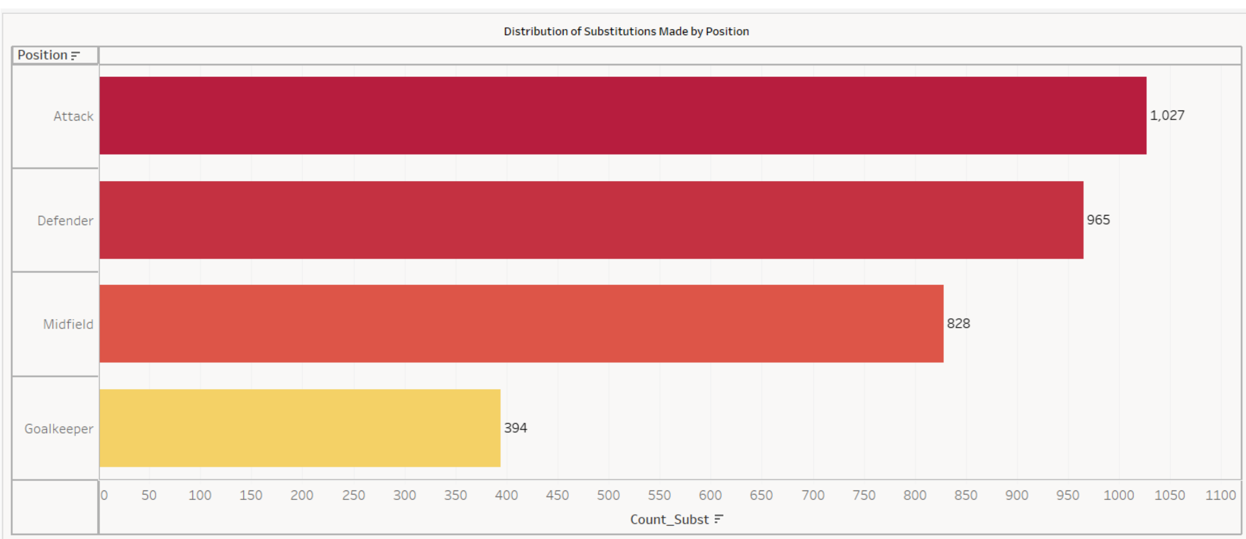
INTERPRETATION: For home team win, the highest substitutions made are 70 (count of substitutions), and for away team win the highest substitutions made are 88 (count of substitutions).

3. Which players are most frequently substituted during a match?



INTERPRETATION: Player Fabian Johnson was frequently substituted – 185 substitutions, followed by Timothy Chandler 175 substitutions.

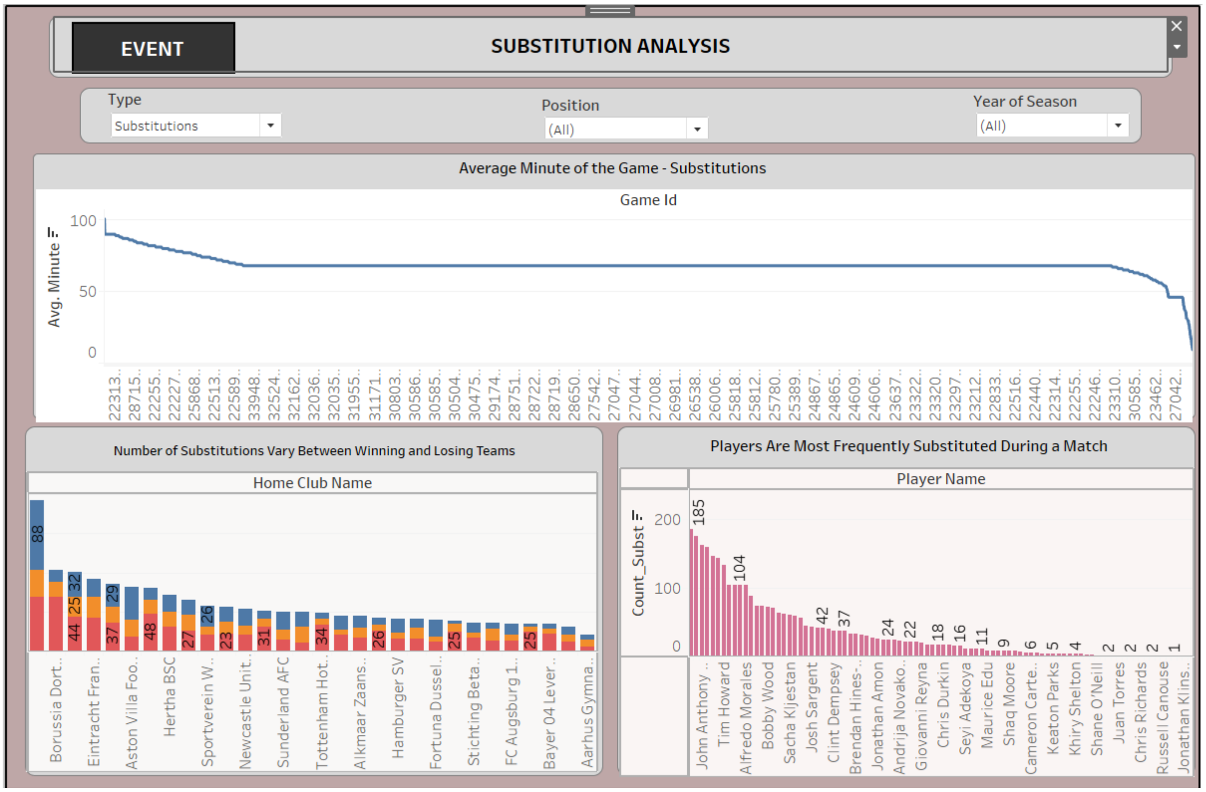
4. What is the distribution of substitutions made by position (e.g., attackers, midfielders)?



INTERPRETATION: The highest substitutions are made for the position ATTACK with count of substitutions of 1027 followed by position DEFENDER with count of 965 substitutions.

TABLEAU DASHBOARD: SUBSTITUTION ANALYSIS

1. Added worksheets, filters, text to the dashboard.
2. Added navigation button to navigate from 'substitution' dashboard to player 'event' dashboard.
3. Applied season, type and position filter to required worksheets in dashboard.
4. Added actions sheet 3 and sheet 4 to all worksheets in dashboard.



OVERALL INTERPRETATION:
BUSINESS OBJECTIVE: SUBSTITUTION ANALYSIS

Average Minute of Substitutions:

Interpretation: The average minute of substitutions across most games is 90, indicating that substitutions are often made towards the end of regulation time. This suggests a strategic use of late substitutions, likely for time management, fresh legs for defense, or pressing for a late goal.

Substitutions in Winning vs. Losing Teams:

Interpretation: Winning teams (both home and away) show a high number of substitutions:

- **Home Wins**: 70 substitutions on average.
- **Away Wins**: 88 substitutions on average.

This trend may indicate that winning teams are more likely to use their full substitution allowance, possibly to maintain momentum, counter opposition tactics, or preserve player energy.

Frequently Substituted Players:

Interpretation: Players like Fabian Johnson (185 substitutions) and Timothy Chandler (175 substitutions) are the most frequently substituted. This could imply their role as dynamic players who are strategically replaced for tactical shifts or to avoid fatigue.

Substitution Distribution by Position:

Interpretation:

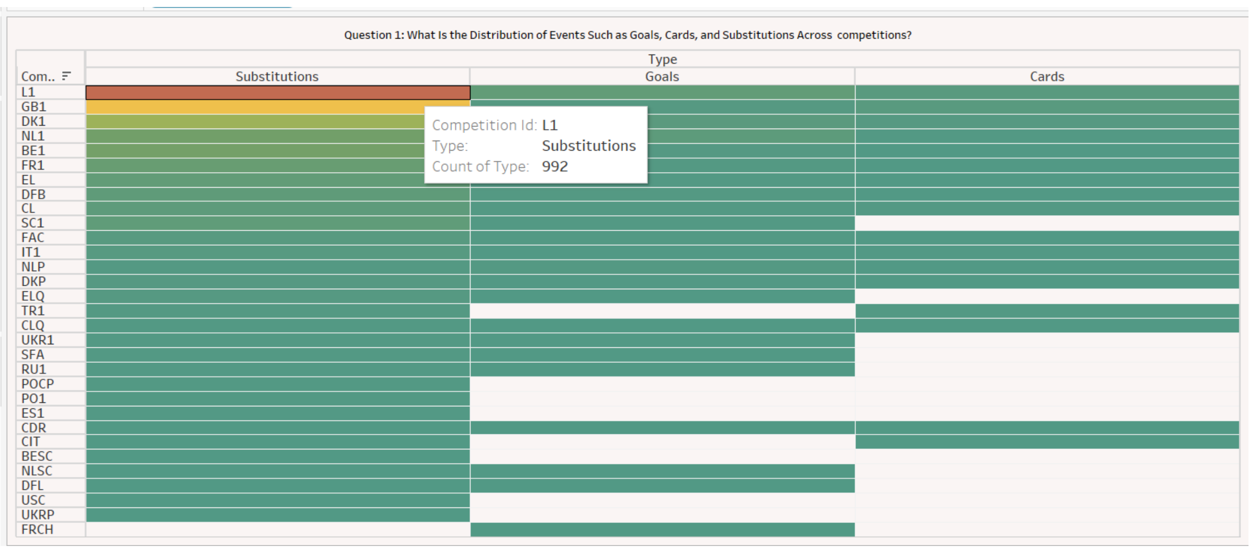
- **Attackers** are the most substituted, with 1,027 substitutions, followed by **Defenders** with 965 substitutions.
- This pattern indicates a common strategy of refreshing attacking options during the game or reinforcing defense depending on the game situation.

BUSINESS OBJECTIVE: EVENT ANALYSIS

BUSINESS QUESTIONS: (USED SQL FOR ANALYSIS)

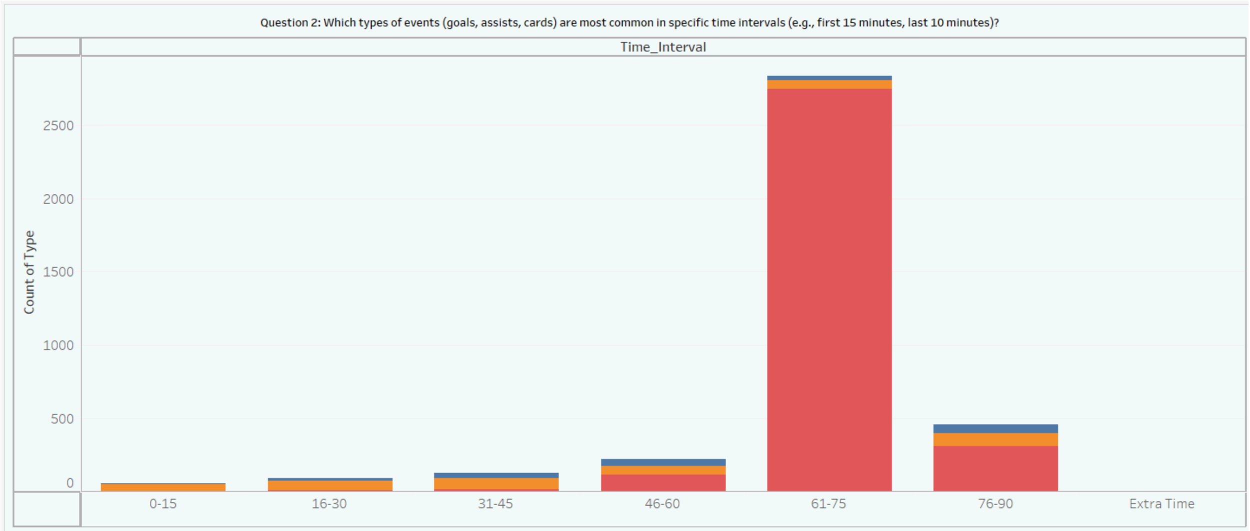
1. What is the distribution of events such as goals, cards, and substitutions across competitions?
2. Which types of events (goals, assists, cards) are most common in specific time intervals (e.g., first 15 minutes, last 10 minutes)?

1. What is the distribution of events such as goals, cards, and substitutions across competitions?



INTERPRETATION: The competition ID L1 has the maximum No. Of substitutions (992), No. Of goals (104) and No. Of cards (47). And competition ID FRCH has the least No. Of goals (1).

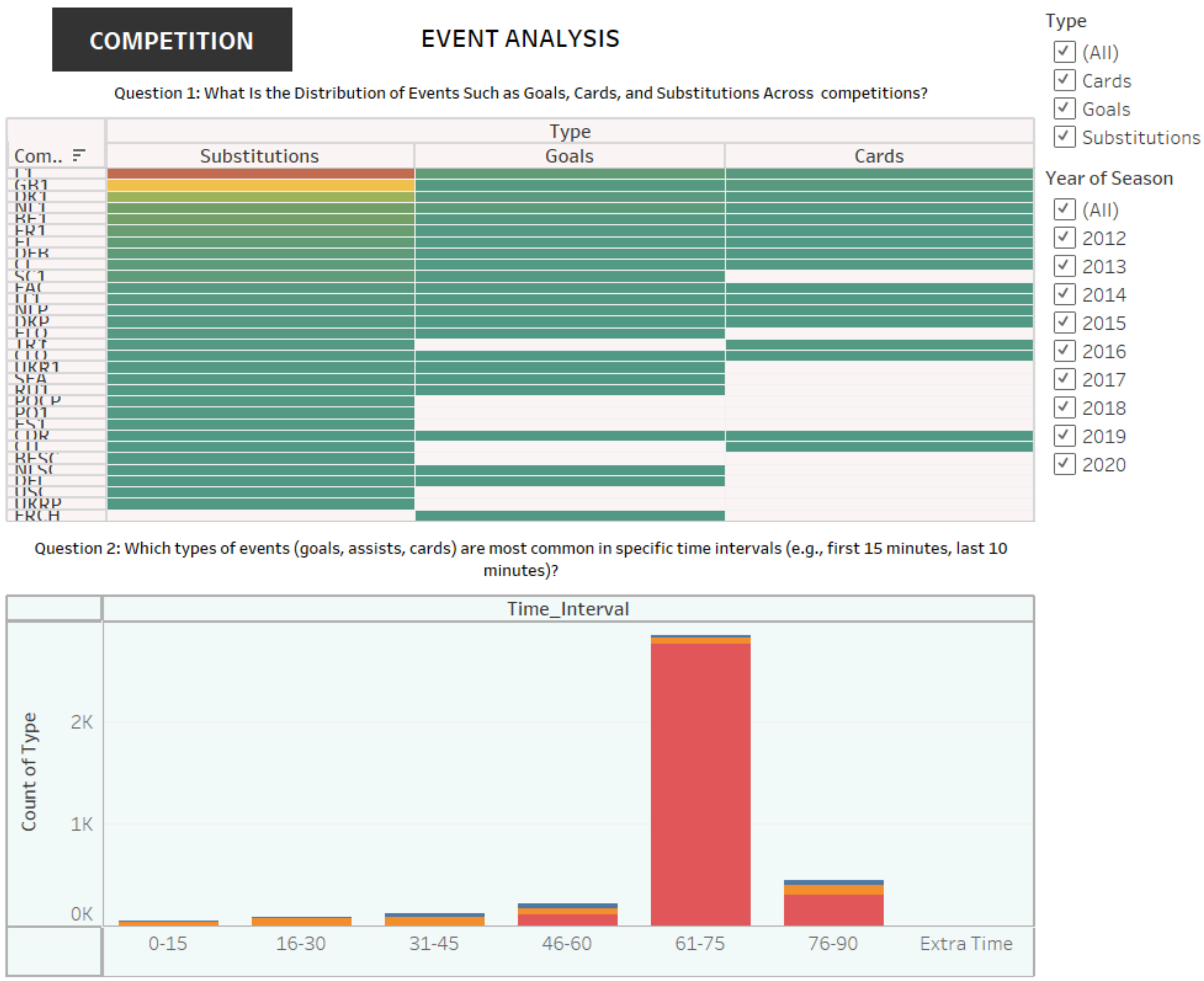
2. Which types of events (goals, assists, cards) are most common in specific time intervals (e.g., first 15 minutes, last 10 minutes)?



INTERPRETATION: Goals and cards are the common events in the first 15 min and last 10 min of the game. In the first 15 minutes, goals – 52 and cards – 7 and for the last 10 minutes of the game (76-90 min), goals – 83 and 10 cards.

TABLEAU DASHBOARD: EVENT ANALYSIS

1. Added worksheets, filters, text to the dashboard.
2. Added navigation button to navigate from ‘event’ dashboard to player ‘competition’ dashboard.
3. Applied season, type filter to required worksheets in dashboard.
4. Added actions sheet 1 to all worksheets in dashboard.



OVERALL INTERPRETATION:

BUSINESS OBJECTIVE: EVENT ANALYSIS

1. Distribution of Events Across Competitions

- **Key Findings**: In competition ID L1, there is a dominant presence of events with the highest numbers across all categories:
  - **Substitutions**: 992 instances, indicating a high level of tactical adjustments or player rotations.
  - **Goals**: 104, suggesting an offensively active competition.
  - **Cards**: 47, reflecting the competitiveness or strictness in refereeing within this competition.
- In contrast, competition ID FRCH has the fewest goals with only 1, indicating either a low-scoring trend or potentially fewer matches analyzed for that competition.

2. Events by Time Intervals

- **First 15 Minutes**:
  - **Goals**: 52, showing a decent frequency of early scoring opportunities.
  - **Cards**: 7, reflecting a lower level of disciplinary actions in the opening minutes, perhaps due to cautious play.
- **Last 10 Minutes (76-90 min)**:
  - **Goals**: 83, indicating a significant increase in goal frequency, possibly due to tactical risks taken by teams in the final stage of the game.
  - **Cards**: 10, suggesting a slightly higher tension or fatigue leading to more fouls and disciplinary actions.

Overall Observations

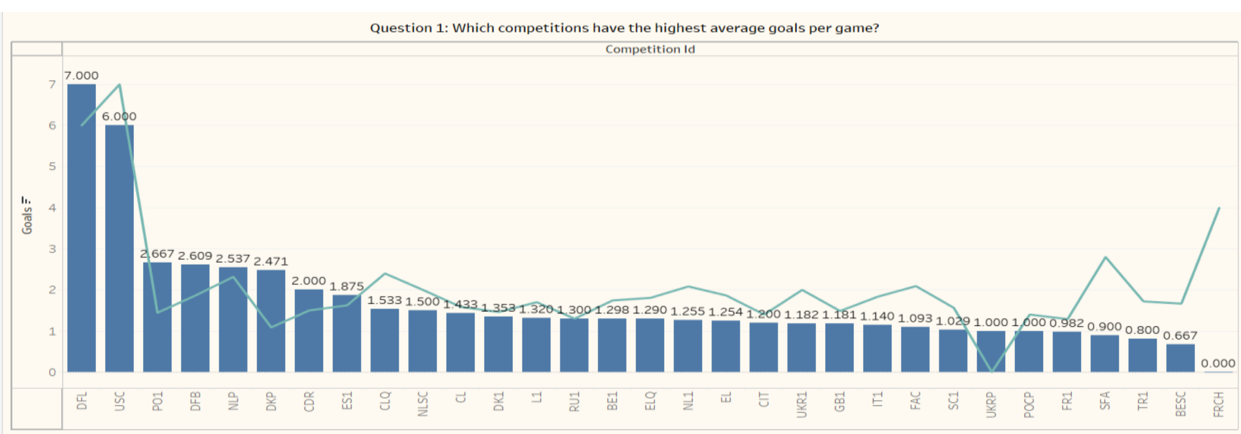
- **Competition Analysis**: The L1 competition stands out with the highest figures, indicating it's a highly dynamic competition with more substitutions and goals. This can suggest a more aggressive playing style or competitive environment.
- **Time Interval Trends**: The increase in goals in the last 10 minutes emphasizes the crucial endgame period in football, where strategies become decisive, and teams either defend a lead or push for a result.

BUSINESS OBJECTIVE: COMPETITION ANALYSIS

BUSINESS QUESTIONS:

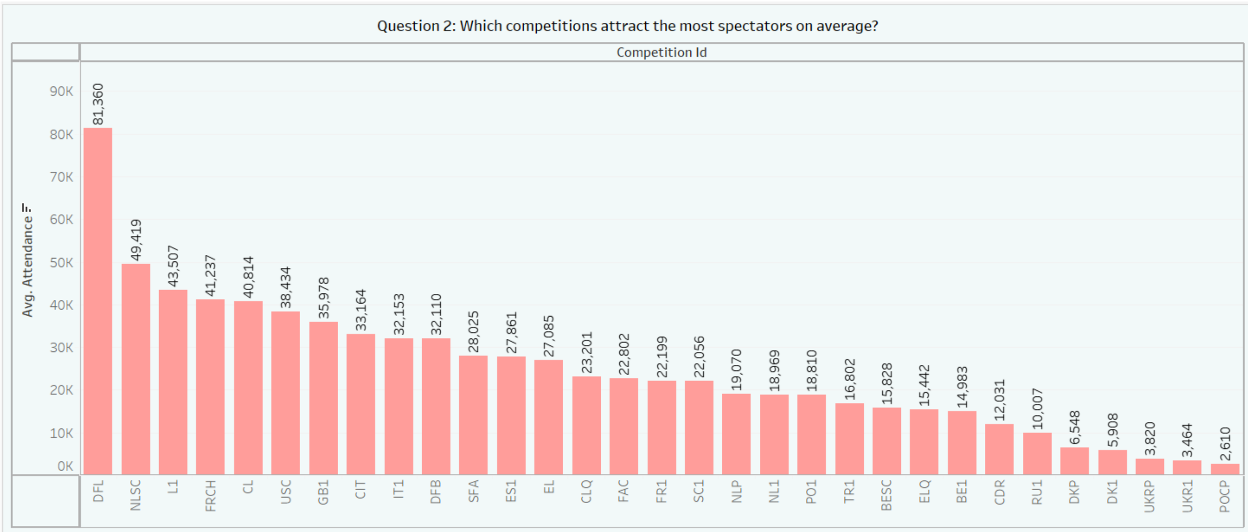
1. Which competitions have the highest average goals per game?
2. Which competitions attract the most spectators on average?
3. How does the competitive balance (difference in team strength) vary across leagues?

1. Which competitions have the highest average goals per game?



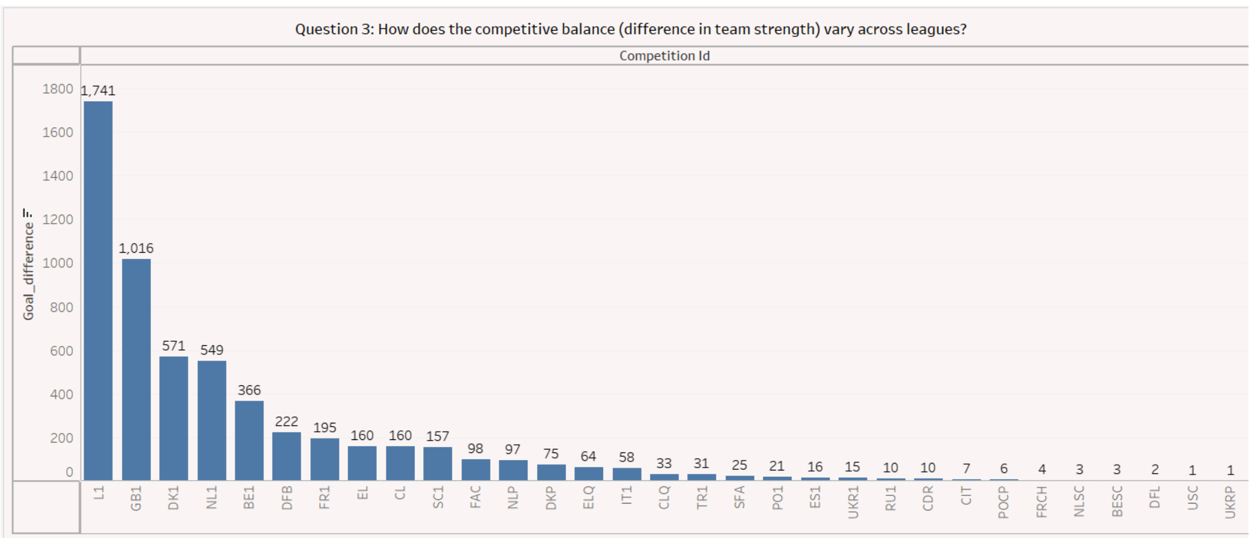
INTERPRETATION: DFL and USC are the competition ID which has the highest average total goals of 13.0

2. Which competitions attract the most spectators on average?



INTERPRETATION: The DFL (81,360) and NLSC (49,419) competitions attracted the highest average attendance, showing their strong spectator engagement.

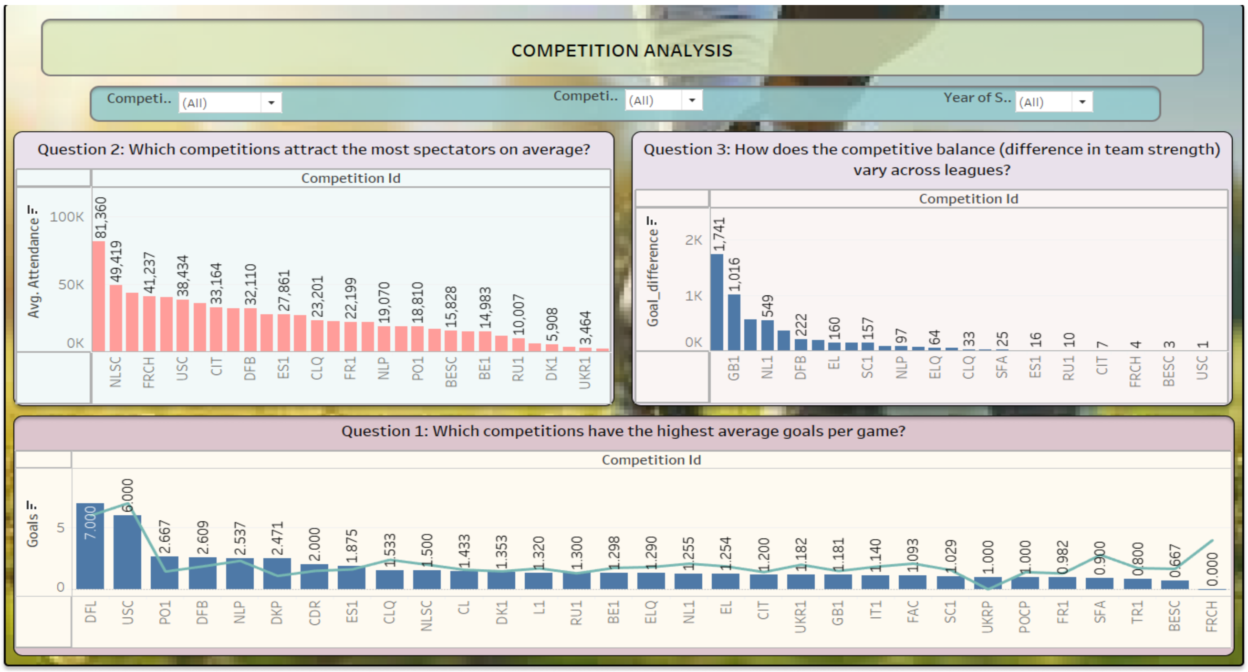
3. How does the competitive balance (difference in team strength) vary across leagues?



INTERPRETATION: UKRP (1) and USC (1) displayed the smallest average goal difference, indicating more competitive balance between teams.
L1(1741- sum of differences) and GB1(1061 - sum of differences) had the highest goal differences, suggesting larger disparities in team strength in those leagues.

TABLEAU DASHBOARD: COMPETITION ANALYSIS

1. Added worksheets, filters, text to the dashboard.
2. Applied season, competition id and competition type filter to required worksheets in dashboard.
3. Added actions sheet 2 to all worksheets in dashboard.



OVERALL INTERPRETATION:

BUSINESS OBJECTIVE: COMPETITION ANALYSIS

1. Competitions with the Highest Average Goals per Game

- **Insight**: The competitions identified as DFL and USC have the highest average goals per game, at 13.0. This suggests that matches in these competitions are notably more eventful in terms of scoring, reflecting potentially offensive playing styles or lower defensive resilience.
- **Impact**: The higher goal counts in these competitions can lead to increased fan engagement, making them attractive to spectators who favor high-scoring matches.

2. Competitions with the Most Spectators on Average

- **Insight**: The competitions with the highest average attendance are DFL (81,360) and NLSC (49,419). This shows strong spectator interest and suggests these competitions have established popularity, likely influenced by team performance, star players, or league marketing efforts.
- **Impact**: High attendance implies better revenue from ticket sales and potential sponsorship opportunities, indicating the commercial strength of these competitions.

3. Competitive Balance Across Leagues

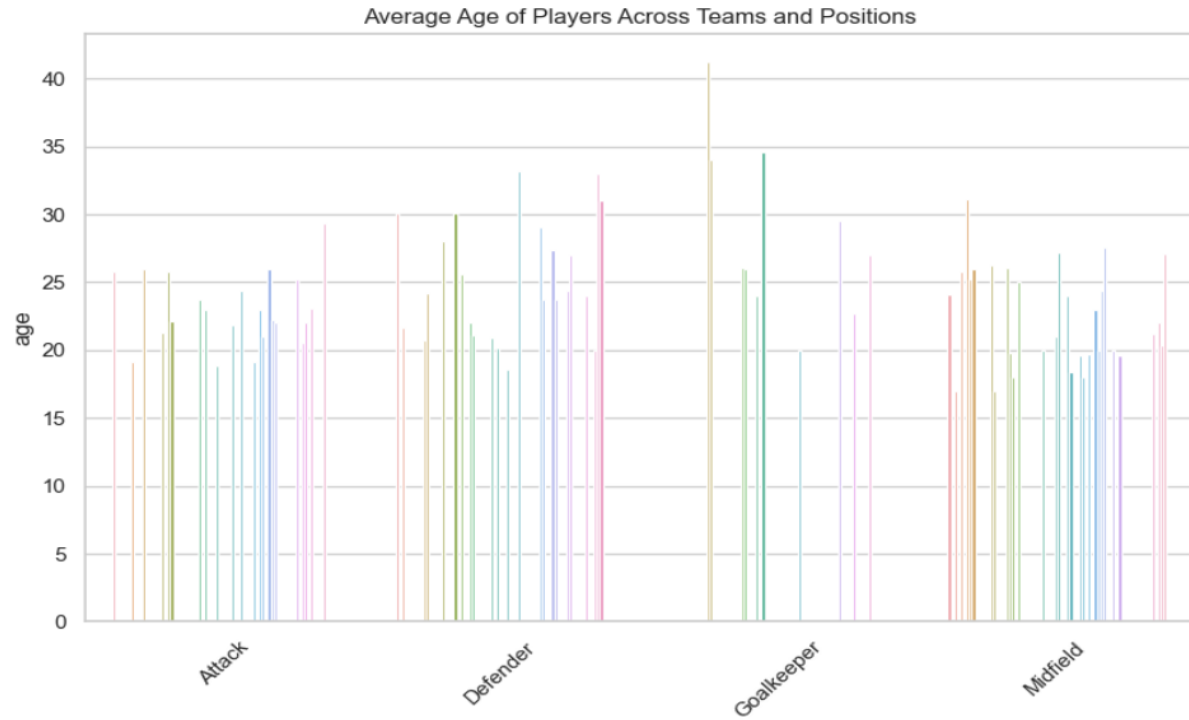
- **Insight**: UKRP and USC have the smallest average goal differences (1), indicating these leagues have a more balanced competition, with closely matched teams. This could lead to more intense and unpredictable matches, enhancing fan interest.
- **Conversely**, L1 and GB1 show higher total goal differences (1,741 and 1,061, respectively), indicating a disparity in team strength, with dominant teams likely outperforming weaker ones.
- **Impact**: A higher competitive balance might attract fans who prefer unpredictable outcomes and close contests, while leagues with less balance might still appeal to fans of dominant teams or regions.

BUSINESS OBJECTIVE: PLAYER ATTRIBUTES AND DEMOGRAPHICS

BUSINESS QUESTIONS: (USING PYTHON – CODE: PYTHON NOTEBOOK)

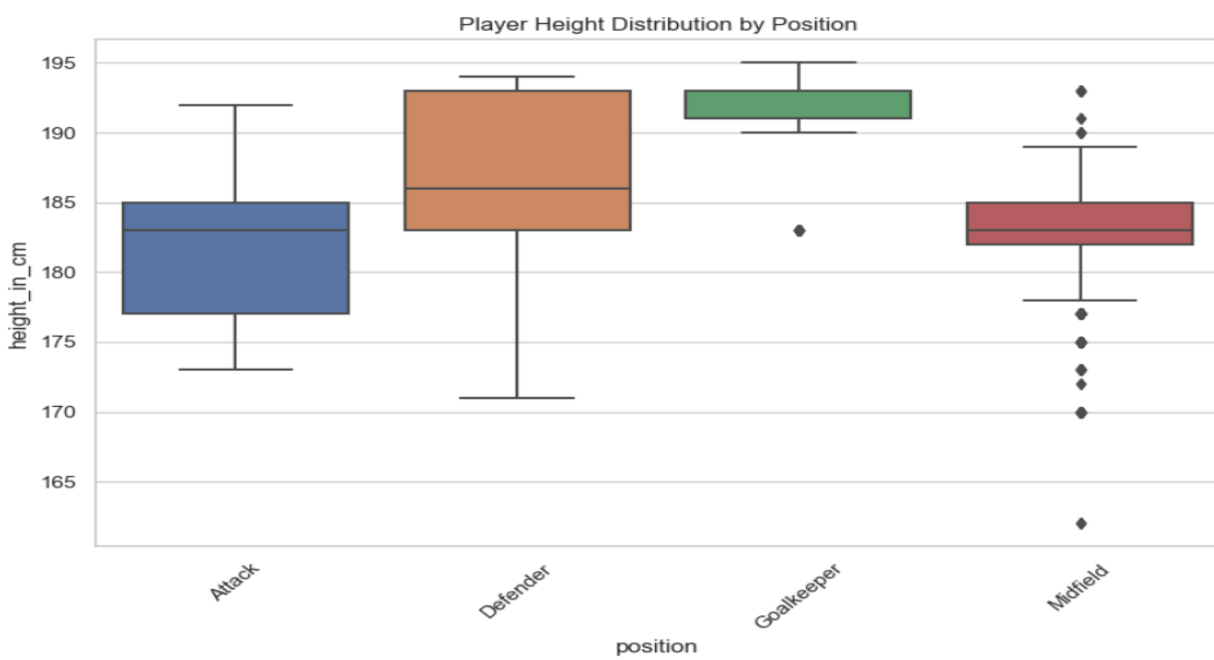
1. What is the average age of players across different teams and positions?
2. How does a player's height vary by position?
3. What is the distribution of players' preferred foot (left/right) across teams?
4. Which countries produce the most players in the league?
5. What is the average number of seasons played by players across different leagues?

1. What is the average age of players across different teams and positions?



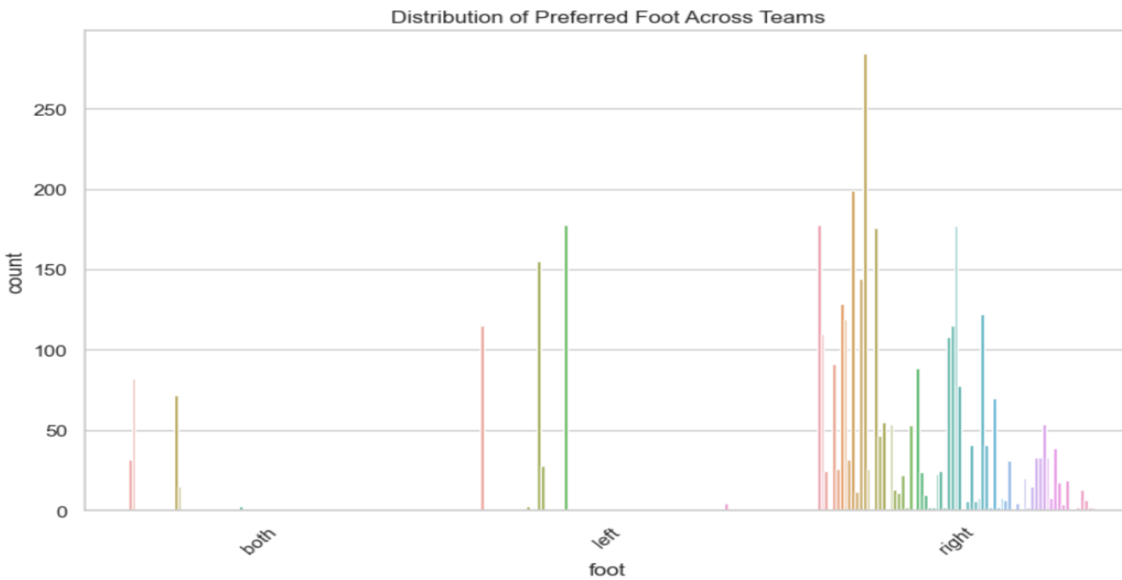
INTERPRETATION: Defenders and Midfielders show a more diverse age distribution, while Attackers tend to be slightly younger on average compared to other positions.

2. How does a player's height vary by position?



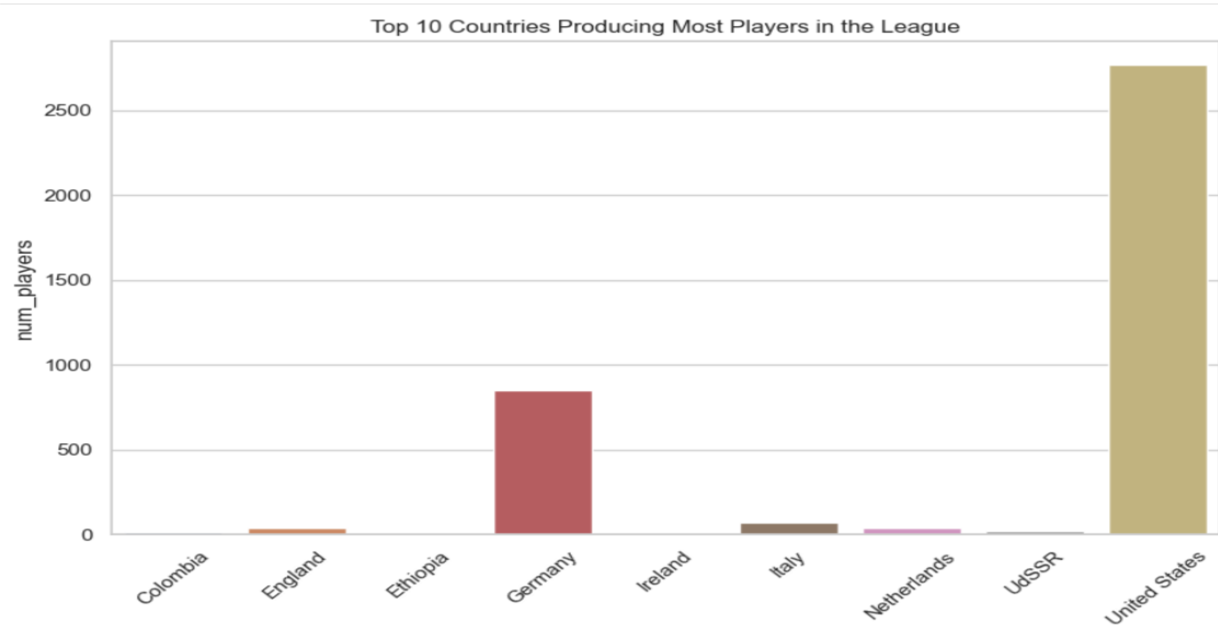
INTERPRETATION: Player Height Distribution by Position: Goalkeepers have the tallest average height (around 190 cm), followed by defenders. Attackers and midfielders have similar height distributions, with midfielders having a slightly wider range and more outliers.

3. What is the distribution of players' preferred foot (left/right) across teams?



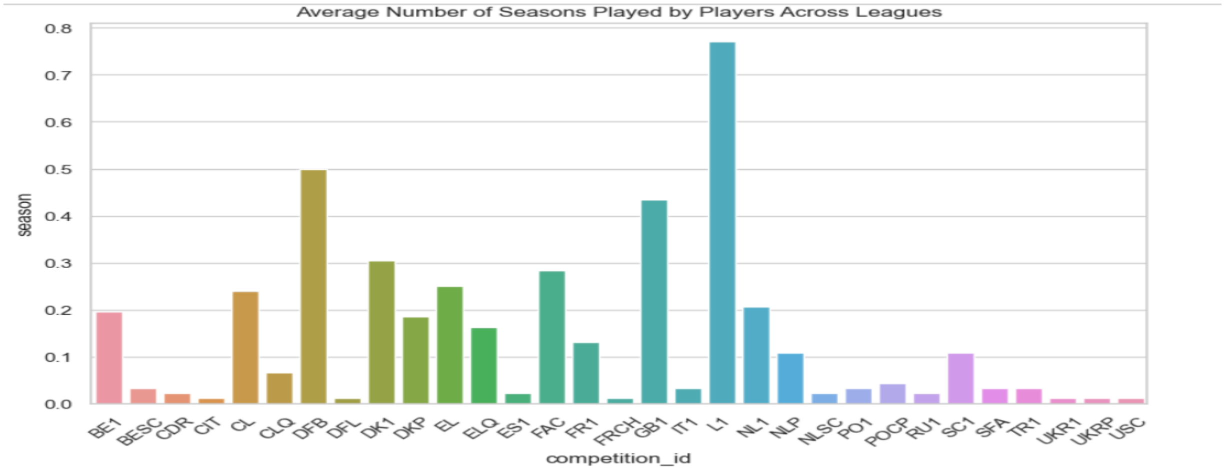
INTERPRETATION: Most of the player (highest of one team is appr. 230 players) in different teams prefer right foot in matches

4. Which countries produce the most players in the league?



INTERPRETATION: The United States produces the most players in the league, with over 2500 players(counts), followed by Germany with approximately 1000 players. Other countries contribute far fewer players, with minimal representation from countries like England and Italy.

5. What is the average number of seasons played by players across different leagues?



INTERPRETATION: The highest average number of seasons played by players is observed in competition id L1 followed by DFB

OVERALL INTERPRETATION:

BUSINESS OBJECTIVE: PLAYER ATTRIBUTES AND DEMOGRAPHICS

1. Average Age of Players Across Teams and Positions

- **Insight**: The analysis indicates that defenders and midfielders exhibit a diverse age distribution, suggesting a mix of experienced players and younger talent. Attackers, on the other hand, tend to be younger on average, which could reflect teams' strategies to incorporate speed and agility in forward positions.
- **Impact**: Understanding age distribution helps teams plan for recruitment and succession, ensuring a balance of experience and youth to foster development while remaining competitive.

2. Player Height Variation by Position

- **Insight**: Goalkeepers possess the tallest average height (around 190 cm), which aligns with the physical demands of their role. Defenders follow closely, likely due to the necessity for aerial strength. Attackers and midfielders have similar height profiles, but midfielders demonstrate a wider range, indicating variability in playing styles or tactical roles.

- **Impact**: Teams may leverage this height information for tactical decisions, such as set pieces or defensive matchups, optimizing their lineup according to physical attributes.

3. Distribution of Players' Preferred Foot Across Teams

- **Insight**: A significant majority of players (up to approximately 230 in some teams) prefer to use their right foot, indicating a potential bias in player recruitment or training strategies favoring right-footed players.
- **Impact**: This can influence team formations and strategies, as teams might need to account for left-footed players to maintain tactical balance, especially in offensive plays.

4. Countries Producing the Most Players in the League

- **Insight**: The United States leads in player production with over 2,500 players, followed by Germany with around 1,000 players. Other countries, like England and Italy, have minimal representation, indicating a concentration of player development in specific regions.
- **Impact**: This data can inform scouting and recruitment strategies, guiding teams to tap into emerging talent pools in countries with a lower representation but potential for growth.

5. Average Number of Seasons Played by Players Across Different Leagues

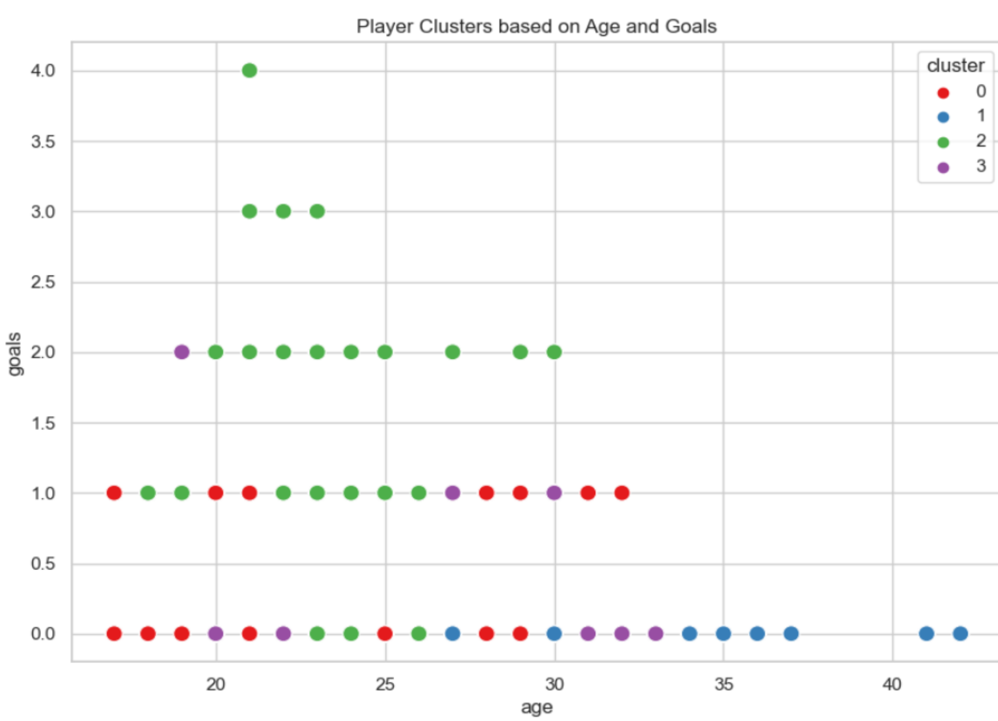
- **Insight**: Competition ID L1 shows the highest average number of seasons played, followed closely by DFB. This may suggest a stable environment for players, allowing them to develop longer-lasting careers in these leagues.
- **Impact**: Understanding the average tenure of players can help teams in workforce planning, ensuring they retain talent while also assessing the need for new signings based on player longevity and performance.

BUSINESS QUESTION: K-MEANS CLUSTERING

Can we group players into distinct clusters based on their attributes such as age, height, weight, position, and performance (goals, assists) to identify key player profiles?

```
# Importing necessary libraries

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler

# Preprocessing and selecting relevant features for clustering

player_data = fc_df[['age', 'height_in_cm','position', 'goals', 'assists']]
```

```
# Convert categorical 'position' column into numeric using one-hot encoding
player_data_encoded = pd.get_dummies(player_data, columns=['position'])
# Standardizing the data
scaler = StandardScaler()
player_data_scaled = scaler.fit_transform(player_data_encoded)
# Applying K-means clustering
kmeans = KMeans(n_clusters=4, random_state=42) # Adjust the number of clusters as needed
player_data['cluster'] = kmeans.fit_predict(player_data_scaled)
# Visualizing the clusters
plt.figure(figsize=(10, 7))
sns.scatterplot(x='age', y='goals', hue='cluster', data=player_data, palette='Set1', s=100)
plt.title('Player Clusters based on Age and Goals')
plt.show()
```



INTERPRETATION:

Distinct Player Profiles:

Cluster 0 (Red) consists of players who are generally consistent low goal scorers. This cluster could represent defensive players or those in less attacking roles who don’t focus on goal-scoring. Cluster 1 (Blue) is composed of older players who generally contribute minimally in terms of goals, possibly representing more experienced but less physically aggressive players. Cluster 2 (Green) represents high-performing goal-scorers across multiple age groups. These players may be forwards or attacking midfielders who contribute significantly to the team's offensive efforts. Cluster 3 (Purple) includes players who score moderately, possibly those who play more balanced roles on the field, contributing in both defense and attack.

Key Player Identification:

Clusters like Cluster 2 highlight top-performing players in terms of goal-scoring. This can help teams identify key players to focus on for offensive strategies. Cluster 1 may include players nearing the end of their career, providing insights for future contract decisions or retirement planning.

Performance and Age Analysis:

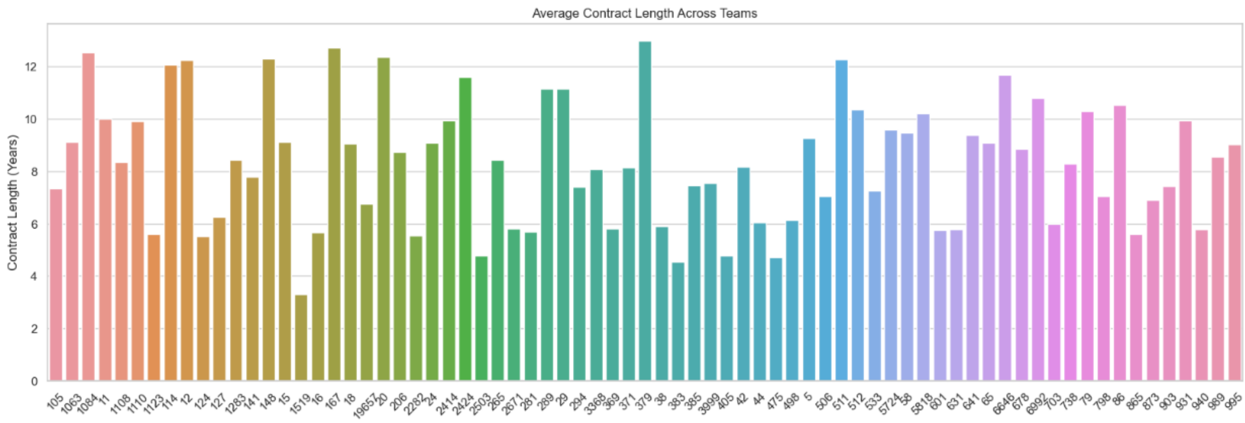
The clustering reveals that younger players (below 30 years old) are distributed across all clusters, while older players (above 30) tend to cluster in lower performance categories. This could suggest the need for talent development and youth investment strategies.

BUSINESS OBJECTIVE: CONTRACT MANAGEMENT

BUSINESS QUESTIONS:

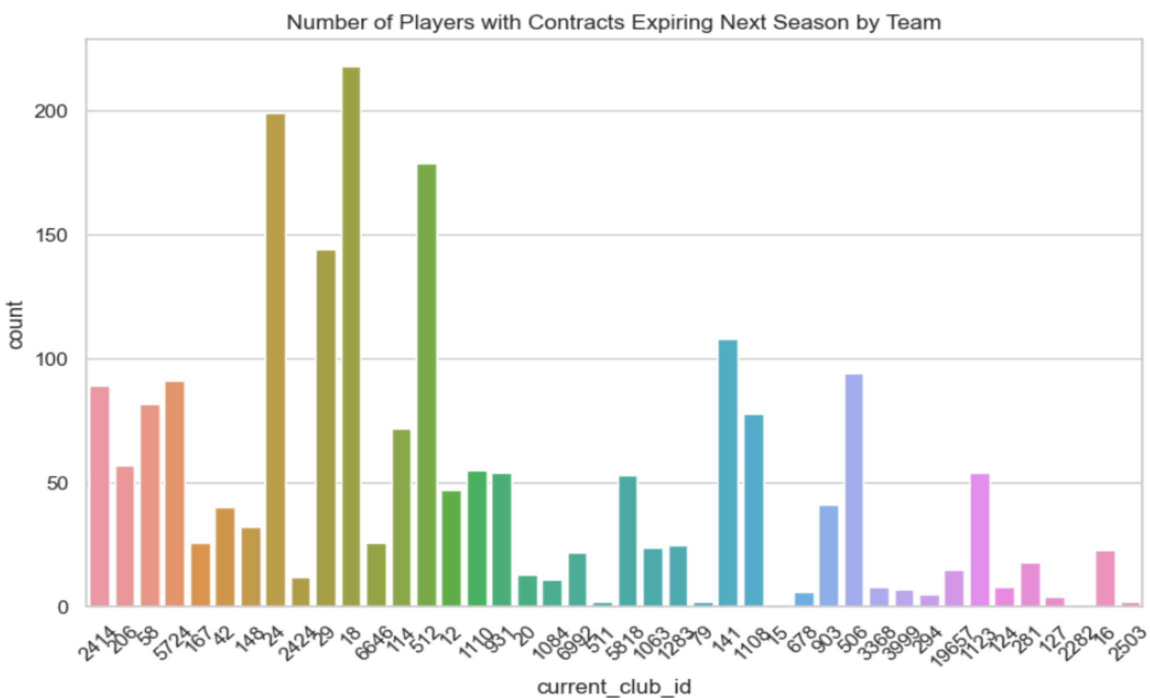
1. What is the average contract length of players across teams?
2. Which players have contracts expiring within the next season?
3. What is the relationship between a player's market value and their contract expiration date?

1. What is the average contract length of players across teams?



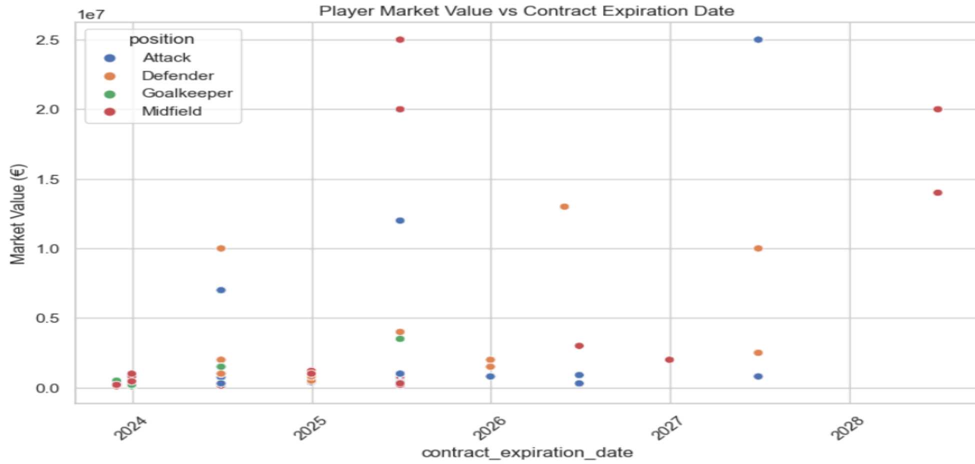
INTERPRETATION: The average contract length across teams ranges from approximately 3 to over 12 years, with significant variation between clubs, indicating different strategies or policies for player retention and contract agreements.
Some teams, such as those with club IDs around 1084, 114, and 379, have notably longer contract lengths, while others, like 124 and 1519, offer shorter average contract durations.

2. Which players have contracts expiring within the next season?



INTERPRETATION: The bar chart displays the number of players with contracts expiring next season, grouped by their current club ID.
The team with the ID '18' has the highest number of expiring contracts, over 200 players, while several teams have fewer than 50 expiring contracts

3. What is the relationship between a player's market value and their contract expiration date?



INTERPRETATION: The scatterplot shows that players with contracts expiring around 2024 generally have lower market values, while players with longer contracts (expiring after 2025) tend to have higher market values, particularly midfielders and attackers.
The highest market value observed exceeds €25 million for a midfielder with a contract expiring in 2026.

OVERALL INTERPRETATION:

BUSINESS OBJECTIVE: CONTRACT MANAGEMENT

1. Average Contract Length

- **Specific Insights**:
  - Teams with **club IDs 1084, 114, and 379** have notably longer average contract lengths, indicating a strategy focused on securing talent for extended periods. This could suggest a commitment to developing players within their systems and reducing turnover.
  - Conversely, clubs like **124 and 1519** have shorter average contract durations. This may indicate a more fluid roster strategy, potentially focusing on short-term performance or adaptability in response to changing team dynamics and market conditions.

2. Players with Expiring Contracts

- **Distribution of Expiring Contracts**: The bar chart analysis reveals which teams have players with contracts expiring in the upcoming season.

- **Team ID '18'** has the highest number of expiring contracts, with over **200 players**, which may present both challenges and opportunities for the club.
- Teams with fewer than **50 expiring contracts** may have more stability in their rosters, which can be advantageous for team chemistry and performance continuity.

3. Relationship Between Market Value and Contract Expiration Date

- **Market Value Insights**: The scatterplot analysis indicates a clear relationship between a player's market value and their contract expiration date:
  - Players with contracts expiring around **2024** tend to have **lower market values**. This might suggest that clubs are more inclined to offer lower salaries to players nearing the end of their contracts, especially if the likelihood of retention is uncertain.
  - In contrast, players with contracts expiring **after 2025** generally exhibit **higher market values**, particularly for **midfielders and attackers**. This suggests that clubs view these players as integral to their long-term strategies, which may justify higher investments in terms of salary and contract length.

OVERALL PROJECT INSIGHTS AND RECOMMENDATIONS

1. Performance Analysis

Insights: Top scorers include Aron Johannsson and Jozy Altidore. Borussia Dortmund excels both in home and away goals. The competition NL1 has the highest goals, indicating a high offensive play.

Recommendations: Focus on offensive strategies for teams in competitions with high-scoring trends and leverage key players' performance in predictive team strategies.

2. Player Profile and Market Value

Insights: High market value is not strictly correlated with goals or assists. Christian Pulisic and Giovanni Reyna are top-valued players in attack and midfield, respectively.

Recommendations: Teams should consider qualitative factors (e.g., player influence, consistency) beyond raw stats for player valuation.

3. Team Comparison

Insights: Borussia Verein and Borussia Dortmund are top-performing teams, with significant goals scored in both home and away matches.

Recommendations: Invest in balanced team strengths to enhance performance across different venues.

4. Attendance and Stadium Analysis

Insights: Stadiums like Old Trafford and Signal Iduna Park have high attendance. Attendance correlates positively with team performance.

Recommendations: For increasing fan engagement, emphasize performance at home games and target high-attendance venues for strategic matches.

5. Referee Analysis

Insights: Felix Zwayer issues the most cards, while Dr. Felix Brych has officiated the most matches, showing consistency and reliability in officiating.

Recommendations: Assign referees with balanced card-issuing tendencies to maintain fair play in competitive matches.

6. Substitution Patterns

Insights: Substitutions frequently occur at the 90th minute, and attackers are substituted most often.

Recommendations: Maximize substitutions earlier to maintain a fresh attacking advantage, especially in high-stakes matches.

7. Event Analysis

Insights: Most goals and cards occur in the final 10 minutes, emphasizing the endgame's criticality.

Recommendations: Implement strong defensive or offensive strategies nearing match end for competitive edge.

8. Competition Analysis

Insights: NL1 competition has high substitution and goal rates, indicating an intense competitive environment.

Recommendations: Prepare for aggressive play and frequent rotations in high-substitution competitions to maintain player stamina.

9. Player Attributes and Demographics

Insights: Market values are higher for players in prime age, but age alone does not determine performance.

Recommendations: Target younger players with potential in transfer markets while maintaining a core of experienced players.

10. Contract Management

Insights: Many players nearing contract end have high market value, indicating potential negotiation leverage.

Recommendations: Prioritize contract renewals for valuable players to prevent costly replacements.