REPORT

CANADIAN PREMIER LEAGUE: DREAM TEAM

By DATA LEGION:

Mohammed Faisal Sharief

Mohammed Ansar Kader Meera Sahib

Krithick Balakrishnan

Mohammad Abdul Aquib Khan

INTRODUCTION

CPL is a professional men's soccer league in Canada known as the Canadian Premier League (CPL or CanPL; French: Première ligue canadienne). It serves as the country's main national soccer league competition and is at the top of the Canadian soccer league structure. There are seven clubs in the league, representing five of Canada's 10 provinces.

The Canadian Premier League (CPL) was established in 2017 with the aim of providing Canadian soccer players with a platform to showcase their skills and develop the sport at the grassroots level. Prior to the creation of the CPL, Canadian players had limited opportunities, often having to seek opportunities abroad or play in lower-tier leagues in other countries. The league's inaugural season took place in 2019 and has since grown rapidly, attracting investment and partnerships with international organizations. And we will be analyzing the data from its inaugural 2019 season to find a Dream Team.

Participating Clubs in the 2019 Season

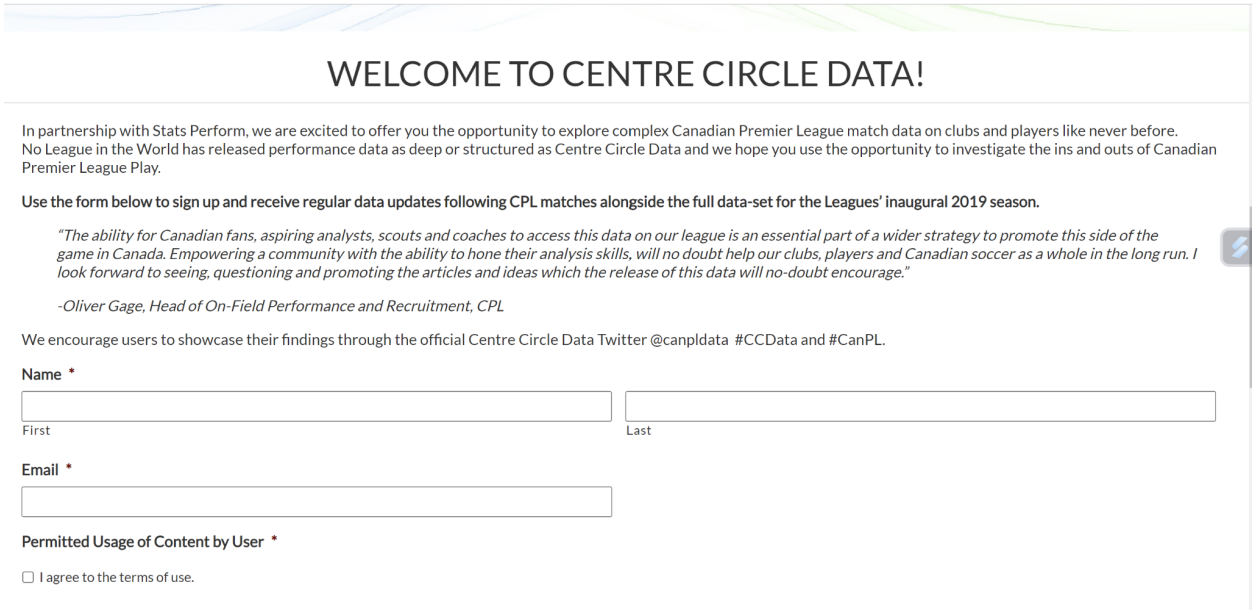
- **Pacific FC**. Vancouver Island.
- **Cavalry FC**. Calgary.
- **Valour FC**. Winnipeg.
- **Forge FC**. Hamilton.
- **HFX Wanderers FC**. Halifax.
- **York United FC**. Greater Toronto Area
- **FC Edmonton**. Alberta

PURPOSE

The purpose of this project is to develop a data-driven approach using Python and Power BI to create a Dream Team for the Canadian Premier League (CPL) for the 2019 season. The aim is to collect and analyze data on player performances and team statistics to identify the best players for each position, and to create an interactive dashboard that provides stakeholders with insights into the performance of individual players and teams. The project also aims to identify potential Key Performance Indicators (KPIs) through clinical analytics, which will help team managers make informed decisions about player selection and tactics. The stakeholders of this project include team members and CPL fans, who will benefit from the insights gained from this data-driven approach to player selection and performance evaluation, leading to a more competitive and engaging league.

DATA COLLECTION

The data for this project was collected from the official website of CPL.
https://canpl.ca/centre-circle-data/



WELCOME TO CENTRE CIRCLE DATA!

In partnership with Stats Perform, we are excited to offer you the opportunity to explore complex Canadian Premier League match data on clubs and players like never before. No League in the World has released performance data as deep or structured as Centre Circle Data and we hope you use the opportunity to investigate the ins and outs of Canadian Premier League Play.

Use the form below to sign up and receive regular data updates following CPL matches alongside the full data-set for the Leagues' inaugural 2019 season.

"The ability for Canadian fans, aspiring analysts, scouts and coaches to access this data on our league is an essential part of a wider strategy to promote this side of the game in Canada. Empowering a community with the ability to hone their analysis skills, will no doubt help our clubs, players and Canadian soccer as a whole in the long run. I look forward to seeing, questioning and promoting the articles and ideas which the release of this data will no-doubt encourage."

-Oliver Gage, Head of On-Field Performance and Recruitment, CPL

We encourage users to showcase their findings through the official Centre Circle Data Twitter @canpldata #CCData and #CanPL.

Name *

First Last

Email *

Permitted Usage of Content by User *

I agree to the terms of use.

In the webpage mentioned above, if you provide your details you will receive a copy of the dataset on your email.

The dataset we are dealing with in this project is from the inaugural 2019 season of the Canadian Premier League.

About the the data:

Variables – Team Analysis

Column Name	Attributes	Short Description	Data Type
Red	Straight Red Cards	a player is sent off and must leave the pitch after receiving a red card.	int64
Yellow	First Yellow Card	a caution issued to a player for unsporting behavior or other offenses.	int64
TouchOpBox	Touches in the Opponents Box	the number of times a player touches the ball inside the opponent's penalty area.	int64
Goal	Goals	The act of scoring a point by kicking or hitting the ball into the opponent's net.	int64
SOTInBox	Shots on Target from Inside the Box	Shots on target taken from inside the opponent's penalty area.	int64

SOTOBox	Shots on Target from Outside the Box	Shots on target taken from outside the opponent's penalty area.	int64
PsAtt	Passes Attempted	Used as a measure of a team's overall ball possession and passing accuracy	Int64
PsOnHfScs	Successful Passes in Own Half	Number of passes a team completes within their own half of the field that are considered successful	Int64
PsOpHfScs	Successful Passes in Opponents Half	Number of passes made by a team that successfully reach a player in the opponent's half of the field	Int64
FlComA3	Fouls Committed in the Attacking Third	Number of fouls committed by a team in the attacking third of the pitch.	int64
FlComD3	Fouls Committed in the Defending Third	Number of fouls committed by a team in the defending third of the pitch.	int64
FlComM3	Foulds Committed in the Middle Third	Number of fouls committed by a team in the middle third of the pitch.	int64
FlSufA3	Fouls Suffered in the Attacking Third	Number of fouls suffered by a team in the attacking third of the pitch	int64
FlSufD3	Fouls Suffered in the Defending Third	Number of fouls suffered by a team in the defending third of the pitch.	int64
FlSufM3	Fouls Suffered in the Middle Third	Number of fouls suffered by a team in the middle third of the pitch	int64

Variables – Dream Team

Column Name	Attributes	Short Description	Data Type
Goal	Goals	The act of scoring a point by kicking or hitting the ball into the opponent's net.	int64
PenGoal	Penalties Scored	A goal scored from a penalty kick.	int64
ExpG	Expected Goals	Expected goals, a statistic that calculates the likelihood of a shot resulting in a goal based on historical data.	float64

GoalHead	Headed Goals	Score a goal by heading the ball into the net from close range	int64
Ast	Assists	A pass or play that leads to a goal being scored by a teammate.	Int64
ExpA	Expected Assists	Expected assists, a statistic that calculates the likelihood of a pass or play leading to a goal based on historical data.	Float64
Chance	Total Chances	An opportunity for a player to score a goal or make a pass that could result in a goal.	Int64
BgChncCrt	Big Chances Created	Big chance opportunity created for the teammate to score	Int64
SuflDuels	Successful Duels	Defensive player stopping the progression of the attacking player with the ball and didn't commit a foul	Int64
ThrghBlCmp	Through Balls Completed	A forward pass played between opposition defenders	Int64
SucflTkls	Successful Tackles	Successful tackles, the number of times a player successfully dispossesses an opponent.	Int64
Int	Interceptions	Interceptions, the number of times a player disrupts an opponent's pass or play.	Int64
DefTouch	Defensive Touches	Actions taken by a defending player to make contact with the ball in order to prevent the opposing team from scoring or advancing the ball towards their goal.	Int64
Clrnce	Clearance	Player kicking the ball away from their goal as a form of defense	Int64
Recovery	Recoveries	Player gains possession after control of the ball has been lost by the opposition	Int64
CleanSheet	Games with 0 goals conceded	A match in which a team does not concede any goals.	Int64
Saves	Total Saves	The number of shots a goalkeeper prevents from scoring a goal.	Int64

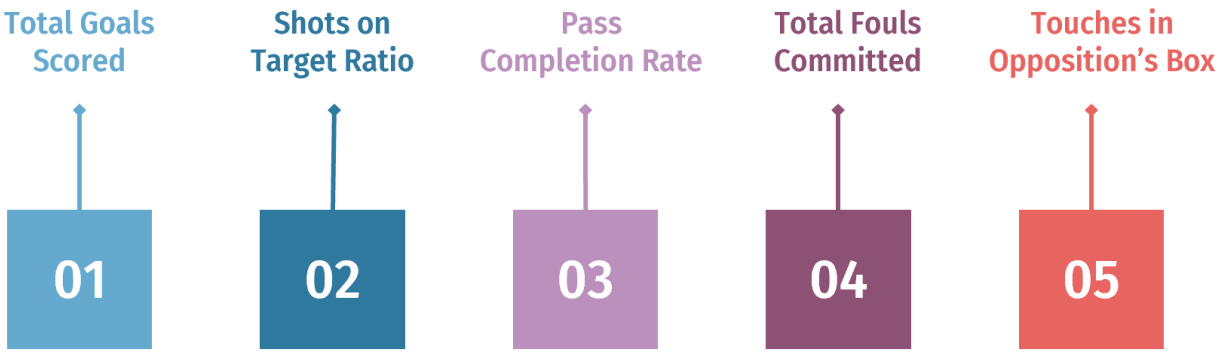
DATA MODELLING

Defining the problem: The goal of the data model is to find the best possible football team based on player statistics. This involves analyzing player data such as goals, assists, penalties, chances created, tackles, interceptions, and so on.

Cleaning and preprocessing the data: We cleaned and preprocessed the data to ensure that it is accurate and consistent. Used only the necessary attributes as mentioned above and removed unwanted columns. And also normalized or standardized the data, transforming certain features.

Developing Key Performance Indicators (KPIs):

KPIs for Team Analysis :-



KPIs for Dream Team :-

	Criteria A	Criteria B	Criteria C
Centre Forwards	Goals	Penalties	Expected Goals
Left Midfielder & Right Midfielder	Assists	Expected Assists	Chances Created
Central Attacking Midfielder	Goals	Assists	Chances Created
Defensive Midfielder	Successful Tackles	Assists	Interceptions
Left Back & Right Back	Interceptions	Successful Duels	Successful Tackles
Centre Backs	Clearance	Successful Tackles	Interceptions
Goalkeeper	Clean Sheets	Saves	-

CLINICAL ANALYSIS

In order to achieve our objective of finding a Dream Team we analyzed the data of all the Teams first. This analysis was done based on the KPIs such as Total Goals Scored by each team, Shots on Target Ratio, Pass Completion Rate, Total Fouls Committed, Red Card, Yellow Cards and Touches in Opposition's Box. Based on which an Overall Score was determined which was further used to rank each team.

Team Analysis

Total Goals Scored:

```
goals = df.groupby('team')['Goal'].sum().sort_values(ascending=False)

goals.head(7)

print (goals)

goals.plot(kind = 'bar', figsize = (15,5))
```

This code is used to group data in a DataFrame by a categorical variable ('team') and calculate a summary statistic (sum of 'Goal' column) for each group. It then sorts the resulting Series of summary statistics in descending order and selects the top 7 groups with the highest values.

Shots on Target Ratio:

```
shots_on_target = (df.groupby('team')['SOTInBox'].sum() /


df.groupby('team')['SOTOBox'].sum()).sort_values(ascending=False)

print(shots_on_target)



plt.stem(shots_on_target.index, shots_on_target)

plt.xticks(rotation=45)

plt.xlabel('Team')

plt.ylabel('Shots on Target Ratio')

plt.title('Shots on Target Ratio per Team')

plt.show()
```

This code is used to calculate and visualize the ratio of shots on target in the box to shots on target outside the box for each team in a DataFrame. The groupby() method is used to group the data by the 'team' column and the 'SOTInBox' and 'SOTOBox' columns are summed for each group. The resulting Series of sums are divided to calculate the ratio of shots on target. The shots_on_target variable stores the resulting Series, which is then sorted in descending order based on the ratio of shots on target.

Pass Completion Rate:

```
pass_completion = (df.groupby('team')['PsCmpM3', 'PsCmpA3',
'PsCmpD3'].sum().sum(axis=1) /


df.groupby('team')['PsAttM3','PsAttD3'].sum().sum(axis=1)).sort_values(asc
ending=False)

print(pass_completion)



plt.figure(figsize=(10, 8))

sns.set_style("whitegrid")

sns.barplot(x=pass_completion.values, y=pass_completion.index,
palette="PuBuGn_d")

plt.title("Pass Completion Rate by Team")

plt.xlabel("Pass Completion Rate")

plt.ylabel("Team")

plt.show()
```

This code calculates and visualizes the pass completion rate of each team in a given DataFrame. The groupby() method is used to group the data by the 'team' column, and the sum of the 'PsCmpM3', 'PsCmpA3', and 'PsCmpD3' columns is calculated for each group. The resulting DataFrame is then summed along the rows using the sum(axis=1) method to get the total number of successful passes for each team. The same is done for the total number of pass attempts by summing the 'PsAttM3' and 'PsAttD3' columns. The two resulting Series are then divided element-wise to get the pass completion rate of each team.

Total Fouls Committed:

```
total_fouls = df.groupby('team')[['FlComA3', 'FlComD3',
'FlComM3']].sum().sum(axis=1)


total_fouls = total_fouls.sort_values(ascending=False)


print(total_fouls)


fig, ax = plt.subplots(figsize=(10,8))

ax.barh(total_fouls.index, total_fouls.values)




ax.set_xlabel('Total fouls')

ax.set_ylabel('Team')

ax.set_title('Total Fouls Committed by Each Team')


plt.show()
```

This code calculates and visualizes the total number of fouls committed by each team in a given DataFrame. The groupby() method is used to group the data by the 'team' column, and the sum of the 'FlComA3', 'FlComD3', and 'FlComM3' columns is calculated for each group. The resulting DataFrame is then summed along the rows using the sum(axis=1) method, resulting in a Series that contains the total number of fouls committed by each team.

Touches in Opposition's Box:

```
touches_in_box =
df.groupby('team')['TouchOpBox'].sum().sort_values(ascending=False)

print(touches_in_box)


plt.figure(figsize=(10, 8))
```

```
sns.set_style("whitegrid")

sns.barplot(x=touches_in_box.values, y=touches_in_box.index,
palette="Blues_d")

plt.title("Total Number of Touches in Opposition Penalty Box by Team")

plt.xlabel("Number of Touches")

plt.ylabel("Team")

plt.show()
```

This code calculates and visualizes the total number of touches made by each team inside the opposition penalty box in a given DataFrame. The groupby() method is used to group the data by the 'team' column, and the sum of the 'TouchOpBox' column is calculated for each group. The resulting DataFrame is then summed along the rows using the sum() method to get the total number of touches inside the opposition penalty box for each team.

Overall Score:

```
scores = (goals + shots_on_target + pass_completion + touches_in_box -
total_fouls - yellow_cards - red_cards)


rankings = scores.sort_values(ascending=False)


print(rankings[:8])


plt.figure(figsize=(10, 8))

sns.set_style("whitegrid")

sns.barplot(x=rankings.values, y=rankings.index, palette="GnBu_d")

plt.title("Overall Scores for Teams")

plt.xlabel("Overall Score")

plt.ylabel("Team")

plt.show()
```

This code calculates and visualizes the overall score for each team in a given DataFrame. First, the scores variable is created by adding up different metrics including goals, shots_on_target, pass_completion, touches_in_box and subtracting attributes such as total_fouls, yellow_cards, and red_cards. This provides a comprehensive picture of each team's performance based on various aspects of their play.

Dream Team Analysis

After analyzing the team dataset, we analyzed the player dataset and wrote functions to find the best player for each position based on the KPIs mentioned above.

Center Forwards:

```
def find_top_scorers():
    with open('C:/CPL/cpl_player.csv') as f:
        reader = csv.DictReader(f)
        top_scorer = None
        top_score = 0
        second_top_scorer = None
        second_top_score = 0
        for row in reader:
            if row['Position'] == 'Centre Forward':
                goals = int(row['Goal'])
                penalties = int(row['PenGoal'])
                expected = float(row['ExpG'])
                score = goals + penalties - expected
                if score > top_score:
                    second_top_score = top_score
                    second_top_scorer = top_scorer
                    top_scorer = row['firstName'] + ' ' + row['lastName']
                    top_score = score
                elif score > second_top_score:
                    second_top_score = score
```

```
                    second_top_scorer = row['firstName'] + ' ' +
row['lastName']

        return [top_scorer, second_top_scorer]


top_scorers = find_top_scorers()

print(top_scorers)
```

For each row, the function first checks if the player's position is 'Centre Forward' using the 'if' statement. If the player is a center forward, the function calculates the player's score using the formula: score = goals + penalties - expected

where 'goals' is the number of goals scored by the player, 'penalties' is the number of penalty goals scored by the player, and 'expected' is the expected number of goals based on the player's performance.

In this function, the player's score is calculated as goals + penalties - expected, which means that the expected goals are subtracted from the actual goals scored by the player. This allows the function to determine how much better or worse a player performs compared to what is expected of them based on their scoring opportunities.

Left Midfielder & Right Midfielder:

```
def find_top_left_midfielders(num_players):

    with open('C:/CPL/cpl_player.csv', newline='') as f:

        reader = csv.DictReader(f)

        players = []

        for row in reader:

            if row['Position'] == 'Left Midfielder':

                assists = int(row['Ast'])

                exp_assists = float(row['ExpA'])

                chances_created = int(row['Chance'])

                score = assists * 2 + exp_assists + chances_created * 0.5

                players.append((row['firstName'] + ' ' + row['lastName'],
score))
```

```
        players.sort(key=lambda x: x[1], reverse=True)

        top_players = [player[0] for player in players[:num_players]]

        return top_players


top_left_midfielders = find_top_left_midfielders(1)

print(top_left_midfielders)
```

For each row, the function checks if the player's position is 'Left Midfielder' using the if statement. If the player is a left midfielder, the function calculates the player's score using the formula: score = assists * 2 + exp_assists + chances_created * 0.5

where assists is the number of assists by the player, exp_assists is the expected number of assists based on the player's performance, and chances_created is the number of chances created by the player. The formula gives double weight to actual assists compared to expected assists and half the weight to chances created.

Note: The same function is used to find the right midfielder. In order to do that the position is changed to right midfielder instead of left midfielder. Rest everything remains the same.

Central Attacking Midfielder:

```
def find_top_centre_attacking_midfielders(num_players):

    with open('C:/CPL/cpl_player.csv', newline='') as f:

        reader = csv.DictReader(f)

        players = []

        for row in reader:

            if row['Position'] == 'Centre Attacking Midfielder':

                goals = int(row['Goal'])

                assists = float(row['Ast'])

                chances_created = int(row['Chance'])

                score = goals * 2 + assists + chances_created * 0.5

                players.append((row['firstName'] + ' ' + row['lastName'],
score))
```

```
        players.sort(key=lambda x: x[1], reverse=True)

        top_players = [player[0] for player in players[:num_players]]

        return top_players
```

This function iterates over each row of the CSV file and checks if the player's position is 'Centre Attacking Midfielder'. For each player that meets this condition, the function calculates a score using a weighted formula that takes into account the player's goals, assists, and chances created. The formula assigns a weight of 2 to goals, 1 to assists, and 0.5 to chances created.

Defensive Midfielder:

```
def find_top_defensive_midfielder(num_players):

    with open('C:/CPL/cpl_player.csv', newline='') as f:

        reader = csv.DictReader(f)

        players = []

        for row in reader:

            if row['Position'] == 'Defensive Midfielder':

                assists = int(row['Ast'])

                successful_tackles = int(row['SucflTkls'])

                interceptions = int(row['Int'])

                score = assists + successful_tackles * 2 + interceptions *
0.5

                players.append((row['firstName'] + ' ' + row['lastName'],
score))

        players.sort(key=lambda x: x[1], reverse=True)

        top_players = [player[0] for player in players[:num_players]]

        return top_players


top_defensive_midfielders = find_top_defensive_midfielder(1)

print(top_defensive_midfielders)
```

This function iterates over each row and checks if the player's position is "Defensive Midfielder". If the player's position is a defensive midfielder, the function calculates the player's score using the formula assists + successful_tackles * 2 + interceptions * 0.5, where assists is the number of assists made by the player, successful_tackles is the number of successful tackles made by the player, and interceptions is the number of interceptions made by the player.

Left Back & Right Back:

```
def find_left_back(num_players):
    with open('C:/CPL/cpl_player.csv', newline='') as f:
        reader = csv.DictReader(f)
        players = []
        for row in reader:
            if row['Position'] == 'Left Back':
                interceptions = int(row['Int'])
                successful_tackles = int(row['SucflTkls'])
                successful_duels = int(row['SucflDuels'])
                score = interceptions * 2 + successful_tackles * 0.5 +
successful_duels
                players.append((row['firstName'] + ' ' + row['lastName'],
score))
        players.sort(key=lambda x: x[1], reverse=True)
        top_players = [player[0] for player in players[:num_players]]
        return top_players


top_left_backs = find_left_back(1)
print(top_left_backs)
```

This function iterates over each row in the CSV file and checks if the Position column matches the string 'Left Back'. If it does, the function calculates a score for the player based on their Interceptions, Successful Tackles, and Successful Duels statistics, which are obtained from the

respective columns in the CSV file. The score is calculated as interceptions * 2 successful_tackles * 0.5 + successful_duels. The player's name and score are appended to a list of tuples called players.

Note: The same function is used to find the right back. In order to do that the position is changed to right back instead of left back. Rest everything remains the same.

Center Backs:

```
def find_left_centre_back(num_players):
    with open('C:/CPL/cpl_player.csv', newline='') as f:
        reader = csv.DictReader(f)
        players = []
        for row in reader:
            if row['Position'] == 'Left Centre Back':
                interceptions = int(row['Int'])
                successful_tackles = int(row['SucflTkls'])
                clearance = int(row['Clrnce'])
                score = interceptions + successful_tackles * 2 + clearance
* 3
                players.append((row['firstName'] + ' ' + row['lastName'],
score))
        players.sort(key=lambda x: x[1], reverse=True)
        top_players = [player[0] for player in players[:num_players]]
        return top_players


top_left_centre_backs = find_left_centre_back(1)
print(top_left_centre_backs)
```

This function extracts data on players whose position is listed as 'Left Centre Back'. For each of these players, it calculates a 'score' based on their number of interceptions, successful tackles and clearances. The score is computed as the sum of interceptions, twice the number of successful

tackles, and three times the number of clearances. The function then sorts the players in descending order based on their score and returns a list of the names of the top left center back.

Note: The same function is used to find the right center back. In order to do that the position is changed to right center back instead of left center back. Rest everything remains the same.

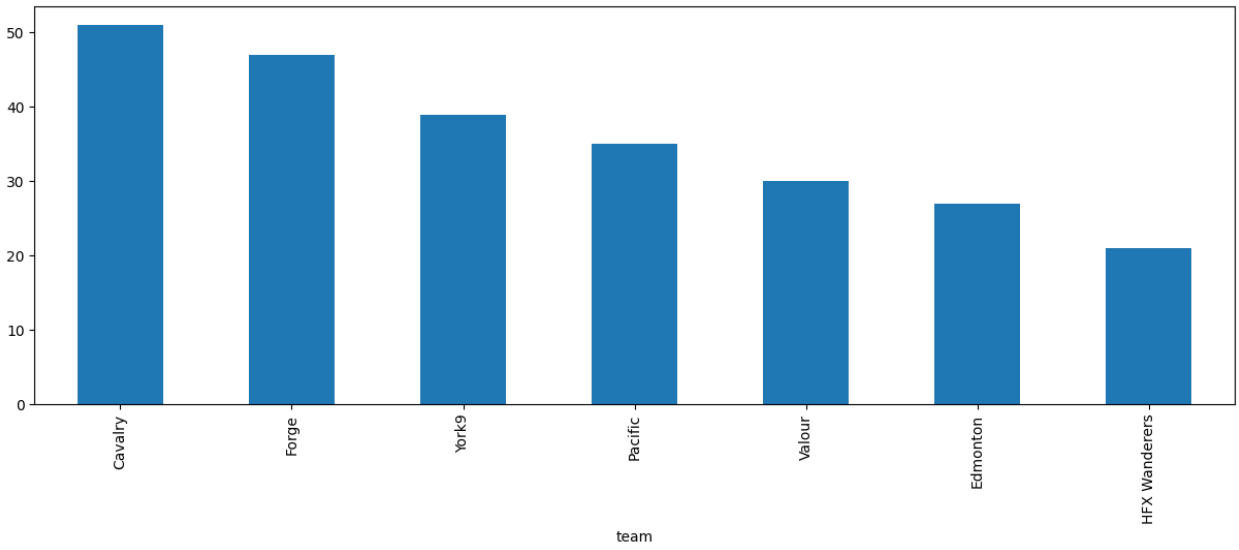
Goalkeeper:

```
def find_top_goalkeeper(num_players):

    with open('C:/CPL/cpl_player.csv', newline='') as f:

        reader = csv.DictReader(f)

        players = []

        for row in reader:

            if row['Position'] == 'Goalkeeper':

                saves = int(row['Saves'])

                clean_sheets = int(row['CleanSheet'])

                score = saves * 2 + clean_sheets * 3

                players.append((row['firstName'] + ' ' + row['lastName'],
score))

        players.sort(key=lambda x: x[1], reverse=True)

        top_players = [player[0] for player in players[:num_players]]

        return top_players


top_goalkeepers = find_top_goalkeeper(1)

print(top_goalkeepers)
```

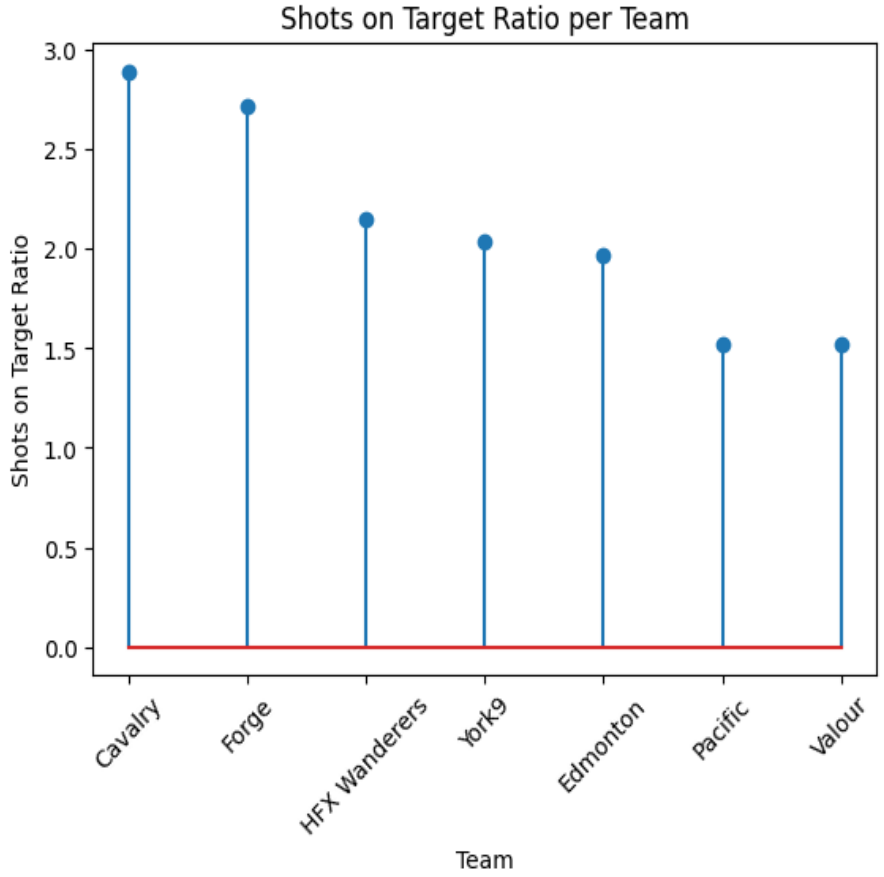
This function iterates over each row in the CSV file using a for loop. For each row, the function checks if the player's position is 'Goalkeeper' and if so, it calculates a score based on the number of saves and clean sheets they have. The score is calculated as saves * 2 + clean_sheets * 3. The function then appends the player's name and score as a tuple to the players list.

DATA VISUALIZATION

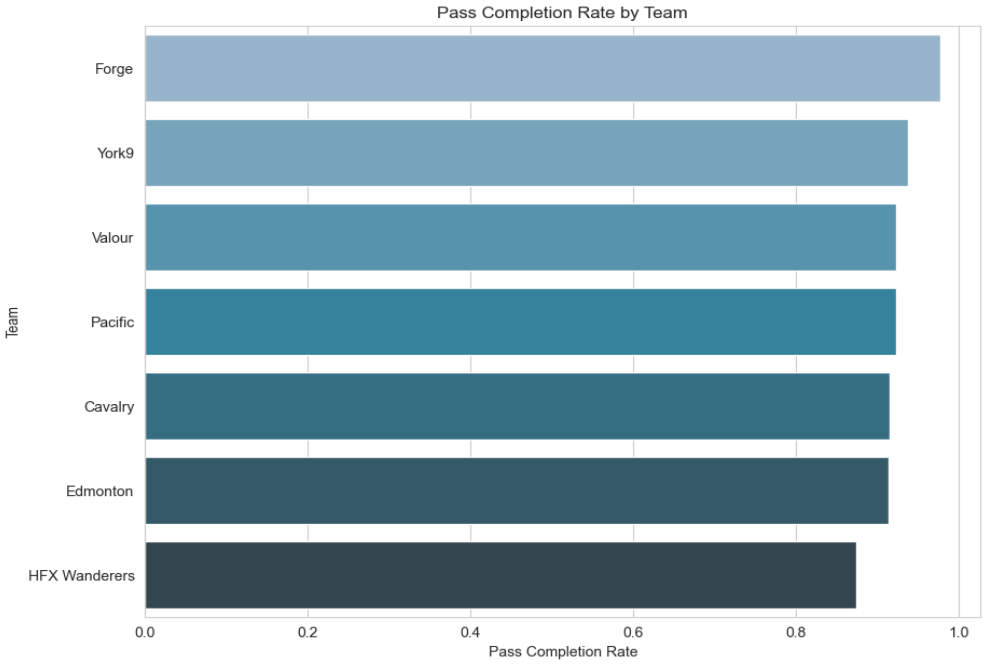
Total Goals Scored by each Team



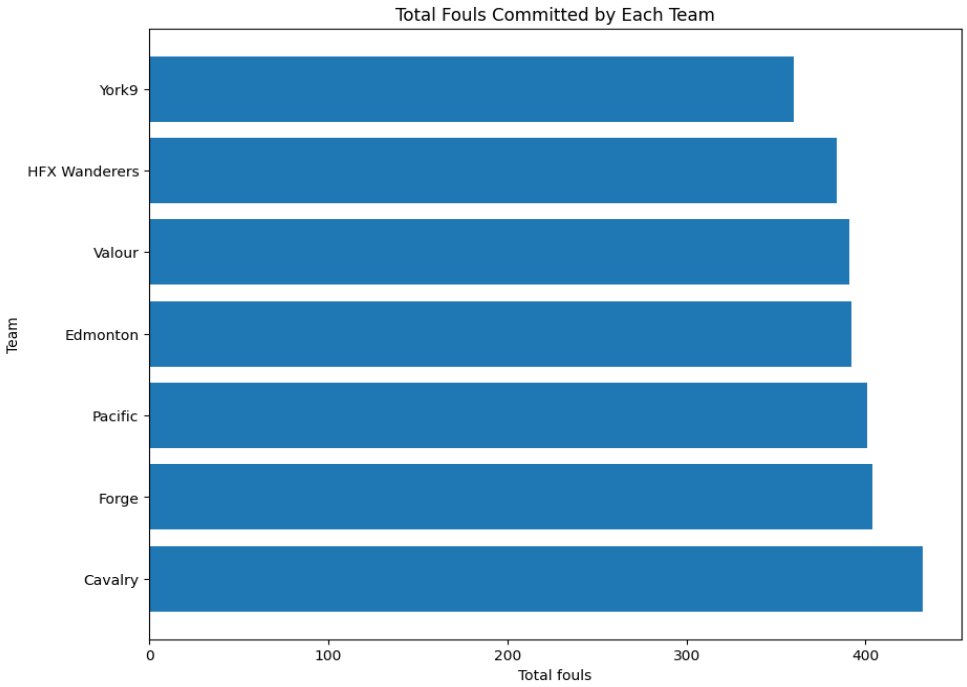
Shots on Target Ratio of each Team



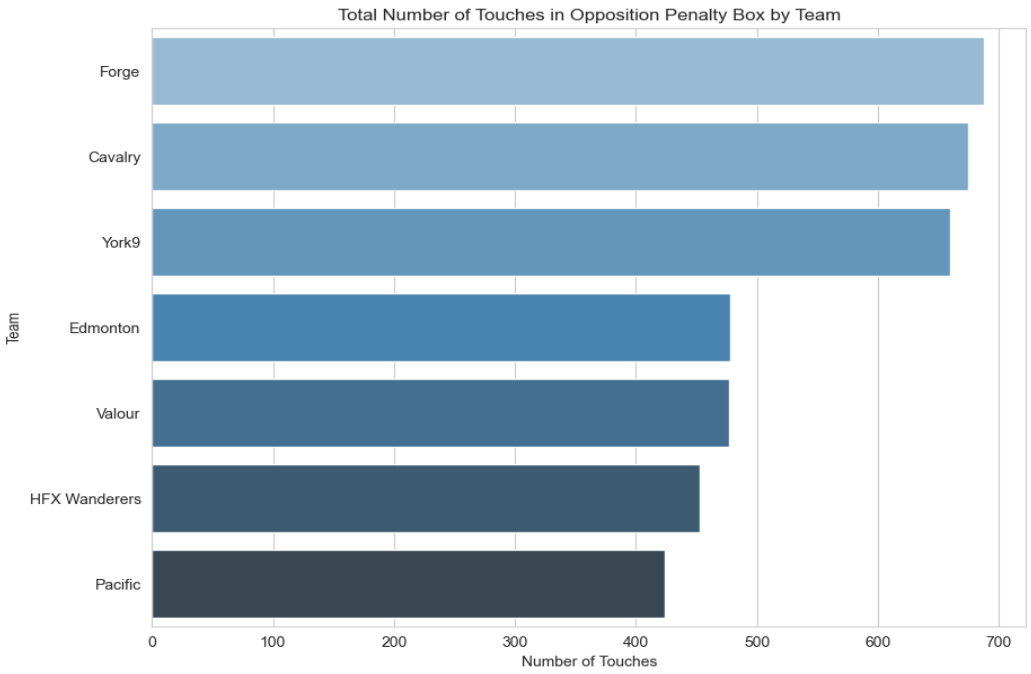
Pass Completion Rate of each Team



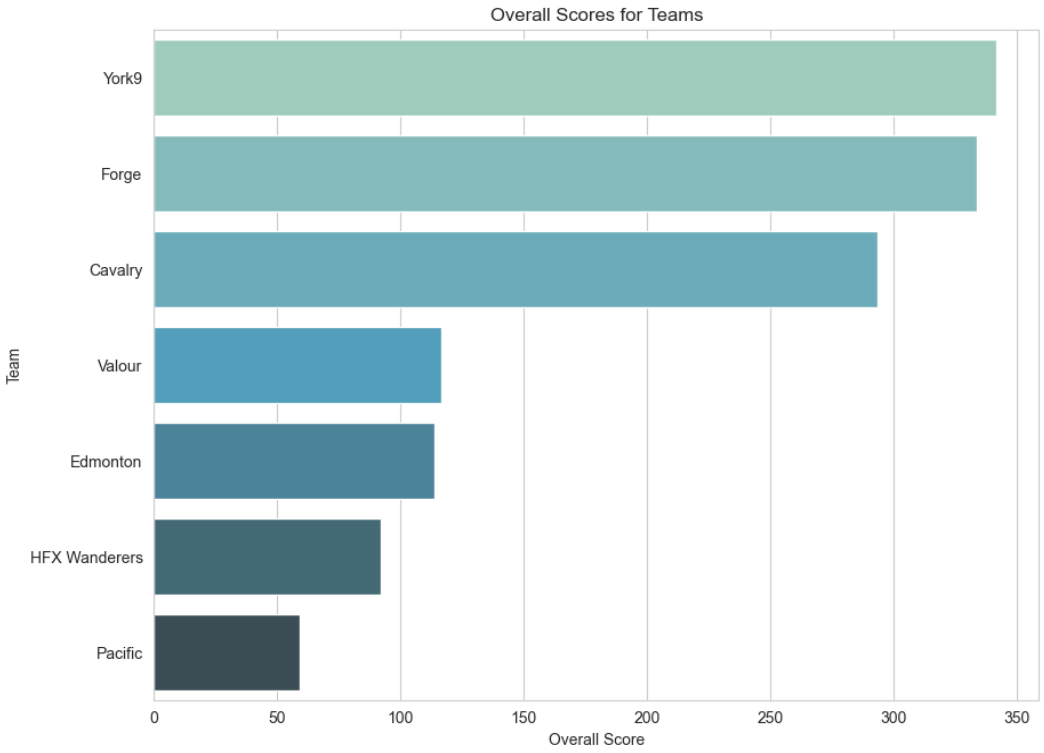
Total Fouls Committed by each Team



Touches in Opposition's Box

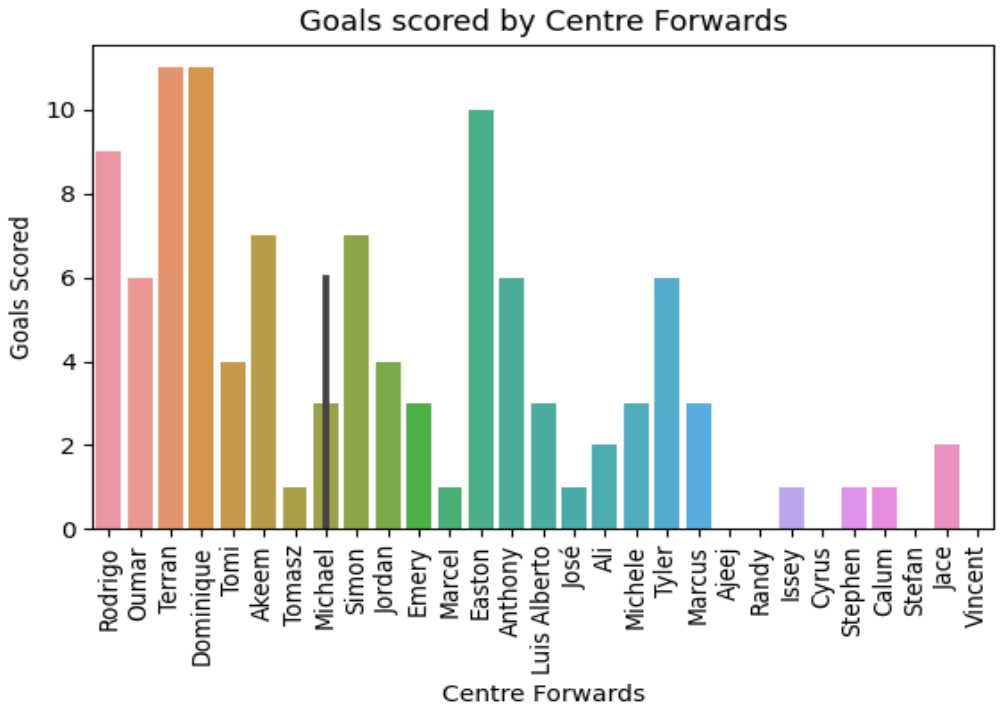


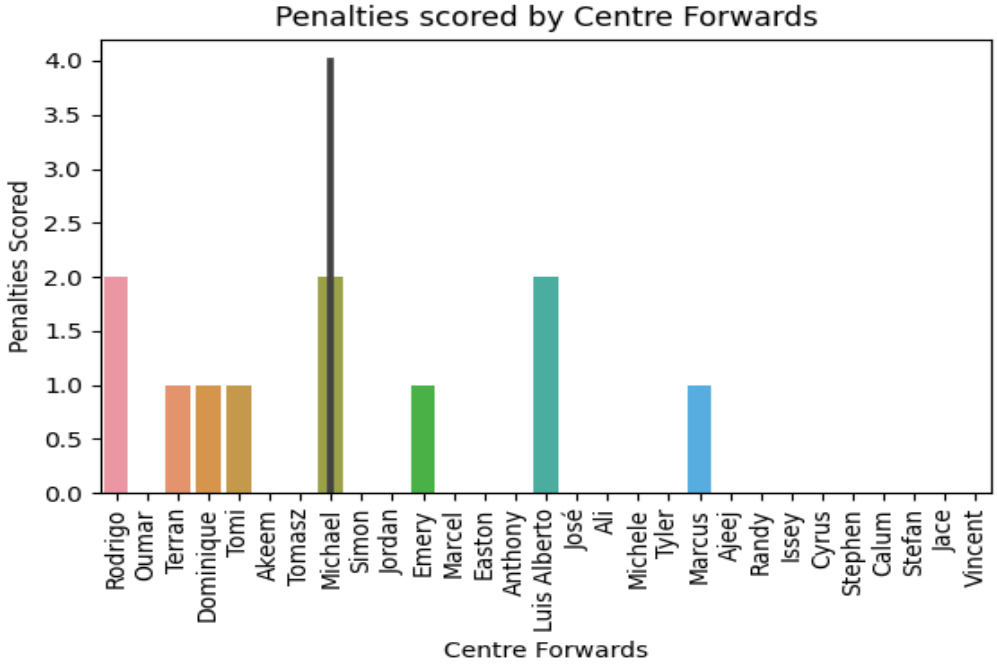
Overall Score

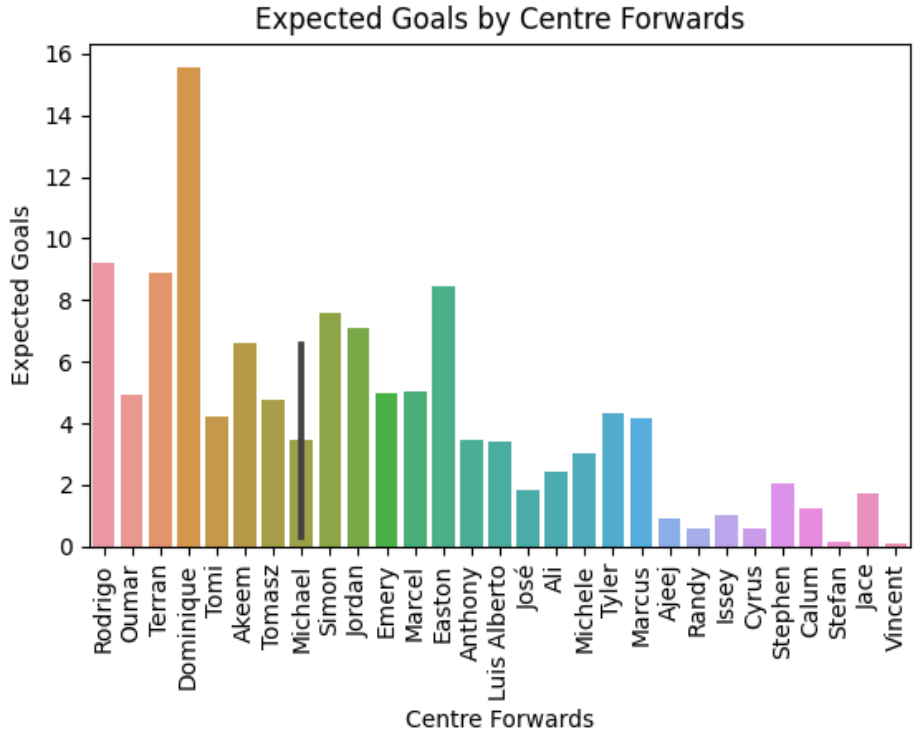


Dream Team:

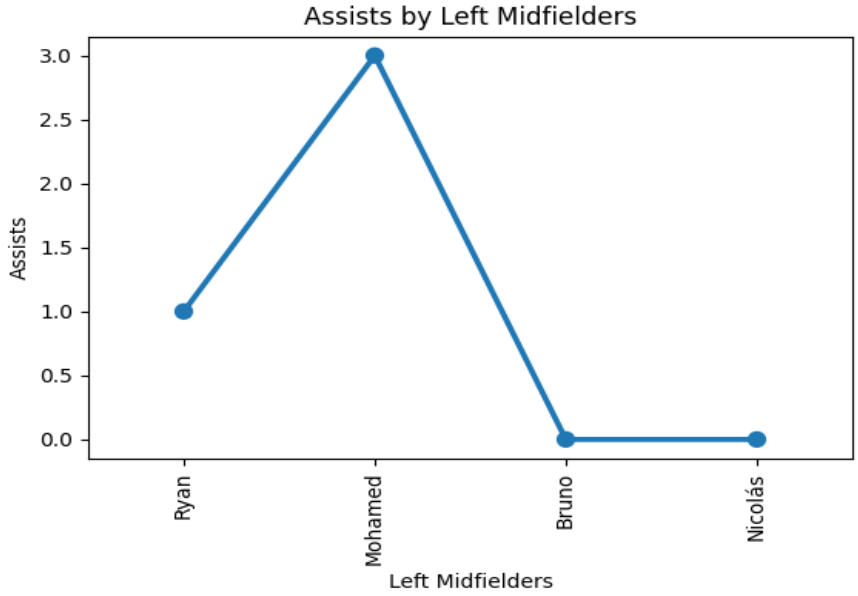
Center Forwards: Michael & Terran *(score = goals + penalties - expected)*

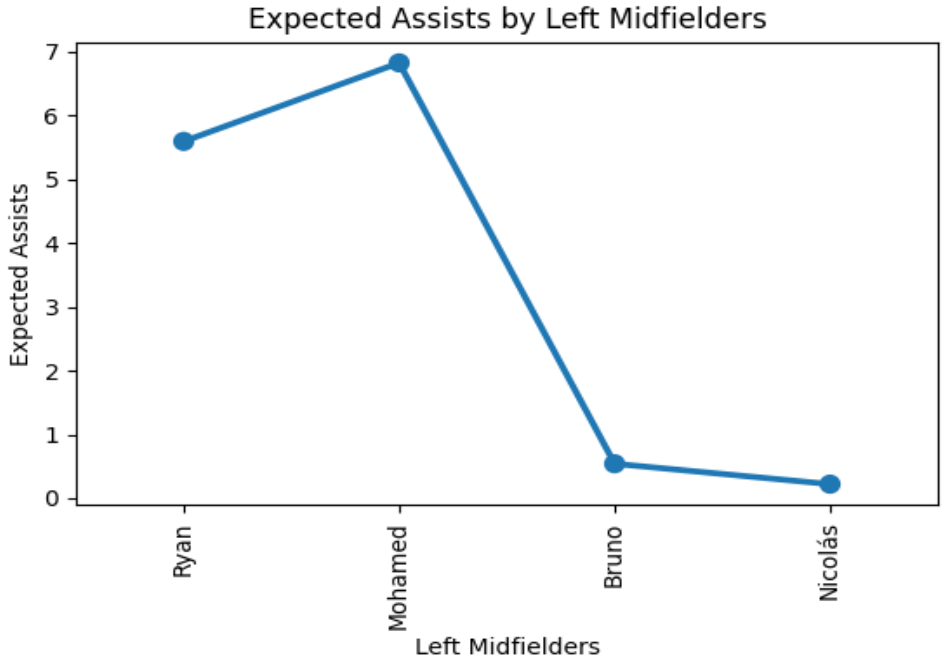


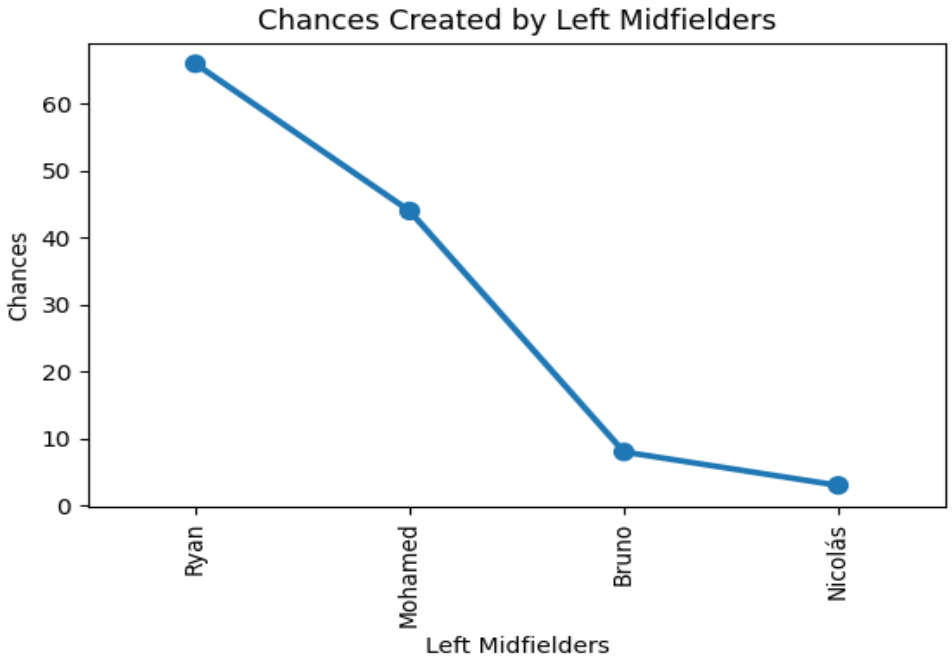




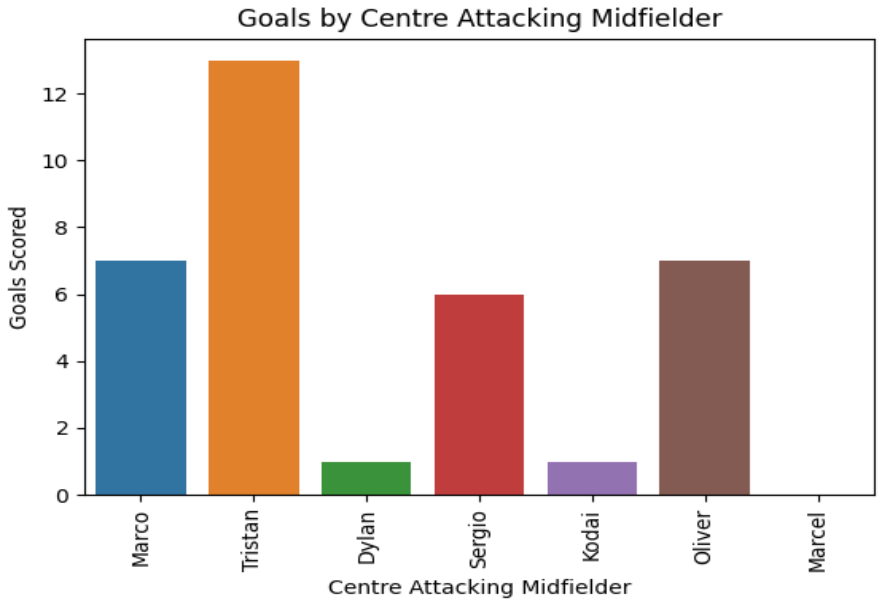
Left Midfielder: Ryan *(score = assists * 2 + exp_assists + chances_created * 0.5)*

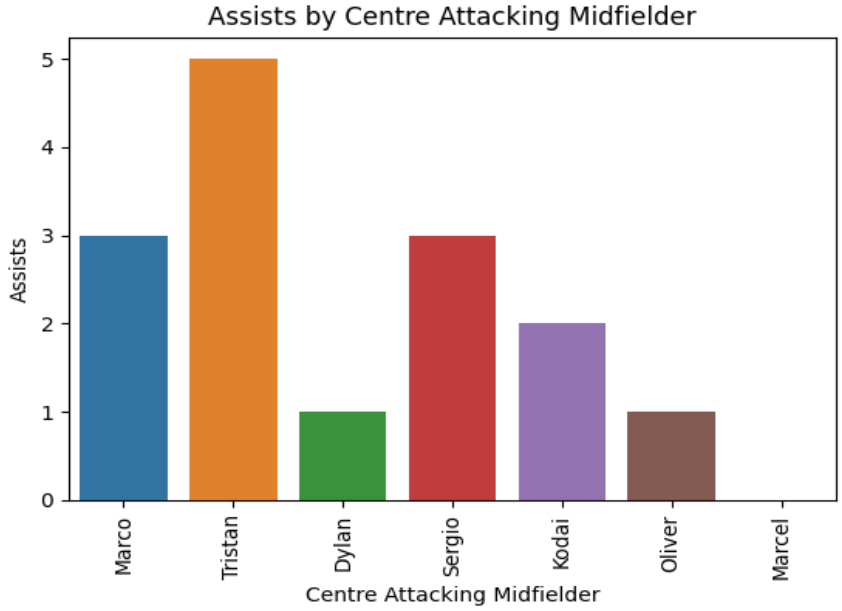


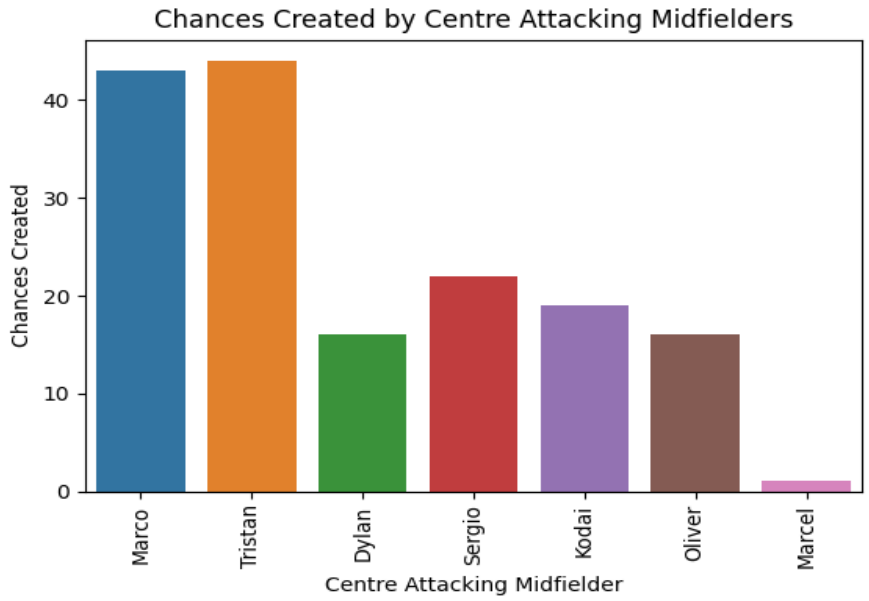

Expected Assists by Left Midfielders
Expected Assists
0
1
2
3
4
5
6
7
Ryan
Mohamed
Bruno
Nicolás
Left Midfielders


Chances Created by Left Midfielders
Chances
0
10
20
30
40
50
60
Ryan
Mohamed
Bruno
Nicolás
Left Midfielders

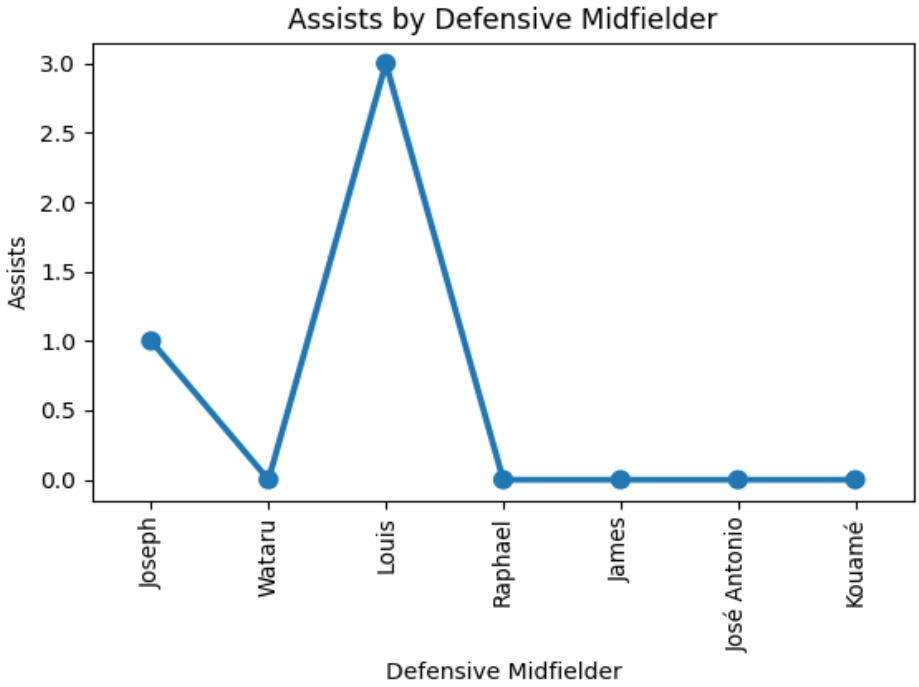
Central Attacking Midfielder: Tristan *(score = goals * 2 + assists + chances_created * 0.5)*

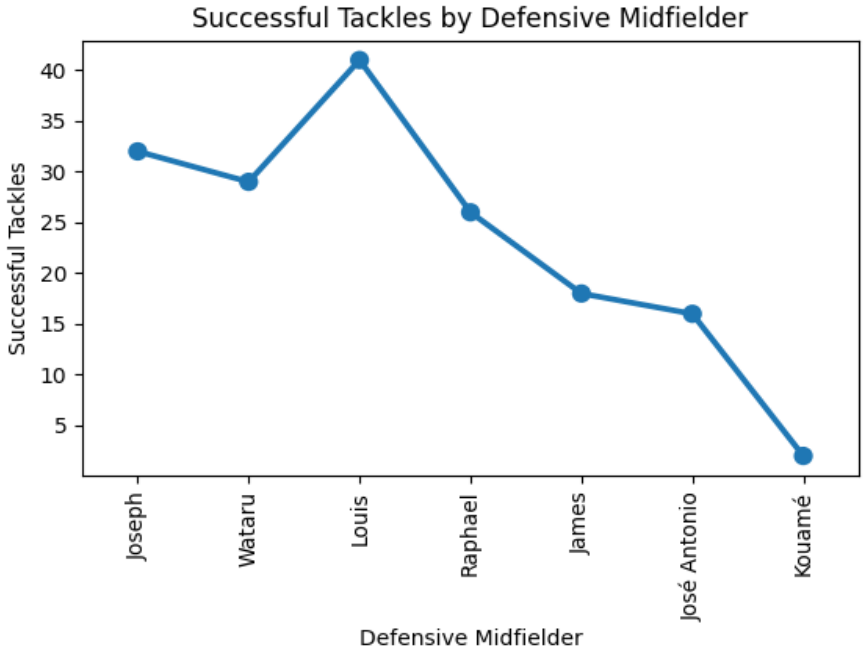


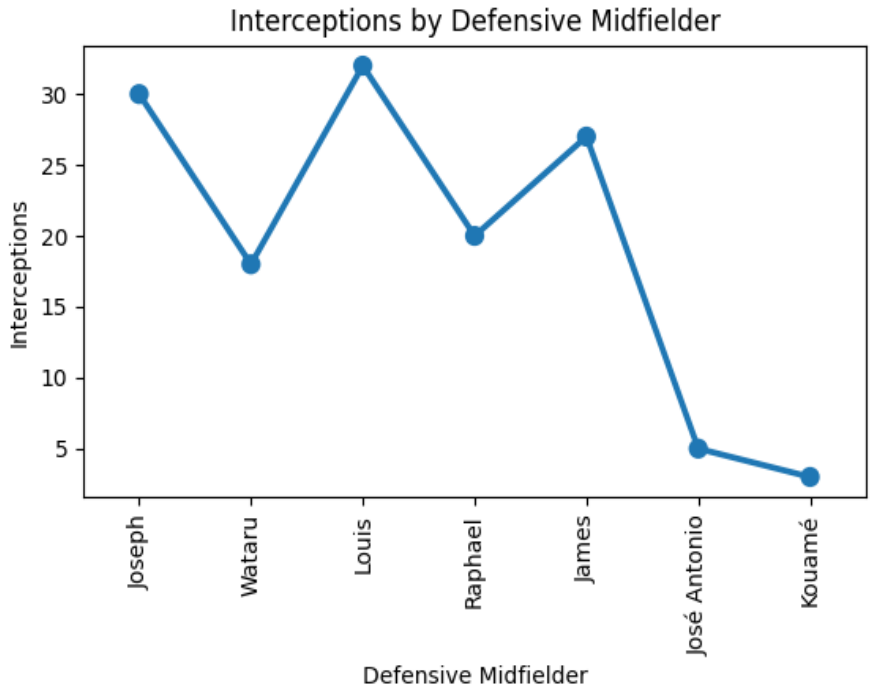




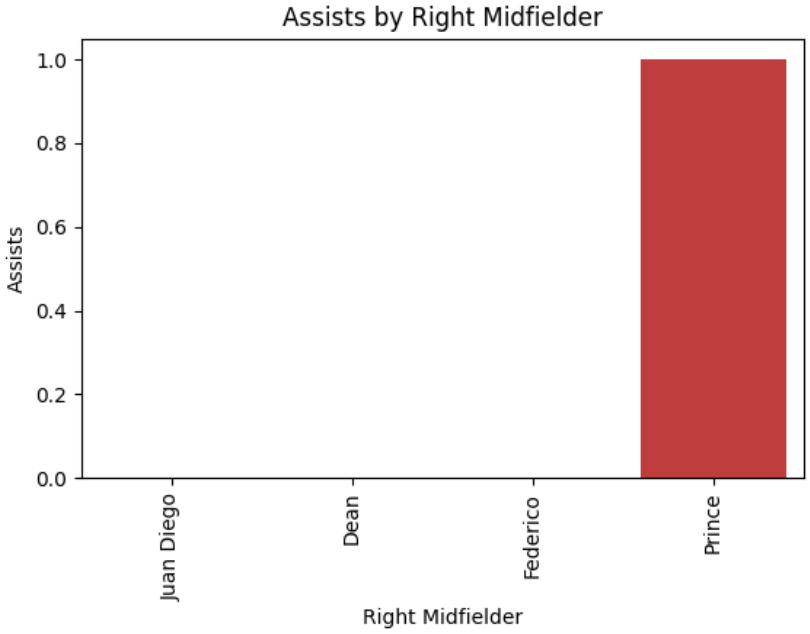
Defensive Midfielder: Louis *(score = assists + successful_tackles * 2 + interceptions * 0.5)*

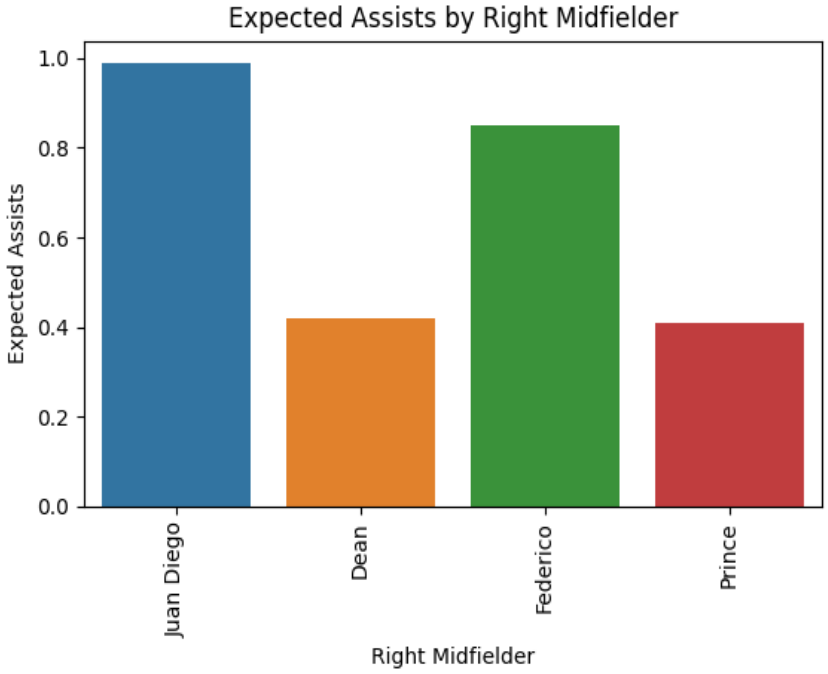


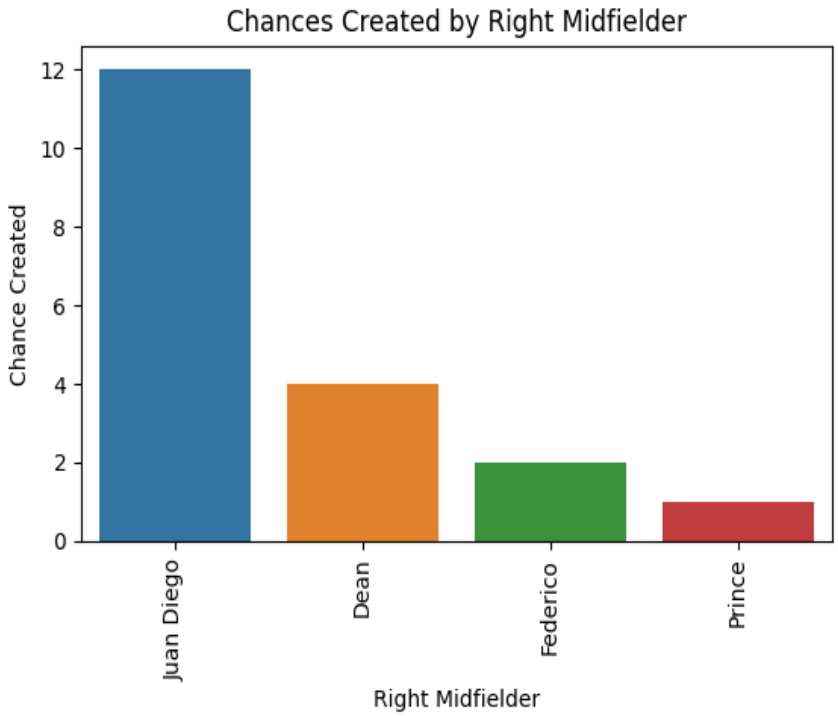

Successful Tackles by Defensive Midfielder
Successful Tackles
40
35
30
25
20
15
10
5
Joseph
Wataru
Louis
Raphael
James
José Antonio
Kouamé
Defensive Midfielder


Interceptions by Defensive Midfielder
Interceptions
30
25
20
15
10
5
Joseph
Wataru
Louis
Raphael
James
José Antonio
Kouamé
Defensive Midfielder

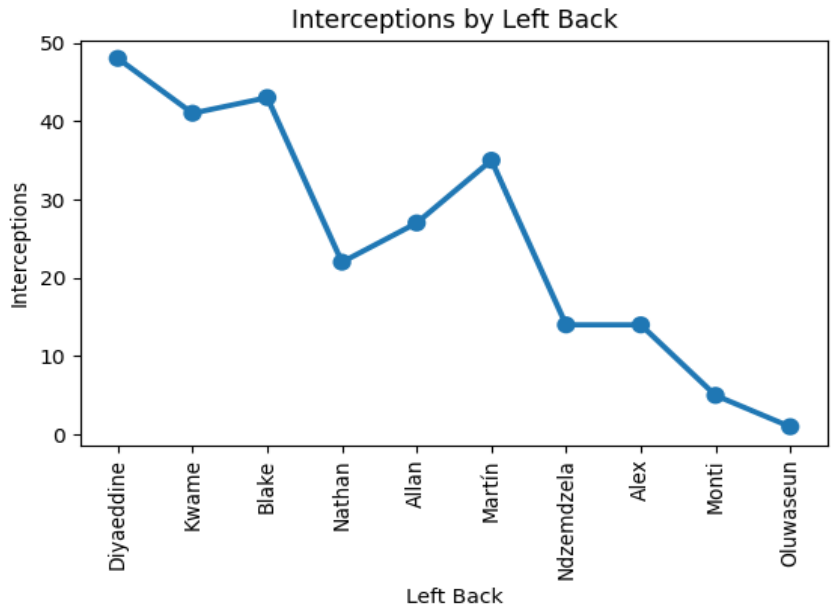
Right Midfielder: Juan *(score = assists * 2 + exp_assists + chances_created * 0.5)*

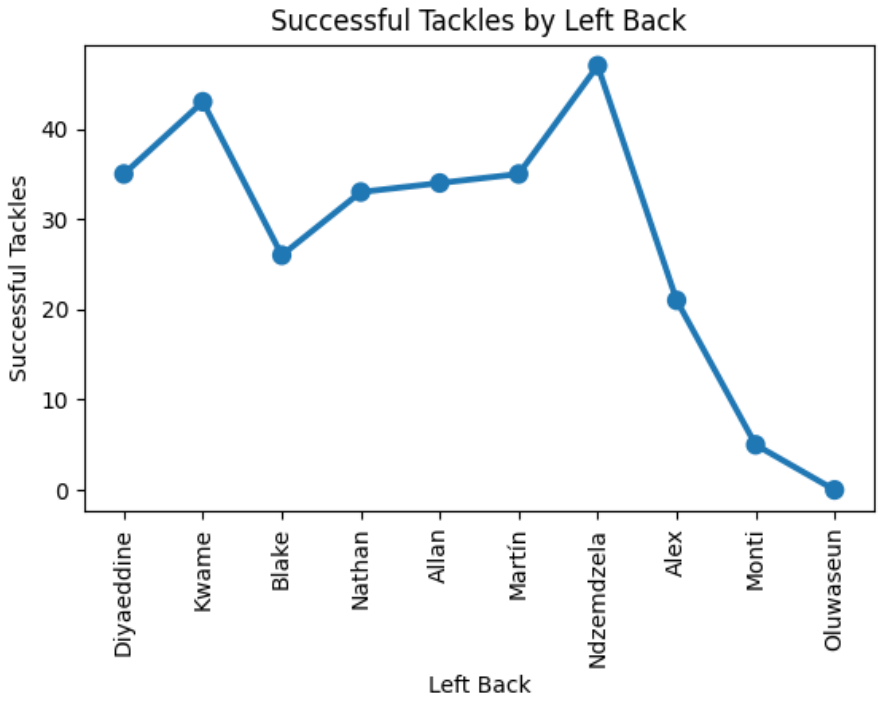


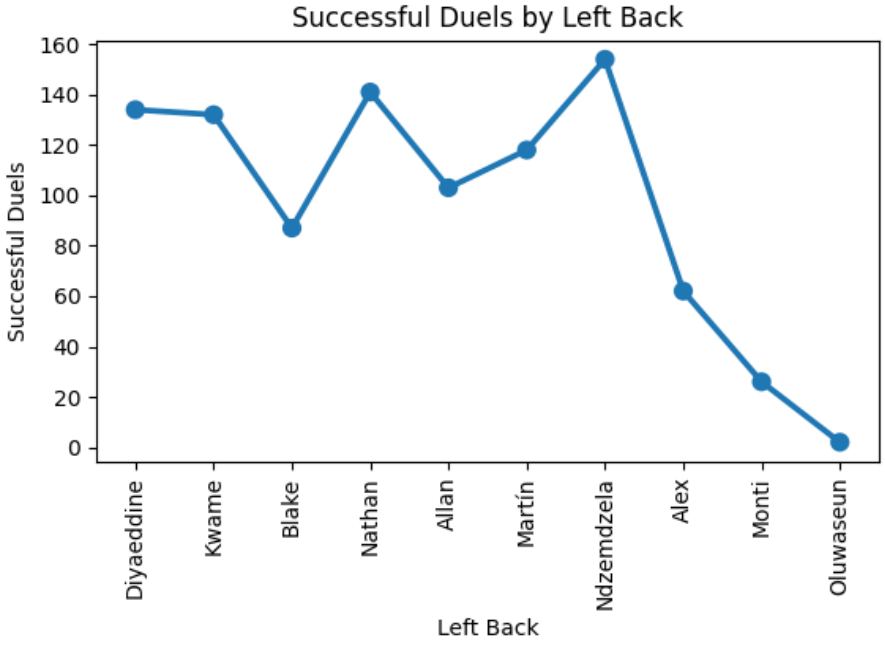




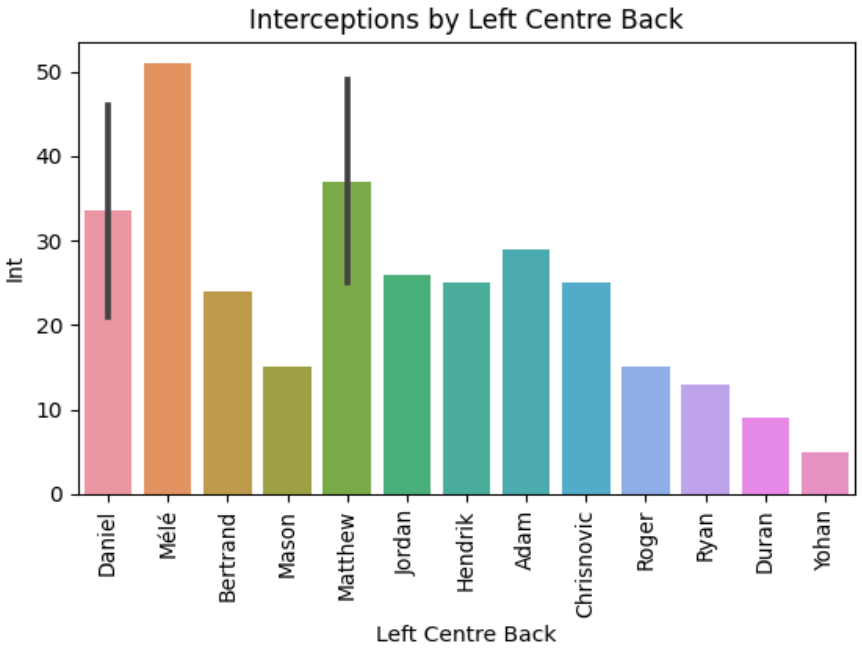
Left Back: Diyaeddine *(score = interceptions * 2 + successful_tackles * 0.5 + successful_duels)*

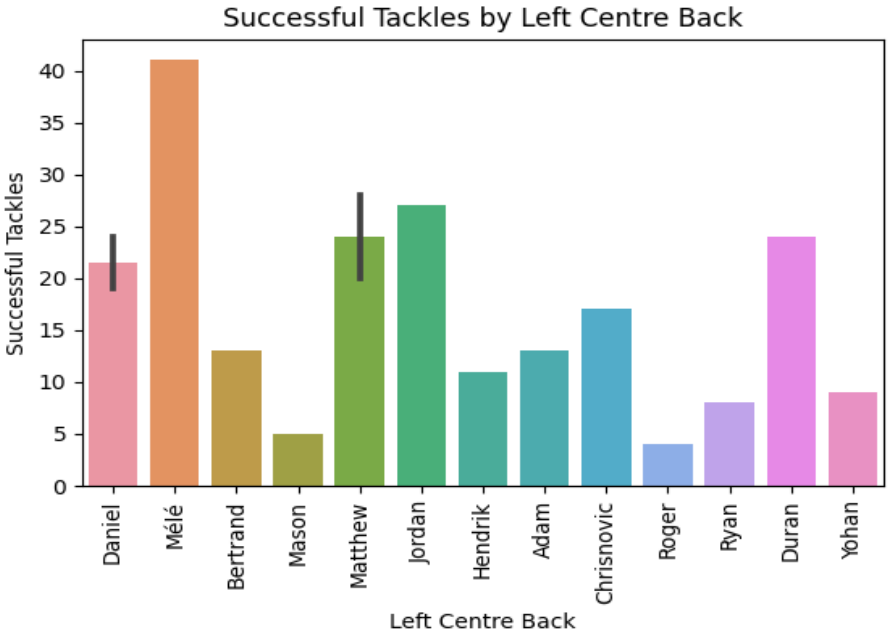


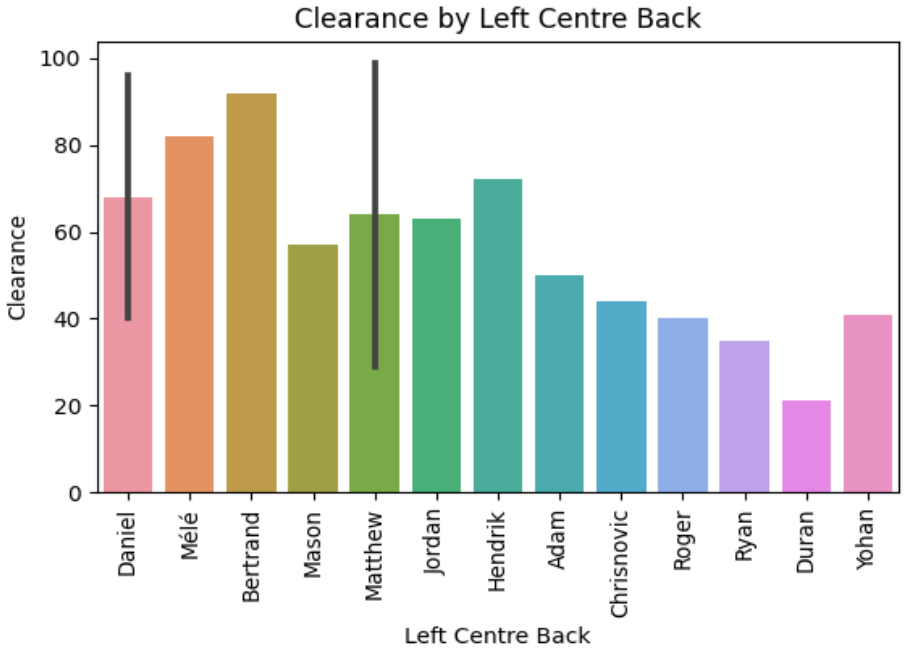

Successful Tackles by Left Back
Successful Tackles
0
10
20
30
40
Diyaeddine
Kwame
Blake
Nathan
Allan
Martín
Ndzemdzela
Alex
Monti
Oluwaseun
Left Back


Successful Duels by Left Back
Successful Duels
0
20
40
60
80
100
120
140
160
Diyaeddine
Kwame
Blake
Nathan
Allan
Martín
Ndzemdzela
Alex
Monti
Oluwaseun
Left Back

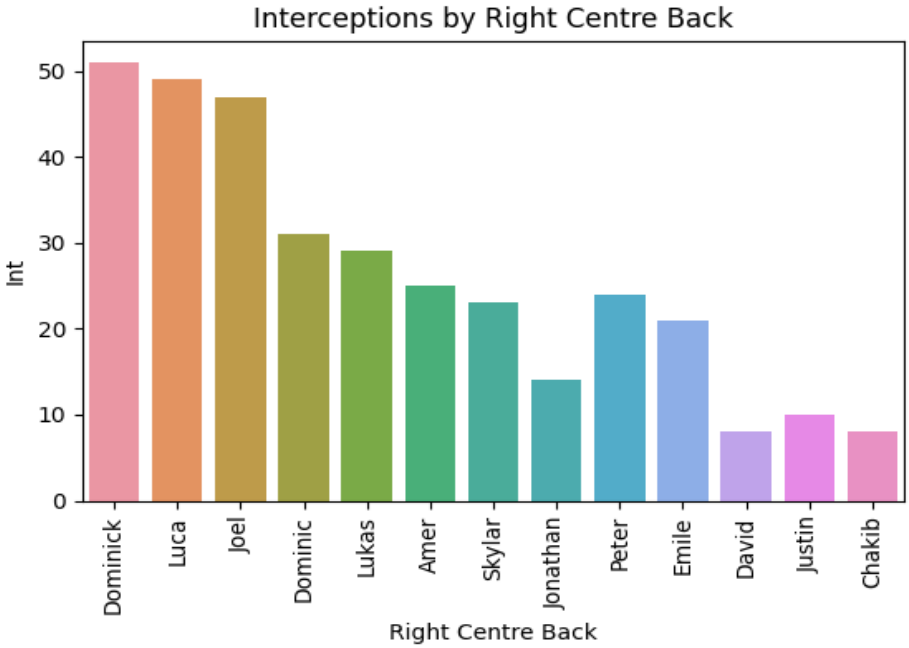
Left Centre Back: Matthew *(score = interceptions + successful_tackles * 2 + clearance * 3)*

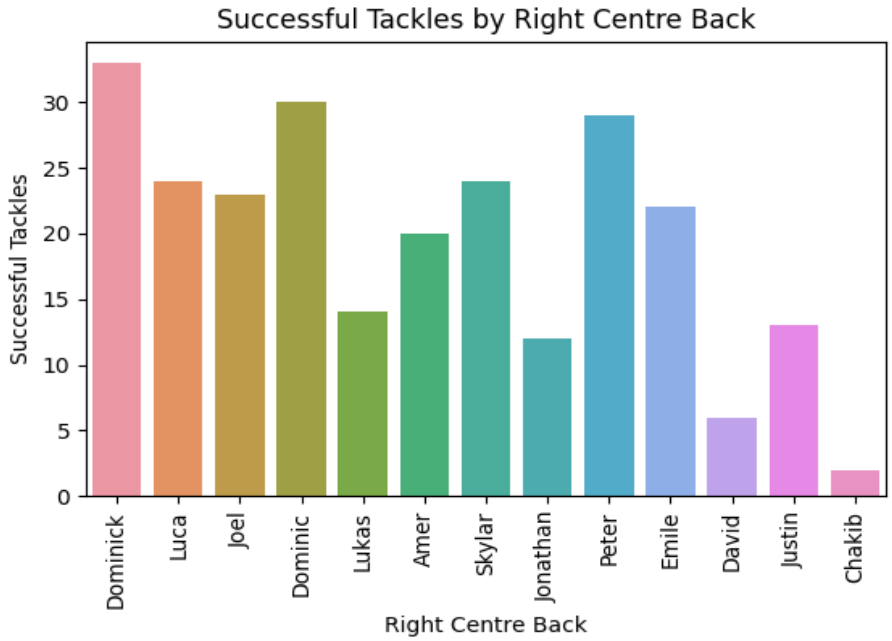


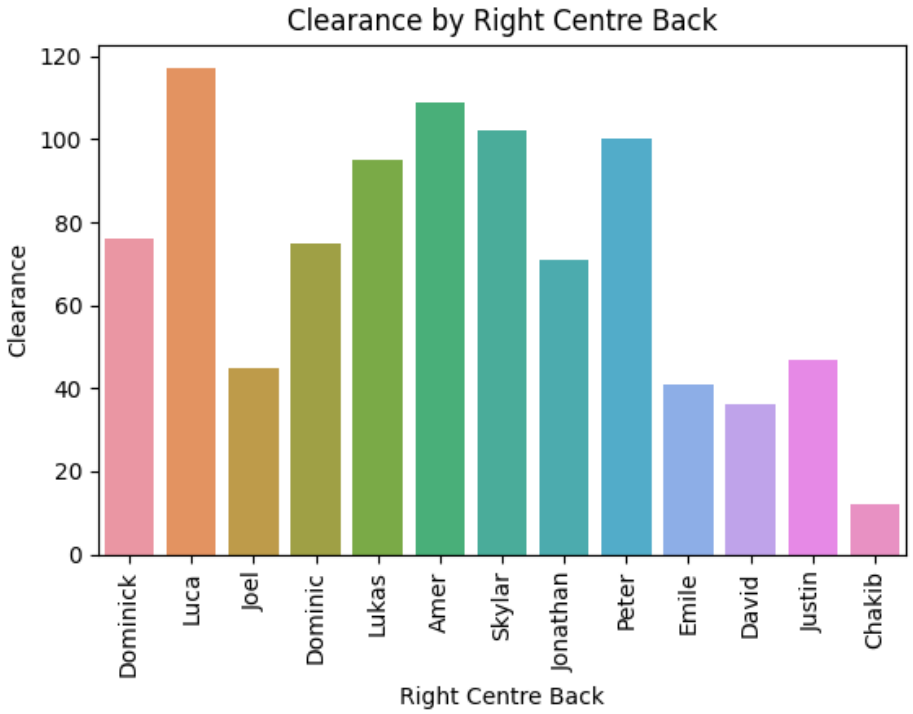




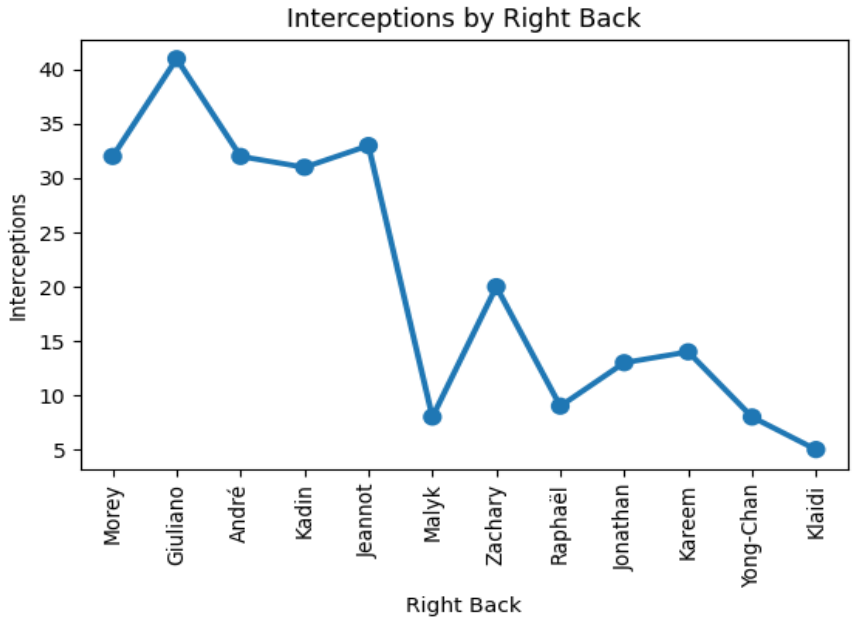
Right Center Back: Luca *(score = interceptions + successful_tackles * 2 + clearance * 3)*

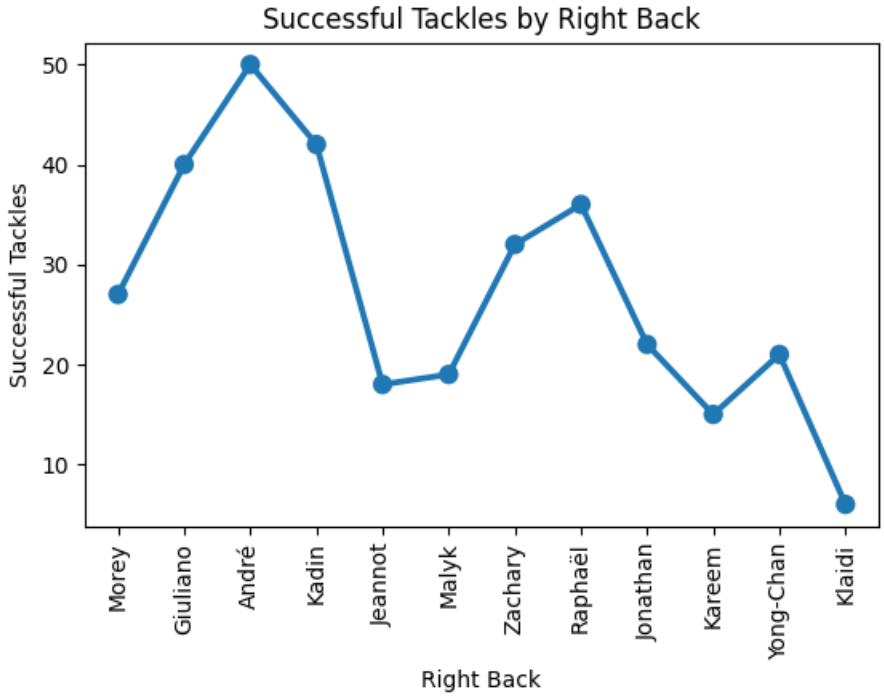


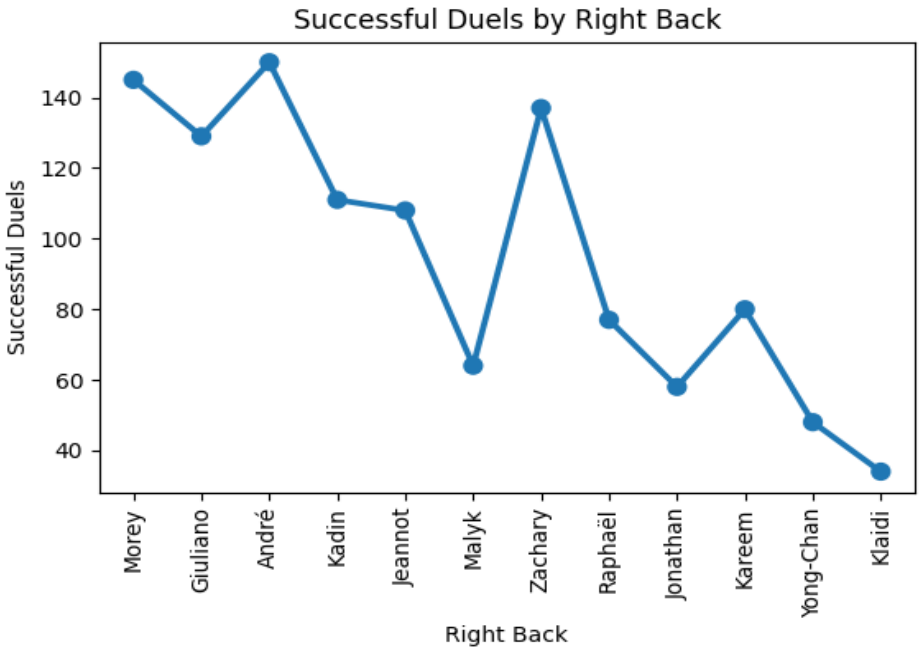

Successful Tackles by Right Centre Back
Successful Tackles
0
5
10
15
20
25
30
Dominick
Luca
Joel
Dominic
Lukas
Amer
Skylar
Jonathan
Peter
Emile
David
Justin
Chakib
Right Centre Back


Clearance by Right Centre Back
Clearance
0
20
40
60
80
100
120
Dominick
Luca
Joel
Dominic
Lukas
Amer
Skylar
Jonathan
Peter
Emile
David
Justin
Chakib
Right Centre Back

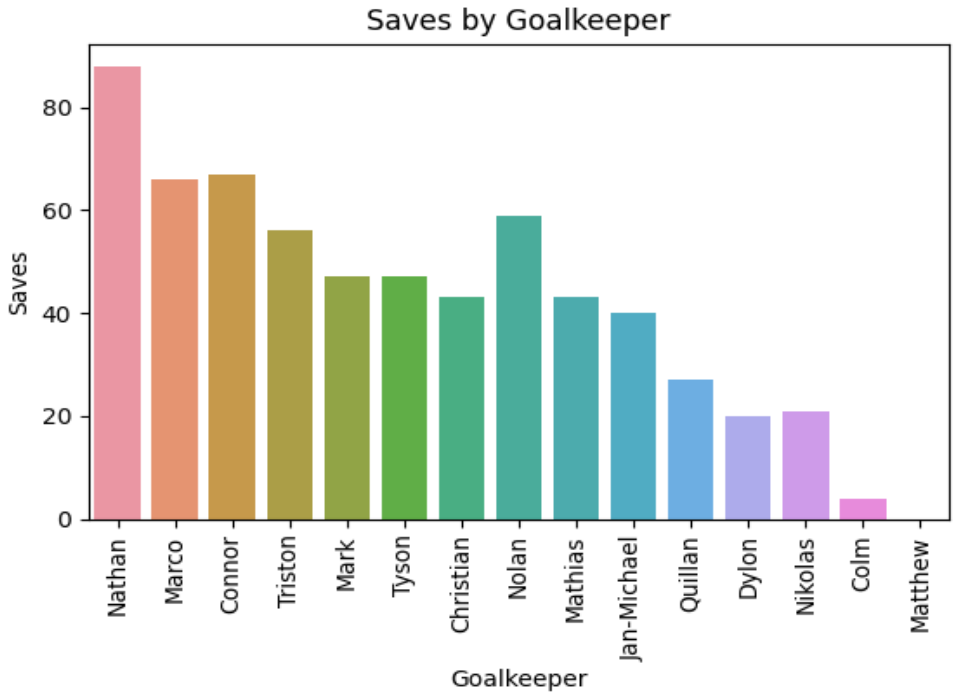
Right Back: Andre *(score = interceptions * 2 + successful_tackles * 0.5 + successful_duels)*

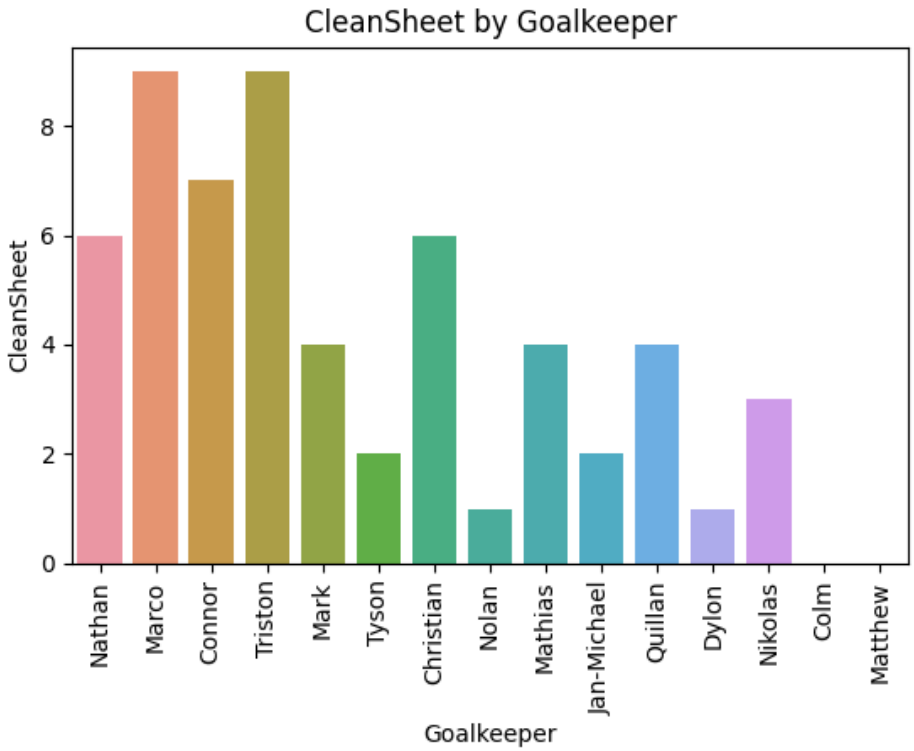






Goalkeeper: Nathan *(score = saves * 2 + clean_sheets * 3)*





REPORTING

The first step in the reporting process is to analyze the attributes of the Canadian Premier League data using Python. This involves using various data analysis libraries like NumPy, Pandas, and Matplotlib to explore the data, clean it, and identify patterns and trends. This is done to gain a better understanding of the data and to prepare it for visualization.

The reporting part of the Canadian Premier League: Dream Team project involves analyzing the data using Python and then creating visualizations using Power BI to communicate the findings in a clear and meaningful way.

Steps

- The first step in the reporting process is to analyze the attributes of the Canadian Premier League data using Python.
- Once the data is analyzed, the next step is to create visualizations.I.
- To create a report.
- Once we have created your visualizations and report, we can arrange them into a dashboard that provides an overview of the data.

Results:

Team Stats:

Total Goals Scored by each Team	Cavalry
Shots on Target Ratio of each Team	Cavalry
Pass Completion Rate of each Team	Forge
Total Fouls Committed by each Team	Cavalry
Touches in Opposition's Box	Forge
Yellow cards	Cavalry
Red cards	Cavalry
Overall	York9

Dream Team:

Centre Forward 1	Michael Petrasso
Centre Forward 2	Terran Campbell

Left MidFielder	Ryan Telfer
Central Attacking MidFielder	Tristan Borges
Defensive MidFielder	Louis Béland-Goyet
Right MidFielder	Juan Diego Gutiérrez
Left Back	Diyaeddine Abz
Left Center Back	Matthew Arnone
Right Center Back	Luca Gasparott
Right Back	André Bona
Goalkeeper	Nathan Ingham

COLLABORATION

To promote teamwork and collaboration within the team working on a CPL dream team analysis project, several techniques were implemented.

Firstly, the project used python to analyze all the problem statements and will use Power BI to create a dashboard that can be accessed by all four team members. The dashboard will allow for real-time editing and sharing of insights, encouraging collaboration and ensuring everyone is on the same page.

Additionally, the team will have edit access to the dashboard, enabling them to make updates and add new features as needed.

Finally, the project will also utilize mobile collaboration tools, allowing team members to work together and stay connected even when they are not in the same location. By implementing these techniques, the team can maximize their collective talents and insights, and create a successful football analysis project.

DEPLOYMENT

The deployment part of the project refers to the process of taking the analysis and insights generated through Python and Power BI and making them accessible to others. This could involve creating reports, dashboards, or other visualizations that allow stakeholders to easily understand and interact with the data.

Here's a step-by-step breakdown of the deployment process

Export the data

Connect to Power BI

Create visualizations

Design a dashboard

Share the dashboard

CONCLUSION

In conclusion, after analyzing the team stats and player stats of the Canadian Premier League for the 2019 season, we have identified a standout team and several standout players. Through this analysis, we have discovered that the winning team had a strong defense and an effective attacking strategy. We also found that the league's top performers were players who demonstrated exceptional skill in their respective positions.

Furthermore, this project has highlighted the importance of data analysis in sports. By utilizing statistical tools and techniques, we were able to gain valuable insights into team and player performance, which can be used to inform coaching decisions and player recruitment strategies.

Overall, this project has provided a comprehensive overview of the Canadian Premier League's 2019 season and has shown the potential of data analysis in sports. We hope that this analysis can contribute to the ongoing development of the league and help to create a more competitive and exciting football environment in Canada.

The Dream Team in 4-4-2 Formation: